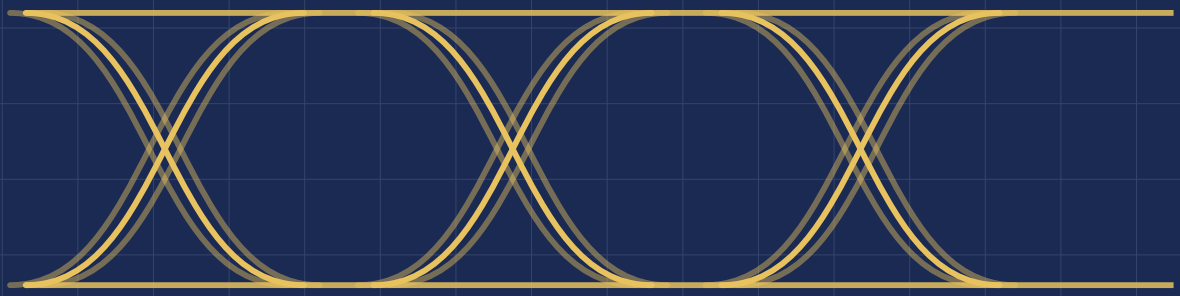
OPEN-SERDES · IHP SG13G2 · 5 Gb/s NRZ

Designing a Serial Link from Scratch

A design tutorial for undergraduate engineers building an open-source high-speed SerDes in 130 nm SiGe BiCMOS

Channels · Link budgets · Equalization · TX and RX circuits · Jitter, PLLs and CDR · Open-source design flow · Hands-on labs



BRICD

Biomedical Research in Integrated Circuit Design · thebricd.org

Open-SERDES Project
Draft v0.1 · October 2026

Preface

Every fast interface, from a PCIe slot to the backplane of an Ethernet switch, moves its data over a **serial link**: a transmitter, a lossy wire and a receiver that recovers both the bits and the clock. This book teaches you to design one by following a single, concrete chip, the **Open-SERDES** transceiver: a 5 Gb/s NRZ link built in the open-source IHP SG13G2 130 nm SiGe BiCMOS process [1] with open-source tools [2].

Who this book is for. Third- and fourth-year electrical engineering undergraduates who will help design the Open-SERDES chip, and anyone who wants a practical first course in high-speed links. After reading it you should be able to:

- predict the eye a channel will produce from its loss and pulse response;
- choose and size the TX FFE, CTLE and DFE for a channel, and defend the split;
- turn a link budget into specifications for the driver, front end, samplers, PLL and CDR;
- model the link in Python, then carry it through schematic, layout and sign-off with open tools.

Prerequisites. Basic circuits (KCL, small-signal MOS and BJT models, the differential pair), signals and systems (Fourier, Laplace, convolution) and some Python. Transmission lines, equalization, jitter, PLLs and CDRs are built up from first principles.

How the book is organized

The book has five parts, shown in Figure 1. **Part I, Foundations** (Chapters 1–4), gives you the vocabulary of links (pulse response, eye diagram, bit-error rate), the physics of the channel and the tools for a link budget. **Part II, Equalization** (Chapters 5–8), shows how the TX FFE, the CTLE and the DFE undo the channel, and how to split the work between them. **Part III, Transceiver Circuits** (Chapters 9–10), turns the system model into HBT and CMOS circuits. **Part IV, Clocking** (Chapters 11–13), covers jitter, the PLL and clock and data recovery. **Part V, Putting It Together** (Chapters 14–16), assembles the full architecture, walks through the open-source design flow and collects the hands-on labs.

Two reading paths. Everyone should read Part I first. After that, Figure 2 suggests two routes. The **system and modeling path** suits readers who will own the link budget and the serdespy model. The **circuit path** suits readers who will design transistors and layout. Both paths meet again at the architecture chapter and the labs.

How to read the boxes

Chapters open with what you will learn and close with key takeaways. In between, six kinds of boxes mark the role of each piece of text:



Figure 1. Your path through this tutorial. The five parts are milestones: Foundations, Equalization, Transceiver circuits, Clocking, and Putting it together. Each part builds on the numbers and models of the one before it.

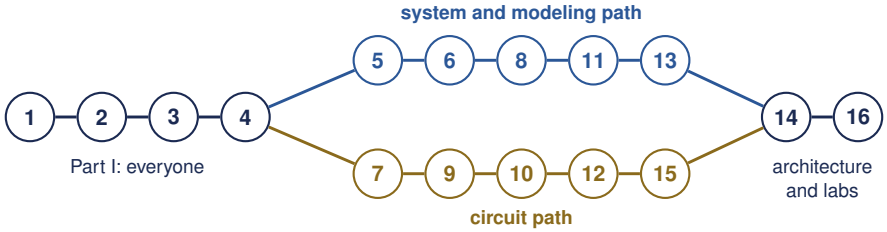


Figure 2. Two ways to read the book after Part I. Circles are chapter numbers. The system path (blue) follows the equalization theory, jitter and the CDR; the circuit path (gold) follows the CTLE circuits, the TX and RX, the PLL and the design flow. Both meet at the architecture (Chapter 14) and the labs (Chapter 16).

KEY EQUATION · Bit period

$$T_b = \frac{1}{f_b} = \frac{1}{5\,\mathrm{Gb/s}} = 200\,\mathrm{ps}$$

WORKED EXAMPLE P.1 · A number you will use on every page

At $5\,\mathrm{Gb/s}$, $f_N = f_b/2 = 2.5\,\mathrm{GHz}$; channel loss is quoted there.

▶ What this means for your design

An *insight* box says what a result means for the Open-SERDES design.

! Common pitfall

A *pitfall* box warns about a common mistake, such as quoting loss at $5\,\mathrm{GHz}$.

>_ Lab 0: set-up

A *try-it* box points to a lab script (Section 16.1).

Open-SERDES target

A *specnote* box states an Open-SERDES target: 5 Gb/s NRZ at BER $\leq 10^{-12}$.

Longer algebra sits in grey *derivation* boxes.

How the labs work

The labs of Chapter 16 reproduce the key plots of the book with **serdespy** [3], a Python library for system-level SerDes modeling, plus NumPy and SciPy. The scripts live in `tutorial/code/` and share one file, `common.py`, that defines the link, the reference channel and the helpers; the figure scripts in `tutorial/figsrc/` import the same model, so labs and figures use the same numbers. Run a lab, change a parameter and compare your plot with the text.

Sources and acknowledgments

The link-modeling chapters follow the structure of Prof. Sam Palermo's ECEN 720 *High-Speed Links* course at Texas A&M University [4]. The labs are built on serdespy by Richard Barrie and colleagues at the University of Toronto [3]. The chip itself is possible because IHP opened its SG13G2 process design kit [1] and the Institute for Integrated Circuits at JKU Linz maintains the IIC-OSIC-TOOLS container and the analog chip template [2, 5].

This is a **living draft (v0.1)**; numbers will be refined as the design matures. If you find an error, an unclear derivation or a figure that does not explain what it should, please email bricd.org@gmail.com or open an issue or a pull request on the project's GitHub repository, linked from thebricd.org.

How this book was made

This book, its figures and lab code, and the accompanying slides were developed with the help of AI tools. The authors reviewed the text, figures and code, but errors may remain; please report any you find as described above, and we will correct them. The topics, structure and many of the results follow the sources acknowledged above and listed in the References.

Contents

Preface i

I Foundations 1

1 Why Serial Links? 2

1.1 From parallel buses to serial links . . . 2
1.2 Anatomy of a SerDes link . . . 3
1.3 Where SerDes are used . . . 4
1.4 The Open-SERDES project . . . 6
1.5 IHP SG13G2 at a glance . . . 7
1.6 How to use the rest of the book . . . 9

2 Signals, Systems and the Language of Links 10

2.1 NRZ signaling: bits, symbols and the unit interval . . . 10
2.2 PRBS test patterns . . . 12
2.3 Linear systems in one section . . . 13
2.4 The pulse response . . . 14
2.5 ISI: how pulse spreading closes the eye . . . 16
2.6 Eye diagrams . . . 17
2.7 Noise and the bit-error rate . . . 18
2.8 Exercises . . . 20

3 The Channel 21

3.1 What is the channel? . . . 21
3.2 Transmission lines . . . 21
3.3 Loss mechanisms . . . 23
3.4 Reflections and termination . . . 25
3.5 S-parameters . . . 28
3.6 From frequency to time . . . 29
3.7 The Open-SERDES reference channel . . . 30
3.8 Exercises . . . 32

4 Link Budgets and Statistical Analysis **34**

4.1 Pulse-response link analysis . . . 34
4.2 Peak distortion analysis . . . 34
4.3 The statistical eye . . . 36
4.4 The system noise and margin budget . . . 38
4.5 Modulation choice: NRZ or PAM4? . . . 40
4.6 Exercises . . . 41

II Equalization **43**

5 Equalization Fundamentals **44**

5.1 Why equalize? . . . 44
5.2 The zero-forcing idea . . . 44
5.3 Where equalizers live . . . 46
5.4 Preview: who does how much? . . . 48

6 Transmit FFE **49**

6.1 A FIR filter in the driver . . . 49
6.2 De-emphasis, pre-emphasis and the swing limit . . . 49
6.3 Solving for the taps . . . 51
6.4 Practical notes . . . 53

7 Continuous-Time Linear Equalizer **55**

7.1 From RC to RLC: three passive ideas . . . 55
7.2 The source-degenerated differential pair . . . 56
7.3 The CTLE on the REF channel . . . 58
7.4 An HBT CTLE in SG13G2 . . . 59

8 Decision-Feedback Equalization and Adaptation **62**

8.1 The DFE principle . . . 62
8.2 The tap-1 timing loop . . . 64
8.3 Adapting the taps . . . 65
8.4 The DFE on the REF channel . . . 66
8.5 Splitting the work . . . 66
Exercises . . . 69

III Transceiver Circuits 70

9 The Transmitter 71

9.1 The transmitter at a glance . . . 71
9.2 The serializer . . . 71
9.3 Termination . . . 74
9.4 The CML output driver . . . 74
9.5 The FFE inside the driver . . . 76
9.6 Pad, ESD and bandwidth . . . 78
9.7 TX design checklist . . . 79

10 Receiver Front-End and Samplers 81

10.1 The receiver front-end . . . 81
10.2 Samplers . . . 82
10.3 Metastability . . . 85
10.4 Offset and calibration . . . 87
10.5 Noise and the sensitivity budget . . . 89
10.6 The deserializer . . . 89
10.7 RX design checklist . . . 90

IV Clocking 92

11 Jitter 93

11.1 What jitter is . . . 93
11.2 Random and deterministic jitter . . . 95
11.3 The dual-Dirac model and total jitter . . . 96
11.4 From phase noise to jitter . . . 97
11.5 The Open-SERDES jitter budget . . . 98
11.6 Jitter transfer, generation and tolerance . . . 99

12 Clock Generation: The Phase-Locked Loop 100

12.1 The charge-pump PLL . . . 100
12.2 Loop dynamics . . . 101
12.3 PLL noise and the optimum bandwidth . . . 103
12.4 The LC VCO . . . 104
12.5 Quadrature clocks and distribution . . . 106

13 Clock and Data Recovery **109**

13.1 Why the receiver needs a CDR . . . 109
13.2 Phase detectors . . . 110
13.3 The linear CDR model and jitter transfer . . . 111
13.4 Bang-bang CDR dynamics . . . 111
13.5 Jitter tolerance . . . 113
13.6 The Open-SERDES PI-based CDR . . . 114
13.7 Exercises . . . 117

V Putting It Together **118**

14 The Open-SERDES Architecture **119**

14.1 What Open-SERDES delivers . . . 119
14.2 The complete transceiver . . . 120
14.3 Block specifications . . . 122
14.4 Design for 10, guarantee 5 . . . 124
14.5 From transceiver to test chip . . . 126
14.6 Power budget . . . 128
14.7 Design decisions . . . 130
14.8 Roadmap: from blocks to silicon . . . 130
14.9 Risks and open questions . . . 130

15 Design and Verification Flow with Open Tools **133**

15.1 The open toolchain . . . 133
15.2 The model–circuit loop . . . 133
15.3 Testbenches . . . 136
15.4 Layout for high speed . . . 138
15.5 Testing the chip in the lab . . . 140

16 Hands-on Labs with serdespy **142**

16.1 Lab 0: set-up . . . 142
16.2 Lab 1: the channel . . . 143
16.3 Lab 2: a PRBS eye through the channel . . . 143
16.4 Lab 3: TX FFE . . . 145
16.5 Lab 4: CTLE . . . 145
16.6 Lab 5: DFE . . . 146

16.7 Lab 6: jitter, noise and BER 147
16.8 Lab 7: a bang-bang PI-CDR 148
16.9 Where to go next 150

A Formula Sheet **151**

B Tables and Constants **153**

C Design-Review Checklists **154**

D Glossary **155**

E Further Reading **156**

References **157**

PART I

Foundations

What a serial link is, the language we use to describe it, and the channel that makes it hard. Everything later in this tutorial is built on the pulse response, the eye diagram and the bit-error rate introduced here.

CHAPTER 1

Why Serial Links?

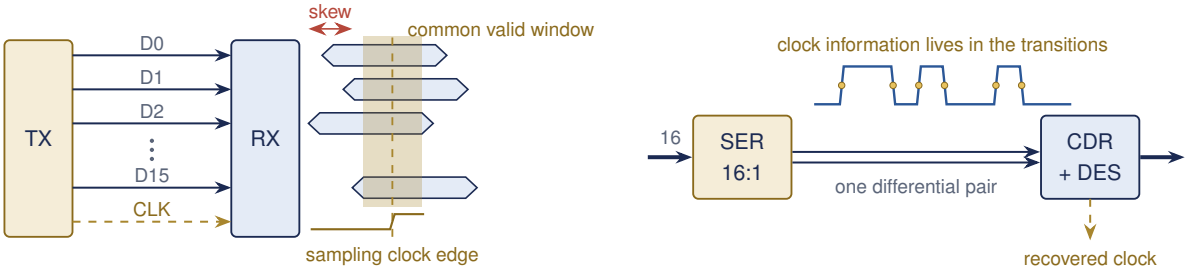
Chips have far more bandwidth inside than pins to move it out, so we send bits one after another, very fast, over a few carefully engineered wires.

IN THIS CHAPTER YOU WILL LEARN

- explain why high-speed I/O moved from wide parallel buses to serial links with an embedded clock;
- name every block of a SerDes link and the data rate at each point of the Open-SERDES design;
- place 5 Gb/s NRZ among real standards such as PCIe, USB and backplane Ethernet;
- state the Open-SERDES targets and why 5 Gb/s NRZ in SG13G2 is the right first chip;
- compare the speed of SG13G2 HBTs with 130 nm CMOS, and describe the back-end metal stack.

1.1 From parallel buses to serial links

The obvious way to move data between chips is a **parallel bus**: one wire per bit plus a clock wire. A 16-bit bus at 312.5 Mb/s per wire moves 5 Gb/s. As rates grew, four problems stopped it from scaling (Figure 1.1a).



(a) Parallel bus, forwarded clock

(b) Serial link, embedded clock

Figure 1.1. A parallel bus is limited by skew; a serial link carries its own clock. (a) Each lane of a bus arrives with a slightly different delay. The receiver can only sample inside the window where every lane is valid, and that window shrinks as the bit time shrinks. (b) A serial link sends one fast stream on one differential pair. The receiver extracts the clock from the data transitions (gold dots), so there is no clock wire to match.

- **Skew.** Lanes differ in length, load and driver strength, so bits launched together arrive at different times; the usable window is the bit time minus the worst skew.
- **Pin count.** Pins and board area are expensive and grow far more slowly than on-chip bandwidth, and a wide bus needs many return pins.
- **Power and noise.** Many single-ended drivers switching together cause supply bounce and crosstalk, and the energy per bit stays high.
- **The clock wire.** The forwarded clock must be matched to every lane across process, voltage and temperature.

A **serial link serializes** the word into one full-rate stream on a single **differential pair**, which carries its own return path and rejects common-mode noise. No clock is sent: a **clock and data recovery** (CDR) circuit in the receiver locks a local clock to the data transitions, so the clock is **embedded** in the data and lane-to-lane skew no longer matters.

WORKED EXAMPLE 1.1 · Skew budget: parallel bus versus serial link

A signal on FR-4 ($\varepsilon_r \approx 4$) travels at $v = c/\sqrt{\varepsilon_r} \approx 0.15\,\text{mm/ps}$, or about 6.7 ps per millimeter. Suppose two lanes differ in length by 10 mm, a modest mismatch on a crowded board. The skew is $10\,\text{mm} \times 6.7\,\text{ps/mm} \approx 67\,\text{ps}$.

- 16-bit bus at 312.5 Mb/s: the bit time is 3.2 ns, so the skew costs $67/3200 \approx 2\%$ of it.
- The same bus pushed to 5 Gb/s per lane: the bit time is 200 ps and the same skew eats *one third* of it, before any jitter or noise.

The pins tell a similar story: 16 data wires plus a clock and their returns, against one differential pair (2 pins) for the serial link.

The price is speed: each bit lasts only $T_\text{b} = 1/f_\text{b}$, and the wire that looked ideal at 312.5 Mb/s now smears each bit into its neighbors. Paying that price efficiently is measured by the energy per bit.

KEY EQUATION · Energy efficiency of a link

$$E_b = \frac{P_\text{link}}{f_\text{b}}, \qquad 1\,\text{mW per Gb/s} = 1\,\text{pJ/b} \tag{1.1}$$

A transceiver burning 50 mW at 5 Gb/s spends 10 pJ per bit.

▶ What this means for your design

Serial links win because they trade many slow, skew-limited wires for one fast, self-clocked wire. The cost moves into the transceiver: it must drive, equalize and re-time a 200 ps bit after a lossy channel. Every chapter that follows is about doing that within a power budget.

1.2 Anatomy of a SerDes link

Figure 1.2 shows the complete Open-SERDES link. Follow a 16-bit word from left to right.

Transmitter (TX). The **serializer** works in stages: CMOS logic does 16:4, current-mode logic (CML) does 4:2, and a **half-rate** 2:1 CML multiplexer clocked at 2.5 GHz emits one bit per clock edge (Section 9.2). The **driver** launches a $1\,V_\text{ppd}$ signal into the 50 Ω-per-side channel through a 3-tap **feed-forward equalizer** (FFE) that pre-distorts it against the channel loss (Chapter 6 and Section 9.4). A **phase-locked loop** (PLL) with an LC oscillator makes the 2.5 GHz clock (Chapter 12).

Channel. Packages, board (PCB) traces and vias and a connector together act as a low-pass filter with reflections; the Open-SERDES reference channel loses 17.4 dB at 2.5 GHz (Chapter 3).

Receiver (RX). A 100 Ω **termination** with ESD protection absorbs the wave (Section 9.3). A **continuous-time linear equalizer** (CTLE) restores the lost high frequencies (Chapter 7). Four

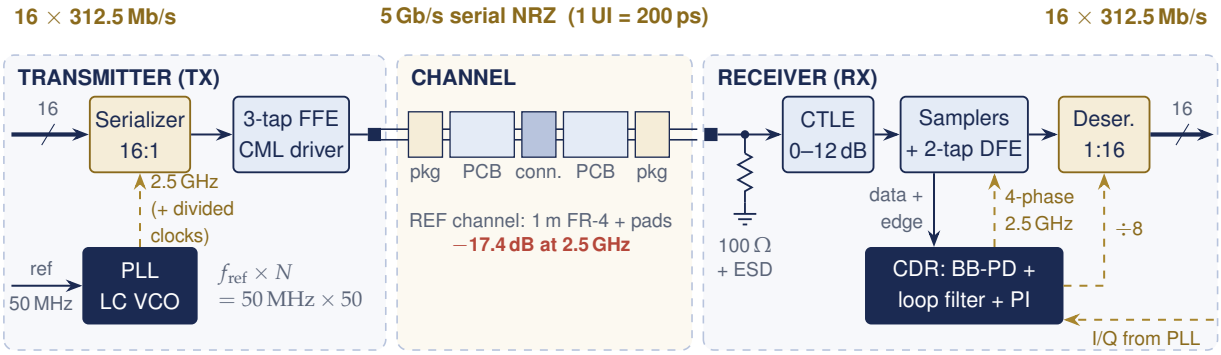


Figure 1.2. Anatomy of the Open-SERDES link, with the data rate at every point. Sixteen bits at 312.5 Mb/s are serialized to one 5 Gb/s NRZ stream, driven through a 3-tap FFE driver, attenuated by the package, board and connector, then equalized (CTLE and DFE), sampled with a recovered clock and deserialized back to 16 bits. Gold dashed lines are clocks: the TX PLL multiplies a 50 MHz reference to 2.5 GHz, and the RX CDR steers four sampling phases using I/Q clocks from a local PLL. All high-speed paths are differential.

samplers on the 0°/90°/180°/270° phases of a 2.5 GHz clock take data and edge samples, while a 2-tap **decision-feedback equalizer** (DFE) subtracts the echo of previous bits (Chapter 8 and Section 10.2). The **CDR** steers a phase interpolator to keep the samples centered (Chapter 13), and the **deserializer** rebuilds the 16-bit word (Section 10.6).

WORKED EXAMPLE 1.2 · Following the rates through the link

- Parallel side: $16 \times 312.5\,\text{Mb/s} = 5\,000\,\text{Mb/s} = 5\,\text{Gb/s}$.
- Serial side: $T_b = 1/5\,\text{Gb/s} = 200\,\text{ps}$, the **unit interval** (UI).
- Half-rate clock: 2.5 GHz has a 400 ps = 2 UI period; both edges give one bit per 200 ps. The four CDR phases are 0.5 UI apart: two data centers and two edges.
- PLL: $f_{ref} \times N = 50\,\text{MHz} \times 50 = 2.5\,\text{GHz}$. The deserializer output clock is $2.5\,\text{GHz}/8 = 312.5\,\text{MHz}$.

The half-rate clock frequency equals the Nyquist frequency $f_N = f_b/2$ of the data (Section 2.1).

Figure 1.3 summarizes the link with the four numbers you will meet on almost every page.

1.3 Where SerDes are used

Serial links connect a CPU to its graphics card (PCIe), a computer to its drives (USB, SATA) and the line cards of a network switch (Ethernet). Table 1.1 lists common standards. **GT/s** counts symbols on the wire and Gb/s counts bits; they are equal for NRZ, while PAM4 carries two bits per symbol (Section 4.5). Line rates include coding overhead: 8b/10b sends 10 line bits per 8 data bits, 64b/66b sends 66 per 64.

The last row is a useful counterexample: UCIe links chiplets millimeters apart in one package, where thousands of short, matched wires are cheap, so the parallel bus with a forwarded clock wins again. Serial links win when the wire is long, lossy and expensive, which is the situation Open-SERDES targets.

Figure 1.4 shows per-lane rates doubling roughly every three to four years. Around 50 Gb/s the

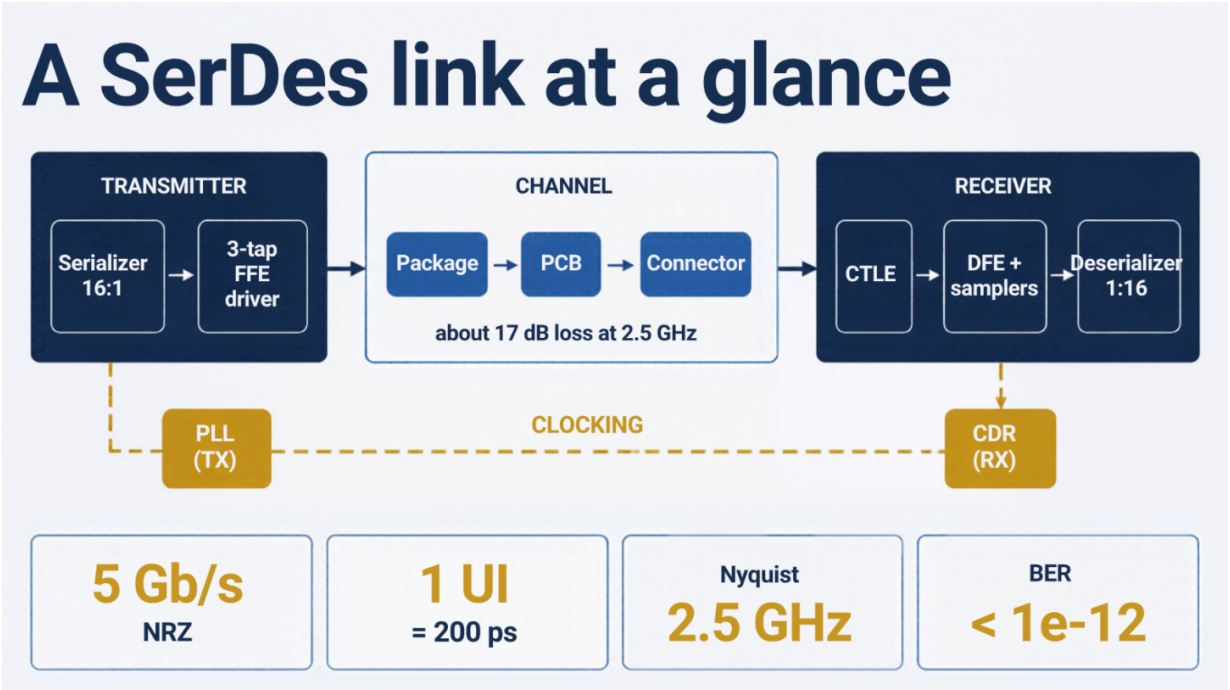


Figure 1.3. A SerDes link at a glance. The transmitter (serializer, FFE driver), the channel (package, PCB, connector) and the receiver (CTLE, DFE, samplers, deserializer), held together by the PLL and the CDR. Keep the four key numbers in mind: 5 Gb/s, a 200 ps UI, a 2.5 GHz Nyquist frequency and a target BER of 10^{-12}.

Table 1.1. Per-lane line rates of common serial standards. The Nyquist frequency is half the symbol rate. Open-SERDES, at 5 Gb/s NRZ with $f_N = 2.5$ GHz, sits next to PCIe 2.0, USB 3.2 Gen 1 and 5GBASE-KR.

Standard (year)	Rate per lane	Modulation and coding	f_N (GHz)
PCIe 1.0a / 2.0 (2003 / 2007)	2.5 / 5 GT/s	NRZ, 8b/10b	1.25 / 2.5
PCIe 3.0 / 4.0 / 5.0 (2010–19)	8 / 16 / 32 GT/s	NRZ, 128b/130b	4 / 8 / 16
PCIe 6.0 (2022)	64 GT/s	PAM4, 32 GBd, FEC	16
USB 3.2 Gen 1 / Gen 2 (2008 / 2013)	5 / 10 Gb/s	NRZ, 8b/10b / 128b/132b	2.5 / 5
USB4 v2 (2022)	40 Gb/s	PAM3, 25.6 GBd	12.8
SATA I / II / III (2003–09)	1.5 / 3 / 6 Gb/s	NRZ, 8b/10b	0.75 / 1.5 / 3
2.5GBASE-KX (802.3cb, 2018)	3.125 Gb/s	NRZ, 8b/10b	1.56
5GBASE-KR (802.3cb, 2018)	5.156 Gb/s	NRZ, 64b/66b	2.58
10GBASE-KR (802.3ap, 2007)	10.3125 Gb/s	NRZ, 64b/66b	5.16
100GBASE-KR4 (802.3bj, 2014)	25.78 Gb/s	NRZ, RS-FEC	12.9
400GBASE-KR4 (802.3ck, 2022)	106.25 Gb/s	PAM4, 53.1 GBd, FEC	26.6
UCIe 1.0–3.0 (2022–25)	4–64 GT/s	NRZ, single-ended, fwd. clock	≤ 32

industry moved to PAM4, because the loss at the NRZ Nyquist frequency became too high.

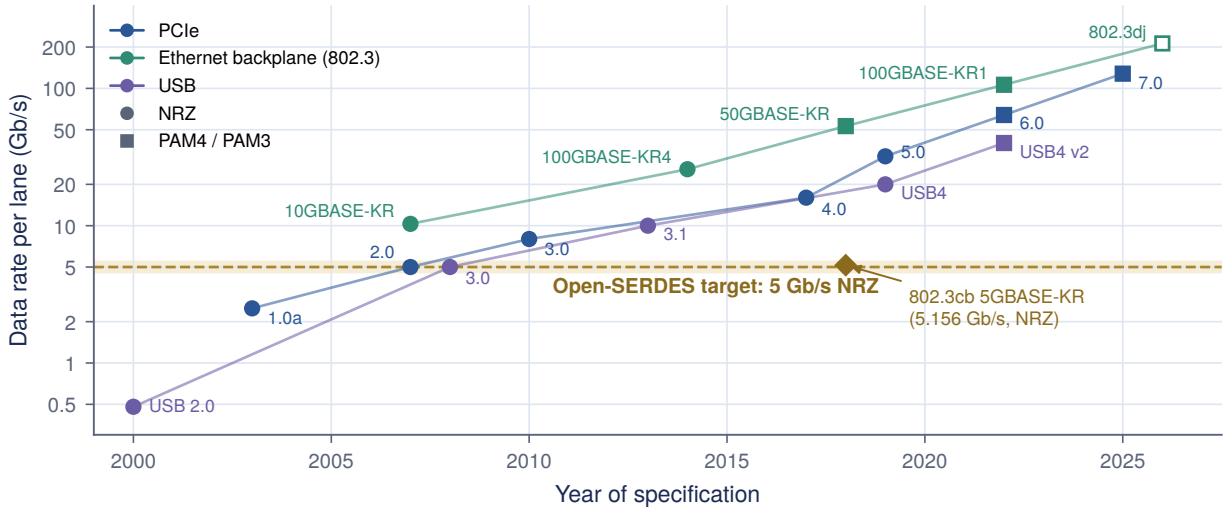


Figure 1.4. Per-lane data rates double every three to four years. PCIe generations, IEEE 802.3 backplane Ethernet lanes and USB, on a log axis. Squares mark multi-level signaling (PAM4, or PAM3 for USB4 v2); the open square is 802.3dj (200 Gb/s per lane), whose completion is planned for 2026. The gold band marks the Open-SERDES target, close to 5GBASE-KR (802.3cb) and PCIe 2.0. Sources: [6–8].

1.4 The Open-SERDES project

Open-SERDES is an open project to design a complete 5 Gb/s serial-link IP for Ethernet-class backplane links in IHP SG13G2, using only open-source PDKs and tools, with every schematic, layout, model and test bench public. The scope runs from specifications and architecture, through a Python link-simulation framework, to the circuits, custom analog layout, synthesized and routed digital blocks, and post-layout verification with DRC and LVS. The **deliverables** are a tape-out-ready GDSII, Verilog RTL with test benches, a demonstration of the modeling framework, a documented GitHub repository with a thesis, and a paper on the whole flow. The specnote collects the targets; Chapter 14 expands them.

Open-SERDES target (reference design)

Data rate	5 Gb/s NRZ, $T_b = 1\,\mathrm{UI} = 200\,\mathrm{ps}$, $f_N = 2.5\,\mathrm{GHz}$
Target BER	$\leq 10^{-12}$ ($Q = 7.03$); PRBS7 for bring-up, PRBS31 for compliance
Channel	REF: 1 m FR-4 stripline + 300 fF pads, −17.4 dB at 2.5 GHz
TX	16:1 serializer (half-rate 2:1 CML mux), HBT CML driver, 3-tap FFE, 1 V_{ppd}, 100 Ω diff.
RX	100 Ω termination + ESD, HBT CTLE (0–12 dB), half-rate 2-tap DFE, 4 samplers, 1:16 deserializer
Clocking	integer-N CP-PLL, LC VCO at 2.5 GHz (f_{ref} = 50 MHz, N = 50); PI-based bang-bang CDR, 64 PI steps per 2 UI
Test	on-chip PRBS7/31 generator and checker, near-end loopback

Why 5 Gb/s NRZ is the right first target.

- **Fast enough to be real.** It matches PCIe 2.0, USB 3.2 Gen 1 and 5GBASE-KR, and a 1 m board loses 17 dB at 2.5 GHz: the raw eye is closed, so every equalization idea in this book is

needed.

- **Slow enough to be safe.** A 200 ps UI is easy for SG13G2 HBTs but at the edge of 130 nm CMOS (Section 1.5), which shows clearly why BiCMOS helps.
- **NRZ is simple and robust.** PAM4 costs about 9.5 dB of eye, three times the samplers and better linearity, which pays off only on much lossier channels (Section 4.5).
- **Proven blocks.** The team's 2.4 GHz LC-VCO PLL in 130 nm BiCMOS [9] moves to 2.5 GHz with little change.

Prior art. A previous undergraduate project modeled a 5 Gb/s SerDes in a commercial 28 nm library [10]. The Purdue **OpenSerDes** [11] was the first open-source SerDes: an all-digital, synthesizable SkyWater 130 nm CMOS design with inverter-based drivers and an oversampling CDR. In post-layout simulation it ran at 2 Gb/s and drew about 438 mW, mostly in the serializer, deserializer and CDR logic; standard 130 nm CMOS cells capped its speed. Neither project produced a silicon-proven, open, analog 5 Gb/s SerDes. Open-SERDES keeps OpenSerDes's open flow but moves the fastest circuits to SiGe HBTs, which breaks the CMOS speed limit at the same 130 nm node.

The plan. We organize the scope into five phases (Figure 1.5): specifications, channel and supply plan (Phase 0); the serdespy system model that checks every later decision (Phase 1); circuits and digital RTL (Phase 2); layout, place-and-route, extraction and EM simulation (Phase 3); and sign-off, tape-out and lab test (Phase 4).

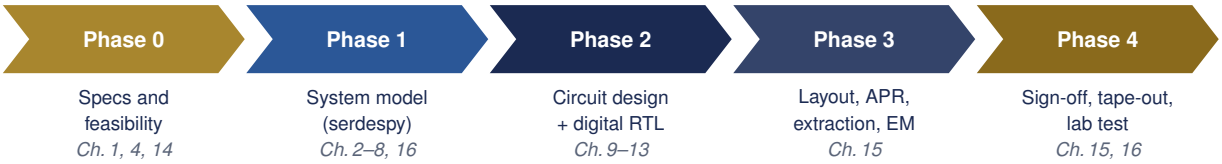


Figure 1.5. The Open-SERDES project in five phases. Each phase ends in a reviewable result: a specification, a system model, schematics, layouts and a signed-off GDSII. The italic line names the chapters that support each phase.

1.5 IHP SG13G2 at a glance

SG13G2 is IHP's 130 nm SiGe:C BiCMOS process, and its PDK is open source [1]. It offers two kinds of transistor on one die: standard 130 nm CMOS (1.2 V thin-oxide core devices and 3.3 V thick-oxide I/O devices), and the **npn13G2** SiGe heterojunction bipolar transistor (HBT), with $f_T \approx 300$ GHz and $f_{max} \approx 450$–500 GHz [12].

HBT versus CMOS speed. The transit frequency f_T, where the current gain falls to one, sets the device's intrinsic time scale $1/(2\pi f_T)$: about 0.5 ps for the HBT. A 130 nm NMOS peaks on the order of 100 GHz, and only at its best bias, and logic gates never operate there. A better yardstick for logic is the **FO4 delay**, the delay of an inverter driving four copies of itself: about 45 ps in 130 nm CMOS. Figure 1.6 puts these times next to the 200 ps UI.

Why CML with HBTs. The fast blocks (final mux, driver, CTLE, samplers, clock buffers) use **current-mode logic**: a differential pair steers a constant tail current into one of two load resistors. The swing is small (a few hundred millivolts), the supply current is constant, so little switching

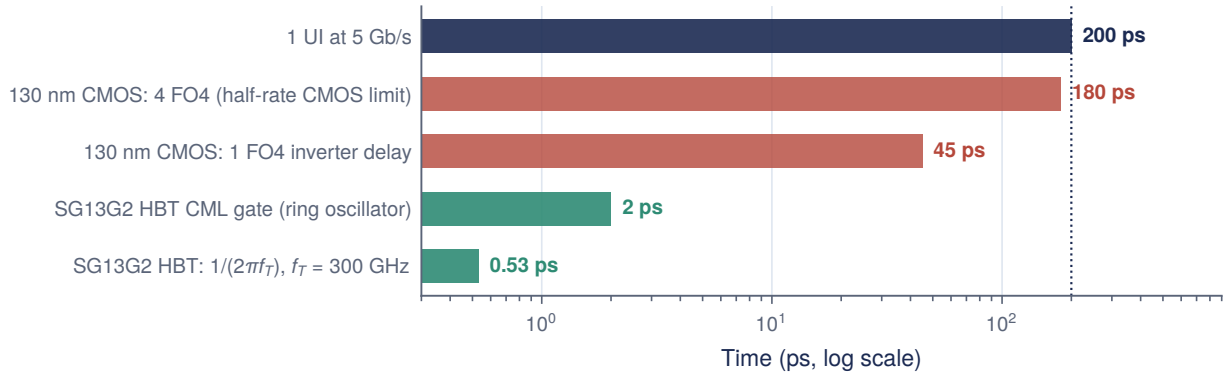


Figure 1.6. A 200 ps UI is about four CMOS gate delays, but about a hundred HBT gate delays. Half-rate CMOS circuits need a bit time of roughly 3–4 FO4 delays, so 130 nm CMOS has almost no margin at 5 Gb/s. The SG13G2 CML ring-oscillator gate delay of about 2 ps [12] is a best case; a loaded CML gate is several times slower and still fits comfortably. FO4 is a typical value for the node.

noise is injected, and an HBT pair switches fully with only about $4V_T \approx 100\,\text{mV}$ of input, because $g_\text{m} = I_C/V_T$ is several times a MOSFET's per unit current (Sections 7.4 and 9.4).

The back end. SG13G2 has seven aluminum metal layers (Figure 1.7): thin Metal1–Metal5 for dense wiring, and thick **TopMetal1** (2 µm) and **TopMetal2** (3 µm), whose low resistance suits spiral inductors (LC VCO, inductive peaking), on-chip transmission lines and supply rails. A MIM capacitor of roughly 1.5 fF/µm^2 (check the PDK) sits below TopMetal1.

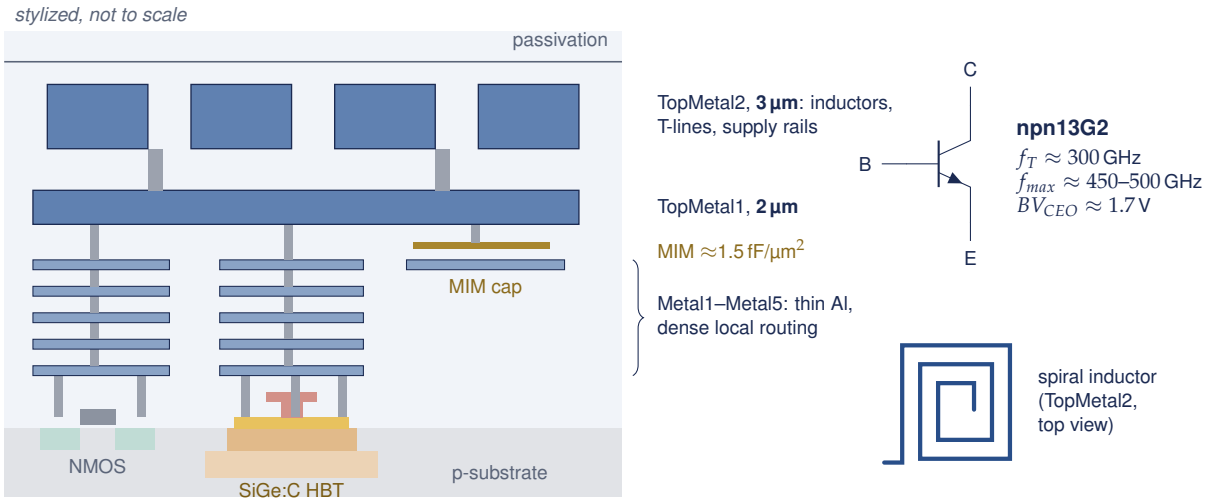


Figure 1.7. The SG13G2 stack: fast HBTs at the bottom, thick aluminum on top. Stylized cross-section (not to scale) with an NMOS, a SiGe:C HBT, five thin metals, the MIM capacitor and the two thick top metals; right, the HBT symbol with its headline figures and a spiral inductor drawn in TopMetal2. Values are approximate; use the PDK documentation [1] for design.

Supplies. CMOS logic (slow SER/DES stages, PRBS, digital CDR loop) runs from 1.2 V. HBT CML stacks a load, a switching pair and a tail source, so it uses a separate analog supply, nominally 2.5 V, to be confirmed in Phase 0.

! Respect BV_{CEO}

The npn13G2 breakdown voltage BV_{CEO} is only about 1.7 V. On a 2.5 V supply, an HBT that sees the whole supply (at power-up, or when a tail source saturates) can break down. Check every V_{CE} in every corner and transient, not just at the operating point.

▶ What this means for your design

Use HBTs where the UI is short (the last mux stage, the driver, the CTLE, the samplers, the VCO and the clock buffers), and CMOS where the rate is low and density matters (the slow serializer and deserializer stages, the PRBS logic and the CDR filter). Put inductors and long high-speed routes on the thick top metals.

1.6 How to use the rest of the book

Chapter 2 builds the vocabulary of links (pulse response, eye, BER), Chapter 3 explains the 17.4 dB channel and Chapter 4 turns pulse responses into link budgets. Part II (Chapters 5 to 8) opens the closed eye with equalization. Part III (Chapters 9 and 10) designs the transmitter and receiver circuits. Part IV (Chapters 11 to 13) handles jitter and clocking. Part V (Chapters 14 to 16) assembles the architecture, runs the open-source flow and provides the labs.

KEY TAKEAWAYS

- Parallel buses stop scaling because of skew, pin count and switching noise; a serial link sends one fast differential stream and recovers the clock from the data.
- A SerDes link is a serializer, an FFE driver, a lossy channel, a CTLE, samplers with a DFE, a deserializer, a PLL and a CDR.
- In Open-SERDES, 16 bits at 312.5 Mb/s become 5 Gb/s NRZ with a 200 ps UI and a 2.5 GHz Nyquist frequency, using a half-rate 2.5 GHz clock.
- 5 Gb/s NRZ matches PCIe 2.0, USB 3.2 Gen 1 and 5GBASE-KR, and the 17.4 dB reference channel forces real equalization.
- The targets are BER $\leq 10^{-12}$ over the REF channel with a 3-tap FFE, a 0–12 dB CTLE, a 2-tap DFE, an LC PLL and a PI-based CDR.
- SG13G2 HBTs ($f_T \approx 300$ GHz) far outrun 130 nm CMOS (FO4 ≈ 45 ps), so the fast blocks use HBT CML; $BV_{CEO} \approx 1.7$ V limits the stack, and the thick top metals carry inductors and lines.

CHAPTER 2

Signals, Systems and the Language of Links

Five ideas carry the whole book: the unit interval, the pulse response, the cursors, the eye diagram and the bit-error rate.

IN THIS CHAPTER YOU WILL LEARN

- describe an NRZ stream by its unit interval, bit rate and Nyquist frequency, and explain why most of its energy lies below f_N;
- generate PRBS test patterns with an LFSR and know what PRBS7 and PRBS31 stress;
- predict how a simple low-pass channel slows edges, using impulse, step and frequency responses;
- build a received waveform from shifted pulse responses and read the main, pre- and post-cursors;
- read an eye diagram (height, width, crossing, slope) and turn an eye opening and an rms noise into a BER.

Here the channel is deliberately simple, a one-pole RC filter or a short line; the realistic REF channel waits for Section 3.7, and the tools carry over unchanged. This chapter is standard background, covered in more depth in Palermo's ECEN 720 lectures [4, 13] and in Dally and Poulton [14].

2.1 NRZ signaling: bits, symbols and the unit interval

A link carries **bits** $b_k \in \{0, 1\}$. The simplest way to put them on a wire is **non-return-to-zero** (NRZ) signaling, also called PAM2: each bit becomes a **symbol** $a_k = 2b_k - 1 \in \{-1, +1\}$, and the transmitter holds the voltage $A\,a_k$ for one bit period (Figure 2.1). In Open-SERDES, $A = 0.5\,\mathrm{V}$ ($1\,\mathrm{V_{ppd}}$). One symbol lasts one **unit interval** (UI). NRZ sends one bit per symbol, so the **bit rate** f_b equals the **baud rate**; PAM4 sends two (Section 4.5).

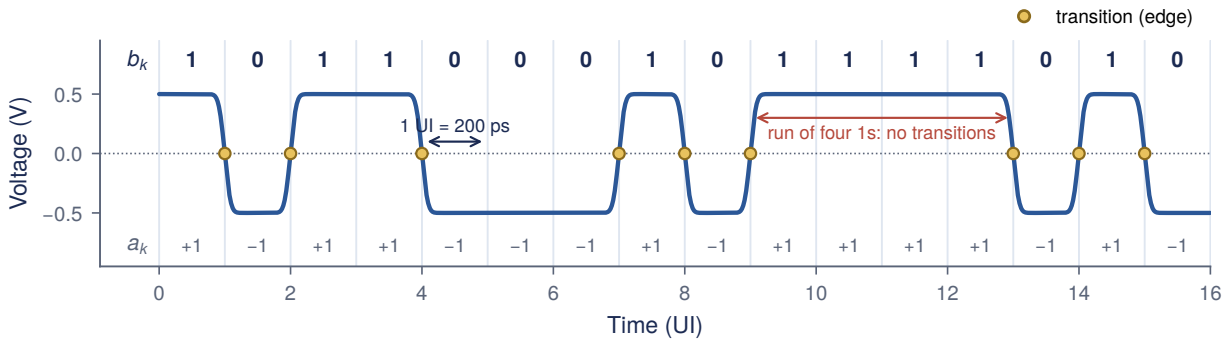


Figure 2.1. An NRZ waveform is a sequence of $\pm A$ levels, one per UI. Each bit b_k maps to a symbol $a_k = \pm 1$ held for $T_b = 200$ ps. Gold dots mark the transitions that clock recovery feeds on; a run of identical bits carries no timing information.

The fastest NRZ pattern, `1010...`, is a square wave with a period of $2T_b$. Its fundamental frequency is the **Nyquist frequency**, the single most important frequency in link design.

KEY EQUATION · Unit interval and Nyquist frequency

$$T_b = 1\,\mathrm{UI} = \frac{1}{f_b}, \qquad f_N = \frac{1}{2T_b} = \frac{f_b}{2} \tag{2.1}$$

For Open-SERDES: $f_b = 5\,\mathrm{Gb/s}$, $T_b = 200\,\mathrm{ps}$ and $f_N = 2.5\,\mathrm{GHz}$.

The spectrum of random NRZ. An NRZ waveform is a sum of rectangular pulses,

$$x(t) = A\sum_n a_n\, g(t - nT_b),$$

where $g(t)$ is a 1-UI rectangle with Fourier transform $G(f) = T_b\,\mathrm{sinc}(fT_b)\,e^{-j\pi fT_b}$ and $\mathrm{sinc}(u) = \sin(\pi u)/(\pi u)$. If the symbols are independent and equally likely to be ± 1, then $\mathrm{E}[a_n a_m]$ is 1 for $n = m$ and 0 otherwise. The cross terms average away, and each pulse contributes its own energy spectrum $A^2|G(f)|^2$ once every T_b seconds. The power spectral density (PSD) is therefore $A^2|G(f)|^2/T_b$:

KEY EQUATION · PSD of random NRZ

$$S_x(f) = A^2\,T_b\,\mathrm{sinc}^2(fT_b), \qquad \text{nulls at } f = f_b, 2f_b, \ldots \tag{2.2}$$

Figure 2.2a plots Equation (2.2). The main lobe runs from DC to the first null at $f_b = 5\,\mathrm{GHz}$, and at f_N the PSD is $\mathrm{sinc}^2(0.5) = (2/\pi)^2$, or −3.9 dB. Integrating the PSD (Figure 2.2b) shows that 77% of the power lies below f_N and 90% below f_b.

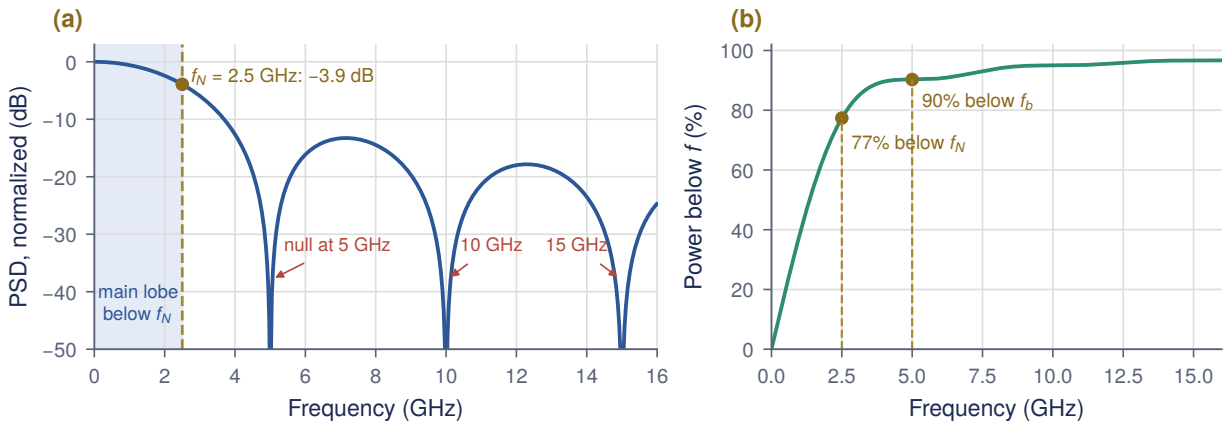


Figure 2.2. Random NRZ puts most of its power below the Nyquist frequency. (a) The PSD follows $\mathrm{sinc}^2(fT_b)$, with nulls at multiples of $f_b = 5$ GHz and only 3.9 dB of roll-off at $f_N = 2.5$ GHz. (b) Cumulative power: 77% lies below f_N and 90% below f_b. This is why channel loss is always quoted at f_N.

WORKED EXAMPLE 2.1 · Where the power of a 5 Gb/s stream sits

The fraction of power below f_N is $2\int_0^{1/2}\mathrm{sinc}^2(u)\,du = 0.774$, so an ideal low-pass filter at 2.5 GHz would still pass three quarters of the power. The REF channel's 17.4 dB at 2.5 GHz, a factor of 7.4 in amplitude, attacks exactly that band.

▶ What this means for your design

Treat $f_N = 2.5\,\text{GHz}$ as the frequency your link lives at: quote channel loss there, and give every block enough bandwidth around it (Section 2.5). The energy above f_b is under 10% and mostly edge sharpness.

2.2 PRBS test patterns

Testing needs data that looks random but is reproducible, so the receiver can check every bit. A **pseudo-random bit sequence** (PRBS) does this. It comes from a **linear-feedback shift register** (LFSR): a chain of n flip-flops whose input is the XOR of two of its outputs. Figure 2.3 shows PRBS7, defined by the polynomial $x^7 + x^6 + 1$: the new bit is stage 6 XOR stage 7. With the right taps, the register steps through all $2^n - 1$ non-zero states before it repeats, producing a **maximal-length sequence**.

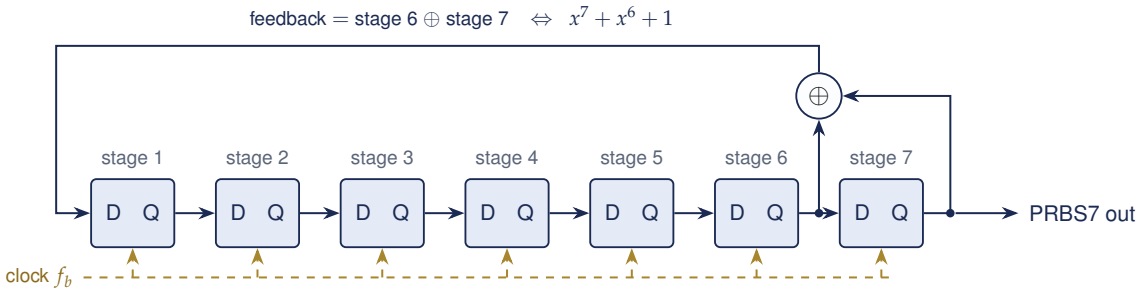


Figure 2.3. A 7-stage LFSR generates PRBS7. Each clock edge shifts the bits right and feeds stage 6 XOR stage 7 into stage 1; from any non-zero state the output repeats after $2^7 - 1 = 127$ bits. A 31-stage register with taps 28 and 31 gives PRBS31.

A PRBS-n sequence has a few properties worth remembering (Figure 2.4):

- **Period** $2^n - 1$ bits, during which every n-bit word except all zeros appears exactly once.
- **Nearly balanced:** 2^{n-1} ones and $2^{n-1} - 1$ zeros (64 and 63 for PRBS7).
- **Run lengths:** the longest run is n ones and $n - 1$ zeros. Half of all runs have length 1, a quarter length 2, and so on, like a fair coin.
- **Line spectrum:** because the pattern repeats, its spectrum is a set of lines spaced $f_b/(2^n - 1)$ apart under the same sinc^2 envelope as random NRZ.

Choosing a PRBS. PRBS7 repeats every 25.4 ns and never holds a level longer than 7 UI: ideal for bring-up. PRBS31 ($x^{31} + x^{28} + 1$) has runs of up to 31 bits and a period of 0.43 s; its low-frequency content stresses AC coupling (baseline wander), long ISI tails and the CDR's ability to hold phase without transitions, so it is the compliance pattern. Model work uses PRBS13 (8191 bits), far longer than the channel's ISI memory. Real protocols also *encode* the data: **8b/10b** (PCIe 1/2, USB 3.2 Gen 1, SATA) maps each byte to 10 bits with at most five equal bits in a row and long-term DC balance, at 25% overhead.

! A short pattern can hide a long problem

If the link remembers more bits than the pattern varies over, the eye looks too good: PRBS7 can pass where PRBS31 fails on baseline wander or a long reflection. Confirm designs with

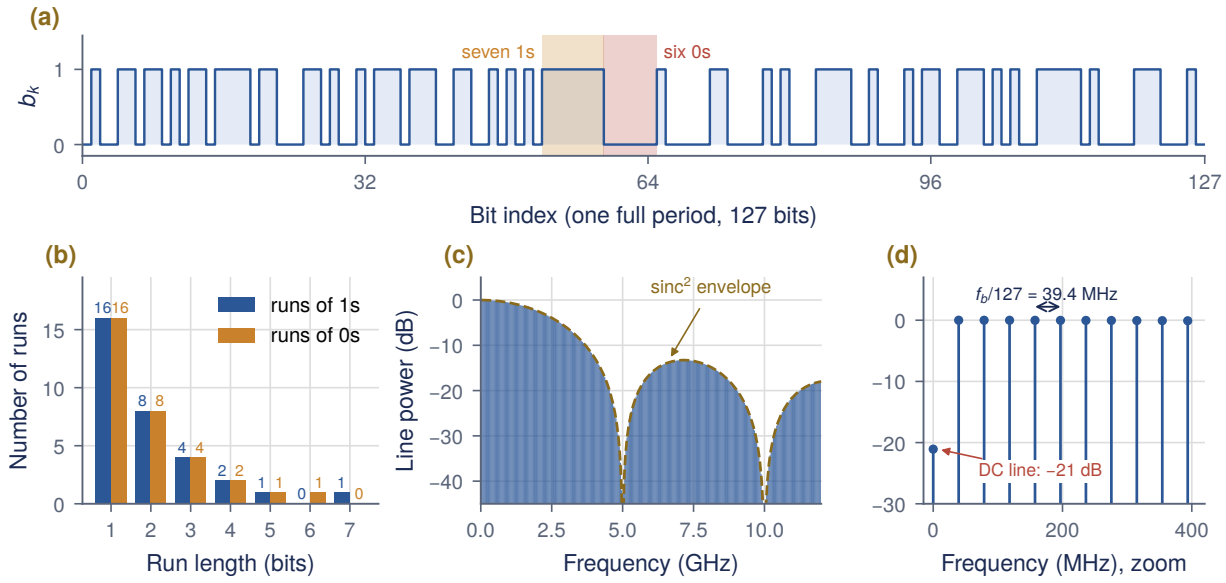


Figure 2.4. PRBS7 behaves like random data with short memory. (a) One full period of 127 bits, with the longest runs (seven 1s, six 0s) shaded. (b) Run-length histogram: 32 runs of length 1, 16 of length 2, halving at each step, as for coin flips. (c) The line spectrum follows the sinc^2 envelope of Figure 2.2. (d) Zoom near DC: lines every $f_\mathrm{b}/127 = 39.4$ MHz, with an almost empty DC line because the pattern is balanced.

a pattern longer than every memory in the link.

2.3 Linear systems in one section

Between the TX output and the RX sampler, the link is close to a **linear time-invariant** (LTI) system, fully described by its **impulse response** $h(t)$, the output for a very short unit-area pulse. For any input $x(t)$ the output is the **convolution**

$$y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau, \qquad Y(f) = X(f)\, H(f), \tag{2.3}$$

where the **frequency response** $H(f)$ is the Fourier transform of $h(t)$. Convolution becomes multiplication, so cascaded blocks multiply. Gains are quoted in decibels, $|H|_\mathrm{dB} = 20\log_{10}|H(f)|$: −3 dB is a factor 0.707, −6 dB one half, −20 dB one tenth, and in a cascade the dB values add. A **Bode plot** shows $|H|_\mathrm{dB}$ against log frequency.

The one-pole system. A resistor driving a capacitor is the simplest low-pass filter and a good first model of a slow channel or a CML stage with load $R_\mathrm{L}C_\mathrm{L}$ (Figure 2.5):

KEY EQUATION · One-pole low-pass filter

$$H(f) = \frac{1}{1 + jf/f_{3\mathrm{dB}}}, \quad h(t) = \frac{1}{\tau}e^{-t/\tau}, \quad s(t) = 1 - e^{-t/\tau}, \quad t_r^{10\text{–}90} = \tau \ln 9 \approx \frac{0.35}{f_{3\mathrm{dB}}} \tag{2.4}$$

Here $\tau = RC = 1/(2\pi f_{3\mathrm{dB}})$, $s(t)$ is the unit step response (for $t \geq 0$) and t_r is the 10–90% rise time. Above $f_{3\mathrm{dB}}$ the gain falls by 20 dB per decade.

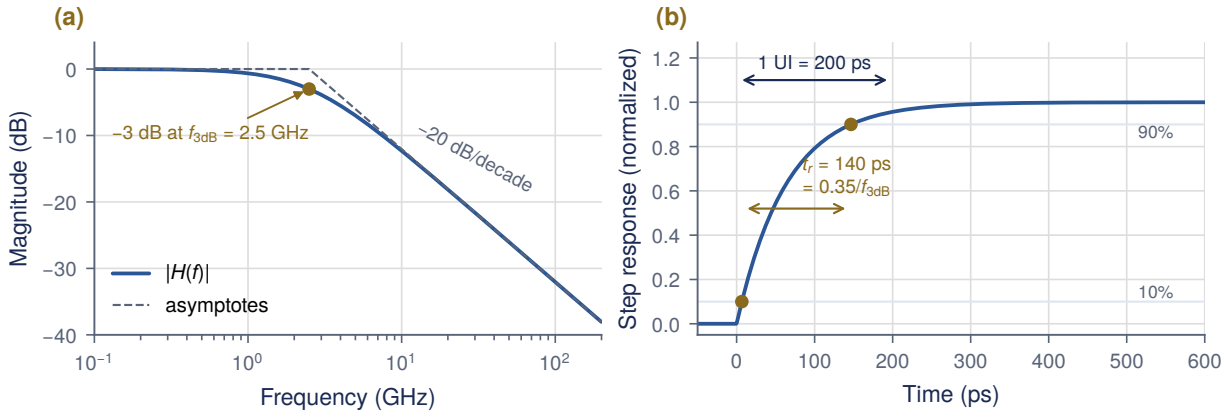


Figure 2.5. Bandwidth and rise time are two views of the same pole. (a) A one-pole response with $f_{3\mathrm{dB}} = 2.5$ GHz: –3 dB at the corner, then –20 dB per decade. (b) Its step response rises from 10% to 90% in 140 ps $= 0.35/f_{3\mathrm{dB}}$, which is 0.7 UI at 5 Gb/s.

WORKED EXAMPLE 2.2 · What bandwidth keeps ISI small at 5 Gb/s?

For a bit to barely affect the next, the output must almost settle within one UI. One UI after a step, a one-pole system still has an error of $e^{-T_\mathrm{b}/\tau}$. Asking for at most 5%:

- $e^{-T_\mathrm{b}/\tau} \le 0.05 \;\Rightarrow\; T_\mathrm{b}/\tau \ge \ln 20 = 3.0 \;\Rightarrow\; \tau \le 66.8\,\mathrm{ps}$;
- $f_{3\mathrm{dB}} = 1/(2\pi\tau) \ge 3.0/(2\pi \times 200\,\mathrm{ps}) = 2.38\,\mathrm{GHz}$;
- the rise time is then $0.35/2.38\,\mathrm{GHz} = 147\,\mathrm{ps} \approx 0.73\,\mathrm{UI}$.

A one-pole bandwidth close to the Nyquist frequency, $f_{3\mathrm{dB}} \approx f_\mathrm{N} = 2.5\,\mathrm{GHz}$, keeps the ISI to a few percent. Section 2.5 turns this into an exact eye opening.

2.4 The pulse response

A link never sends impulses; it sends 1-UI rectangles. The response to one isolated rectangle of 1 V and width T_b is the **pulse response** $p(t)$, the most useful single picture of a link:

$$p(t) = h(t) * \Pi_{T_\mathrm{b}}(t) = \int_{t-T_\mathrm{b}}^{t} h(\tau)\,d\tau = s(t) - s(t-T_\mathrm{b}), \tag{2.5}$$

where $\Pi_{T_\mathrm{b}}(t)$ is 1 for $0 \le t < T_\mathrm{b}$: a step up, then a step down one UI later. An NRZ waveform is a sum of shifted, signed rectangles and the channel is linear, so the received signal is a sum of shifted, signed pulse responses (Figure 2.7). Sample it once per UI at $t_\mathrm{s} + nT_\mathrm{b}$, with t_s usually at the peak of $p(t)$. The samples of the pulse response at integer UI offsets are the **cursors** $p_k = p(t_\mathrm{s} + kT_\mathrm{b})$ (Figure 2.6). The **main cursor** p_0 carries the wanted bit. **Post-cursors** p_k, $k > 0$, are the tail of a pulse sent k bits *earlier*. **Pre-cursors** p_k, $k < 0$, are the leading edge of a pulse sent $|k|$ bits *later*.

KEY EQUATION · Superposition of pulse responses

$$y(t) = A\sum_n a_n\,p(t-nT_\mathrm{b}) \quad\Longrightarrow\quad y_n = y(t_\mathrm{s}+nT_\mathrm{b}) = \underbrace{A\,a_n\,p_0}_{\text{wanted}} + \underbrace{A\sum_{k\neq 0} a_{n-k}\,p_k}_{\text{intersymbol interference (ISI)}} \tag{2.6}$$

The channel acts like an FIR filter whose taps are the cursors; everything except the main cursor is **intersymbol interference** (ISI), the echo of neighboring bits at the sampling instant.

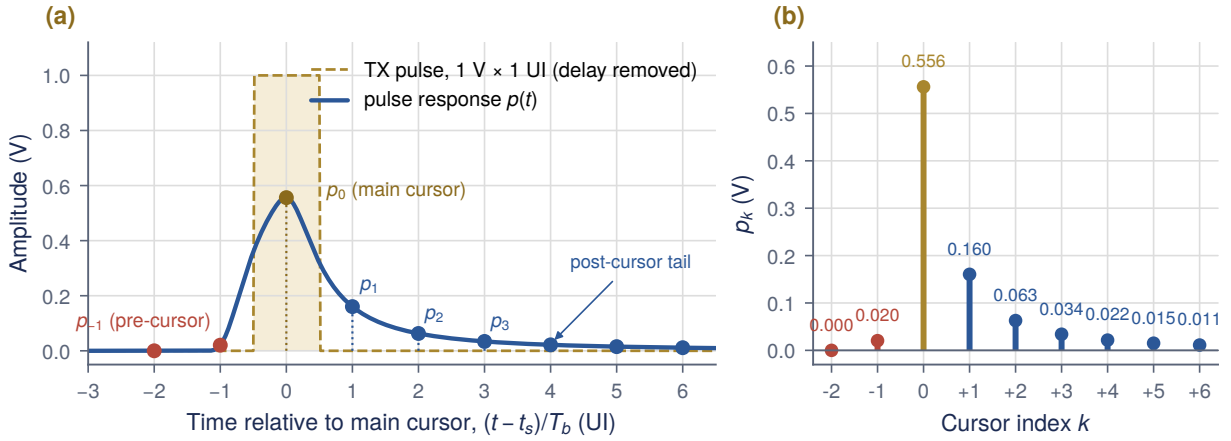


Figure 2.6. The pulse response and its cursors. (a) A 1 V, 1-UI pulse through a mild 10 dB-at-Nyquist line (0.57 m of the loss family) comes out lower, wider and with a long tail. Sampling it once per UI around the peak gives the cursors. (b) The main cursor $p_0 = 0.556$ is the useful signal; $p_{-1} = 0.020$ is a pre-cursor and $p_1 = 0.160$, $p_2 = 0.063, \ldots$ are post-cursors.

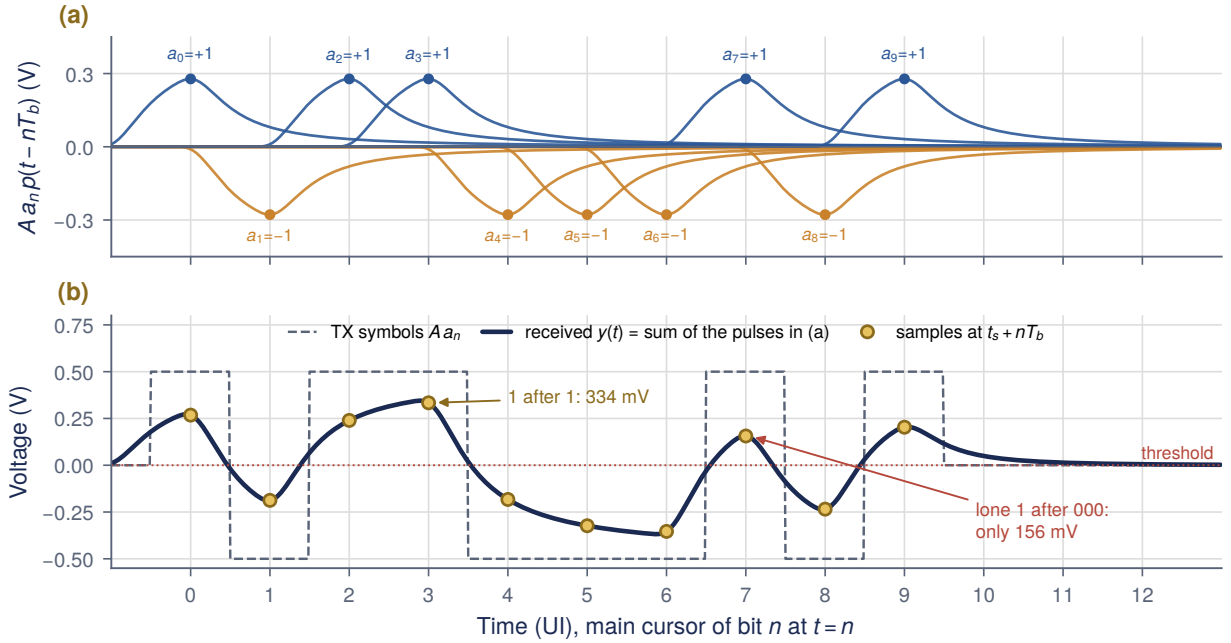


Figure 2.7. A received waveform is a sum of shifted pulse responses. (a) Each bit launches its own pulse $A\,a_n\,p(t - nT_\mathrm{b})$ (blue for $+1$, amber for -1). (b) Their sum is the received signal; gold dots are the samples at $t_\mathrm{s} + nT_\mathrm{b}$. A 1 that follows another 1 gets help from its neighbor's tail (334 mV), while a lone 1 after three 0s is pulled down to 156 mV. The line is idle before bit 0.

WORKED EXAMPLE 2.3 · Reading a sample from the cursors

Bit 7 in Figure 2.7 is a lone 1 after three 0s, followed by a 0. With $A = 0.5\,\mathrm{V}$ and the cursors of Figure 2.6:

- wanted signal: $Ap_0 = 0.5 \times 0.5561 = 278\,\mathrm{mV}$;
- the three previous 0s and the next 0 subtract: $-A(p_1 + p_2 + p_3 + p_{-1}) = -0.5\,(0.1605 +$

$0.0629 + 0.0340 + 0.0205) = -139\,\text{mV}$;

- the older bits ($a_3 = a_2 = a_0 = +1$, $a_1 = -1$) add $0.5\,(0.0216 + 0.0151 - 0.0112 + 0.0087) = 17\,\text{mV}$.

The sample is $278 - 139 + 17 = 156\,\text{mV}$, as simulated: ISI took 44% of the wanted signal.

▶ What this means for your design

The cursors are the channel's fingerprint, and every equalizer in Part II acts on them: the TX FFE reshapes the pulse (Chapter 6), the CTLE sharpens it (Chapter 7) and the DFE subtracts post-cursors (Chapter 8). Faced with a new channel, plot its pulse response first.

>_ Lab 1: pulse response and cursors

`lab1_channel.py` turns the REF channel's S_{21} into a pulse response and its cursors. Lengthen the line and watch p_1/p_0 grow (Section 16.2).

2.5 ISI: how pulse spreading closes the eye

The one-pole channel shows in closed form how bandwidth turns into ISI. Its pulse response peaks at the end of the bit, $t = T_\text{b}$. Sampling there and writing $r = e^{-T_\text{b}/\tau}$ gives $p_0 = 1 - r$ and $p_k = (1 - r)\,r^k$ for $k \geq 1$, with no pre-cursors. The post-cursors form a geometric series that sums to r. The worst case for a 1 is that every earlier bit is a 0, so each tail pulls it down. The smallest 1 is then $A(p_0 - \sum_{k\geq 1} p_k)$, and by symmetry the eye opening at the sampling instant is

KEY EQUATION · Worst-case eye of a one-pole channel

$$\text{EH} = 2A\left(p_0 - \sum_{k\geq 1} p_k\right) = 2A\left(1 - 2e^{-T_\text{b}/\tau}\right), \qquad \frac{T_\text{b}}{\tau} = 2\pi f_{3\text{dB}} T_\text{b} = \pi \frac{f_{3\text{dB}}}{f_\text{N}} \tag{2.7}$$

Figure 2.8 shows the pulse and the eye for three time constants. With $\tau = 0.2\,\text{UI}$ ($f_{3\text{dB}} = 4.0\,\text{GHz}$) the eye is 98.7% open. At $\tau = 0.5\,\text{UI}$ (1.6 GHz) it is 73% open, and at $\tau = 1\,\text{UI}$ (0.8 GHz) only 26%. The eye closes completely when $r = 1/2$, that is $\tau = T_\text{b}/\ln 2 = 1.44\,\text{UI}$, or $f_{3\text{dB}} = 552\,\text{MHz}$. For the bandwidth of Worked Example 2.2, $f_{3\text{dB}} = f_\text{N}$, the exponent is π and $\text{EH} = 2A(1 - 2e^{-\pi}) = 0.91 \times 2A$.

Subtracting the magnitudes of all other cursors from the main cursor works for any channel; this is **peak-distortion analysis** (Section 4.2). For the 10 dB line of Figure 2.6 it gives 230 mV out of 1 V; for the REF channel it is negative: the eye is closed.

▶ Design rule of thumb

Each one-pole block in the data path needs $f_{3\text{dB}} \gtrsim f_\text{N} = 2.5\,\text{GHz}$ to cost less than about 10% of eye height. A cascade of several poles has less bandwidth than any single one, so in practice each CML stage of Open-SERDES should reach 0.7–1 f_b (3.5–5 GHz).

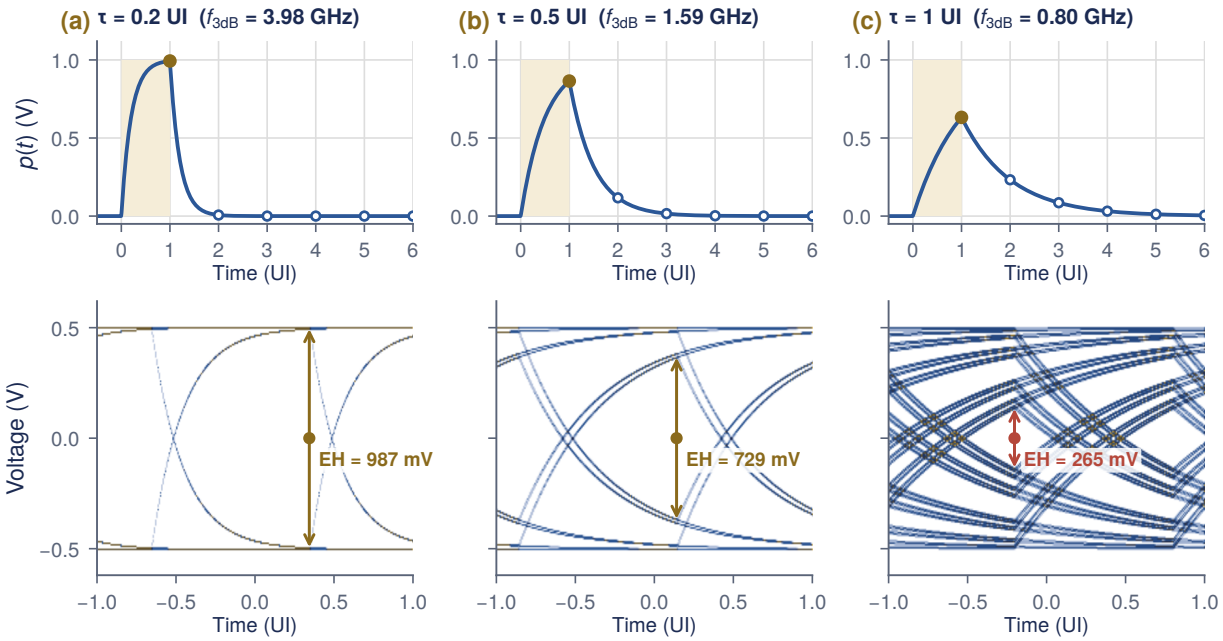


Figure 2.8. Slower channels spread each pulse over more UIs and close the eye. Top: one-pole pulse responses for $\tau = 0.2$, 0.5 and 1 UI; filled dots are the main cursor, open dots the post-cursors. Bottom: PRBS7 eyes through the same channels ($2A = 1$ V). The measured openings, 987, 729 and 265 mV, match $2A(1 - 2e^{-T_b/\tau})$ from Equation (2.7). The arrow sits at the best sampling instant, which for a one-pole channel is the pulse peak at the end of the bit, not the visual center of the eye.

2.6 Eye diagrams

An **eye diagram** cuts the received waveform into 2-UI windows aligned to the bit clock and overlays them (Figure 2.9). Every bit pattern draws its own trace, so the overlay shows at a glance how much the patterns differ at each instant.

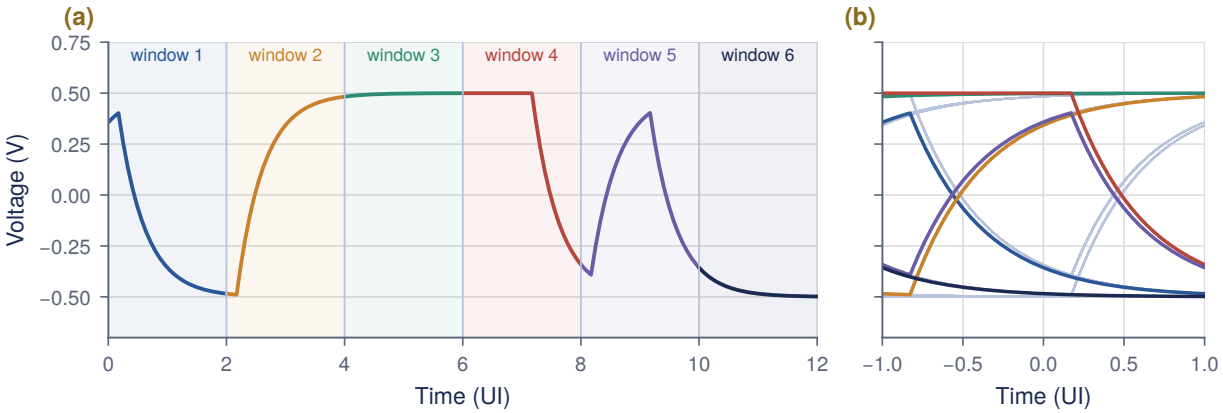


Figure 2.9. An eye diagram folds the waveform every 2 UI. (a) Twelve UI of PRBS7 after a one-pole channel ($\tau = 0.45$ UI), cut into six 2-UI windows. (b) The same segments, same colors, overlaid on 600 others (grey).

Figure 2.10 names the parts you will measure throughout this book.

- **Eye height** (EH): the vertical opening at the sampling instant, between the lowest 1 and the highest 0. It must beat noise, offset and sampler sensitivity.
- **Eye width** (EW): the horizontal opening at the threshold, between adjacent crossings. It must beat clock jitter and the sampler aperture.

- **Crossing points**: where edges cross the threshold. Their spread in time is **jitter**; their mean is what the CDR locks to.
- **Optimal sampling point**: the instant and threshold with the most margin both ways.
- **Slope** near the crossing: it converts timing error into voltage error, $\Delta v \approx (dV/dt)\,\Delta t$.

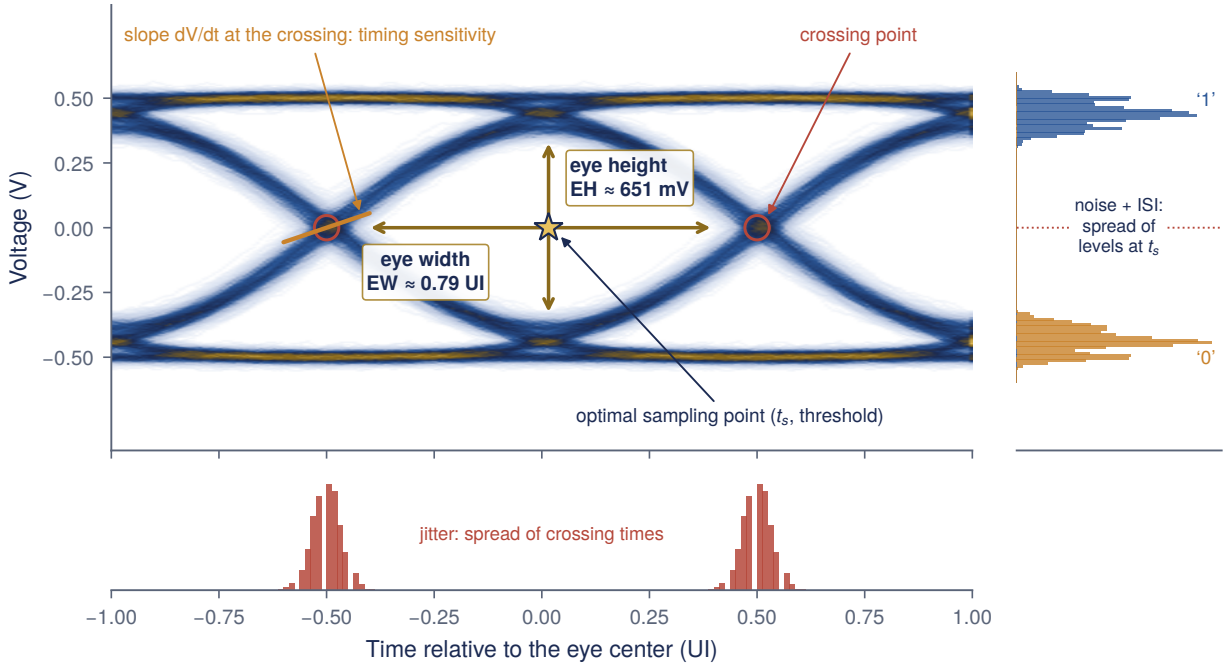


Figure 2.10. Anatomy of an eye diagram. A PRBS7 stream through a smooth band-limited channel, with 6 ps rms edge jitter and 15 mV rms noise. Eye height and eye width are measured between the 0.1% tails of the traces. The right histogram shows the spread of the sampled levels (noise plus ISI) at t_s; the bottom histogram shows the spread of the crossing times (jitter). The orange line is the edge slope, here about 2.8 mV/ps.

With the slope of Figure 2.10, sampling 4 ps off near an edge costs about $2.8 \times 4 \approx 11$ mV. Noise eats the eye vertically and jitter horizontally; the next section asks how much is tolerable.

! An eye is only as deep as the number of bits behind it

A simulated eye of 10^4 bits only shows events more frequent than about 10^{-4}; a 7σ noise peak will almost never appear. An open eye in a short simulation says nothing about 10^{-12}. Section 4.3 computes the eye at that probability directly.

>_ Lab 2: PRBS7 eye through the REF channel

`lab2_prbs_eye.py` draws the PRBS7 eye before and after the REF channel. The output eye is closed (−78 mV): the motivation for Part II (Section 16.3).

2.7 Noise and the bit-error rate

The receiver compares each sample y_n with a threshold at 0, and an error happens when noise pushes a sample across it. With a worst-case eye of height EH, a 1 arrives at least EH/2 above the threshold and a 0 at least EH/2 below it. Add Gaussian noise $n \sim \mathcal{N}(0, \sigma_n^2)$ from the

termination, the front end and the sampler (Figure 2.11a). A 1 is read as a 0 when $n < -\mathrm{EH}/2$:

$$P(\text{error} \mid 1) = \int_{-\infty}^{-\mathrm{EH}/2} \frac{1}{\sqrt{2\pi}\,\sigma_\mathrm{n}} e^{-v^2/(2\sigma_\mathrm{n}^2)}\,dv = Q\!\left(\frac{\mathrm{EH}}{2\sigma_\mathrm{n}}\right),$$
$$Q(x) = \int_{x}^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}}\,du = \tfrac{1}{2}\,\mathrm{erfc}\!\left(\frac{x}{\sqrt{2}}\right). \tag{2.8}$$

By symmetry $P(\text{error} \mid 0)$ is the same. With 1s and 0s equally likely, the BER is the average of the two:

KEY EQUATION · Bit-error rate with Gaussian noise

$$\mathrm{BER} = Q\!\left(\frac{\mathrm{EH}}{2\,\sigma_\mathrm{n}}\right), \qquad \mathrm{SNR} \equiv \left(\frac{\mathrm{EH}}{2\,\sigma_\mathrm{n}}\right)^2 \;\Rightarrow\; \mathrm{BER} = Q\!\left(\sqrt{\mathrm{SNR}}\right) \tag{2.9}$$

The argument of Q is half the eye in units of rms noise. With the worst-case EH, Equation (2.9) is an upper bound, since most patterns leave more margin (Section 4.3 computes the average). Table 2.1 lists common values. Remember $Q = 7.03$ for 10^{-12}: the eye must be at least $2Q = 14.07$ times the rms noise.

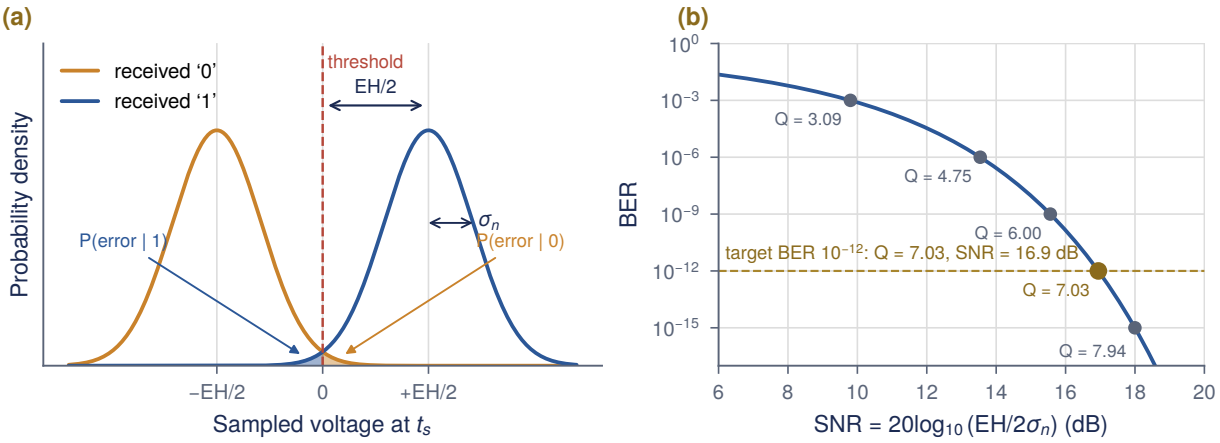


Figure 2.11. BER is the area of a Gaussian tail beyond half the eye. (a) Received 1s and 0s spread by noise of rms σ_n around $\pm\mathrm{EH}/2$; the shaded tails beyond the threshold are errors. (b) BER against $\mathrm{SNR} = 20\log_{10}(\mathrm{EH}/2\sigma_\mathrm{n})$. The curve is a cliff: near 10^{-12} (Q = 7.03, 16.9 dB), one extra dB of SNR buys about three decades of BER.

WORKED EXAMPLE 2.4 · How much noise can the eye tolerate?

The target is $\mathrm{BER} \le 10^{-12}$, so $\mathrm{EH}/(2\sigma_\mathrm{n}) \ge 7.03$ and $\sigma_\mathrm{n} \le \mathrm{EH}/14.07$.

- A sampler with $\sigma_\mathrm{n} = 2\,\mathrm{mV}$ rms needs an eye of at least $14.07 \times 2\,\mathrm{mV} = 28\,\mathrm{mV}$.
- Lab 5 opens the REF channel to $\mathrm{EH} = 143\,\mathrm{mV}$ with an 8 dB CTLE and a 3-tap DFE. The noise budget is then $\sigma_\mathrm{n} \le 143/14.07 = 10.2\,\mathrm{mV}$ rms, for all noise sources together.
- At $\sigma_\mathrm{n} = 15\,\mathrm{mV}$, $Q = 4.77$ and $\mathrm{BER} \approx 9 \times 10^{-7}$: $1.5\times$ the noise costs six decades.

A first look at the bathtub. Jitter moves the crossings, so the BER also depends on *when* we sample. Sweeping the sampling instant across the UI gives a **bathtub curve** (Figure 2.12); its width at 10^{-12} is the usable eye width. With 4 ps rms random jitter (0.02 UI), 0.72 UI remains; doubling the jitter leaves 0.45 UI. Chapter 11 covers this in full (Section 11.3).

Table 2.1. Q values for common BER targets. BER $= Q(x)$; $2Q$ is the eye-to-noise ratio needed.

BER	Q	$2Q$	SNR (dB)
10^{-3}	3.09	6.18	9.8
10^{-6}	4.75	9.51	13.5
10^{-9}	6.00	12.00	15.6
10^{-10}	6.36	12.72	16.1
10^{-12}	**7.03**	**14.07**	**16.9**
10^{-15}	7.94	15.88	18.0

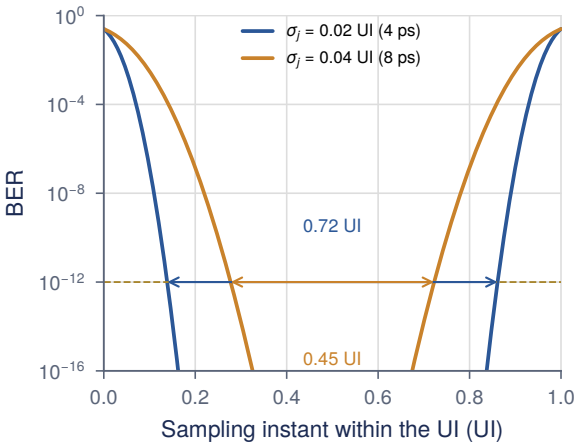


Figure 2.12. Bathtub curves. BER against sampling instant for Gaussian edge jitter of 0.02 and 0.04 UI rms (half the edges are transitions). The opening at 10^{-12} shrinks from 0.72 to 0.45 UI.

! You cannot measure 10^{-12} in a coffee break

Showing BER $< 10^{-12}$ at 95% confidence takes $3/\text{BER} = 3 \times 10^{12}$ error-free bits: ten minutes on the bench, far beyond any simulation. Design with statistical models; verify with PRBS checkers.

2.8 Exercises

2.1 Rates. A 2.5GBASE-KX lane runs at 3.125 Gb/s with 8b/10b. Find T_b, f_N, the first PSD null and the payload rate.

2.2 PRBS9 ($x^9 + x^5 + 1$). Draw its LFSR; give its period, the counts of 1s and 0s, the longest runs and the line spacing at 5 Gb/s.

2.3 A CML stage as a pole. For $R_\text{L} = 200\,\Omega$, $C_\text{L} = 150$ fF, find $f_{3\text{dB}}$, t_r and $1 - 2e^{-T_\text{b}/\tau}$. What C_L costs 10% of the eye?

2.4 Cursors by hand. With the cursors of Figure 2.6 ($p_{-1}..p_3$ = 0.0205, 0.5561, 0.1605, 0.0629, 0.0340) and $A = 0.5$ V, find the sample of the bracketed 1 in `1 1 1 [1] 0` and in `0 0 0 [1] 1`. Compare the worst-case eye from these cursors with the 230 mV of the full tail.

2.5 BER. A receiver sees EH = 60 mV with $\sigma_\text{n} = 4$ mV. Find Q and the BER. What eye height or noise gives 10^{-15}, and how long must the link run error-free to show it?

KEY TAKEAWAYS

- At 5 Gb/s, 1 UI = 200 ps and $f_\text{N} = 2.5$ GHz; 77% of NRZ power lies below f_N.
- PRBS-n (an n-stage LFSR) repeats every $2^n - 1$ bits with runs up to n; PRBS7 for bring-up, PRBS31 for compliance.
- A one-pole block has $t_r \approx 0.35/f_{3\text{dB}}$ and a worst-case eye of $2A(1 - 2e^{-T_\text{b}/\tau})$; keeping ISI small needs $f_{3\text{dB}} \gtrsim f_\text{N}$.
- The received signal is $y(t) = A\sum_n a_n\, p(t - nT_\text{b})$: main cursor plus ISI from the cursors p_k, with worst-case eye $2A(p_0 - \sum_{k\neq 0}|p_k|)$.
- With Gaussian noise, BER $= Q(\text{EH}/2\sigma_\text{n})$, so 10^{-12} needs EH $\geq 14.07\,\sigma_\text{n}$.

CHAPTER 3

The Channel

Copper is not a wire at 5 Gb/s: it is a lossy, reflective transmission line, and its pulse response sets almost every number in the rest of this book.

IN THIS CHAPTER YOU WILL LEARN

- Decide when a trace is a transmission line, and compute Z_0 and the delay from its RLGC parameters.
- Explain why conductor loss grows as $\sqrt{f}$ and dielectric loss as f, and estimate FR-4 loss in dB per inch.
- Predict reflections from terminations, pads, bondwires and via stubs, and read them in a TDR plot.
- Use S-parameters, including the mixed-mode S_{dd21} of a differential pair.
- Turn $S_{21}(f)$ into a causal impulse response and a pulse response.
- Know the Open-SERDES reference channel REF and its cursors.

3.1 What is the channel?

The **channel** is everything between the transmitter's output transistors and the receiver's input transistors. Figure 3.1 follows a bit across a typical board-to-board link. Long traces and cables attenuate high frequencies (**loss**). Short lumps such as bondwires, pads and vias change the impedance and send part of the wave back (**reflections**). Dense pins and connectors couple neighboring lanes (**crosstalk**). We treat each mechanism in turn and then combine them into one transfer function $H_{\mathrm{ch}}(f)$ and one pulse response $p(t)$, the two objects that the link-analysis tools of Chapter 4 work with.

3.2 Transmission lines

3.2.1 When is a wire a transmission line?

A wire is **lumped** when its voltage is the same everywhere at any instant. It becomes a **transmission line** when the signal changes noticeably while the wave travels along it. The usual

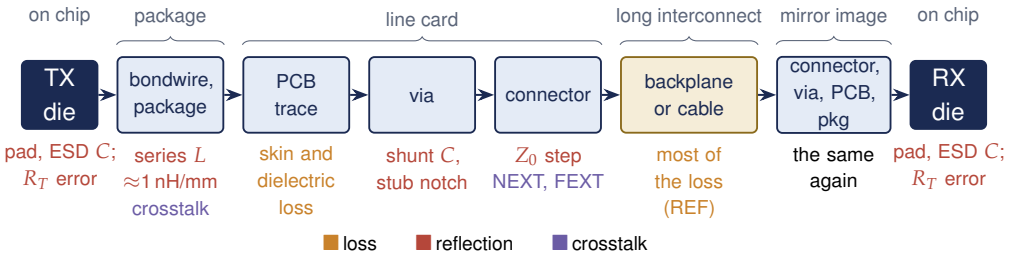


Figure 3.1. The physical path of one bit, with the main impairment of each segment. Long, uniform segments (gold) cause most of the loss; short lumps cause reflections; dense pin fields cause crosstalk. The Open-SERDES reference channel REF lumps the long interconnect into 1 m of FR-4 stripline with a 300 fF pad at each end.

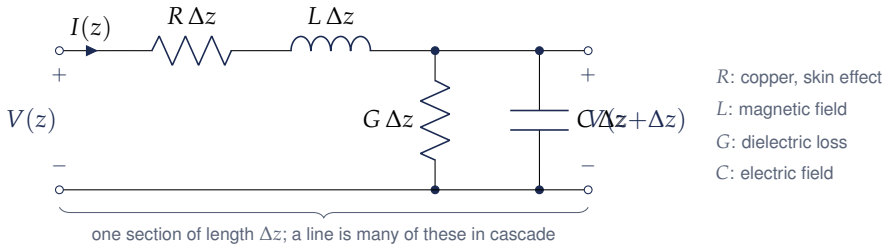


Figure 3.2. One RLGC section of a transmission line. The series elements carry the current and the shunt elements hold the voltage. On a board at 5 Gb/s, $\omega L \gg R$ and $\omega C \gg G$: the line is an LC ladder with small loss terms.

rule compares the one-way delay t_d with the rise time t_r [15, 16]:

$$t_d \gtrsim \frac{t_r}{6} \quad \Longleftrightarrow \quad \ell \gtrsim \frac{t_r\, v}{6}, \tag{3.1}$$

where v is the propagation velocity. Equivalently, a line longer than about a tenth of a wavelength at the highest frequency of interest is a transmission line.

3.2.2 The telegrapher model

A transmission line is a ladder of tiny identical sections (Figure 3.2). Each section of length Δz has a series resistance $R\,\Delta z$ (the copper), a series inductance $L\,\Delta z$ (the magnetic field around the current), a shunt capacitance $C\,\Delta z$ (the electric field to the return plane) and a shunt conductance $G\,\Delta z$ (leakage through the dielectric). KVL and KCL around one section, in the limit $\Delta z \to 0$, give the **telegrapher equations**

$$\frac{\partial V}{\partial z} = -(R + j\omega L)\, I, \qquad \frac{\partial I}{\partial z} = -(G + j\omega C)\, V. \tag{3.2}$$

Eliminating I gives $\partial^2 V/\partial z^2 = \gamma^2 V$, whose solution is a forward and a backward wave, $V(z) = V^+ e^{-\gamma z} + V^- e^{+\gamma z}$.

KEY EQUATION · Propagation constant, impedance and velocity

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)}, \qquad Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \xrightarrow{\text{low loss}} \sqrt{\frac{L}{C}},$$
$$v = \frac{1}{\sqrt{LC}} = \frac{c}{\sqrt{\varepsilon_r}} \quad \text{(lossless)}. \tag{3.3}$$

α (Np/m) is the attenuation and β (rad/m) the phase constant. The **characteristic impedance** Z_0 is the ratio V^+/I^+ of one traveling wave: the impedance the wave sees before it meets the far end, not something an ohmmeter can measure. Inductance takes over from resistance above $f = R/(2\pi L)$. For the REF line ($R_{DC} = 7\,\Omega$/m, $L = 334$ nH/m) that is only 3.3 MHz, so every board trace is an LC line at gigahertz frequencies. Thin on-chip wires have far more resistance per meter and stay RC lines.

WORKED EXAMPLE 3.1 · FR-4 delay and the transmission-line length

Given: FR-4 stripline, $\varepsilon_r = 4.0$, $Z_0 = 50\,\Omega$, and a 5 Gb/s CML edge with $t_r \approx 50$ ps (0.25 UI, a reasonable value for an SG13G2 HBT driver).

Solution: $v = c/\sqrt{4} = 1.50 \times 10^8\,\mathrm{m/s}$, so the delay is 6.67 ps/mm (169 ps per inch); one meter of REF takes 6.67 ns, or 33.4 UI. The line has $L = Z_0/v = 334\,\mathrm{nH/m}$ and $C = 1/(Z_0 v) = 133\,\mathrm{pF/m}$. By Equation (3.1), a trace is a transmission line beyond $t_r v/6 = 1.25\,\mathrm{mm}$. The wavelength rule agrees: $\lambda/10$ at 7.5 GHz, the third harmonic of f_N, is 2.0 mm.

Conclusion: everything off-chip longer than about a millimeter, package traces included, is a transmission line at 5 Gb/s. On-chip routes of a few hundred micrometers are lumped.

3.3 Loss mechanisms

A line loses energy in the copper and in the dielectric. In the low-loss limit ($R \ll \omega L$, $G \ll \omega C$), a first-order expansion of Equation (3.3) splits the attenuation into two independent terms, $\alpha \approx \alpha_R + \alpha_D$.

Skin effect: conductor loss grows as $\sqrt{f}$. At DC the current fills the whole cross-section of the trace. At high frequency, the changing magnetic field inside the conductor induces eddy currents that cancel the current in the middle, and the current crowds into a surface layer of thickness

$$\delta = \sqrt{\frac{\rho}{\pi f \mu_0}}, \tag{3.4}$$

the **skin depth** (ρ is the resistivity, $\mu_0 = 4\pi \times 10^{-7}\,\mathrm{H/m}$). For copper ($\rho = 1.72 \times 10^{-8}\,\Omega\,\mathrm{m}$) at 2.5 GHz, $\delta = 1.32\,\mu\mathrm{m}$. Once δ is below half the conductor thickness, the current flows in a sheet δ thick around the effective perimeter P, so

$$R(f) \approx \frac{\rho}{P\,\delta} = \frac{\sqrt{\pi f \mu_0 \rho}}{P} \propto \sqrt{f}. \tag{3.5}$$

The numerator $R_s = \rho/\delta$ is the **surface resistance** in ohms per square. Figure 3.3 shows both effects for a 5-mil (127 µm) wide, half-ounce (17.5 µm) copper trace. On chip, the SG13G2 thick top metals are aluminum, with $\delta = 1.6\,\mu\mathrm{m}$ at 2.5 GHz, close to half the 3 µm TopMetal2 thickness. That matters for inductors and long clock lines, not for the board channel.

Dielectric loss grows as f. The alternating field also rotates the dipoles in the board material, and each rotation dissipates a little heat. A lossy dielectric acts like a capacitor with a conductance in parallel, $G = \omega C \tan\delta$, where the **loss tangent** $\tan\delta$ is roughly constant over frequency (0.02 for FR-4, 0.002–0.005 for low-loss laminates). Because G grows with ω, dielectric loss grows linearly with frequency. (The loss angle δ is unrelated to the skin depth δ.)

KEY EQUATION · Conductor and dielectric attenuation

$$\alpha_R = \frac{R(f)}{2Z_0}, \qquad \alpha_D = \frac{GZ_0}{2} = \frac{\pi f \sqrt{\varepsilon_r}\,\tan\delta}{c}\ [\mathrm{Np/m}], \qquad \mathrm{IL}(f) = 8.686\,(\alpha_R + \alpha_D)\,\ell\ [\mathrm{dB}]. \tag{3.6}$$

Figure 3.4 plots both terms in dB per inch (1 Np = 8.686 dB). Conductor loss dominates only below about 0.5 GHz. At the 2.5 GHz Nyquist frequency of Open-SERDES, dielectric loss is already more than twice as large, and the gap widens because f outruns $\sqrt{f}$.

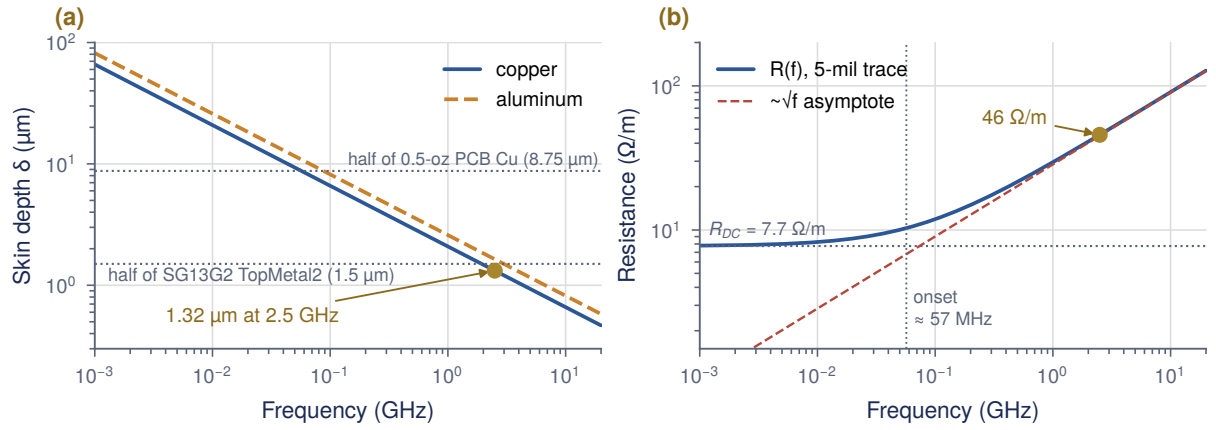


Figure 3.3. Skin effect makes resistance grow as $\sqrt{f}$. (a) Skin depth falls as $1/\sqrt{f}$; below half the conductor thickness (dotted lines) the current no longer uses the whole cross-section. (b) Resistance of a 5-mil, half-ounce stripline: flat at $R_{\mathrm{DC}} = 7.7\,\Omega$/m up to about 57 MHz, then $\propto \sqrt{f}$, reaching about 46 Ω/m at the 2.5 GHz Nyquist frequency.

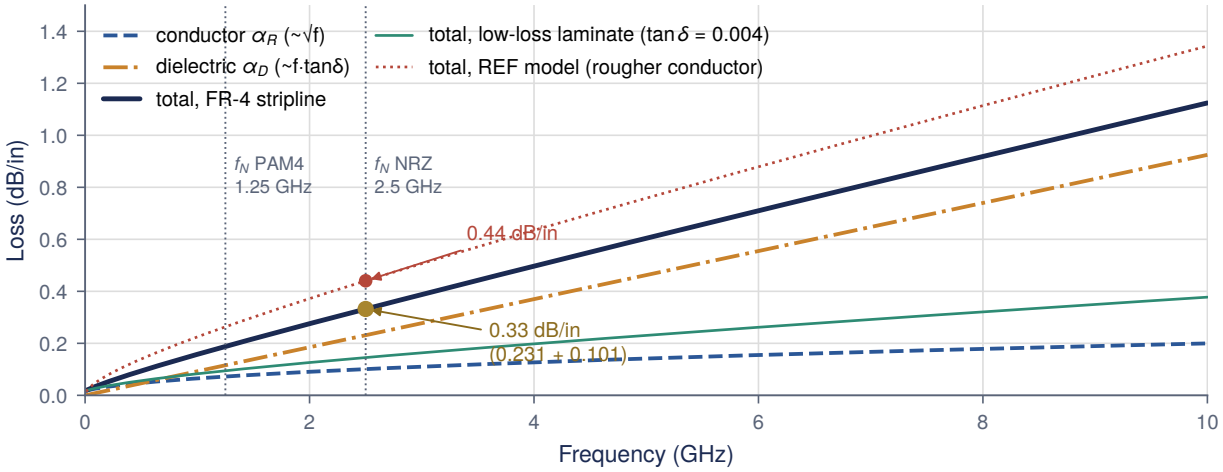


Figure 3.4. Above about 0.5 GHz, dielectric loss dominates FR-4. Conductor loss (dashed) grows as $\sqrt{f}$, dielectric loss (dash-dot) as f. At the NRZ Nyquist frequency a smooth 5-mil stripline loses 0.33 dB/in; a low-loss laminate cuts that to 0.15 dB/in. The REF model (dotted) has a rougher, lossier conductor and loses 0.44 dB/in.

WORKED EXAMPLE 3.2 · Loss per inch and reach for a 10 dB budget

Given: the 5-mil, half-ounce, 50 Ω FR-4 stripline of Figure 3.3 ($\varepsilon_r = 4.0$, $\tan\delta = 0.02$) at $f_{\mathrm{N}} = 2.5\,\mathrm{GHz}$.

Dielectric: $\alpha_D = \pi(2.5\times10^9)(2)(0.02)/(3\times10^8) = 1.047\,\mathrm{Np/m}$: 9.10 dB/m or **0.231 dB/in**.

Conductor: $R_s = \rho/\delta = 13.0\,\mathrm{m\Omega}$ per square. With $P \approx 2(w+t) = 289\,\mu\mathrm{m}$, $R \approx 45\,\Omega/\mathrm{m}$, so $\alpha_R = 45/(2\cdot50) = 0.45\,\mathrm{Np/m}$ or **0.10 dB/in**. Copper roughness and return current in the planes add tens of percent in real boards.

Total and reach: $\alpha \approx 0.33\,\mathrm{dB/in}$. A 10 dB budget allows $10/0.33 = 30\,\mathrm{in}$ (76 cm) of bare trace, or 21 in (54 cm) if 3 dB is set aside for two packages and a connector pair. The REF model reaches 10 dB at 0.572 m (22.5 in, the 10 dB entry of `LOSS_FAMILY`) because its conductor term is twice as large (0.21 dB/in) and it includes the pads.

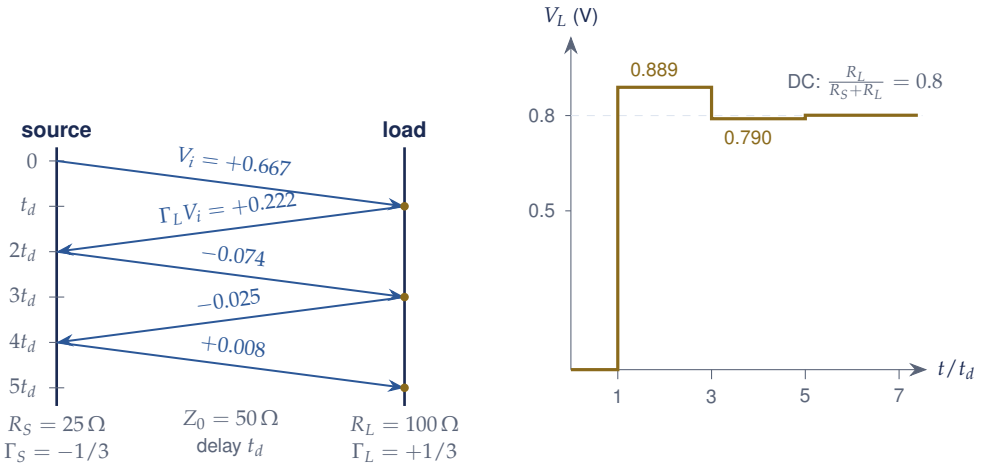


Figure 3.5. A lattice (bounce) diagram tracks every reflection. Left: a 1 V step from a 25 Ω source into a 50 Ω line ending in 100 Ω. Each arrow is one traveling wave with its amplitude. Right: the load voltage overshoots to 0.889 V, undershoots to 0.790 V and settles to 0.8 V, one step every $2t_d$. In a link, the same echoes appear as bumps in the pulse response many UI after the main cursor.

▶ Why everyone quotes the loss at Nyquist

NRZ puts most of its power below $f_N = f_b/2$ (Section 2.1), and the fastest pattern, 1010…, is a square wave whose fundamental is exactly f_N. The loss at f_N therefore predicts how much the channel closes the eye, and how much equalization you need, far better than the length or the loss at f_b. For Open-SERDES, budget *insertion loss at 2.5 GHz*: REF has 17.4 dB. Up to about 10 dB, a small TX FFE or a CTLE is enough; from 10 to 25 dB you need a CTLE and a DFE (Chapter 5).

3.4 Reflections and termination

A wave on a line of impedance Z_0 that meets a load Z_L must satisfy $V = Z_L I$ at the load and $V/I = Z_0$ in the wave. One wave cannot do both, so a reflected wave is launched back toward the source, with relative amplitude given by the **reflection coefficient**.

KEY EQUATION · Reflection coefficient and return loss

$$\Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0}, \qquad \Gamma_S = \frac{R_S - Z_0}{R_S + Z_0}, \qquad V_i = V_S \frac{Z_0}{R_S + Z_0}, \qquad \mathrm{RL} = -20\log_{10}|\Gamma|. \tag{3.7}$$

A matched load gives $\Gamma = 0$, an open $+1$ and a short -1. V_i is the wave launched by a source of open-circuit voltage V_S and resistance R_S. The **return loss** RL is $|\Gamma|$ in positive decibels: larger is better. With both ends mismatched, the wave bounces back and forth. The **lattice diagram** of Figure 3.5 tracks each bounce, with position across and time down. Each arrival at the load adds $(1 + \Gamma_L)$ times the incident wave, and the load voltage converges to the DC divider value only after several round trips of $2t_d$.

Termination choices. **Source termination** ($R_S = Z_0$, open load) absorbs reflections at the driver. **Load termination** ($R_L = Z_0$) absorbs the wave at the receiver but only works if nothing in between reflects. **Double termination** ($R_S = R_L = Z_0$) absorbs every returning echo at either end. It halves the received swing, but echoes are worse than a smaller signal, so serial links use

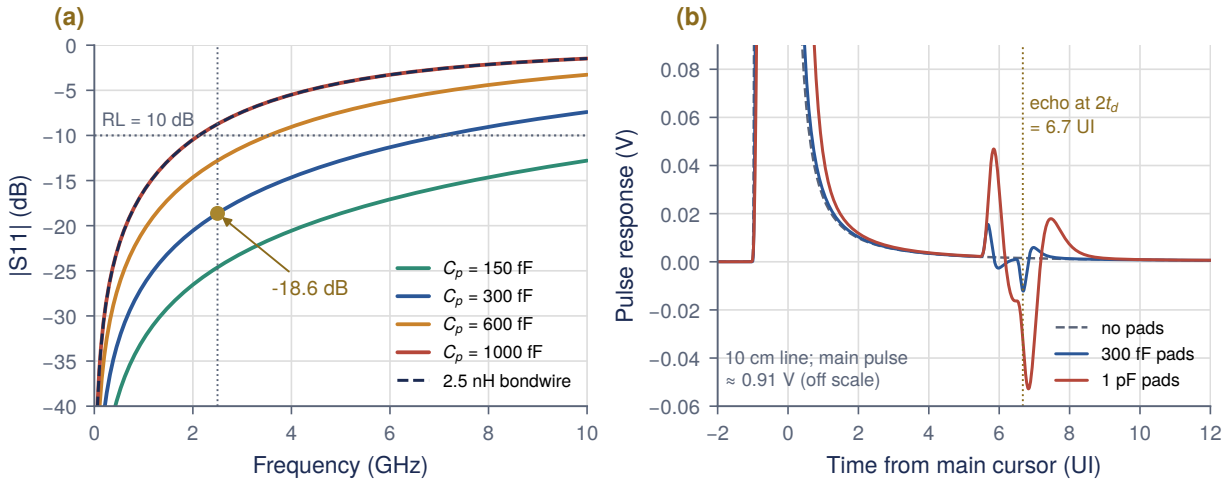


Figure 3.6. Pad capacitance spoils the termination at high frequency, and the reflections return as echoes. (a) Return loss of a 50 Ω termination with shunt C_p; a 2.5 nH bondwire (dashed) is as bad as 1 pF. (b) Pulse response of a 10 cm REF-type line. With 1 pF pads, the wave reflected at the RX pad returns from the TX pad and arrives $2t_d = 6.7$ UI after the main cursor. A short, low-loss line does not attenuate this echo.

it. Open-SERDES has 50 Ω per side (100 Ω differential) at TX and RX; Section 9.3 builds these resistors in SG13G2. A ±20% resistor error (40–60 Ω) gives $|\Gamma| \le 0.11$, i.e. RL ≥ 19 dB, which is acceptable. The parasitics around the resistor are the bigger problem.

Pad capacitance. The bond pad, the ESD diodes and the input devices add a shunt capacitance C_p across the termination. With $R_T = Z_0$, the load is $Z_0 \parallel 1/(j\omega C_p)$ and Equation (3.7) becomes

$$\Gamma(f) = \frac{-j\omega Z_0 C_p}{2 + j\omega Z_0 C_p}, \qquad |\Gamma| = \frac{\omega Z_0 C_p}{\sqrt{4 + (\omega Z_0 C_p)^2}}, \tag{3.8}$$

plotted in Figure 3.6(a) for several values of C_p.

WORKED EXAMPLE 3.3 · Pad capacitance and return loss

Given: $R_T = 50\,\Omega$ in parallel with $C_p = 300$ fF (the REF pad), at 2.5 GHz.

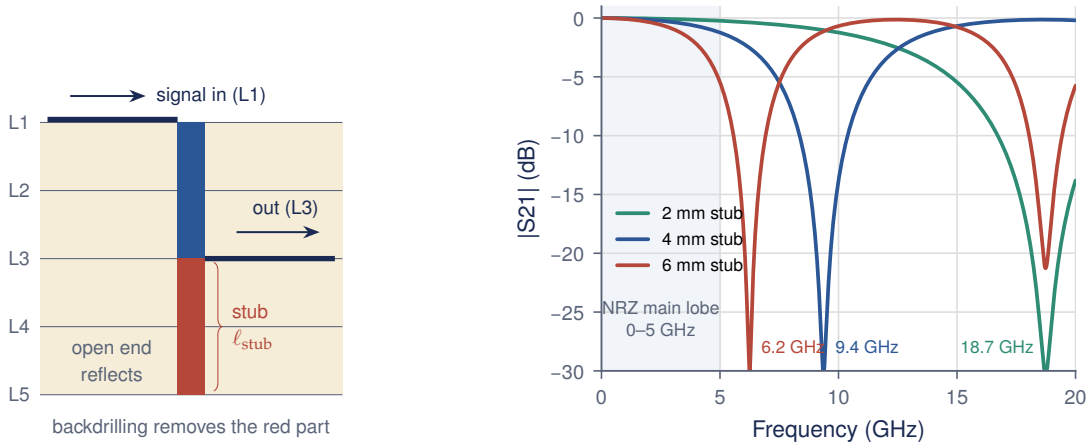
Solution: $\omega Z_0 C_p = 2\pi(2.5\times10^9)(50)(300\times10^{-15}) = 0.236$, so $|\Gamma| = 0.236/\sqrt{4.056} = 0.117$ and RL = 18.6 dB (12.8 dB at 5 GHz).

Budget: inverting Equation (3.8), $\omega Z_0 C_p = 2|\Gamma|/\sqrt{1-|\Gamma|^2}$. RL ≥ 10 dB at 2.5 GHz allows $C_p \le 0.85$ pF; RL ≥ 15 dB allows $C_p \le 0.46$ pF. Pad, ESD and input devices share this budget (Section 9.6).

WORKED EXAMPLE 3.4 · Bondwire inductance

Given: a 2.5 mm bondwire, $L \approx 2.5$ nH (about 1 nH/mm), in series between two 50 Ω environments, at 2.5 GHz.

Solution: a series impedance $Z = j\omega L$ between matched lines gives $S_{11} = (Z/2Z_0)/(1 + Z/2Z_0)$ and $S_{21} = 1/(1 + Z/2Z_0)$. Here $\omega L = 39.3\,\Omega$ and $\omega L/2Z_0 = 0.393$. So $|S_{21}| = 1/\sqrt{1.154} = 0.931$ (−0.62 dB) but $|S_{11}| = 0.393/1.075 = 0.366$: **RL = 8.7 dB**, failing a 10 dB target on its own.



(a) Via with an unused stub **(b)** Notch at $v/(4\ell_{stub})$ in FR-4

Figure 3.7. A via stub is a quarter-wave resonator that notches $|S_{21}|$. (a) The signal turns at L3, so the barrel below is an open stub. (b) Through-path $|S_{21}|$ with 2, 4 and 6 mm stubs, labeled with their null frequencies. A 6 mm stub puts the notch at 6.2 GHz, close enough to the 0–5 GHz NRZ main lobe (shaded) to distort the pulse.

> **Fix:** keep bondwires below about 1 mm, use flip-chip bumps, or absorb the inductance into a matched section: 2.5 nH with 1.0 pF of pad capacitance forms an artificial line with $\sqrt{L/C} = 50\,\Omega$ and a cutoff of $1/(\pi\sqrt{LC}) = 6.4$ GHz.

Via stubs. A through-hole via connects traces on different layers. If the signal enters on the top layer and leaves on an inner layer, the rest of the barrel is an unused **stub**: a short open-ended line hanging off the signal path (Figure 3.7a). Where the stub is a quarter wavelength long, its open end is transformed into a short at the signal path, and almost nothing gets through:

$$f_{\text{null}} \approx \frac{v}{4\,\ell_{\text{stub}}}. \tag{3.9}$$

WORKED EXAMPLE 3.5 · Via-stub notch

Given: FR-4 ($v = 1.5 \times 10^8$ m/s), stubs of 1.5, 3 and 5 mm.

Solution: Equation (3.9) gives $f_{\text{null}} = 25$, 12.5 and 7.5 GHz. To keep the first null above $3f_{\text{N}} = 7.5$ GHz, stubs must be shorter than $v/(4 \times 7.5\,\text{GHz}) = 5$ mm. The barrel's extra capacitance slows the wave, so in practice keep stubs below about 3 mm, route on layers near the bottom, or backdrill. The Open-SERDES test board should follow this rule for every high-speed via (Section 15.4).

▶ What this means for your design

Every discontinuity sends back a small, filtered copy of the signal. Two discontinuities form a cavity, and the copy that bounces between them reaches the receiver $2t_d$ late, about 1.7 UI per inch of FR-4 between them. In the pulse response this shows up as *ripples and bumps in the tail*, often 5–20 UI after the main cursor (Figure 3.6b). A two-tap DFE cannot cancel ISI that far out, so control reflections at the source: accurate termination, small pad capacitance, short bondwires and no long via stubs. A short, low-loss channel does not attenuate its echoes, so it can be harder to equalize than a longer one.



(a) Port numbering used in this book **(b)** Near- and far-end crosstalk

Figure 3.8. Differential channels are 4-ports, and neighboring lanes couple. (a) Ports 1 and 3 are the P and N inputs, 2 and 4 the outputs. Check the port map of every Touchstone file: many tools number the pair (1,2) in and (3,4) out. (b) An aggressor couples into the victim at the near end (NEXT) and the far end (FEXT).

3.5 S-parameters

At gigahertz frequencies we cannot measure with opens and shorts, because a short wire is already a transmission line. A vector network analyzer instead drives each port from 50 Ω, terminates the others in 50 Ω, and measures traveling waves: a_j into port j and b_i out of port i. The **scattering parameters** are $S_{ij} = b_i/a_j$ with all other inputs matched. For a two-port, S_{11} is the input reflection (return loss $-20\log_{10}|S_{11}|$) and S_{21} the transmission; the **insertion loss** is $\mathrm{IL} = -20\log_{10}|S_{21}|$, a positive number. Because a doubly terminated link has the same 50 Ω reference, S_{21} is the transfer function $H_{\mathrm{ch}}(f) = 2V_{\mathrm{out}}/V_S$ that we need. To cascade elements, convert each to an ABCD matrix, multiply, and convert back; `link.py` builds REF this way.

Differential channels and mixed-mode S-parameters. Open-SERDES is differential: the information is $V_P - V_N$. A differential channel has four single-ended ports, numbered as in Figure 3.8(a): ports 1 and 3 are the P and N inputs, 2 and 4 the P and N outputs. Driving the input differentially ($a_1 = -a_3$) and reading the differential output $(b_2 - b_4)/\sqrt{2}$ gives the **mixed-mode** parameters, referenced to 100 Ω differential:

$$S_{dd21} = \tfrac{1}{2}\left(S_{21} - S_{23} - S_{41} + S_{43}\right), \qquad S_{dd11} = \tfrac{1}{2}\left(S_{11} - S_{13} - S_{31} + S_{33}\right). \tag{3.10}$$

$S_{cd21} = \frac{1}{2}(S_{21} - S_{23} + S_{41} - S_{43})$ measures how much differential signal turns into common mode, for example through a P/N length mismatch. For a symmetric, weakly coupled pair, $S_{dd21} \approx S_{21}$ of one leg, which is why REF can be modeled as a single-ended 50 Ω line.

Crosstalk. Mutual capacitance C_m and inductance L_m couple an **aggressor** line into a **victim** (Figure 3.8b). **Near-end crosstalk** (NEXT) travels back toward the aggressor's source and is the dangerous kind in a transceiver: a full-swing local TX lane couples into a local RX lane whose own signal has already lost 17 dB. A 0.8 Vppd aggressor at −30 dB NEXT injects about 25 mVpp, comparable with the residual ISI after equalization. **Far-end crosstalk** (FEXT) travels with the victim's signal, is attenuated with it, and nearly cancels in a homogeneous stripline. Route TX and RX on different layers or connector rows, and prefer stripline.

Time-domain reflectometry. A **TDR** launches a fast step and records the reflected voltage at the source. A reflection at round-trip time t comes from distance $x = vt/2$, and its amplitude gives the local impedance through $Z = Z_0(1+\rho)/(1-\rho)$, with $\rho = V_r/V_i$. Figure 3.9 shows the two signature shapes: a series inductance bumps above 50 Ω and a shunt capacitance dips below. For a lumped element, the size of the feature depends on the rise time: with a 25 ps step a 1 nH pin reads 97 Ω, with a 100 ps step only 61 Ω. A TDR result always comes with its rise time, which sets its resolution.

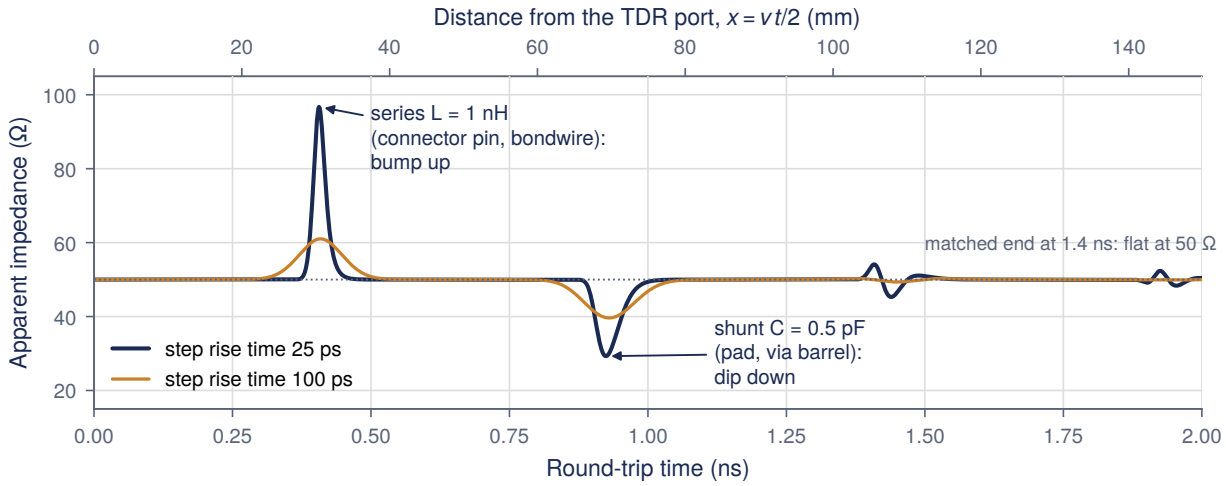


Figure 3.9. TDR: inductance bumps up, capacitance dips down. Simulated TDR of a 50 Ω line with a 1 nH series inductance at 15 mm and a 0.5 pF shunt capacitance at 68 mm, ending in a matched load. The top axis converts round-trip time to distance. A slower step (amber) blurs and shrinks both features. The wiggles near 1.4 ns are the wave re-reflected between the two discontinuities.

3.6 From frequency to time

The tools of Chapter 4 need the pulse response $p(t)$ of Section 2.4. We get it from $S_{21}(f)$ in two steps.

Causality. A physical channel cannot respond before it is driven. In the frequency domain this means the magnitude and phase of $H_{\text{ch}}(f)$ are linked: a given loss curve comes with a specific phase (dispersion). A model built from $|S_{21}|$ alone with a pure delay gives a symmetric impulse response that starts long before the wave could arrive (Figure 3.10). A dielectric model with ε_r and $\tan\delta$ held exactly constant has the same flaw in a milder form. REF uses a causal dielectric whose capacitance varies as $(j\omega/\omega_0)^{-2\tan\delta/\pi}$, the simplest of the causal FR-4 models [17]. A non-causal model produces *fake pre-cursors*, which then mislead the design of the TX FFE pre-tap.

Impulse and pulse response. The impulse response is the inverse Fourier transform of the conjugate-symmetric spectrum, $h(t) = \mathcal{F}^{-1}\{H_{\text{ch}}(f)\}$ with $H_{\text{ch}}(-f) = H_{\text{ch}}^{*}(f)$. The pulse response is $h(t)$ convolved with a 1-UI rectangle.

KEY EQUATION · From S_{21} to the pulse response

$$p(t) = \int_{t-T_{\text{b}}}^{t} h(\tau)\,d\tau \quad\Longleftrightarrow\quad P(f) = H_{\text{ch}}(f)\,T_{\text{b}}\,\text{sinc}(fT_{\text{b}})\,e^{-j\pi fT_{\text{b}}} \tag{3.11}$$

On a grid of N points spaced Δf up to $f_{\max}$, the IFFT gives a time step $\Delta t = 1/(2f_{\max})$ and a window $T_w = 1/\Delta f$. `common.py` uses 0–160 GHz in 10 MHz steps: $\Delta t = 3.125$ ps (64 samples per UI) and $T_w = 100$ ns, far longer than the REF delay plus tail. Figure 3.11 compares the two responses: the 1-UI rectangle averages h over 200 ps, so $p(t)$ is smoother and lower, and its tail coincides with that of $h(t) \cdot T_{\text{b}}$.

Figure 3.12 shows how the pulse response changes with loss, using the `LOSS_FAMILY` lines of `link.py`. From 5 to 25 dB of Nyquist loss, the main cursor falls from 0.77 to 0.22 V while the first post-cursor stays near 0.16–0.18 V, so p_1/p_0 grows from 0.12 to 0.71. That ratio, more than the loss itself, is what the equalizer must fight.

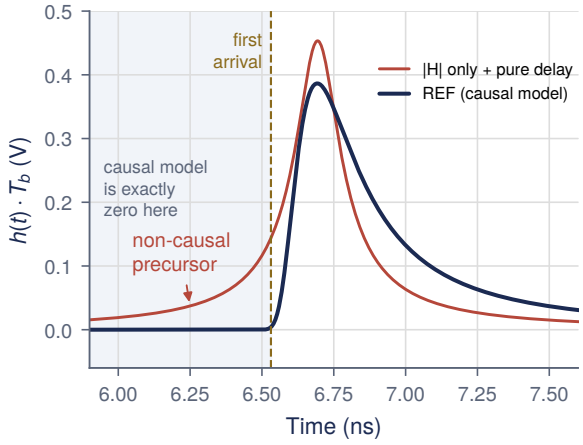


Figure 3.10. A magnitude-only model is not causal. Keeping $|S_{21}|$ of REF but replacing its phase by a pure delay gives a symmetric response with a long precursor before the first arrival.

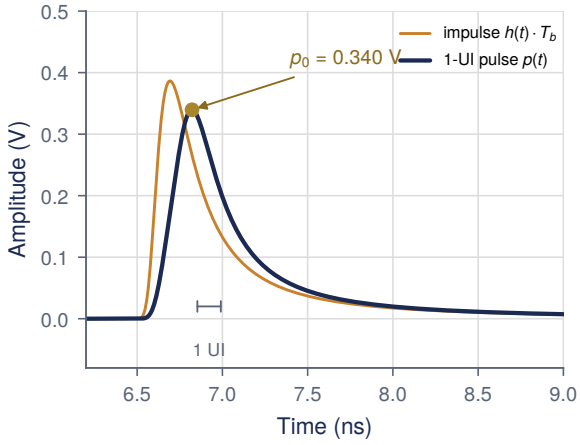


Figure 3.11. The pulse response is the impulse response averaged over one UI. REF, with $h(t)$ scaled by T_b so both curves are in volts. The peak $p_0 = 0.340\,\mathrm{V}$ arrives 6.82 ns after launch.

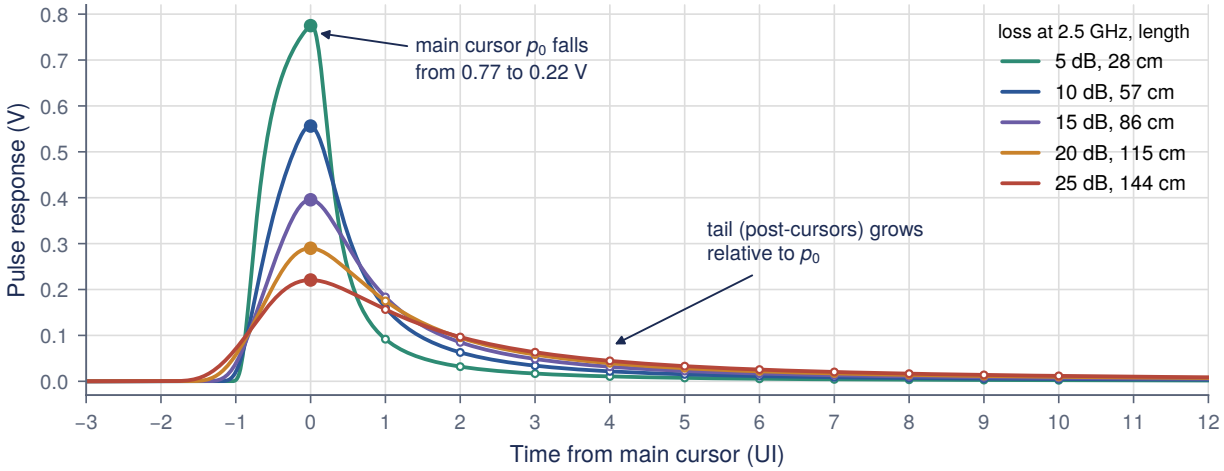


Figure 3.12. More loss means a smaller main cursor and a relatively larger tail. Pulse responses of the `LOSS_FAMILY` channels, aligned at their peaks; open dots mark the post-cursors $p_1 \ldots p_{10}$. At 15 dB (close to REF) the first post-cursor is already 46% of the main cursor.

! IFFT mistakes that change your cursors

Forgetting conjugate symmetry gives a complex $h(t)$. A coarse Δf makes the window shorter than the delay plus tail, so the tail wraps around and looks like pre-cursor ISI. Data cut abruptly at its last frequency rings (Gibbs), and data without a DC point must be extrapolated. And the cursors are samples of $p(t)$, not of $h(t)$.

3.7 The Open-SERDES reference channel

Every model, figure and lab in this book uses one synthetic channel, **REF**: a 50 Ω source, a 300 fF pad, 1 m of lossy FR-4 stripline, another 300 fF pad and a 50 Ω load. It is built from serdespy's ABCD functions [3] by `channel(1.0)` in `tutorial/figsrc/link.py` and `tutorial/code/common.py`. Table 3.1 lists the parameters, and Figure 3.13 shows the insertion loss: −17.4 dB at 2.5 GHz. The pads cost only 0.1 dB of that; their real damage is the reflections they add.

Table 3.1. REF model parameters and the matching `channel()` arguments.

Quantity	Value	Arg.
Length	1.0 m	`length`
Z_0 per side	50 Ω	`z0`
ε_r	4.0 (6.67 ns/m)	`er`
$\tan\delta$	0.02 at 1 GHz, causal	`tand`
R_{DC}	7 Ω/m	`rdc`
Skin R	60 Ω/m at 1 GHz, $\propto\sqrt{f}$	`kr`
Pads	300 fF each end	`c_pad`
Source, load	50 Ω each	
Grid	0–160 GHz, 10 MHz	`F`
Loss	0.59 / 10.4 / 17.4 / 30.1 dB	
	at 0 / 1.25 / 2.5 / 5 GHz	

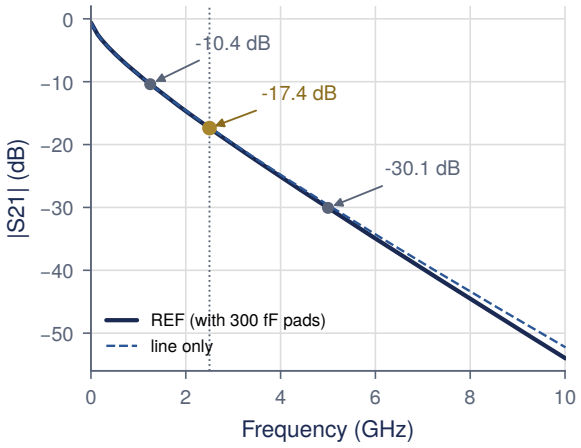


Figure 3.13. REF loses 17.4 dB at Nyquist. $|S_{21}|$ with and without the 300 fF pads, which barely change the loss.

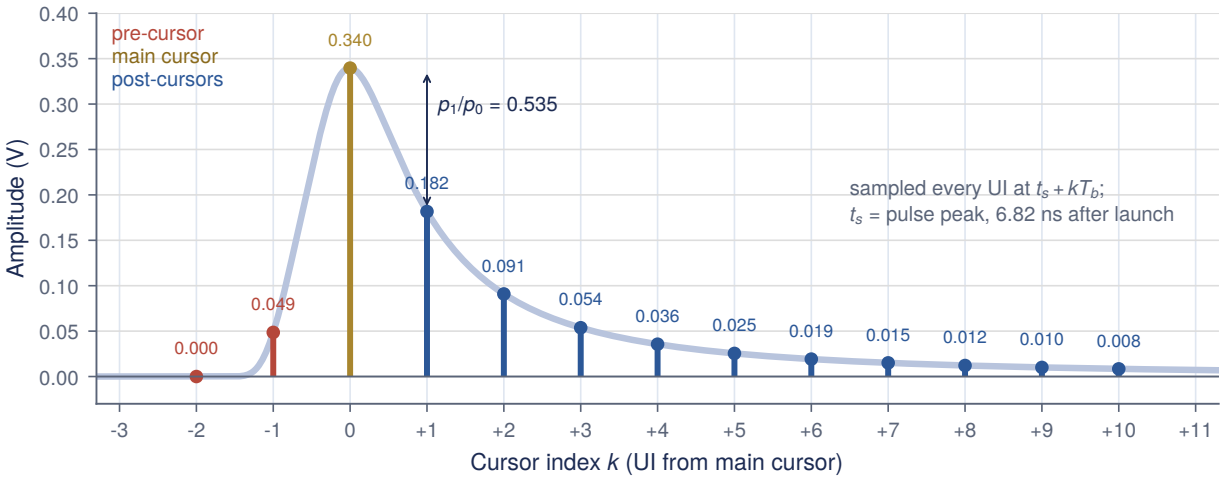


Figure 3.14. The REF pulse response and its cursors. A 1 V, 1-UI pulse arrives as a 0.340 V peak with a long tail. Sampling every UI from the peak gives the pre-cursor p_{-1} (red), the main cursor p_0 (gold) and the post-cursors $p_1, p_2, \ldots$ (blue). The first post-cursor alone is 53.5% of the main cursor.

Figure 3.14 shows the REF pulse response sampled every UI from its peak, and Table 3.2 lists the cursors. These thirteen numbers are the most-used data in this book.

Why this channel is hard. Three numbers tell the story. First, $p_0 = 0.34$: only a third of the transmitted pulse arrives in its own UI. Second, $p_1/p_0 = 0.535$: the previous bit contributes more than half as much as the current one. Third, the ISI magnitudes $\sum_{k\neq 0}|p_k|$ of these cursors add up to 0.501 V, more than p_0. A worst-case pattern can push a "1" below zero, and the peak-distortion eye of Section 4.2 is $1\,\text{V} \times (0.340 - 0.501) = -0.161\,\text{V}$: *closed*. A PRBS7 simulation agrees (Section 16.3). REF cannot be received without equalization, and the Open-SERDES link must still reach BER $\leq 10^{-12}$ over it, with margin up to the 20–25 dB end of `LOSS_FAMILY`

Table 3.2. REF cursors in volts for a 1 V, 1-UI pulse, sampled at the pulse peak ($t_s = 6.82$ ns), from `link.cursors`.

k	−2	−1	**0**	1	2	3	4	5	6	7	8	9	10
p_k	0.000	0.0486	**0.3396**	0.1817	0.0909	0.0537	0.0356	0.0255	0.0192	0.0150	0.0121	0.0100	0.0084

(Section 8.5, Table 14.2).

Replacing REF with real S-parameters. REF is a teaching model. Once the Open-SERDES test board is measured, or its traces are simulated with **openEMS** or **Palace**, you will swap in a 4-port Touchstone file. **scikit-rf** [18] reads it, converts it to mixed mode and extrapolates it to DC; the rest of the flow is unchanged:

```
import skrf as rf
from common import *                          # grid F, impulse(), pulse(), SPS
ch = rf.Network("board.s4p")                  # measured or EM-simulated 4-port
ch.renumber([1, 2], [2, 1])                   # book map (1,3 in; 2,4 out) -> skrf (0,1 in)
ch.se2gmm(p=2)                                # mixed mode: ports d-in, d-out, c-in, c-out
ch = ch.extrapolate_to_dc(kind="linear")      # add 0 Hz, uniform frequency step
sdd21 = ch.s[:, 1, 0]
H = np.interp(F, ch.f, sdd21.real, right=0) + 1j * np.interp(F, ch.f, sdd21.imag, right=0)
h, t = impulse(H); p = pulse(h)               # then cursors(), pda_eye(), ... as for REF
```

Check the port map first (Figure 3.8), and taper the data toward its highest frequency instead of cutting it to zero. serdespy's `four_port_to_diff` does the same job but assumes through paths $1 \to 2$ and $3 \to 4$ and ignores its `port_def` argument.

>_ Lab 1: the channel

Run `lab1_channel.py` (Section 16.2). It plots $|S_{21}|$ of REF, converts it to $h(t)$ and $p(t)$, and prints the cursors of Table 3.2. Then set `c_pad` to 1 pF and `length` to 0.3 m and watch the reflections of Figure 3.6(b) appear in the tail.

3.8 Exercises

3.1 Is it a line? A package trace is 8 mm long on a substrate with $\varepsilon_r = 3.5$, driven with $t_r = 40$ ps. Compute its delay. Must it be modeled as a transmission line?

3.2 Better laminate. Repeat Worked Example 3.2 for a low-loss laminate ($\varepsilon_r = 3.7$, $\tan\delta = 0.004$) with the same copper. How long can the trace be for 17.4 dB at 2.5 GHz? Which loss term now dominates?

3.3 ESD budget. The RX must have RL ≥ 15 dB at 2.5 GHz. The pad is 120 fF and the input devices add 60 fF. How much capacitance is left for ESD protection?

3.4 Via stub. A 3.2 mm thick board routes a signal from L1 to an inner layer 0.8 mm deep, and the via's capacitance gives $\varepsilon_{r,\mathrm{eff}} = 6$. Where is the stub null? Is it acceptable for Open-SERDES? What if the signal exits on the bottom layer instead?

3.5 Cursor ratio. With `channel()` from `link.py`, find the length at which $p_1/p_0 = 0.5$ and its Nyquist loss. Repeat with `tand=0.01` and explain why the answer moves.

KEY TAKEAWAYS

- Off-chip, every trace longer than about 1 mm is a transmission line at 5 Gb/s. FR-4 delay is 6.67 ps/mm (169 ps/in); REF takes 6.67 ns, or 33 UI.
- Conductor loss grows as $\sqrt{f}$ (skin depth 1.32 µm at 2.5 GHz); dielectric loss grows as $f \tan\delta$ and dominates FR-4 above 0.5 GHz. A smooth FR-4 stripline loses about 0.33 dB/in at 2.5 GHz.
- Quote and budget insertion loss at Nyquist, 2.5 GHz. REF loses 17.4 dB there.
- Discontinuities reflect: 300 fF of pad gives RL = 18.6 dB, a 2.5 nH bondwire only 8.7 dB, and a via stub notches S_{21} at $v/(4\ell)$. Echoes return as late bumps in the pulse response.
- Use mixed-mode S_{dd21} for differential channels and check the port map. On a TDR, inductance bumps up and capacitance dips down.
- A causal model and a correct IFFT turn S_{21} into $h(t)$ and $p(t)$. REF has $p_{-1} = 0.049$, $p_0 = 0.340$ and $p_1 = 0.182$ ($p_1/p_0 = 0.535$), and its unequalized peak-distortion eye is closed (-0.161 V).

CHAPTER 4

Link Budgets and Statistical Analysis

The pulse response is enough to predict the eye: its worst case, its probability at 10^{-12}, and how much margin is left after noise, jitter and the sampler take their share.

IN THIS CHAPTER YOU WILL LEARN

- Explain why superposition lets us predict every received sample from the cursors p_k.
- Derive the peak-distortion (worst-case) eye and the worst-case bit pattern, and apply them to REF.
- Build a statistical eye by convolving ISI, noise and jitter distributions, and read eye height at 10^{-12}.
- Write a top-down margin budget for the Open-SERDES receiver and decide what to spend margin on.
- Decide between NRZ and PAM4 with the 9.54 dB rule.

4.1 Pulse-response link analysis

The channel, the TX FFE and the CTLE are linear and time-invariant, so the received waveform is a sum of shifted, scaled copies of one pulse response (Sections 2.3 and 2.4): $y(t) = A\sum_n a_n\, p(t - nT_\mathrm{b})$. Sampling at $t_\mathrm{s} + nT_\mathrm{b}$ and writing $p_k = p(t_\mathrm{s} + kT_\mathrm{b})$ gives

$$y_n = \underbrace{A\, a_n\, p_0}_{\text{signal}} + \underbrace{A \sum_{k\neq 0} a_{n-k}\, p_k}_{\text{ISI}} + \text{noise}. \tag{4.1}$$

At the sampler, the channel is a finite-impulse-response (FIR) filter whose taps are the cursors (Figure 4.1). Positive k are **post-cursors**, which come from *earlier* bits; negative k are **pre-cursors**, from later bits. Every question in this chapter (how bad the eye can get, how often, and how much margin is left) is answered from the list of p_k, without simulating long bit streams.

4.2 Peak distortion analysis

The worst case. Take a transmitted "1" ($a_n = +1$). By Equation (4.1), its sample is smallest when every ISI term pulls down, $a_{n-k}\, p_k = -|p_k|$ for all $k \neq 0$. The lowest "1" is then $A(p_0 - \sum_{k\neq 0}|p_k|)$,

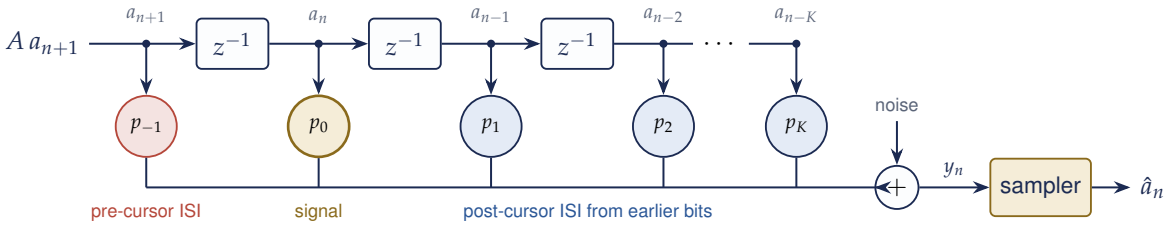


Figure 4.1. Sampled once per UI, the channel is an FIR filter whose taps are the pulse-response cursors. Each received sample is the current symbol times p_0 plus every other symbol times its cursor. REF has one significant pre-cursor ($p_{-1} = 0.049$) and a long post-cursor tail.

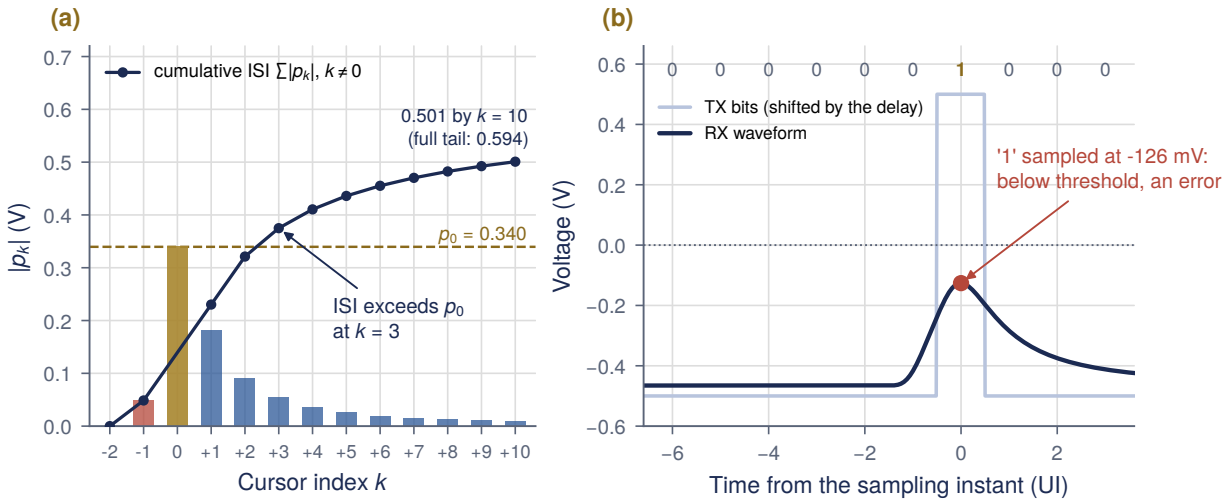


Figure 4.2. Peak distortion on REF: the ISI outweighs the main cursor. (a) Cursor magnitudes and the cumulative ISI (navy). It passes p_0 at $k = 3$ and reaches 0.501 V by $k = 10$; the full tail adds 0.093 V more. (b) The worst-case pattern, a lone "1" after a long run of "0"s. The sample at the main cursor (red dot) is -126 mV, so this "1" is read as a "0".

and by symmetry the highest "0" is its negative. The gap between them is the worst-case inner eye height.

KEY EQUATION · Peak-distortion eye and worst-case pattern

$$\mathrm{EH}(t_s) = 2A\Big(p_0 - \sum_{k\neq 0}|p_k|\Big), \qquad a_{n-k} = -\operatorname{sgn}(p_k)\, a_n. \tag{4.2}$$

This is **peak distortion analysis** (PDA) [13, 19]. It needs only the cursors, it bounds the eye for *any* bit stream, and the second half of Equation (4.2) gives the **worst-case bit pattern**. Because all cursors depend on the sampling instant, EH is a function of t_s. Sweeping t_s across the UI and keeping the range where $\mathrm{EH}(t_s) > 0$ gives the worst-case **eye width**.

WORKED EXAMPLE 4.1 · PDA of the REF channel

Given: the REF cursors of Table 3.2, $2A = 1$ V, t_s at the pulse peak.

Thirteen cursors: $\sum_{k\neq 0}|p_k| = 0.0486 + 0.1817 + 0.0909 + \cdots + 0.0084 = 0.501$ V, so EH $= 1\,\mathrm{V} \times (0.340 - 0.501) = \mathbf{-0.161\ V}$: closed. Figure 4.2(a) shows that the ISI already exceeds p_0 after three post-cursors.

Full tail: the cursors beyond $k = 10$ are small but many, and they add another 0.093 V. Summing out to 300 UI gives EH $= -0.255$ V. A check: all REF cursors are positive, so $\sum_k p_k = H_{\mathrm{ch}}(0) = 0.935$ (the DC gain) and $\mathrm{EH} = 2A(2p_0 - H_{\mathrm{ch}}(0)) = -0.255$ V.

Worst-case pattern: every p_k with $k \neq 0$ is positive, so for $a_n = +1$ all other bits are -1: a lone "1" after a long run of "0"s, followed by a "0". Figure 4.2(b) simulates it: the "1" arrives at -126 mV, below the threshold.

PDA versus a PRBS simulation. PDA is a bound, not a probability. The worst-case pattern for REF needs dozens of identical bits before the "1"; PRBS7 never has more than six "0"s in a row, so a PRBS7 simulation cannot find it. Figure 4.3 compares eye height versus sampling phase from PDA, from PRBS simulations and from the statistical eye of Section 4.3. On REF alone,

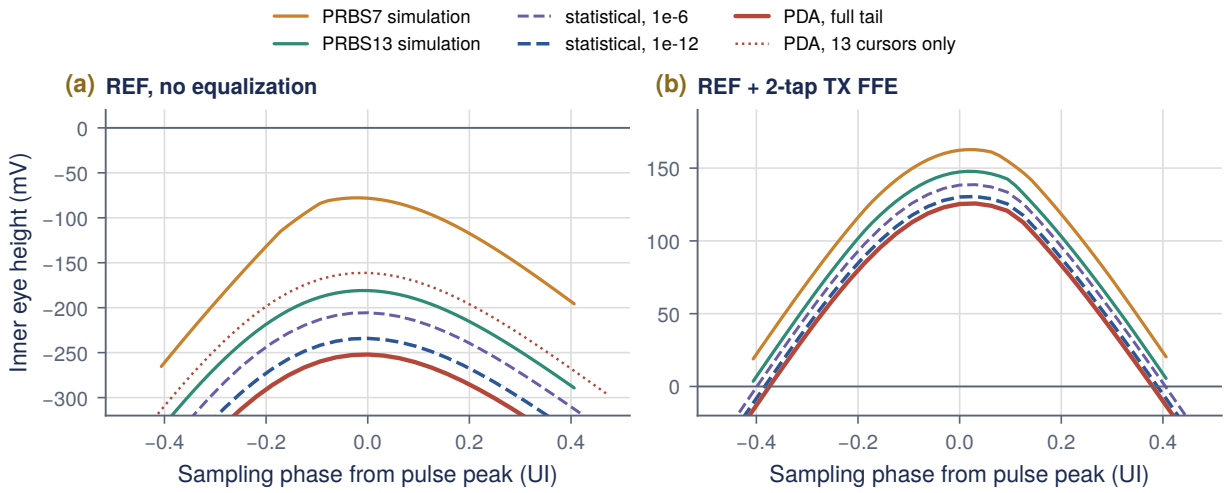


Figure 4.3. PDA is the most pessimistic estimate; short PRBS runs are the most optimistic. Inner eye height versus sampling phase. (a) REF is closed by every measure; truncating PDA at 13 cursors (dotted) hides 90 mV of tail ISI. (b) REF with the 2-tap TX de-emphasis $c_0 = 0.652$, $c_1 = -0.348$ (Lab 3). The statistical eye at 10^{-12} sits only 5 mV above PDA, while PRBS7 overestimates it by 32 mV.

PRBS7 reports −78 mV (Section 16.3), PRBS13 −181 mV and full-tail PDA −252 mV: closed in every view. After a 2-tap TX de-emphasis, the ranking is the same: PRBS7 163 mV, PRBS13 148 mV, PDA 126 mV. A short PRBS is optimistic because it never visits the rare patterns that set the eye at 10^{-12}.

! Truncating the tail

PDA sums $|p_k|$ over *all* k. Many small cursors add up: for REF, the cursors beyond $k = 10$ are each below 0.008 V yet add 0.093 V. Always extend the cursor list until the tail is negligible, and simulate with a PRBS whose period is longer than the channel memory (PRBS13 or longer for REF), or the eye will look better than it is.

4.3 The statistical eye

PDA tells us the worst case but not how often it happens. A BER target is a statement about probability, so we want the *distribution* of y_n. In Equation (4.1) each ISI term $A\,a_{n-k}p_k$ is an independent random variable that takes $+Ap_k$ or $-Ap_k$ with probability $\frac{1}{2}$. The PDF of a sum of independent variables is the convolution of their PDFs, so the ISI PDF is built one cursor at a time (Figure 4.4a): one cursor gives two spikes, two give four, m give up to 2^m, and the shape fills in [19, 20].

KEY EQUATION · Statistical eye

$$f_{\mathrm{ISI}}(v) = \mathop{\ast}_{k\neq 0}\left[\tfrac{1}{2}\delta(v - Ap_k) + \tfrac{1}{2}\delta(v + Ap_k)\right], \qquad f_X = f_{\mathrm{ISI}} * \mathcal{N}(0, \sigma_{\mathrm{n}}^2), \tag{4.3}$$

$$\mathrm{BER}(t_{\mathrm{s}}, v) = \tfrac{1}{2}P(Ap_0 + X < v) + \tfrac{1}{2}P(-Ap_0 + X > v), \tag{4.4}$$

$$\mathrm{BER}_J(t_{\mathrm{s}}, v) = \int \mathrm{BER}(t_{\mathrm{s}} - \tau, v)\, f_J(\tau)\, d\tau. \tag{4.5}$$

Gaussian noise of rms σ_{n} is one more convolution in voltage. Sampling-clock jitter moves t_{s}, so it

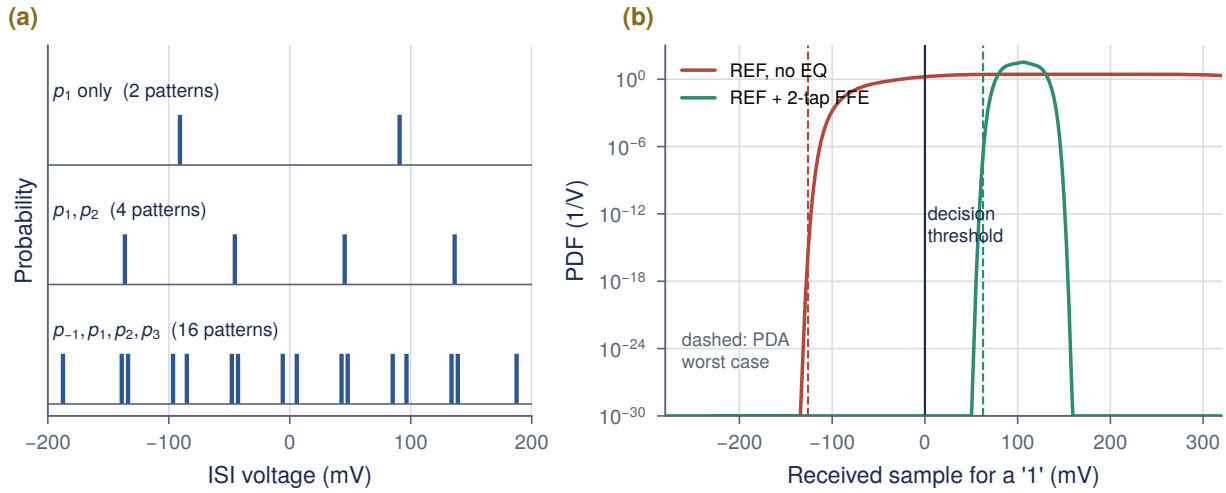


Figure 4.4. The ISI distribution is a convolution of two-point distributions. (a) REF cursors convolved one at a time: each new cursor splits every spike in two. (b) Distribution of the received sample for a "1", with all 153 cursors and 1.5 mV rms noise, on a log scale. Without equalization (red), 6% of all "1"s fall below the threshold. With the 2-tap TX FFE (teal), the distribution ends at 61 mV at 10^{-12}, just inside the PDA worst case (dashed).

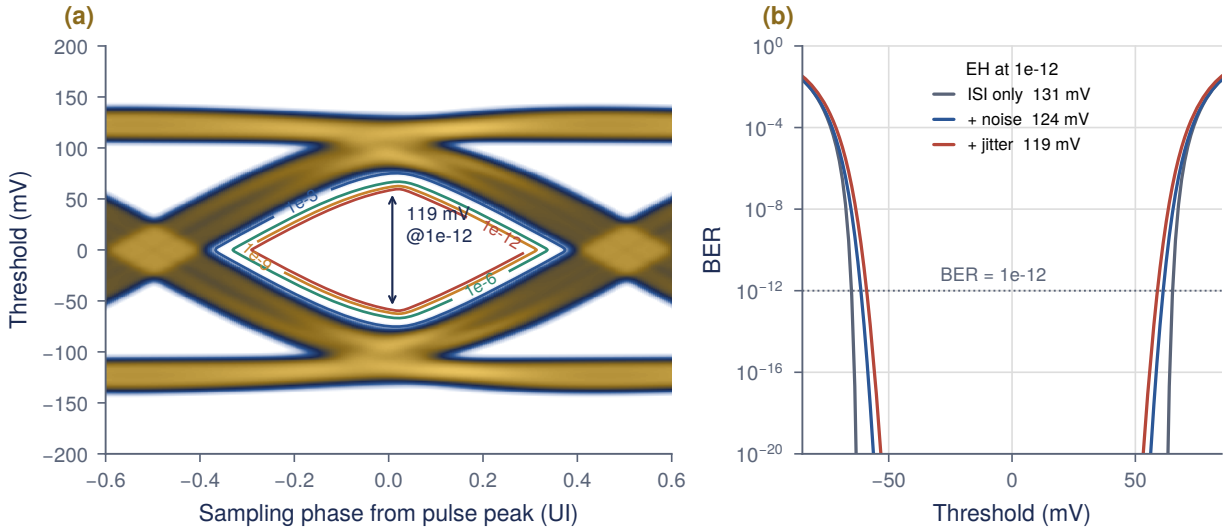


Figure 4.5. The statistical eye gives BER contours instead of one worst case. REF with 2-tap TX de-emphasis, $\sigma_n = 1.5$ mV rms, sampling jitter of 2 ps RJ and 20 ps DJ (dual-Dirac). (a) Signal density (shading) and contours at BER 10^{-3} to 10^{-12}; the 10^{-12} eye is 119 mV high and 0.59 UI wide. (b) Vertical bathtubs at the best phase. Noise and jitter each shave a few millivolts off the 131 mV ISI-only eye.

is a convolution in *time* with the jitter PDF f_J. Here we use the dual-Dirac model [21] (a Gaussian of rms σ_{RJ} at $\pm DJ/2$); Section 11.3 explains where it comes from. Evaluating Equation (4.4) over a grid of (t_s, v) gives the **statistical eye**: contours of constant BER. The **vertical bathtub** is one column of it, BER versus threshold at a fixed phase. The eye height at a BER target is the width of the bathtub at that BER.

The statistical eye needs a channel that is open, so from here on we use **REF with the 2-tap TX de-emphasis** of Lab 3, $c_0 = 0.652$ and $c_1 = -0.348$ (Sections 6.2 and 16.4). It is the simplest equalizer that opens REF, not the final Open-SERDES design. Figure 4.5 shows its statistical eye with noise and jitter. The code is `stat_eye()` in `figsrc/ch04_stateye.py`: a few dozen lines of NumPy that shift and add the two-point PDFs on a 0.05 mV grid.

WORKED EXAMPLE 4.2 · PDA versus statistical eye height

Given: REF with 2-tap de-emphasis. The equalized pulse peaks at $q_0 = 0.209\,\text{V}$, 22 ps earlier than REF. Its residual cursors are $q_{-1} = 0.0155$, $q_1 = 0.0125$, $q_2 = -0.0056$, then a slowly decaying tail of 0.002–0.0045 V per UI. $\sum_{k\neq 0}|q_k| = 0.084\,\text{V}$.

PDA: $\text{EH} = 1\,\text{V} \times (0.209 - 0.084) =$ **126** mV, open over 0.75 UI.

Statistical eye, ISI only: 139 mV at 10^{-6} and **131** mV at 10^{-12}, open over 0.77 UI.

With noise and jitter: $\sigma_\text{n} = 1.5\,\text{mV}$ gives 123 mV; adding 2 ps RJ and 20 ps DJ gives **119** mV and cuts the width to 0.59 UI.

Interpretation: at 10^{-12} the statistical eye is only 4% larger than PDA. A probability of $10^{-12} \approx 2^{-40}$ means roughly 40 cursors line up in their worst direction, and here the cursors beyond the first 40 add only 0.006 V. PDA is far too pessimistic only when the ISI is spread over many comparable cursors. A large gap between PDA and the statistical eye should make you check your tail, not celebrate.

▶ What this means for your design

Use the three tools for different jobs. PDA is instant and gives the worst-case pattern, which is the best stimulus for a transistor-level simulation. The statistical eye answers the actual specification, eye height and width at 10^{-12}, and is where noise and jitter enter. A PRBS7 eye from a short transient is a sanity check, never a sign-off. For REF-like channels, PDA is within a few percent of the 10^{-12} eye, so design to PDA early and confirm with the statistical eye.

>_ Statistical eye in serdespy

Import `stat_eye` and `eye_metrics` from `ch04_stateye.py`, and feed them the pulse response from any lab (Sections 16.3 and 16.4). Change `sigma_n`, `rj` and `dj` and watch which one moves the height and which the width.

4.4 The system noise and margin budget

A **margin budget** starts from the transmitted swing and subtracts, step by step, everything that eats the eye before the sampler decides. Whatever is left is the **margin** that covers what the model does not know: process corners, crosstalk, adaptation error. The terms are:

(i) **TX swing** $2A = 1\,\text{Vppd}$.
(ii) **Channel loss**, which shrinks the main cursor to $2Ap_0$.
(iii) **Equalization**: a TX FFE costs main cursor (Chapter 6); a CTLE and DFE do not, to first order.
(iv) **Residual ISI** at the BER target, from the statistical eye.
(v) **RX noise**: for Gaussian noise of rms σ_n, a BER of 10^{-12} needs $\text{EH} \geq 2Q\sigma_\text{n} = 14.07\,\sigma_\text{n}$ (Equation (2.9)).
(vi) **Jitter**: sampling away from the best phase lowers the eye.
(vii) **Sampler offset** V_os and **sensitivity** V_sens (Sections 10.2 and 10.4).

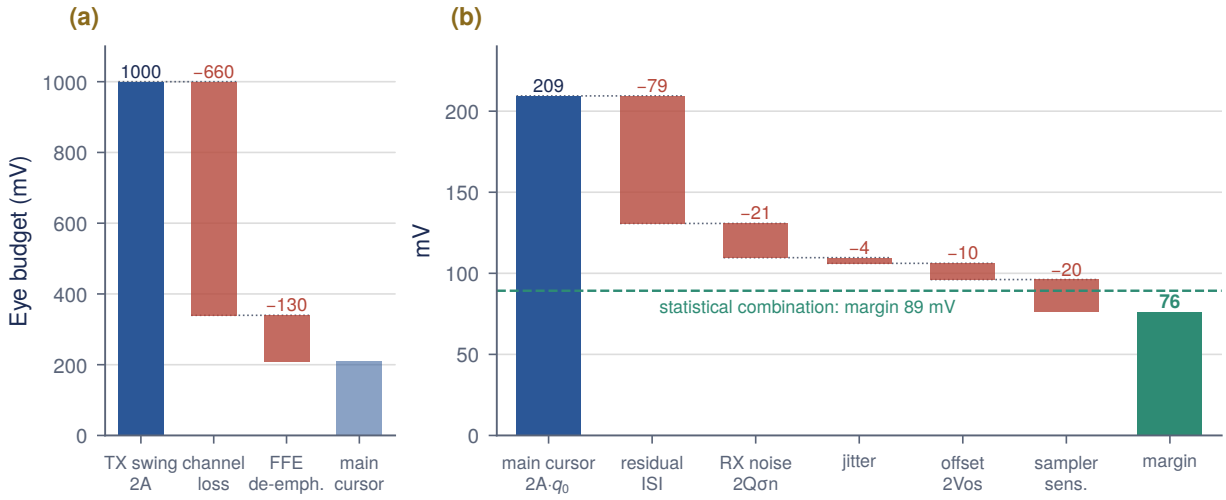


Figure 4.6. A top-down margin budget for REF with 2-tap de-emphasis. (a) The channel and the de-emphasis take 79% of the transmitted swing before any circuit impairment. (b) Zoom on the 209 mV main cursor: residual ISI is the largest single term. Adding the rest linearly leaves 76 mV; combining noise and jitter statistically (dashed) leaves 89 mV.

KEY EQUATION · Linear margin budget at the sampler

$$M = \mathrm{EH}_{\mathrm{ISI}}(10^{-12}) - 2Q\,\sigma_{\mathrm{n}} - \Delta V_{\mathrm{jitter}} - 2V_{\mathrm{os}} - V_{\mathrm{sens}} \;>\; 0. \tag{4.6}$$

An offset moves the threshold toward one rail, so the worse half-eye shrinks by V_{os}; that costs $2V_{\mathrm{os}}$ of symmetric eye height. The sensitivity is the minimum peak-to-peak input the sampler resolves in time. All terms are referred to the RX input.

WORKED EXAMPLE 4.3 · Margin budget for REF with 2-tap de-emphasis

Assumptions: $\sigma_{\mathrm{n}} = 1.5\,\mathrm{mV}$ rms input-referred (CTLE plus samplers); sampling jitter of 2 ps RJ and 20 ps DJ, the latter from PI quantization dither (6.25 ps steps) and CDR wander (Section 11.5); offset after calibration $V_{\mathrm{os}} = 5\,\mathrm{mV}$; sampler sensitivity $V_{\mathrm{sens}} = 20\,\mathrm{mV}$.

Stack-up (Figure 4.6): 1000 mV → channel → 340 mV main cursor → de-emphasis → 209 mV → residual ISI at 10^{-12} → 131 mV. Then noise $2Q\sigma_{\mathrm{n}} = 14.07 \times 1.5 = 21\,\mathrm{mV}$, jitter 3.5 mV, offset $2 \times 5 = 10\,\mathrm{mV}$ and sensitivity 20 mV leave $M \approx 76\,\mathbf{mV}$.

Statistical check: the statistical eye with noise and jitter included is 119 mV, which leaves $119 - 10 - 20 = 89\,\mathrm{mV}$. Noise really costs only 8 mV here, not 21 mV, because the 7σ noise peak rarely coincides with a worst-case ISI pattern. The linear budget is conservative by 13 mV.

▶ What to spend margin on

The largest term is residual ISI (79 mV), not noise (8–21 mV). The best use of design effort is therefore equalization: the CTLE and 2-tap DFE of Chapters 7 and 8 remove most of the residual ISI without the main-cursor cost of a TX FFE (Section 8.5). Keep the margin for things the model leaves out: the 20–25 dB end of `LOSS_FAMILY` (each extra dB of Nyquist loss shrinks the main cursor by about 6%), crosstalk (25 mVpp of NEXT would take a third of it), termination and CTLE drift over corners, and DFE tap error. Do not spend power to push σ_{n} below 1 mV: halving it buys only a few millivolts.

Table 4.1. Loss at the NRZ and PAM4 Nyquist frequencies for REF and the `LOSS_FAMILY` channels, from `link.py`.

Channel	Length (m)	IL at 2.5 GHz (dB)	IL at 1.25 GHz (dB)	Δ (dB)	Rule favors
5 dB	0.285	5.0	3.0	2.0	NRZ
10 dB	0.572	10.0	6.0	4.0	NRZ
15 dB	0.862	15.0	9.0	6.0	NRZ
REF	**1.000**	**17.4**	**10.4**	**7.0**	**NRZ**
20 dB	1.152	20.0	12.0	8.0	NRZ
25 dB	1.441	25.0	15.0	10.0	PAM4 (barely)
30 dB	1.731	30.0	18.0	12.0	PAM4

4.5 Modulation choice: NRZ or PAM4?

PAM4 sends two bits per symbol on four levels, $\{-A, -A/3, +A/3, +A\}$. At the same bit rate the symbol rate halves (2.5 GBd for 5 Gb/s), so the Nyquist frequency halves to 1.25 GHz and the channel loss there is smaller. The price is that the three stacked eyes are each $2A/3$ high instead of $2A$ (Figure 4.7a,b).

KEY EQUATION · The PAM4 rule

$$20\log_{10}3 = 9.54\,\text{dB} \quad\Longrightarrow\quad \text{PAM4 wins only if} \quad \Delta = \text{IL}(f_N) - \text{IL}(f_N/2) > 9.54\,\text{dB}. \tag{4.7}$$

PAM4 recovers its 9.54 dB eye penalty only if the channel loses more than that between the two Nyquist frequencies, which are an octave apart. If the loss were proportional to f (pure dielectric), Δ would be half the NRZ Nyquist loss; if it were proportional to $\sqrt{f}$ (pure skin effect), $\Delta = 0.29\,\text{IL}(f_N)$. Real FR-4 sits in between.

WORKED EXAMPLE 4.4 · NRZ or PAM4 for Open-SERDES?

Given: `channel()` from `link.py` for REF and the `LOSS_FAMILY` lengths.

Solution: Table 4.1 lists the losses at both Nyquist frequencies. For our model $\Delta \approx 0.40\,\text{IL}(f_N)$. REF has $\Delta = 17.4 - 10.4 = 7.0$ dB, so NRZ wins by $9.54 - 7.0 = 2.6$ dB. Δ reaches 9.54 dB at 1.37 m, where the NRZ Nyquist loss is 23.8 dB (Figure 4.7c). Only the 25 and 30 dB channels pass the rule, and only by 0.5 and 2.5 dB.

The rule is a first-order comparison, and it flatters PAM4. A DFE cancels post-cursors for either format, but whatever ISI is left (pre-cursors, tap errors) weighs three times more against a PAM4 eye. PAM4 also needs three comparators per sampling phase (up to twelve samplers instead of the two data and two edge samplers of our half-rate receiver), reference levels that track the signal amplitude, a linear TX driver and CTLE, and a CDR that copes with several zero-crossing times per transition. It is also three times more sensitive to residual ISI, crosstalk and sampler offset.

Open-SERDES decision

Open-SERDES uses **NRZ**. Over REF and every channel up to 20 dB, NRZ wins by the 9.54 dB rule, and it costs a third of the slicers. PAM4 is kept as a Phase-4 option, worth

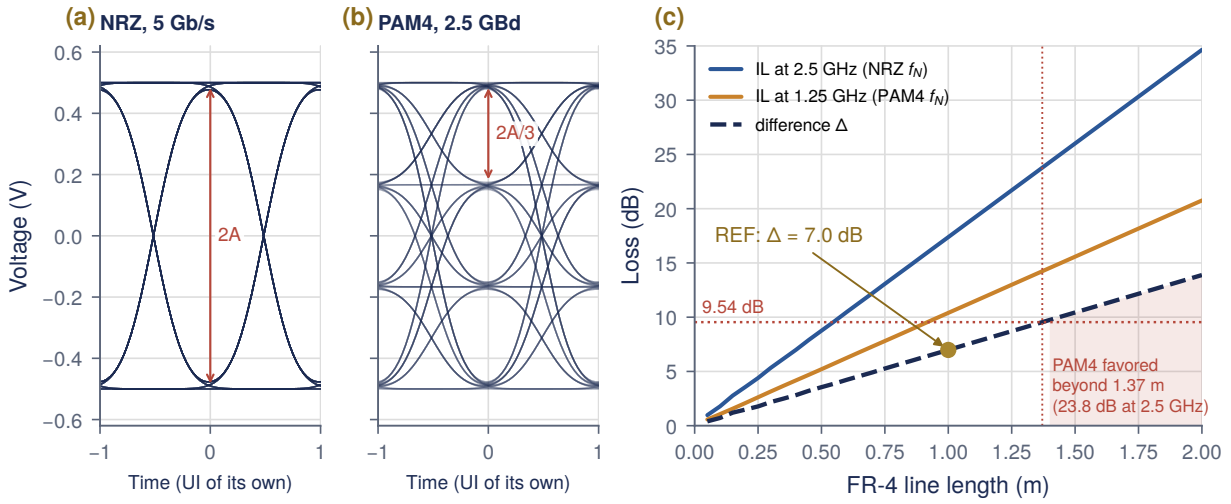


Figure 4.7. PAM4 pays 9.54 dB of eye height to halve the Nyquist frequency. (a, b) Ideal NRZ and PAM4 eyes at the same peak swing: each PAM4 eye is one third as tall. (c) REF-model loss at 2.5 and 1.25 GHz versus length, and their difference Δ. PAM4 is favored only where Δ exceeds 9.54 dB, beyond 1.37 m (23.8 dB at 2.5 GHz). REF (gold) is 2.6 dB short of that.

revisiting only for channels beyond about 24 dB at 2.5 GHz or for a 10 Gb/s follow-on (Table 14.4).

4.6 Exercises

4.1 PDA by hand. A channel has $p_{-1} = 0.05$, $p_0 = 0.50$, $p_1 = 0.20$, $p_2 = -0.04$, $p_3 = 0.02$. Find the PDA eye for $2A = 0.8$ V and the worst-case bit pattern in time order.

4.2 The tail matters. Using `link.cursors` with `n_post` = 10, 30, 100 and 300, plot the REF PDA eye against `n_post`. How many cursors do you need before the answer changes by less than 5 mV?

4.3 Two cursors, exactly. For ISI from only p_1 and p_2, write the ISI PDF and the BER versus threshold in closed form, with Gaussian noise σ_n. Compare with `stat_eye()`.

4.4 Budget. Redo Worked Example 4.3 with $\sigma_n = 3$ mV and an uncalibrated offset of 15 mV. Is the margin still positive? Which single improvement restores the most margin?

4.5 PAM4 at 10 Gb/s. For a 10 Gb/s follow-on over REF, compare NRZ ($f_N = 5$ GHz) with PAM4 ($f_N = 2.5$ GHz) using `loss_db`. Which wins by the rule?

KEY TAKEAWAYS

- At the sampler, the channel is an FIR filter with taps p_k, so the cursors alone predict every received sample.
- PDA gives the worst-case eye $2A(p_0 - \sum_{k\neq 0}|p_k|)$ and the worst-case pattern $a_{n-k} = -\operatorname{sgn}(p_k)\, a_n$. REF is closed: -0.161 V with 13 cursors, -0.255 V with the full tail.
- The statistical eye convolves two-point ISI distributions with noise (in voltage) and jitter (in time) to give BER contours. For REF with 2-tap de-emphasis: PDA 126 mV, statistical 131 mV at 10^{-12}, 119 mV with noise and jitter.
- Short PRBS simulations are optimistic (PRBS7: 163 mV) because they never meet the rare worst-case patterns.
- A linear margin budget for that link leaves about 76 mV (89 mV statistically). Residual ISI, not noise, is the

largest term, so spend design effort on equalization.

- PAM4 pays 9.54 dB of eye height and wins only if the loss between 1.25 and 2.5 GHz exceeds that. REF has 7.0 dB, so Open-SERDES uses NRZ.

PART II

Equalization

Undoing the channel. We look at where equalization can live in the link (TX FFE, CTLE, DFE), how much each can do, what it costs, and how to split the work between them for the Open-SERDES channel.

CHAPTER 5

Equalization Fundamentals

The channel is a low-pass filter. An equalizer is the filter we add so that channel and equalizer together look flat out to the Nyquist frequency.

IN THIS CHAPTER YOU WILL LEARN

- Explain how loss at f_N turns into inter-symbol interference and closes the REF eye.
- State the zero-forcing idea and quantify why an ideal channel inverse amplifies noise.
- Use a simple cable model to connect a loss in dB to the size of the main cursor.
- Name the four places an equalizer can sit in a link, and what each one can and cannot fix.
- Frame the partition question that Section 8.5 answers for Open-SERDES.

5.1 Why equalize?

Section 3.7 showed that the REF channel loses 17.4 dB at $f_N = 2.5$ GHz but only 0.6 dB at DC. A single 1-UI pulse therefore comes out smeared. Its main cursor is only $p_0 = 0.340$ V instead of 1 V, and its energy leaks into the neighbouring bits: $p_1 = 0.182$, $p_2 = 0.091$ and a long tail after it, plus a small pre-cursor $p_{-1} = 0.049$ (Table 3.2). Summed over all bits, this **inter-symbol interference** (ISI) exceeds the main cursor, so the peak-distortion eye of Equation (4.2) is -0.161 V. The eye is closed.

An **equalizer** is a filter $H_{eq}(f)$ placed somewhere in the link so that the cascade $H_{ch}(f)\,H_{eq}(f)$ is flat, at least up to f_N [22]. Flat to f_N is enough. A pulse sampled once per UI is free of ISI when its spectrum, folded around multiples of $1/T_b$, adds up to a constant. A response that is flat to f_N and rolls off smoothly above it comes close to meeting that condition. In the time domain, the same goal reads: *make every cursor except p_0 zero*. Figure 5.1(a) shows the most direct way to do it, which is to build the exact inverse of the channel.

5.2 The zero-forcing idea

A **zero-forcing** (ZF) equalizer chooses $H_{eq} = 1/H_{ch}$ over the band of interest, which forces all ISI to zero. It ignores noise, and that is its weakness. Thermal noise from the termination and from the first amplifier enters *before* the equalizer and is filtered by it. By the Wiener–Khinchin theorem, a noise process with power spectral density $S_n(f)$ passing through H_{eq} comes out with variance:

KEY EQUATION · Noise at the equalizer output

$$\sigma_{out}^2 = \int_0^B |H_{eq}(f)|^2\, S_n(f)\,\mathrm{d}f \xrightarrow{H_{eq}=1/H_{ch},\ S_n=N_0} \sigma_{out}^2 = N_0 B \cdot \underbrace{\frac{1}{B}\int_0^B \frac{\mathrm{d}f}{|H_{ch}(f)|^2}}_{\text{noise enhancement}} \tag{5.1}$$

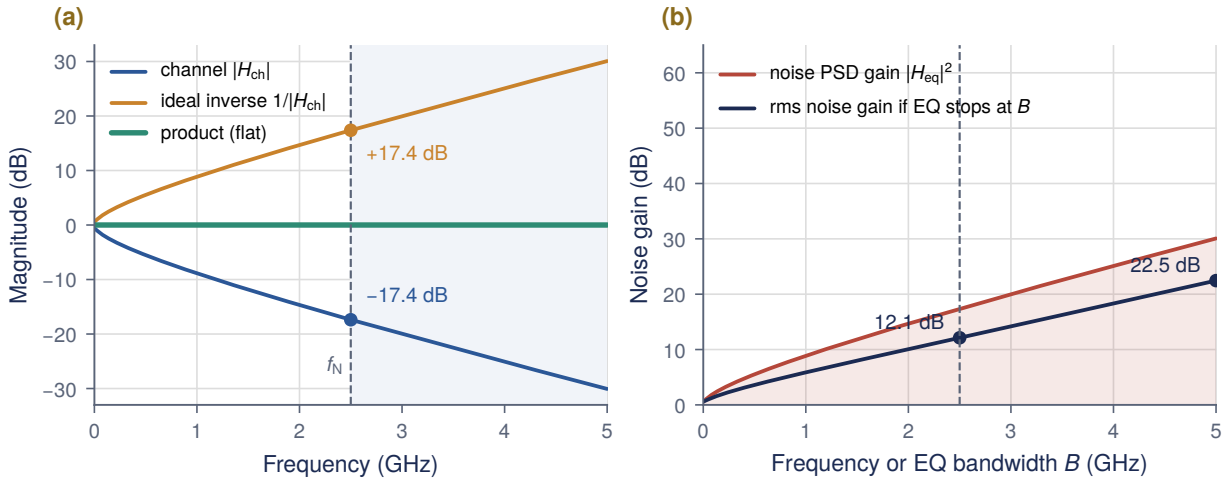


Figure 5.1. Inverting the channel removes ISI but amplifies noise. (a) REF $|H_{\mathrm{ch}}|$, its ideal inverse and their flat product. The inverse needs $+17.4$ dB at f_{N} and keeps rising beyond it. (b) For white noise at the equalizer input, the noise power spectrum is multiplied by $|H_{\mathrm{eq}}|^2$. The rms noise gain is 12.1 dB if equalization stops at f_{N} and 22.5 dB if it extends to $2f_{\mathrm{N}}$.

For REF the enhancement factor is 12.1 dB for $B = f_{\mathrm{N}}$ (Figure 5.1b). The signal is not boosted by that much on average. The main cursor recovers, but the noise integrated over the whole band grows faster than the signal does. Extending the inverse to $2f_{\mathrm{N}}$ costs another 10 dB of noise for almost no gain in eye opening. Practical equalizers are therefore *partial* inverses. They stop near f_{N} and limit their peak boost, or, like the DFE of Chapter 8, they avoid linear filtering of the noise altogether.

A cable model. A model with a closed-form pulse helps build intuition. In a cable, dielectric loss dominates, and the transfer function is close to $C(f) = \exp(-k_d\, l\, |f|)$, with l the length and k_d a loss coefficient. Write $a = k_d l$. The inverse Fourier transform is a Lorentzian,

$$c(t) = \int_{-\infty}^{\infty} e^{-a|f|} e^{j2\pi f t}\, \mathrm{d}f = 2\int_{0}^{\infty} e^{-af} \cos(2\pi f t)\, \mathrm{d}f = \frac{2a}{a^2 + 4\pi^2 t^2}. \tag{5.2}$$

Convolving with a pulse of height V_0 on $0 \le t \le T_{\mathrm{b}}$, and using the antiderivative $\int c(u)\, \mathrm{d}u = \frac{1}{\pi}\arctan(2\pi u/a)$, gives

$$y(t) = \frac{V_0}{\pi}\left[\arctan\frac{2\pi t}{a} - \arctan\frac{2\pi(t - T_{\mathrm{b}})}{a}\right]. \tag{5.3}$$

The pulse is symmetric about $t = T_{\mathrm{b}}/2$, so setting $\mathrm{d}y/\mathrm{d}t = 0$ puts the peak there:

KEY EQUATION · Main cursor of a dielectric-loss cable

$$y_{\max} = \frac{2V_0}{\pi}\arctan\frac{\pi T_{\mathrm{b}}}{k_d\, l}, \qquad \text{loss at } f_{\mathrm{N}}\ (\mathrm{dB}) = 20\log_{10}(e)\, k_d l\, f_{\mathrm{N}} = 8.686\, k_d l\, f_{\mathrm{N}}. \tag{5.4}$$

WORKED EXAMPLE 5.1 · How much does 15 dB of cable loss cost?

Take a cable with 15 dB of loss at $f_{\mathrm{N}} = 2.5$ GHz. Then $k_d l = 15/(8.686 \times 2.5\,\mathrm{GHz}) = 691$ ps, and $\pi T_{\mathrm{b}}/(k_d l) = \pi \cdot 200/691 = 0.909$. Equation (5.4) gives $y_{\max} = (2/\pi)\arctan(0.909) = (2/\pi)(0.738) = 0.470\, V_0$, so the main cursor has fallen to less than half.

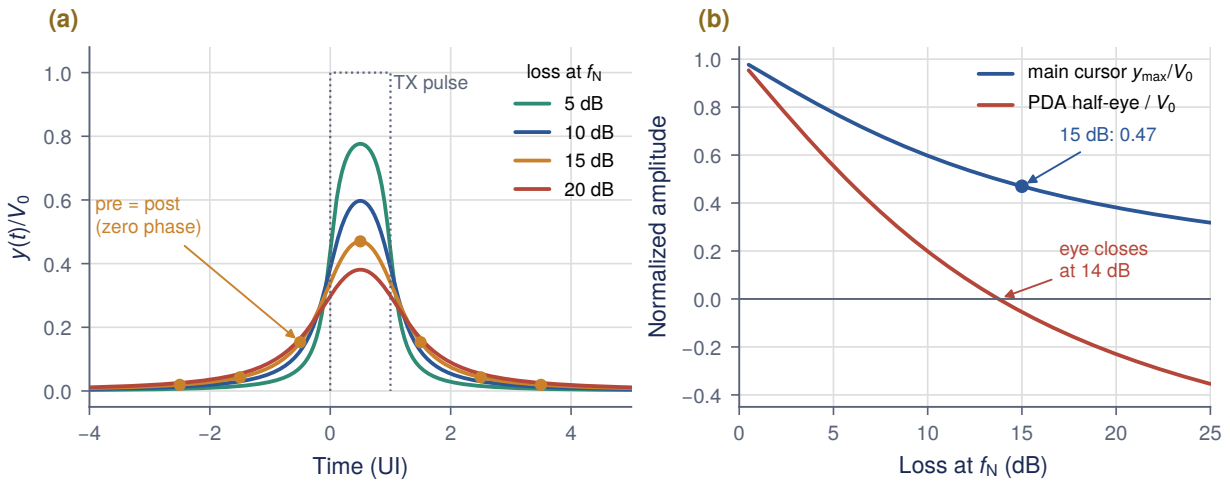


Figure 5.2. A simple cable model ties loss in dB to the pulse. (a) Pulse responses from Equation (5.3) for 5–20 dB of loss at f_N. The dots mark the cursors of the 15 dB case. The model has zero phase, so the pre-cursors equal the post-cursors. (b) Main cursor from Equation (5.4) and PDA half-eye against loss. The tails close the eye at about 14 dB, while the main cursor is still about half its lossless value.

The neighbours follow from Equation (5.3) at $t = T_b/2 + kT_b$: $y_{\pm1} = 0.153$, $y_{\pm2} = 0.043$ and $y_{\pm3} = 0.019$, in units of V_0. Summing all cursors out to ±100 UI gives $\sum|y_k| = 0.527 > 0.470$, so the PDA eye is closed. Figure 5.2(b) shows that the eye of this model closes at about 14 dB.

A zero-forcing inverse needs +15 dB at f_N. By Equation (5.1) with $|C|^{-2} = e^{2af}$, its noise enhancement is $(e^{2af_N} - 1)/(2af_N) = 9.5$ dB up to f_N. Up to $2f_N$ it would be 21.6 dB.

! A zero-phase model is not a real channel

$C(f) = e^{-k_d l|f|}$ is real and even, so its impulse response starts before $t = 0$ and is not causal. That is why its pre- and post-cursors are equal in Figure 5.2. A real channel such as REF is causal, and its post-cursors dominate ($p_1 = 0.182$ against $p_{-1} = 0.049$). Use the cable model for intuition about loss and the main cursor, and use REF (Section 3.7) for design numbers.

▶ What this means for your design

- Every 5 dB of loss at f_N costs a large share of the main cursor and adds tail ISI. Near 15 dB the tails, not the main cursor, are what close the eye.
- Equalize to about f_N and no further. A full inverse to $2f_N$ roughly doubles the noise penalty in dB (12.1 dB becomes 22.5 dB on REF) and buys almost no eye.
- Zero forcing ignores noise and swing. Every practical equalizer in this part of the book is a constrained version of it.

5.3 Where equalizers live

An equalizer can be placed at the transmitter, at the receiver input, or after the receiver's decision. Figure 5.3 places the options in the Open-SERDES link, and Table 5.1 compares them.

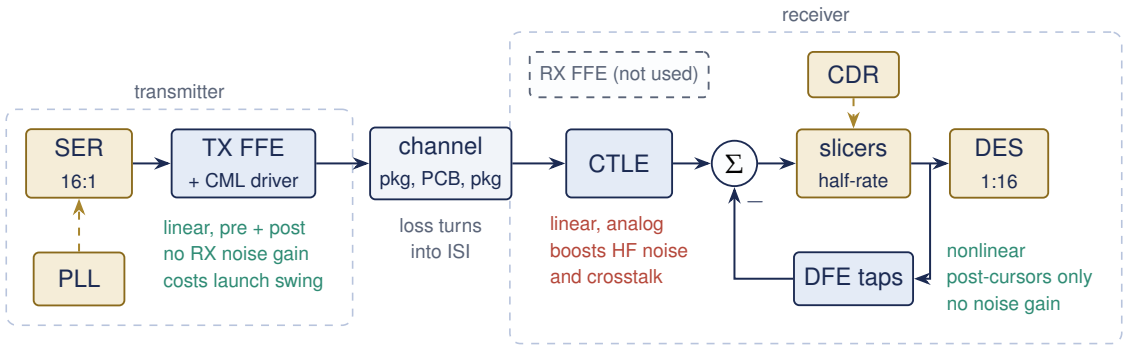


Figure 5.3. Equalization can sit at the TX, at the RX input, or after the decision. The Open-SERDES link uses a TX FFE in the driver, a CTLE as the first receiver stage and a DFE that feeds back from the half-rate slicers. A receiver FIR (RX FFE) is the fourth option. We do not use it.

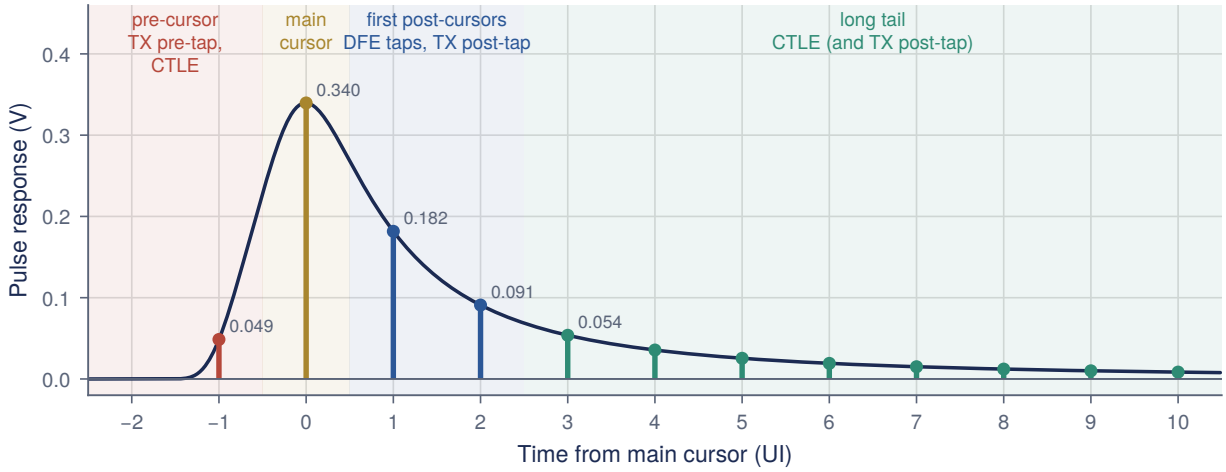


Figure 5.4. Each part of the pulse response has a natural owner. The pre-cursor (red) can only be removed by linear equalization, either the TX pre-tap or the CTLE. The first post-cursors (blue) are the DFE's job, which it does without noise. The long tail (teal) is best flattened by the CTLE or the TX post-tap, because removing it with DFE taps would take many taps. Shown is the unequalized REF pulse with $p_{-1}..p_{10}$.

TX FFE. A FIR filter in the driver pre-distorts the waveform (Chapter 6). It handles pre- and post-cursors and cannot amplify receiver noise, but with a fixed peak swing it can only *lower* the low-frequency level, so the eye arrives smaller.

RX CTLE. An analog high-pass stage at the receiver input (Chapter 7). It is cheap, fast and adaptable, and reshapes the long tail and part of the pre-cursor, but it boosts high-frequency noise and crosstalk with the signal.

RX FFE. A receiver FIR, from analog delay cells or after an ADC. It amplifies noise like a CTLE, and needs accurate 200 ps analog delays or a power-hungry ADC. Open-SERDES does not use it.

DFE. Subtracts the ISI that already-decided bits will cause (Chapter 8). Its feedback is a clean ± 1 decision, so it adds no noise, but it cancels only *post*-cursors and its first tap must close a loop within one UI.

Figure 5.4 reads the REF pulse in these terms. Each zone of cursors has a natural owner.

Table 5.1. Four equalizers, four sets of costs. Power and complexity are relative, for a 5 Gb/s link.

	Cursors it fixes	Frequency view	Noise, crosstalk	Power	Main limit
TX FFE	pre + post (as many as taps)	lowers LF, Nyquist kept	no RX noise gain; launch swing lost	low	peak swing; not adaptive without a back-channel
RX CTLE	long tail, some pre-cursor	boosts HF over LF	amplifies HF noise and crosstalk	low	peaking range and linearity
RX FFE	pre + post	like a CTLE, discrete taps	amplifies noise	high	analog delays or ADC power
DFE	post only (one per tap)	nonlinear (no $H(f)$)	no noise gain	medium	tap-1 timing, error propagation

5.4 Preview: who does how much?

None of the four is free: the TX FFE costs swing, the CTLE noise and linearity, the DFE timing margin and loop power. The design question is how to *split* the work of undoing 17.4 dB between them. Chapters 6–8 build each equalizer, and Section 8.5 answers the split with numbers for REF and for 5–30 dB channels.

▶ Design rule of thumb

Give each equalizer the job it does most cheaply. The TX pre-tap takes the pre-cursor, the DFE takes the first one or two post-cursors without adding noise, and a linear equalizer (CTLE or TX post-tap) flattens the long tail. Any equalizer pushed far beyond its natural job costs more than it returns.

KEY TAKEAWAYS

- Loss at f_N spreads each bit into its neighbours. On REF the ISI sum exceeds the main cursor, so the PDA eye is -0.161 V.
- Equalization aims for a cascade that is flat up to f_N. In the time domain that means zero ISI cursors.
- A zero-forcing inverse multiplies the noise by $\int |H_{eq}|^2$. That is 12.1 dB on REF up to f_N and 22.5 dB up to $2f_N$.
- In the zero-phase cable model, $y_{max} = (2V_0/\pi)\arctan(\pi T_b/k_d l)$. At 15 dB of loss the main cursor is 0.47 and the eye is already closed.
- The TX FFE, CTLE, RX FFE and DFE differ in which cursors they fix, whether they amplify noise, and what limits them.
- The partition between them is a design decision. Section 8.5 makes it for Open-SERDES.

CHAPTER 6

Transmit FFE

The transmitter knows every bit before it leaves the chip, so it can pre-distort the waveform to cancel the ISI the channel is about to add.

IN THIS CHAPTER YOU WILL LEARN

- Describe the TX feed-forward equalizer (FFE) as a 3-tap FIR filter, and write its output and its effect on the pulse response.
- Derive the FFE frequency response and the de-emphasis in dB, and explain the peak-swing constraint $\sum|c_k| = 1$.
- Solve for zero-forcing taps as a small matrix equation, and say when least squares does better.
- Show on the REF channel why zero-forcing taps are not eye-optimal (Lab 3).
- Pick practical settings: standard presets, tap resolution, and the swing the FFE costs.

6.1 A FIR filter in the driver

The TX FFE is a finite-impulse-response filter that acts on the symbol stream $a_n \in \{-1, +1\}$ before the driver (Figure 6.1). With three taps, the driver output during UI n is

$$x_n = A\,(c_{-1}\,a_{n+1} + c_0\,a_n + c_1\,a_{n-1})\,. \tag{6.1}$$

The **pre-tap** c_{-1} weights the *next* bit. The transmitter already knows that bit, because in hardware the whole chain is delayed by one UI. The **main tap** is c_0 and the **post-tap** is c_1. The general $(2M+1)$-tap version is $H(z) = \sum_{k=0}^{2M} \alpha_{-M+k} z^{-k}$ with $z = e^{j\omega T_b}$. It has no feedback, so it is unconditionally stable.

Since the channel is linear, sending x_n is the same as sending three scaled, shifted copies of each bit. The pulse response seen at the receiver is therefore the convolution of the tap list with the cursor list:

$$q_k = \sum_{j=-1}^{1} c_j\,p_{k-j} = c_{-1}p_{k+1} + c_0 p_k + c_1 p_{k-1}. \tag{6.2}$$

Equation (6.2) is all we need. Every FFE question in this chapter is a question about the new cursors q_k.

6.2 De-emphasis, pre-emphasis and the swing limit

The driver's output swing is fixed by its tail current and load at $\pm A$. By Equation (6.1), the largest output occurs when all terms add, so it equals $A\sum|c_k|$. To keep the driver within its swing we require the **peak-power constraint**:

$$|c_{-1}| + |c_0| + |c_1| = 1. \tag{6.3}$$

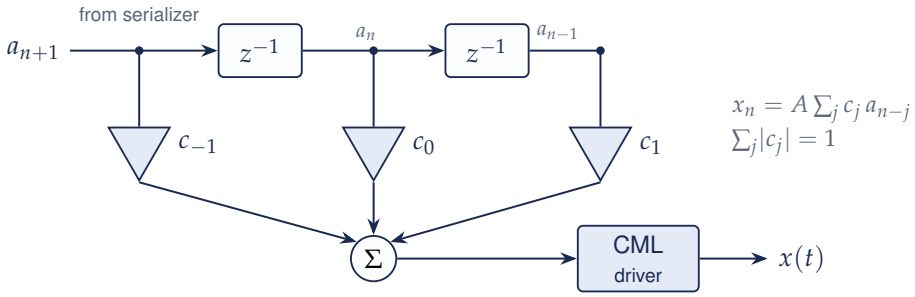


Figure 6.1. A 3-tap TX FFE is a short delay line with weighted taps. Each bit is sent three times: once early with weight c_{-1}, once on time with c_0, and once late with c_1. In a CML driver the taps are parallel current-steering slices that share the output load (Section 9.5).

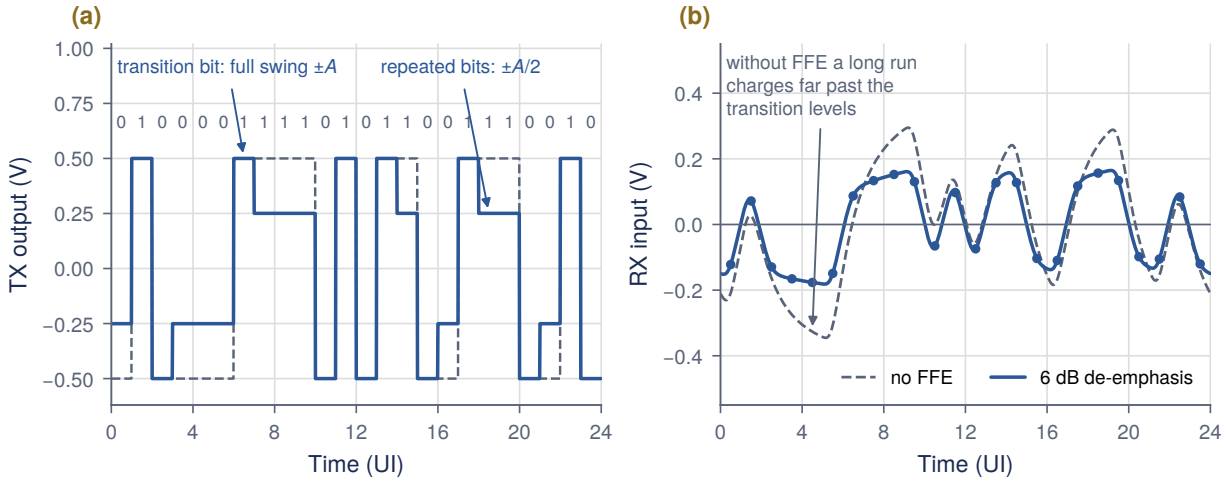


Figure 6.2. De-emphasis keeps transitions at full swing and halves the repeated bits. (a) TX output for a 24-bit pattern without FFE, and with 6 dB of 2-tap de-emphasis ($c_0 = 0.751$, $c_1 = -0.249$). (b) The same signals after the REF channel. Without FFE a long run charges the line far beyond the transition levels, so the next transition cannot cross in time. With de-emphasis the run and transition levels are much closer, and the dots (one sample per UI) form a cleaner pattern.

For a low-pass channel the side taps are negative. Consider a 2-tap FFE with $c_1 < 0$. A *transition* bit ($a_n = -a_{n-1}$) gets $|x_n| = A(c_0 + |c_1|) = A$, the full swing. A *repeated* bit gets $A(c_0 - |c_1|)$, which is smaller. This is **de-emphasis**: transitions keep full swing and runs of identical bits are attenuated (Figure 6.2).

Pre-emphasis has exactly the same filter shape, but it is scaled the other way. It keeps $c_0 = 1$ and lifts the transitions to $A(1 + |c_1|)$. A driver that already runs at its maximum swing has no headroom for that, so a practical TX de-emphasizes.

Frequency response. The FFE transfer function is $H_{\mathrm{FFE}}(f) = \sum_k c_k\, e^{-j2\pi f k T_\mathrm{b}}$. For two taps, $|H|^2 = c_0^2 + c_1^2 + 2c_0c_1\cos(2\pi f T_\mathrm{b})$. Normalized to $c_0 = 1$, $c_1 = \alpha$, with $-1 < \alpha < 0$:

KEY EQUATION · 2-tap FFE response and de-emphasis

$$|H_{\mathrm{FFE}}(f)| = \sqrt{1 + \alpha^2 + 2\alpha\cos(2\pi f T_\mathrm{b})}, \qquad \mathrm{DE} = 20\log_{10}\frac{|H(f_\mathrm{N})|}{|H(0)|} = 20\log_{10}\frac{1-\alpha}{1+\alpha} \quad (6.4)$$

With $\sum|c| = 1$ the response is $c_0 + c_1 = 1 - 2|c_1|$ at DC and $c_0 - c_1 = 1$ at f_N, so $\mathrm{DE} = -20\log_{10}(1 - 2|c_1|)$.

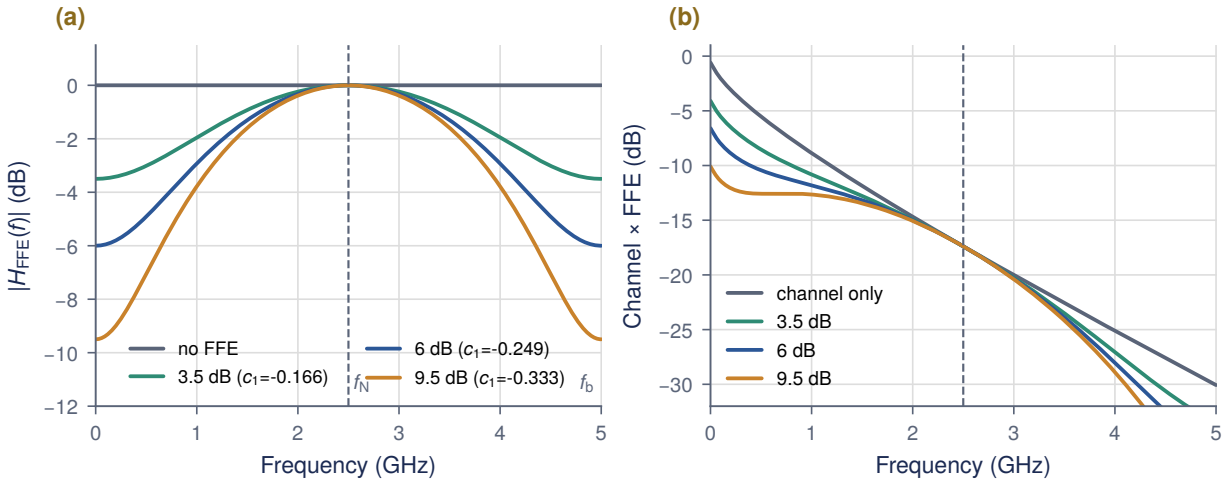


Figure 6.3. A TX FFE pulls the low frequencies down; it never lifts f_N. (a) Equation (6.4) for 0, 3.5, 6 and 9.5 dB of de-emphasis. The response peaks at f_N and is periodic in f_b. (b) REF channel times FFE. Each setting removes as many dB of the low-frequency excess as its de-emphasis, so the cascade gets flatter but stays at -17.4 dB at f_N.

As a check, Equation (6.4) gives $1+\alpha$ at DC and $1-\alpha$ at f_N. The response rises monotonically from DC to f_N and repeats with period $1/T_b = 5$ GHz (Figure 6.3a). It never rises above 0 dB. It only pulls the low frequencies down.

WORKED EXAMPLE 6.1 · The PCIe de-emphasis presets in volts

PCIe Gen2 at 5 GT/s specifies TX de-emphasis of −3.5 dB and −6 dB. For -3.5 dB, $1-2|c_1| = 10^{-3.5/20} = 0.668$, so $c_1 = -0.166$ and $c_0 = 0.834$. For -6 dB, $1-2|c_1| = 0.501$, so $c_1 = -0.249$ and $c_0 = 0.751$.

With $A = 0.5$ V (1 Vppd), transition bits swing ± 500 mV in both cases. Repeated bits swing $\pm 0.668 \times 500 = \pm 334$ mV at -3.5 dB and ± 251 mV at -6 dB. A 9.5 dB setting would need $c_1 = -1/3$ and leave repeated bits at ± 167 mV.

In a CML driver the taps steer a fixed total tail current, so the supply current is the same at every setting. De-emphasis does not save power. It only redistributes the swing.

6.3 Solving for the taps

Zero forcing. Choose the three taps so that the equalized pulse has $q_{-1} = 0$, $q_0 = 1$ and $q_1 = 0$. Writing Equation (6.2) for $k = -1, 0, 1$ gives three linear equations in three unknowns. Each row is one cursor of the equalized pulse, and each column is the pulse shifted by one tap:

KEY EQUATION · Zero-forcing FFE taps

$$\begin{bmatrix} p_0 & p_{-1} & p_{-2} \\ p_1 & p_0 & p_{-1} \\ p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad c_k \leftarrow \frac{c_k}{\sum_j |c_j|}. \tag{6.5}$$

The matrix has constant diagonals: it is a **Toeplitz** matrix. A $(2M+1)$-tap FFE gives a $(2M+1) \times (2M+1)$ system of the same form. The normalization step comes last. It scales the whole equalized pulse down, but it does not change where the zeros are.

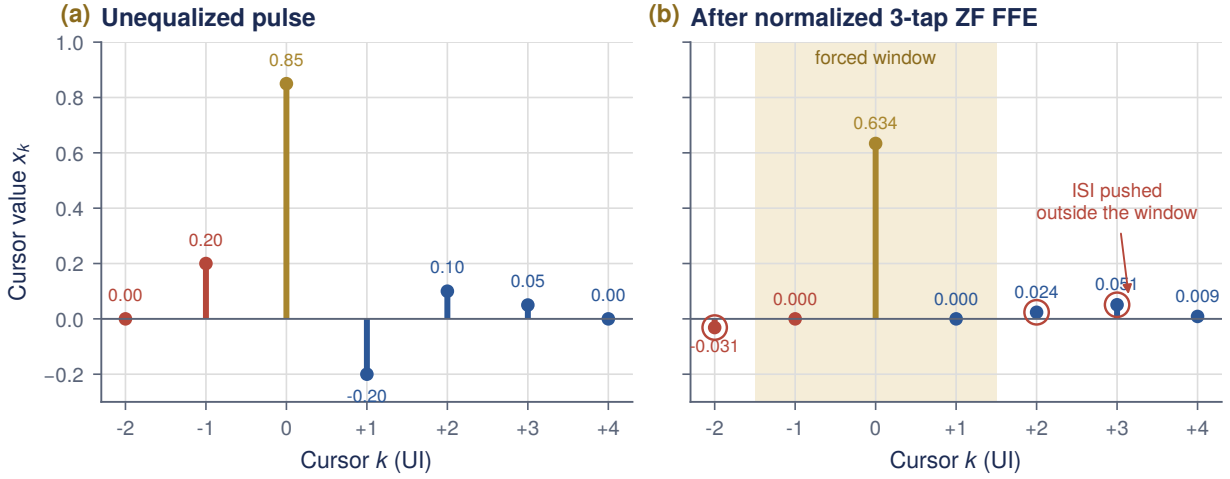


Figure 6.4. Zero forcing zeros only the cursors it is told to. (a) The example pulse. (b) After the normalized 3-tap ZF FFE, the cursors at $k = \pm 1$ are exactly zero, but ISI appears at $k = -2$ and grows at $k = +3$ (circled). Normalization to $\sum|c| = 1$ lowers the main cursor to 0.634.

WORKED EXAMPLE 6.2 · A 3-tap zero-forcing FFE

Take an example pulse with $x_{-1} = 0.2$, $x_0 = 0.85$, $x_1 = -0.2$, $x_2 = 0.1$ and $x_3 = 0.05$. Equation (6.5) reads

$$\begin{bmatrix} 0.85 & 0.2 & 0 \\ -0.2 & 0.85 & 0.2 \\ 0.1 & -0.2 & 0.85 \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \boldsymbol{\alpha} = [-0.248,\ 1.053,\ 0.277].$$

Since $\sum|\alpha| = 1.578$, the normalized taps are $[-0.157,\ 0.667,\ 0.176]$. The equalized cursors for $k = -2..4$ are $[-0.031,\ 0,\ 0.634,\ 0,\ 0.024,\ 0.051,\ 0.009]$. With $2A = 1$, the PDA eye opens from $0.85 - 0.55 = 0.30$ to $0.634 - 0.115 = 0.519$ (Figure 6.4). Three lessons:

- The post-tap is *positive*, because x_1 is negative. ZF inverts whatever ISI is there. This FFE is not even high-pass: it is -3.3 dB at DC and -3.8 dB at f_N.
- ISI is pushed outside the forced window. A new cursor appears at $k = -2$, and $k = 3$ grows from 6 % to 8 % of the main cursor.
- Normalization costs main cursor. It drops from 0.85 to 0.634.

Least squares and MMSE. Zero forcing looks at only $2M + 1$ cursors. **Least squares** (LS) uses the whole pulse. Stack shifted copies of the full pulse as the columns of a matrix $\mathbf{H}$ (one row per output cursor), then minimize $\|\mathbf{Hc} - \mathbf{y}_{\text{des}}\|^2$, where $\mathbf{y}_{\text{des}}$ is 1 at the main cursor and 0 elsewhere. The solution is $\mathbf{c} = (\mathbf{H}^{\mathsf{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{y}_{\text{des}}$, which is then normalized. For a receiver FIR, where noise passes through the filter, the noise-aware **MMSE** form adds $(\sigma_n^2/\sigma_a^2)\mathbf{I}$ to $\mathbf{H}^{\mathsf{T}}\mathbf{H}$. Palermo's 10 Gb/s example gives $\mathbf{c} = [-0.131,\ 0.595,\ -0.274]$ [22]. Its DC gain is $\sum c = 0.190$ (-14.4 dB) and its Nyquist gain is $|\sum(-1)^k c_k| = 1$, so it applies 14.4 dB of de-emphasis, all of it paid in swing.

The REF channel. Lab 3 applies Equation (6.5) to the REF cursors $p_{-1} = 0.0486$, $p_0 = 0.3396$ and $p_1 = 0.1817$. The 3-tap ZF taps are $[-0.0853,\ 0.5959,\ -0.3188]$, which is 14.3 dB of boost. The 2-tap ZF taps, forcing only $q_1 = 0$, are $[0.6515,\ -0.3485]$, or 10.4 dB. Figure 6.5 shows the result for PRBS7. The eye goes from −77.8 mV without FFE to 162.7 mV (0.89 UI) with the 2-tap FFE,

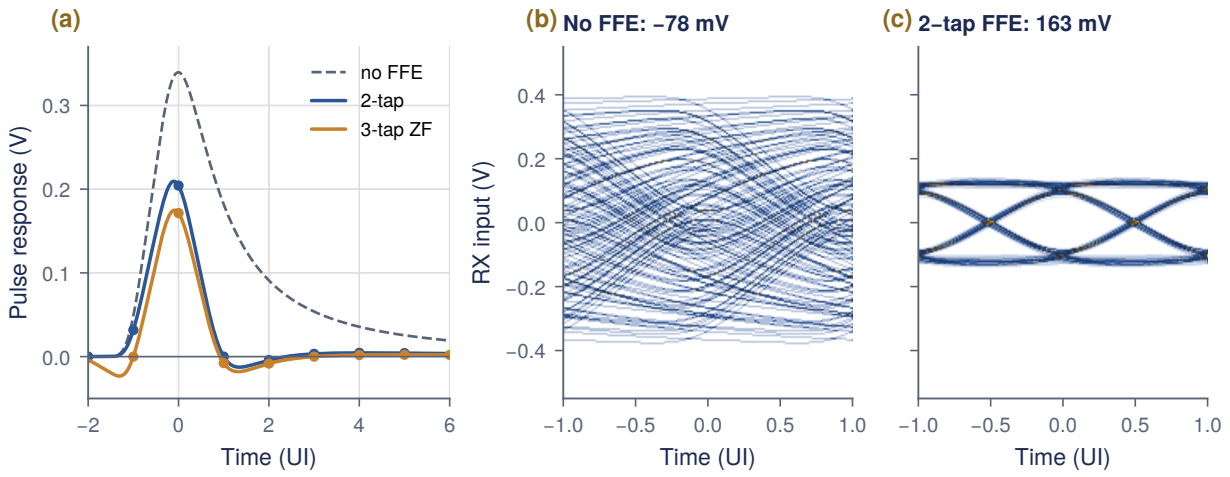


Figure 6.5. On REF, a 2-tap FFE opens the eye; the 3-tap ZF FFE does slightly worse. (a) Pulse responses without FFE, with the 2-tap ZF taps $[0.652, -0.349]$ and with the 3-tap ZF taps $[-0.085, 0.596, -0.319]$. Dots mark the cursors. (b, c) PRBS7 eyes at the RX input, plotted on the same scale. The open eye is 163 mV but only about a third of the unequalized swing. That is the price of equalizing at the TX.

but only to 144.8 mV (0.83 UI) with the 3-tap ZF FFE. (With PRBS13 the 2-tap eye is 148 mV. Its PDA and statistical eyes, 126 mV and 131 mV at 10^{-12}, are worked out in Sections 4.2 and 4.3.)

So the 3-tap zero-forcing FFE is *worse* than the 2-tap one. Under $\sum|c| = 1$, every unit of tap weight comes out of the main cursor, which drops from 0.340 to 0.175 V with three taps. The pre-tap spends 0.085 of the budget to null $p_{-1} = 0.049$ exactly, and it leaks p_2 into q_1. A grid search for the 3-tap setting with the largest PDA eye finds $[-0.04,\ 0.61,\ -0.35]$. Its PRBS7 eye is 163.6 mV, only 1 mV better than the 2-tap FFE. *Zero forcing is a recipe, not an optimum.* Judge taps by the eye they produce.

>_ Lab 3: TX FFE

`lab3_tx_ffe.py` computes the ZF taps with `sdp.forcing_ffe`, normalizes them, and measures the eyes quoted above. Change the pre-tap by hand and watch the eye move in both directions (Section 16.4).

6.4 Practical notes

Presets. Standards define a few TX presets rather than free taps, because a transmitter cannot measure the channel. PCIe Gen2 uses −3.5 dB and −6 dB (Worked Example 6.1). Open-SERDES offers the same two presets plus a small pre-tap. Section 8.5 shows that −6 dB is the right preset for REF once a DFE follows.

Tap resolution. A CML driver builds each tap from N equal current slices, so the taps move in steps of $1/N$. For the REF 2-tap FFE, 8 slices give $[0.625, -0.375]$ and a 162.6 mV eye. 32 slices give $[0.656, -0.344]$ and 162.3 mV. The ideal setting gives 162.7 mV. The eye is flat near its optimum, so 4–5 bits (16–32 slices) are plenty at this data rate. Palermo quotes 4–6 bits as typical [22].

▶ What this means for your design

A TX FFE does not boost the high frequencies. It *lowers the low frequencies*, because the driver is already at its peak swing. The 10.4 dB ZF setting on REF leaves repeated bits at 30 % of full swing, and the eye at the receiver is about a third of the unequalized swing. Receiver noise, crosstalk and sampler offset do not shrink with it, so the SNR falls by the de-emphasis. Keep the TX FFE at moderate de-emphasis (3.5–6 dB) with a small pre-tap, and let the receiver equalizers do the rest. How the driver builds the taps is covered in Section 9.5.

KEY TAKEAWAYS

- A TX FFE is a short FIR filter. Its taps convolve with the cursor list: $q_k = \sum_j c_j p_{k-j}$.
- The peak-swing constraint $\sum|c_k| = 1$ forces de-emphasis. Transitions keep full swing and repeated bits are attenuated.
- A 2-tap FFE has $|H| = \sqrt{1 + \alpha^2 + 2\alpha\cos 2\pi f T_\mathrm{b}}$, with de-emphasis $-20\log_{10}(1 - 2|c_1|)$. PCIe's -3.5 and -6 dB correspond to $c_1 = -0.166$ and -0.249.
- Zero-forcing taps solve a small Toeplitz system. Least squares uses the whole pulse, and MMSE adds noise.
- On REF, 2-tap ZF gives 162.7 mV and 3-tap ZF gives 144.8 mV. Zero forcing is not eye-optimal.
- The FFE costs launch swing and SNR, not power. Use moderate presets and 4–5-bit taps.

CHAPTER 7

Continuous-Time Linear Equalizer

A CTLE is a high-pass filter built into the first receiver stage. It lifts the frequencies the channel attenuated by giving up gain at low frequencies.

IN THIS CHAPTER YOU WILL LEARN

- Compare passive RC, RLC and degenerated-pair CTLEs and know why the degenerated pair is the workhorse.
- Derive the degenerated-pair transfer function, with its zero, two poles, DC gain and peaking.
- Turn a peaking target into R_S, C_S, R_D and C_L, and respect the gain–bandwidth limit.
- Choose the peaking for the REF channel, and recognize over-peaking.
- Design an SG13G2 HBT CTLE, explain why its realized boost falls short of the ideal, and make it programmable from 0 to 12 dB.

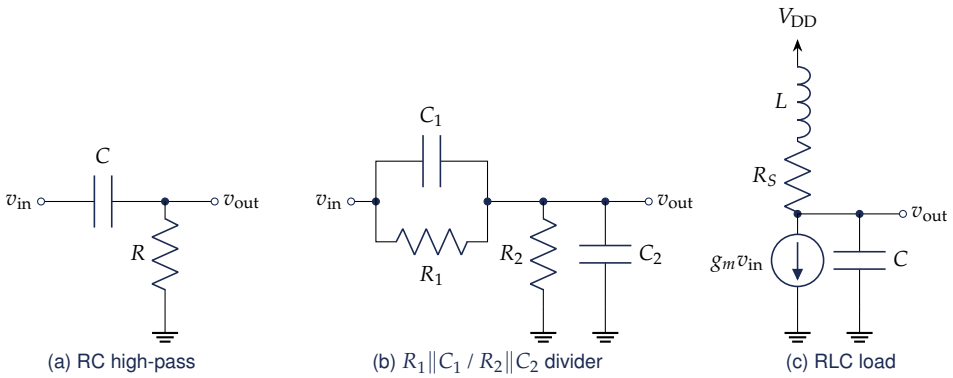
7.1 From RC to RLC: three passive ideas

The channel is a low-pass filter, so its inverse must be a high-pass filter. Figure 7.1 shows three passive ways to build one.

RC high-pass. $H(s) = sRC/(1+sRC)$ has zero gain at DC. A long run of identical bits decays toward 0 V, so the baseline wanders with the data. On its own it is unusable.

Over-compensated divider. With a series $R_1 \| C_1$ and a shunt $R_2 \| C_2$,

$$H(s) = \frac{R_2}{R_1+R_2} \cdot \frac{1+sR_1C_1}{1+s\,(R_1\|R_2)(C_1+C_2)}, \qquad \frac{\omega_p}{\omega_z} = \frac{R_1+R_2}{R_2} \cdot \frac{C_1}{C_1+C_2}. \tag{7.1}$$



(a) RC high-pass (b) $R_1\|C_1$ / $R_2\|C_2$ divider (c) RLC load

Figure 7.1. Three passive ways to make a high-pass equalizer. (a) A plain RC high-pass has no DC gain. (b) A deliberately over-compensated divider has a finite DC gain but attenuates at every frequency. (c) An inductively peaked load gives real boost, but with complex poles that ring.

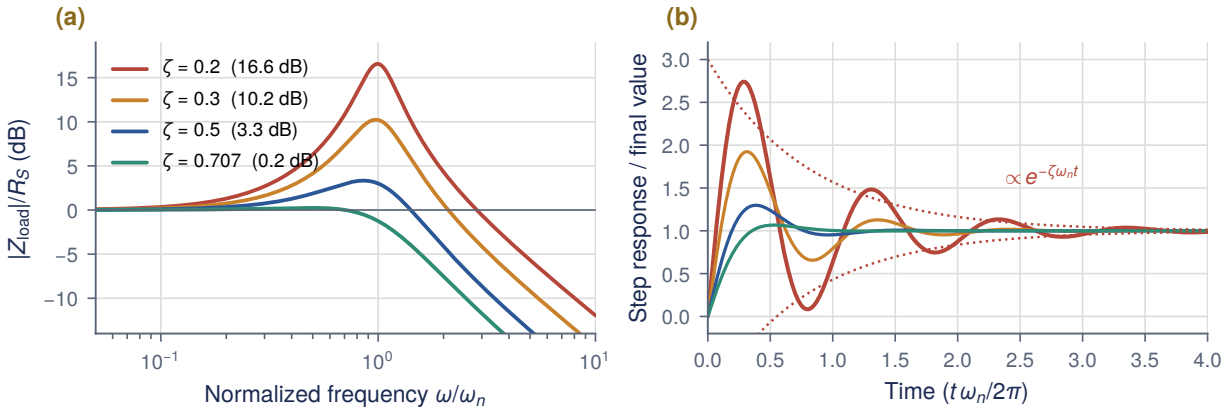


Figure 7.2. An RLC load buys boost with ringing. (a) Load impedance from Equation (7.2) for several damping factors. $\zeta = 0.2$ gives 16.6 dB of peaking at ω_n. (b) Step responses. At $\zeta = 0.2$ the output overshoots to 2.7 times its final value and rings for several periods inside the $e^{-\zeta\omega_n t}$ envelope (dotted).

The DC gain is $R_2/(R_1+R_2)$ and the high-frequency gain is $C_1/(C_1+C_2)$. With $R_1C_1 = R_2C_2$ it is a flat compensated probe; a larger R_1C_1 makes it an equalizer. It is very linear, but its gain is below 1 everywhere, which degrades the SNR at the next stage.

RLC load. Put an inductor in series with the load resistor R_S of a differential pair and let C be the node capacitance. With $\omega_n = 1/\sqrt{LC}$ and $\zeta = \frac{R_S}{2}\sqrt{C/L}$, the load impedance is

$$Z(s) = R_S\,\frac{\omega_n}{2\zeta}\cdot\frac{s+2\zeta\omega_n}{s^2+2\zeta\omega_n s+\omega_n^2}, \qquad \frac{|Z(j\omega_n)|}{R_S} = \frac{\sqrt{1+4\zeta^2}}{4\zeta^2}. \tag{7.2}$$

With $\zeta = 0.2$ the boost is 6.7, or 16.6 dB (Figure 7.2). That is a serious drawback: the complex poles sit close to the $j\omega$ axis, so the step response rings as $e^{-\zeta\omega_n t}\cos(\sqrt{1-\zeta^2}\,\omega_n t)$, and the ringing is ISI of its own. Mild inductive peaking with $\zeta \approx 0.5$–0.7 is still useful. It extends bandwidth without much ringing, and we return to it in Section 7.4.

7.2 The source-degenerated differential pair

The workhorse CTLE is a differential pair with a parallel $R_S\|C_S$ between its sources or emitters (Figure 7.3). At low frequency R_S degenerates the pair and lowers its gain. At high frequency C_S shorts R_S, and the full gain $g_m R_D$ returns. The zero and first pole are set by R_S and C_S, so R_S can be a triode MOSFET and C_S a varactor, giving a tunable CTLE [22].

Transfer function of the degenerated pair

In the half circuit the source sees $Z_S = \frac{R_S}{2}\big\|\frac{1}{2sC_S} = \dfrac{R_S/2}{1+sR_SC_S}$. Series feedback divides the transconductance by $1+g_mZ_S$:

$$G_m(s) = \frac{g_m}{1+g_mZ_S} = \frac{g_m\,(1+sR_SC_S)}{1+g_mR_S/2+sR_SC_S} = \frac{g_m}{1+g_mR_S/2}\cdot\frac{1+sR_SC_S}{1+\dfrac{sR_SC_S}{1+g_mR_S/2}}.$$

The drain load is $Z_D = R_D/(1+sR_DC_L)$, and $H = G_mZ_D$ (the sign inversion is absorbed by the output polarity).

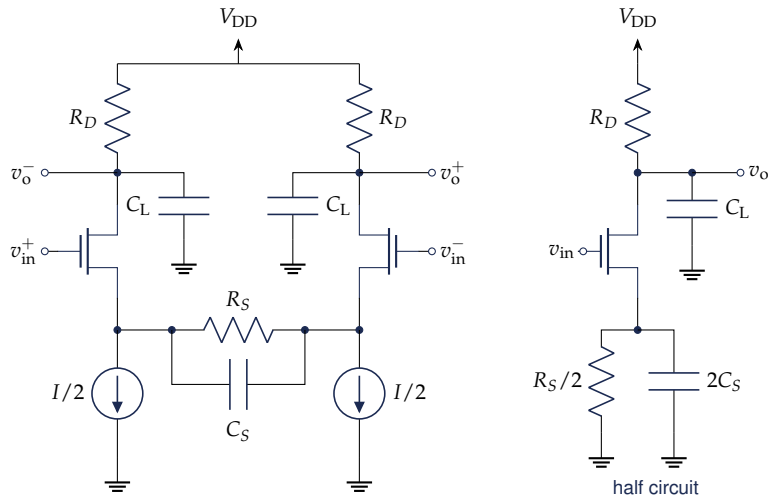


Figure 7.3. The degenerated differential pair is the workhorse CTLE. $R_S \| C_S$ between the sources sets the zero and the first pole. $R_D \| C_L$ at the drains sets the second pole. The split tail ($2 \times I/2$) keeps DC current out of R_S. In the half circuit, the degeneration appears as $R_S/2$ in parallel with $2C_S$.

KEY EQUATION · Degenerated-pair CTLE

$$H_{CTLE}(s) = \frac{g_m R_D}{1 + g_m R_S/2} \cdot \frac{1 + s/\omega_z}{(1 + s/\omega_{p1})(1 + s/\omega_{p2})}, \tag{7.3}$$

$$\omega_z = \frac{1}{R_S C_S}, \qquad \omega_{p1} = \frac{1 + g_m R_S/2}{R_S C_S}, \qquad \omega_{p2} = \frac{1}{R_D C_L}$$

DC gain $= g_m R_D/(1 + g_m R_S/2)$. Gain between ω_{p1} and $\omega_{p2} \approx g_m R_D$. Ideal peaking $= \omega_{p1}/\omega_z = 1 + g_m R_S/2$.

R_S sets the peaking and the DC gain, and barely moves $\omega_{p1} \approx g_m/(2C_S)$ when $g_m R_S/2 \gg 1$. C_S moves ω_z and ω_{p1} together at constant peaking (Figure 7.4a). In this book the "peaking" of a setting means the *ideal* ratio ω_{p1}/ω_z, as in `ctle()` of `link.py`, which fixes $f_{p1} = f_N$, $f_{p2} = 10\,\text{GHz}$ and 0 dB high-frequency gain. The boost realized at f_N is smaller: the 12 dB setting gives 9.0 dB, because the first pole already costs 3 dB there (Figure 7.4b).

WORKED EXAMPLE 7.1 · From a peaking target to component values

Lab 4 finds 14 dB best for REF without a DFE (Section 7.3). Target $a = 10^{14/20} = 5.01$, $f_{p1} = f_N$, $f_{p2} = 10\,\text{GHz}$ and 0 dB high-frequency gain, with $g_m = 20\,\text{mS}$ per side. Peaking: $R_S = 2(a-1)/g_m = 401\,\Omega$. Zero: $f_z = f_{p1}/a = 0.499\,\text{GHz}$, so $C_S = 1/(2\pi f_z R_S) = 0.795\,\text{pF}$. Gain: $g_m R_D = 1$ gives $R_D = 50\,\Omega$, and $C_L = 1/(2\pi f_{p2} R_D) = 318\,\text{fF}$. The DC gain is $g_m R_D/a = 0.20$ (−14 dB). This circuit only *loses* gain at low frequency; a real front end adds gain with more $g_m R_D$ or a following stage.

Gain–bandwidth. Multiplying the mid-band gain by the second pole gives

$$(g_m R_D) \cdot \omega_{p2} = \frac{g_m}{C_L}. \tag{7.4}$$

This is the gain–bandwidth product of a plain differential pair; degeneration does not change it. A high ω_{p2} together with a large peak gain needs more g_m (current) or a smaller C_L (smaller next stage, inductive peaking).

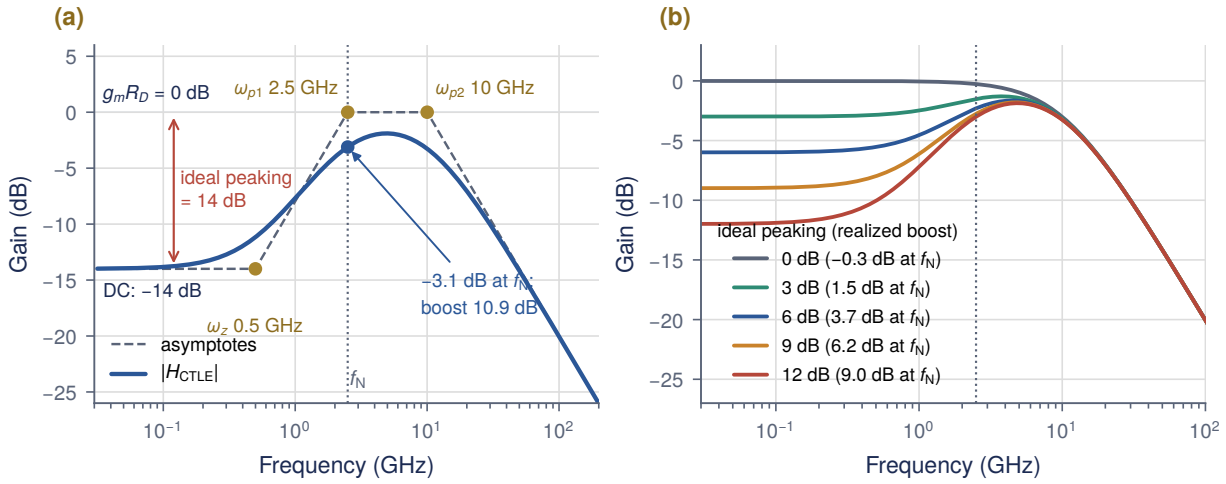


Figure 7.4. Zero, poles and peaking of the degenerated pair. (a) Equation (7.3) for the Lab 4 example circuit ($g_m = 20$ mS, $R_S = 401\,\Omega$, $C_S = 0.795$ pF, $R_D = 50\,\Omega$, $C_L = 318$ fF), with its asymptotes. The ideal peaking is 14 dB, but the boost at f_N is only 10.9 dB. (b) The `ctle()` family from 0 to 12 dB, as used in the REF studies. The legend gives the boost realized at f_N.

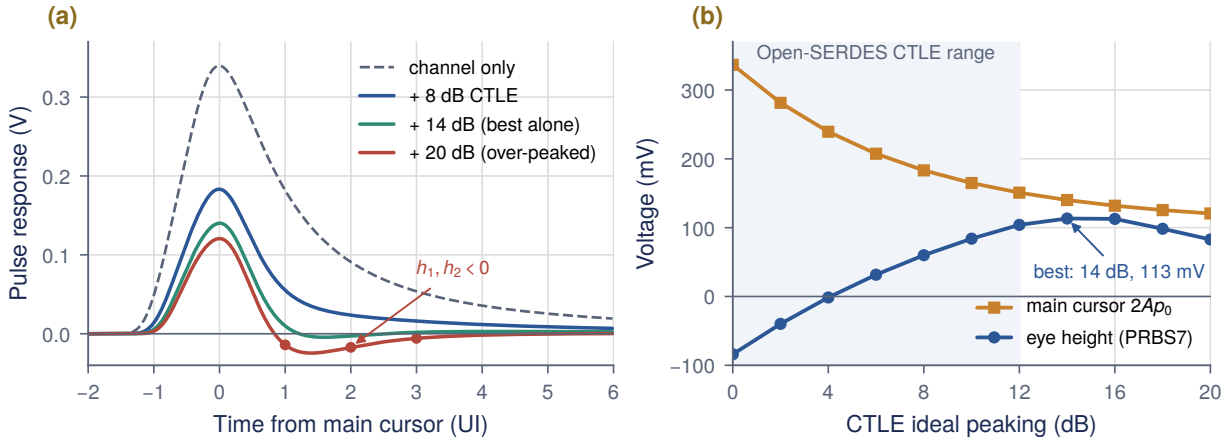


Figure 7.5. More peaking is better only up to a point. (a) REF pulse response alone and after 8, 14 and 20 dB CTLEs. At 20 dB the first post-cursors go negative, which is over-equalization. (b) PRBS7 eye height and main cursor against peaking (Lab 4). The eye is best at 14 dB (113 mV), just outside the Open-SERDES range (shaded). The main cursor shrinks at every step.

7.3 The CTLE on the REF channel

Lab 4 cascades REF with `ctle(pk)` and sweeps the peaking (Figure 7.5). Without a CTLE the PRBS7 eye is −84 mV. It opens at about 4 dB, reaches 104 mV at 12 dB and peaks at 113.3 mV (0.94 UI) at 14 dB. Beyond 16 dB it closes again. Meanwhile the main cursor $2Ap_0$ falls steadily, from 337 mV to 140 mV at 14 dB. This is the CTLE "boosting" by suppressing the low frequencies.

▶ What this means for your design

- **Over-peaking returns the ISI with the opposite sign.** Above 16 dB, p_1 and p_2 turn negative and the eye closes again.
- **Every dB of peaking costs main cursor.** From 0 to 14 dB the main cursor drops by 7.6 dB. Noise that enters after the CTLE, such as sampler and DFE-summer noise, does

not drop with it, and the CTLE's high-frequency gain lifts noise and crosstalk relative to the signal.

- **Pre-cursors shrink but do not vanish.** In our model p_{-1} falls from 0.049 to 0.006 at 14 dB. A larger pre-cursor, for example from a reflection, is a job for the TX pre-tap.
- With a DFE behind it, the CTLE should be tuned for the best eye *after* the DFE. Let the DFE take p_1 instead of peaking it away (Section 8.5).

>_ Lab 4: CTLE

`lab4_ctle.py` sweeps the peaking from 0 to 20 dB, prints the eye and the cursors, and converts the best setting into R_S, C_S, R_D and C_L (Section 16.5).

7.4 An HBT CTLE in SG13G2

Open-SERDES builds its CTLE from SG13G2 SiGe HBTs. A bipolar transistor gives $g_m = I_C/V_T = 38.7$ mS per mA, several times more than a MOSFET at the same current, and its $f_T \approx 300$ GHz keeps device capacitance small. The circuit is Figure 7.3 with npn devices, emitter degeneration $R_E \| C_E$ and collector loads R_C. Each V_{CE} must stay below the BV_{CEO} of about 1.6–1.7 V on the 2.5 V analog supply.

WORKED EXAMPLE 7.2 · An SG13G2 HBT CTLE with 10 dB of ideal peaking

Bias with $I_{tail} = 2$ mA (1 mA per side): $g_m = 1\,\text{mA}/25.85\,\text{mV} = 38.7$ mS.

- 10 dB ideal peaking: $1 + g_m R_E/2 = 3.16$, so $R_E = 2(2.16)/g_m = 112\,\Omega$.
- Put f_{p1} at f_N: $f_z = 2.5/3.16 = 0.79$ GHz, so $C_E = 1/(2\pi f_z R_E) = 1.8$ pF.
- $R_C = 200\,\Omega$ and $C_L = 100$ fF give $f_{p2} = 7.96$ GHz.
- DC gain $= g_m R_C/3.16 = 2.45$ (7.8 dB). $|H(j2\pi f_N)| = 14.8$ dB.

The **realized boost** at f_N is $14.8 - 7.8 = 7.0$ dB, not 10 dB. Each factor of Equation (7.3) evaluated at f_N shows why: the zero contributes $20\log_{10}|1 + j\,2.5/0.79| = +10.4$ dB, the first pole -3.0 dB because it sits at f_N, and the second pole -0.4 dB. The response peaks at 4.3 GHz, well above f_N.

Fix. The boost at f_N is set by f_N/f_z, minus whatever the poles take away. Move f_{p1} up to $2f_N$ and lower f_z to 0.70 GHz. That needs $1 + g_m R_E/2 = 7.1$ (17 dB ideal), so $R_E = 317\,\Omega$ and $C_E = 0.72$ pF. The realized boost is then 10.0 dB. The price is DC gain, which falls to 0.7 dB (Figure 7.6). A second option is inductive (shunt) peaking in the collector loads. It pushes ω_{p2} out and adds a little gain near f_N (Section 7.1).

Programmability. The Open-SERDES CTLE must cover 0–12 dB of peaking for channels shorter and longer than REF. Figure 7.7 shows the usual way to do it: binary-weighted banks of R_E and C_E segments, switched in parallel by MOS switches under a digital code. Closing more R_E segments lowers $R_{E,\text{eff}}$, which reduces the peaking and raises the DC gain. A fully closed bank (0 dB) makes it a flat amplifier. C_E segments are switched together with R_E, so ω_{p1} stays near $2f_N$. Keep the switches small. Their on-resistance adds to R_E and their junction capacitance adds to C_E, and both shift the corners. Include them in the **ngspice** netlist.

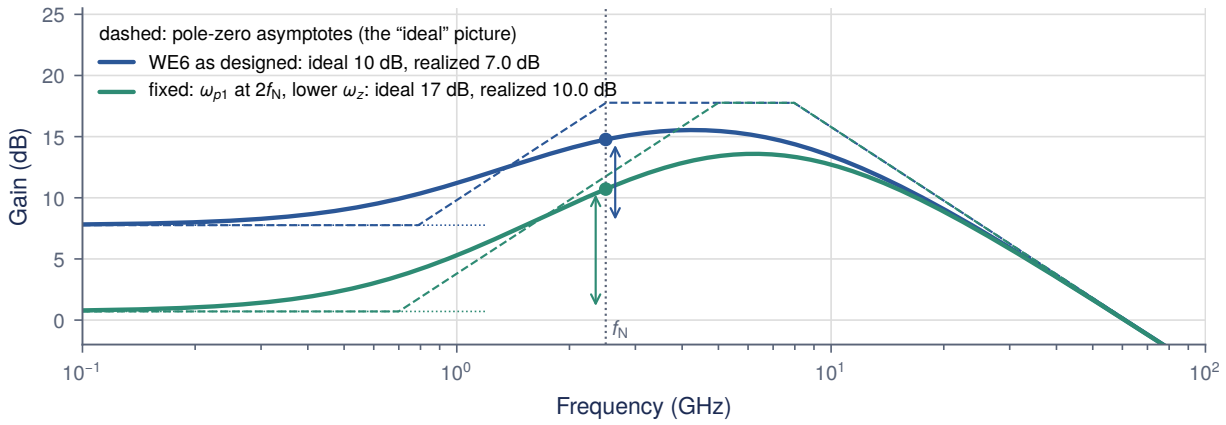


Figure 7.6. The pole-zero “ideal” overstates the boost at f_N. Blue: Worked Example 7.2 as designed. The asymptotes promise 10 dB, but only 7.0 dB is realized at f_N (arrow). Teal: the fixed design, with $f_{p1} = 2f_N$ and $f_z = 0.70$ GHz, realizes 10.0 dB at the cost of DC gain.

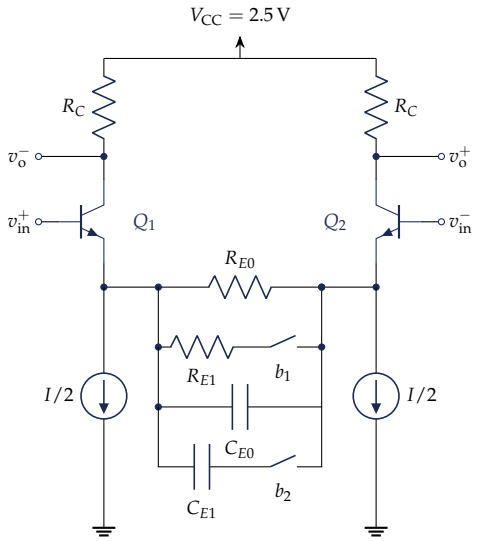


Figure 7.7. A programmable HBT CTLE. An SG13G2 npn pair with emitter degeneration banks. Closing switch b_1 puts R_{E1} in parallel with R_{E0}, lowering the peaking. Switch b_2 adds C_{E1} and lowers the corners. More binary-weighted segments (code $b_{3:0}$) cover 0–12 dB. The split tail keeps DC current out of the banks, so the switches carry no bias current.

Noise. The degeneration resistor adds $\sqrt{4kTR_E} = 1.4\,\text{nV}/\sqrt{\text{Hz}}$ for $R_E = 112\,\Omega$, against $\sqrt{2 \cdot 2kT/g_m} = 0.65\,\text{nV}/\sqrt{\text{Hz}}$ of collector shot noise from the pair. Over 5 GHz that is about 0.1 mV rms, far below the 10 mV slicer noise assumed in Lab 5. Offset, crosstalk and linearity limit a 5 Gb/s CTLE before thermal noise does.

Linearity and overload. With a split tail, the pair fully switches when the differential input exceeds about $I_{tail}R_E/2$ plus a few V_T. For Worked Example 7.2 that is $112+ \sim 100 \approx 0.2$ V. Long runs on REF arrive at nearly the full DC level, $\pm 0.93 \times 0.5\,\text{V} = \pm 0.47\,\text{V}$ differential. The CTLE would compress them, and a compressed CTLE breaks the linear superposition that the DFE taps rely on. Three remedies stack: TX de-emphasis lowers the long-run level (± 0.23 V at -6 dB); the fixed design’s $I_{tail}R_E/2 = 317$ mV widens the linear range; and an input attenuator or gain control handles short channels (Section 10.1). At high frequency C_E shorts R_E and the range shrinks toward that of a bare pair, about $\pm 2V_T$. That is tolerable, because the channel has already attenuated the high-frequency content by 17 dB.

Open-SERDES CTLE targets (derived here)

Ideal peaking 0–12 dB in 1–2 dB steps (≈ 9 dB realized at f_N); $f_{p1} \approx 1$–$2\,f_N$; $f_{p2} \geq 7.5$ GHz; DC gain ≥ 0 dB at 12 dB; LF linear range $\geq \pm 0.25$ V differential. Verify the realized boost in **ngspice**.

KEY TAKEAWAYS

- An RC high-pass has no DC gain and a passive divider has no gain at all. RLC peaking gives boost but rings when ζ is small (16.6 dB at $\zeta = 0.2$).
- The degenerated pair gives $\omega_z = 1/R_S C_S$, $\omega_{p1} = (1 + g_m R_S/2)/R_S C_S$ and $\omega_{p2} = 1/R_D C_L$, with ideal peaking $1 + g_m R_S/2$.
- The CTLE boosts by suppressing the low frequencies. Its gain–bandwidth is still g_m/C_L.
- On REF, the eye is best at 14 dB without a DFE (113.3 mV). Over-peaking turns the post-cursors negative and costs main cursor.
- Realized boost at f_N is well below the ideal when $f_{p1} = f_N$: 7.0 dB instead of 10 dB in Worked Example 7.2. Move f_{p1} up and f_z down to recover it.
- Make the HBT CTLE programmable with switched R_E/C_E banks, and check its linear range against the long-run input level.

CHAPTER 8

Decision-Feedback Equalization and Adaptation

A DFE removes post-cursor ISI by subtracting what the past decisions predict. It equalizes without amplifying noise, and it can learn its own taps.

IN THIS CHAPTER YOU WILL LEARN

- Derive the DFE from a linear IIR equalizer, and explain why the slicer in the loop removes noise enhancement.
- Compute DFE taps from a pulse response and combine a DFE with a TX pre-tap.
- Budget the tap-1 timing loop, and choose between direct feedback, half-rate and loop unrolling.
- Write the LMS and sign-sign LMS updates, and recognize when adaptation wanders.
- Split the equalization work between TX FFE, CTLE and DFE for Open-SERDES, with numbers.

8.1 The DFE principle

From IIR filter to DFE. Start from a one-tap *linear* feedback equalizer, $y(t) = x(t) - \alpha_1 y(t - T_\mathrm{b})$. Its transfer function is $H(z) = 1/(1 + \alpha_1 z^{-1})$, with

$$|H| = \frac{1}{\sqrt{1 + \alpha_1^2 + 2\alpha_1 \cos \omega T_\mathrm{b}}}. \tag{8.1}$$

For $0 < \alpha_1 < 1$ it rises from $1/(1+\alpha_1)$ at DC to $1/(1-\alpha_1)$ at f_N, so it could equalize, but it boosts noise like any linear equalizer and is unstable for $|\alpha_1| \geq 1$.

The key step is to put a **slicer** *inside* the loop. The fed-back value is then a clean decision $\hat{a}_{n-1} \in \{-1, +1\}$, the noise on y is never recirculated, and the DFE adds no noise. With N taps (Figure 8.1):

KEY EQUATION · N-tap DFE

$$z_n = y_n - \sum_{k=1}^{N} w_k \hat{a}_{n-k}, \qquad \hat{a}_n = \mathrm{sgn}(z_n), \qquad w_k = A\, p_k \tag{8.2}$$

Here y_n is the summer input in volts and the slicer threshold is 0. The DFE coefficients are the post-cursors: the ideal w_k is the post-cursor *at the DFE input* (after TX FFE and CTLE) times A.

Post-cursors only. The DFE cancels only the ISI of bits already decided: the pre-cursor of bit $n+1$ hits y_n before $\hat{a}_{n+1}$ exists. Figure 8.2 shows the feedback staircase on the REF pulse after an 8 dB CTLE. It matches the pulse at $k = 1, 2$ and leaves the pre-cursor and the tail untouched.

Error propagation. Instability is a risk for the linear prototype only. With a slicer in the loop the output is bounded, and the real risk is **error propagation**: a wrong decision adds $2w_1$ of error

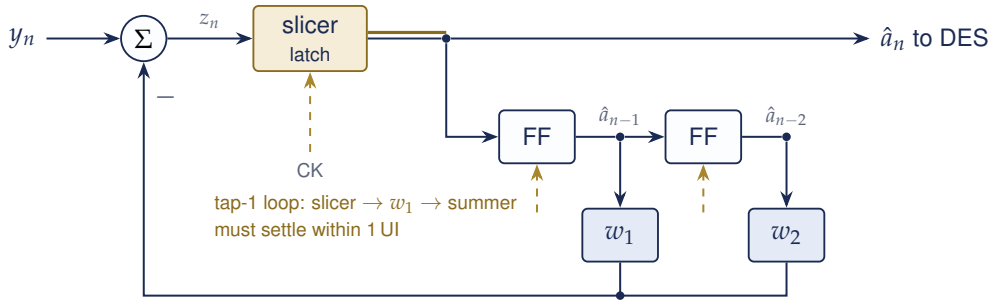


Figure 8.1. A 2-tap DFE. The slicer decision $\hat{a}_n$ is delayed by flip-flops, weighted by w_1 and w_2, and subtracted from the next samples at the summer. The fed-back values are clean ± 1 decisions, so noise is not recirculated. Tap 1 closes a loop that must settle within one UI (Section 8.2).

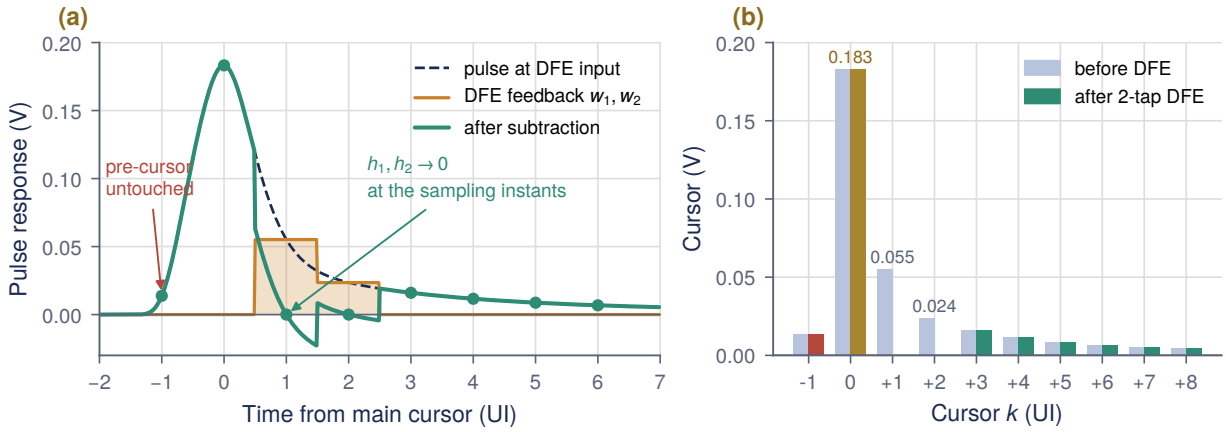


Figure 8.2. A DFE subtracts the post-cursors at the sampling instants. REF + 8 dB CTLE (the Lab 5 setup). (a) Pulse at the DFE input, the 2-tap feedback held for one UI per tap, and the result. Only the sampling instants (dots) matter: h_1 and h_2 become zero, while the pre-cursor and the tail beyond $k = 2$ remain. (b) Cursors before and after the 2-tap DFE. The main cursor is unchanged, which is the advantage over linear equalization.

to the next sample instead of removing w_1. In the Lab 5 setup ($w_1 = 27.6\,\text{mV}$, half-eye 64.7 mV after two taps) that leaves only 9.5 mV of margin on the next bit. At a BER near 10^{-12} first errors are so rare that this at most doubles the BER [22]. Keep $p_1/p_0 \lesssim 0.5$ at the DFE input.

WORKED EXAMPLE 8.1 · A DFE alone, and a DFE with a TX pre-tap

Take the pulse of Worked Example 6.2, $x_{-1} = 0.2$, $x_0 = 0.85$, $x_1 = -0.2$, $x_2 = 0.1$, $x_3 = 0.05$. Forcing the decision sequence $\hat{y} = \{\ldots, 0, 1, 0, 0, \ldots\}$ at $t = T_\text{b}, 2T_\text{b}, 3T_\text{b}$ gives $-\alpha_k + x_k = 0$, so the taps are the post-cursors: $\alpha = [-0.2,\ 0.1,\ 0.05]$. With $2A = 1$ (so $w_k = x_k/2$ in volts), the PDA eye becomes $0.85 - 0.2 = 0.65$. Only the pre-cursor remains.

Now add a TX pre-tap with $c_{-1}/c_0 = -0.2/0.85$, normalized to $c = [-0.190,\ 0.810]$. By Equation (6.2) the cursors for $k = -2..3$ become -0.038, 0, 0.726, -0.181, 0.071 and 0.041. The DFE re-adapts to the last three, and the eye is $0.726 - 0.038 = \mathbf{0.688}$. That beats both the 3-tap ZF FFE alone (0.519, Worked Example 6.2) and the DFE alone (0.65). This is the canonical split: the TX takes the pre-cursor and the DFE takes the post-cursors.

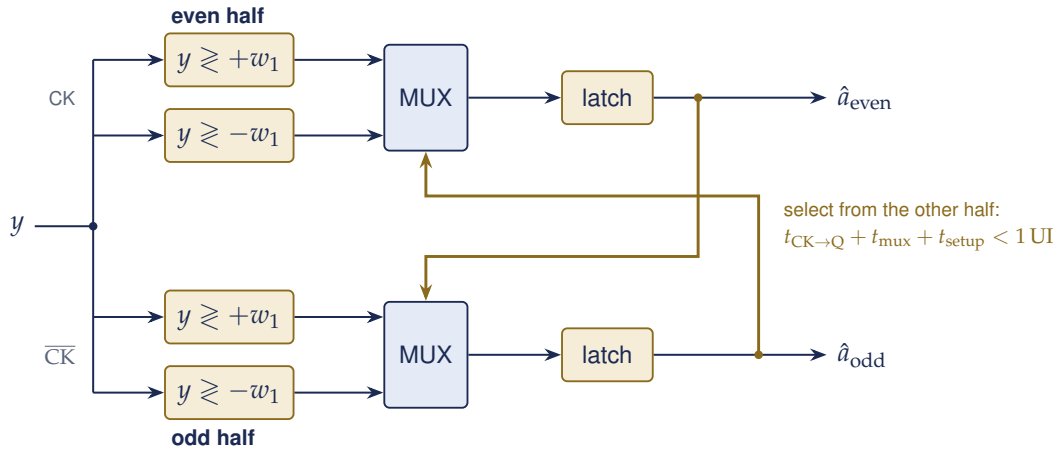


Figure 8.3. Half-rate DFE with tap 1 unrolled. Each half has two comparators with thresholds $\pm w_1$ and a multiplexer. The multiplexer is selected by the latest decision of the *other* half (gold paths). These cross-coupled selects are the critical path and still have only 1 UI, but there is no summer to settle. Tap 2 and higher would still be subtracted at a summer before the comparators.

8.2 The tap-1 timing loop

Decision $\hat{a}_{n-1}$ must be resolved, weighted and subtracted, and the summer settled, before z_n is sampled one UI later:

KEY EQUATION · DFE tap-1 critical path

$$t_{\text{CK}\to\text{Q}} + t_{\text{tap}} + t_{\text{settle}} + t_{\text{setup}} \le T_{\text{b}} = 200\,\text{ps} \tag{8.3}$$

Taps 2 and higher have one or more extra UIs, so tap 1 sets the speed limit.

Half rate. Open-SERDES slices even bits on the rising and odd bits on the falling edge of the 2.5 GHz clock, so each latch has 2 UI to regenerate. Half rate does *not* relax Equation (8.3): the even decision must correct the odd sample one UI later, so the tap-1 path crosses between the halves and still has 1 UI.

Loop unrolling. **Loop unrolling** (**speculation**) removes the summer from the tap-1 path. Two comparators per half precompute both answers, against $+w_1$ and $-w_1$, and a multiplexer selected by the previous decision picks one [22]:

$$\hat{a}_n = \begin{cases} \operatorname{sgn}(y_n - w_1), & \hat{a}_{n-1} = +1 \\ \operatorname{sgn}(y_n + w_1), & \hat{a}_{n-1} = -1 \end{cases} \quad \Rightarrow \quad t_{\text{CK}\to\text{Q}} + t_{\text{mux}} + t_{\text{setup}} \le T_{\text{b}}. \tag{8.4}$$

Figure 8.3 shows the half-rate version. The cost is 2^N comparators for N unrolled taps, twice the load on the CTLE, and taps that live in comparator thresholds.

WORKED EXAMPLE 8.2 · Tap-1 timing budget at 5 Gb/s

Use illustrative SG13G2 CML numbers, to be confirmed post-layout: latch $t_{\text{CK}\to\text{Q}} = 25 - -35\,\text{ps}$; tap switch about 10 ps; summer with $R_L = 250\,\Omega$ and $C = 80\,\text{fF}$, so $\tau = 20\,\text{ps}$ and 95 % settling takes $3\tau = 60\,\text{ps}$; setup and aperture about 20 ps; skew and PVT margin about 20 ps. The total is about 135–145 ps, which is below 200 ps. Direct feedback works at 5 Gb/s, including in the half-rate cross-coupled form, with 55–65 ps to spare (Figure 8.4).

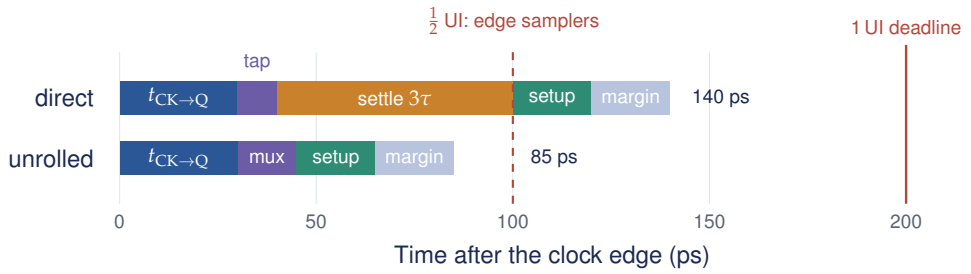


Figure 8.4. The tap-1 budget fits in one UI with direct feedback. Numbers from Worked Example 8.2. Direct feedback (140 ps) meets the 1 UI deadline but not the half-UI one. Unrolling removes the summer settling and fits both.

It does *not* meet the 100 ps half-UI budget that would apply if the 90°/270° edge samplers also had to see a fully corrected signal. In Open-SERDES the edge samplers therefore see a partly settled correction, which works because TX and CTLE keep the pre-DFE eye open (Section 8.5). If post-layout corners push the direct path past about 0.7 UI, unroll tap 1: with an assumed 15 ps mux the unrolled path is 85 ps. Halving C (fewer taps) or an integrating summer saves about 30 ps.

8.3 Adapting the taps

The ideal taps depend on channel, temperature and supply, so the receiver should find them itself. Define the slicer error $e_n = z_n - d_{\text{lev}}\,\hat{a}_n$, where d_{lev} is the expected signal level (ideally $A p_0$). Since $\partial E[e_n^2]/\partial w_k = -2E[e_n \hat{a}_{n-k}]$, stepping against the gradient gives the **LMS** update. Hardware prefers the **sign-sign** form (SS-LMS), which needs only signs [22, 23]:

KEY EQUATION · LMS and sign-sign LMS

$$\begin{aligned} \text{LMS:}\quad & w_k \leftarrow w_k + \mu\, e_n\, \hat{a}_{n-k} \\ \text{SS-LMS:}\quad & w_k \leftarrow w_k + \mu\, \operatorname{sgn}(e_n)\, \hat{a}_{n-k}, \qquad d_{\text{lev}} \leftarrow d_{\text{lev}} + \mu\, \operatorname{sgn}(e_n)\, \hat{a}_n \end{aligned} \tag{8.5}$$

$\operatorname{sgn}(e_n)$ comes from one extra comparator, the **error sampler**, with threshold $\pm d_{\text{lev}}$ chosen by $\hat{a}_n$. The updates are digital counters decimated to the deserialized rate: slow compared with the data, fast compared with temperature drift.

WORKED EXAMPLE 8.3 · Sign-sign LMS on the REF channel

Simulate the Lab 5 setup (REF + 8 dB CTLE) with PRBS13, 4 mV rms noise and a 2-tap DFE adapted by Equation (8.5) with $\mu = 0.25$ mV, starting from $w = 0$ and $d_{\text{lev}} = 50$ mV. The pre-DFE eye is open (60 mV), so decisions are correct from the first bit. w_1 comes within 2 mV of $A p_1 = 27.6$ mV after about 310 UI and w_2 reaches $A p_2 = 11.8$ mV after about 170 UI (Figure 8.5a). Both then dither by a few μ; after 6000 UI they read 29.8 and 11.5 mV, with $d_{\text{lev}} = 94.0$ mV (target 91.6). A larger μ converges faster but dithers more, and dither is a tap error that costs eye.

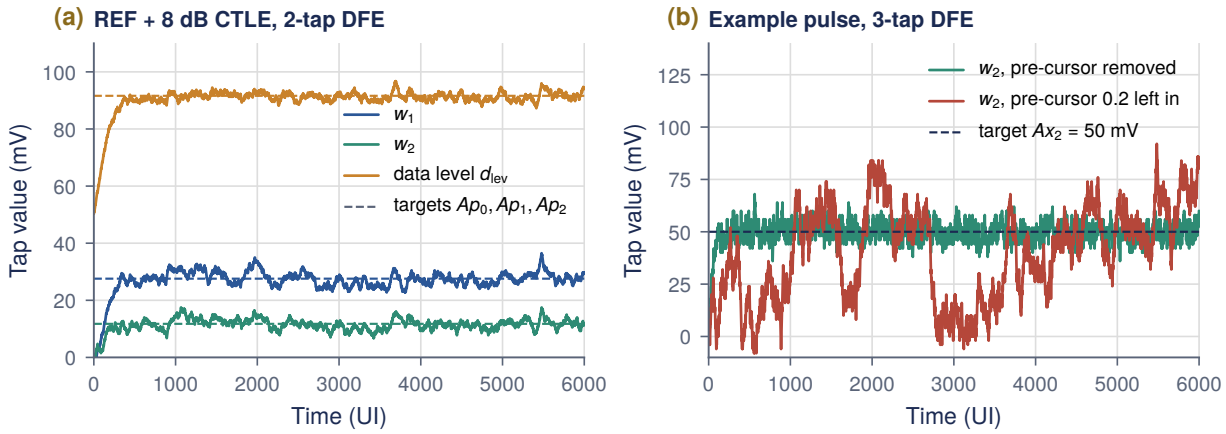


Figure 8.5. Sign-sign LMS finds the taps, unless a pre-cursor is left in. (a) REF + 8 dB CTLE, 2-tap DFE ($\mu = 0.25$ mV, 4 mV noise). The taps and the data level settle within a few hundred UI onto their zero-forcing values (dashed: 91.6, 27.6 and 11.8 mV). (b) The pulse of Worked Example 8.1 with a 3-tap DFE ($\mu = 2$ mV, 10 mV noise). With its 0.2 pre-cursor removed, w_2 stays within 36–62 mV of its 50 mV target. With the pre-cursor left in, it wanders between -6 and 92 mV.

! An uncancelled pre-cursor makes SS-LMS wander

The 0.2 pre-cursor puts $\pm Ax_{-1} = \pm 100$ mV into every error sample. Once the tap errors are smaller than that, $\mathrm{sgn}(e_n)$ is set by $Ax_{-1}\hat{a}_{n+1}$ alone, which is uncorrelated with $\hat{a}_{n-k}$: the average update vanishes and the taps random-walk (Figure 8.5b). Cancel the pre-cursor first (TX pre-tap or CTLE), or use a smaller μ and average.

8.4 The DFE on the REF channel

Lab 5 adds DFE taps one at a time after an 8 dB CTLE (Figure 8.6). The taps are the post-cursors at the DFE input, 0.0552, 0.0235 and 0.0160, so $w_k = 27.6$, 11.8 and 8.0 mV. The first tap lifts the PRBS7 eye from 60.0 to 110.7 mV, the second adds 19 and the third 13.5 mV. Beyond four taps nothing more is gained (153.6, 152.1 and 156.4 mV for 4, 5 and 8 taps). At 10 mV rms slicer noise the estimated BER is 3.2×10^{-14} with three taps and 4.5×10^{-12} with two.

>_ Lab 5: DFE

`lab5_dfe.py` reads the taps off the pulse response, runs `sdp.Receiver.nrz_DFE` with 0–8 taps and estimates the BER. Then try a -6 dB TX preset with 2 taps (Section 16.6).

8.5 Splitting the work

How should Open-SERDES divide the work between TX FFE, CTLE and DFE (Section 5.4)? Table 8.1 and Figure 8.7 compare seven strategies on the `link.py` channel family.

Method. Channels are `channel(LOSS_FAMILY[d])` for 5–30 dB at f_N, plus REF. The CTLE is `ctle(pk)`, 0–20 dB in 1 dB steps. The TX FFE is a grid search over $c_{-1} \in [-0.2, 0]$ and $c_1 \in [-0.5, 0]$ (step 0.02) with $c_0 = 1 - |c_{-1}| - |c_1|$. The DFE is ideal: it removes the first N post-cursors exactly. Each strategy is scored by its PDA eye over $k = -4..40$, maximized over sampling phase and free settings, and also normalized to $2Ap_0$ of the *unequalized* pulse, which exposes the swing cost of

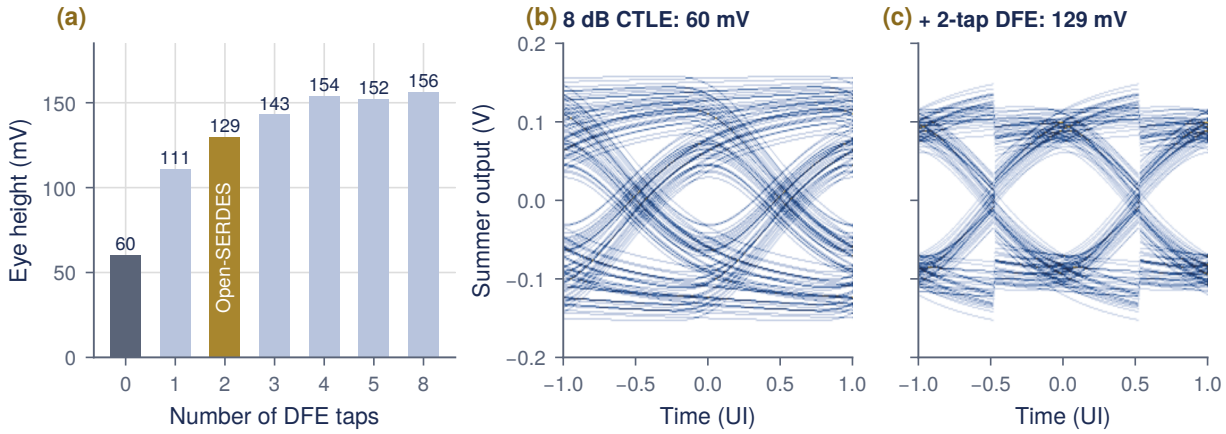


Figure 8.6. The first DFE taps do most of the work. REF + 8 dB CTLE, PRBS7 (Lab 5). (a) Eye height against the number of ideal DFE taps. The first two taps add 70 mV, and beyond four taps the gain is lost in the simulation's resolution. (b, c) Summer output before and after a 2-tap DFE. The feedback switches half a UI after each decision, so the eye shows steps at ± 0.5 UI. Only the sampling instant at 0 UI matters.

Table 8.1. PDA eye height (mV) and eye normalized to the unequalized main cursor, by strategy and channel loss. Regenerated with `link.py`; method in the text. *Open-SERDES limits: $|c_1| \leq 0.25$, $|c_{-1}| \leq 0.1$, CTLE ≤ 12 dB, open pre-DFE eye.

	10 dB		15 dB		REF 17.4 dB		20 dB		25 dB	
Strategy	mV	norm	mV	norm	mV	norm	mV	norm	mV	norm
No equalization	165	0.30	closed		closed		closed		closed	
TX 2-tap FFE	341	0.61	189	0.48	141	0.42	100	0.34	44	0.20
CTLE (0–20 dB)	236	0.42	132	0.33	99	0.29	70	0.24	32	0.15
CTLE + 2-tap DFE	395	0.71	146	0.37	110	0.32	84	0.29	51	0.23
TX 3-tap + CTLE + 2-tap DFE	396	0.71	204	0.52	160	0.47	123	0.42	75	0.34
Open-SERDES limits, 2-tap DFE*	396	0.71	190	0.48	134	0.40	87	0.30	35	0.16
Open-SERDES limits, 5-tap DFE*	467	0.84	239	0.60	170	0.50	120	0.41	43	0.19

TX equalization. The "Open-SERDES limits" rows add $|c_1| \leq 0.25$ (6 dB post-tap de-emphasis), $|c_{-1}| \leq 0.1$, CTLE ≤ 12 dB, and an eye open *before* the DFE (for start-up adaptation and the edge samplers). Because the tail is summed to 40 UI, the unequalized REF eye is -0.23 V rather than the 13-cursor -0.161 V of Table 3.2.

What the numbers say.

- **No single equalizer is enough at REF.** The TX FFE alone reaches 141 mV but needs 12 dB of de-emphasis; the CTLE alone needs 16 dB of peaking, outside our range, for 99 mV.
- **With a DFE present, the noise-free optimum moves the linear work into the TX** (CTLE at 0 dB, 11–16 dB of de-emphasis). In a PDA our normalized CTLE and the TX FFE pay the same price, a lower low-frequency level, and the 1-UI FIR fits a sampled channel better. The PDA cannot see noise, crosstalk, sampler sensitivity or adaptivity, which are what limit large de-emphasis in practice.
- **Within practical limits** the REF eye is 134 mV (0.40 of the unequalized main cursor). Three more DFE taps add 36 mV.
- **Above about 22 dB the 12 dB CTLE limit binds.** At 25 dB, CTLE + 2-tap DFE with 19 dB of

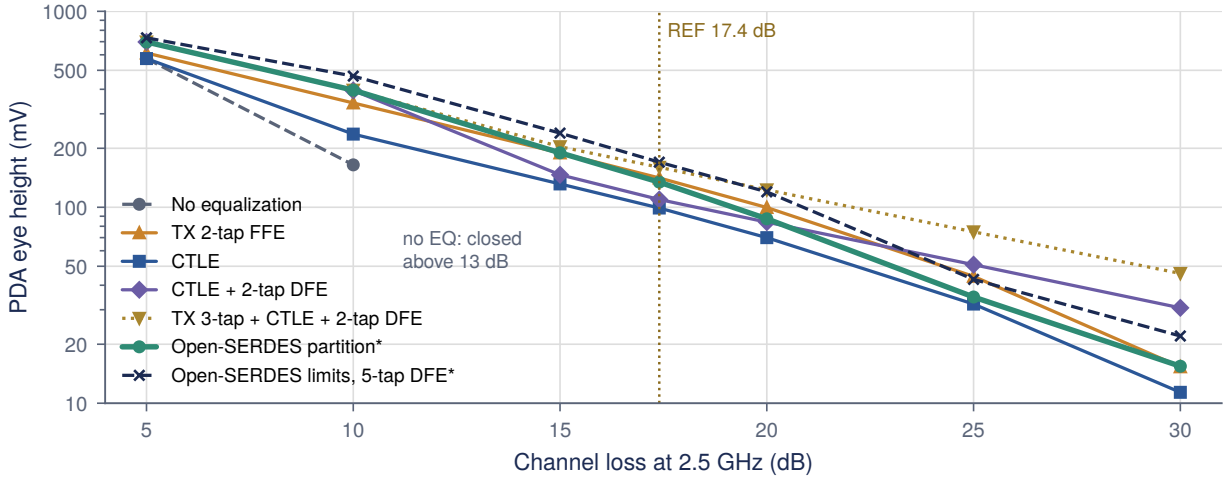


Figure 8.7. Every strategy loses about a factor of two in eye per 5 dB of loss; combinations lose less. PDA eye against loss at f_N (log scale, data of Table 8.1). At REF (dotted) the Open-SERDES partition gives 134 mV. Above about 22 dB, a CTLE with more than 12 dB of peaking (purple) overtakes it, which signals when Open-SERDES would need more equalization.

peaking (51 mV) beats both Open-SERDES rows; more DFE taps help less than more CTLE.

WORKED EXAMPLE 8.4 · The REF operating point in the time domain

Check the partition with the Lab 5 machinery: PRBS7 through REF, a TX preset, `ctle(pk)` and an ideal 2-tap DFE. The BER is estimated at 10 mV rms slicer noise.

TX preset, CTLE	pre-DFE eye	+ 2-tap DFE	p_1/p_0	w_1, w_2 (mV)	BER
none, 8 dB (Lab 5)	60.0	129.5	0.30	27.6, 11.8	4.5×10^{-12}
none, 0 dB	−84.0	167.1	0.54	91.2, 45.6	1.3×10^{-15}
−3.5 dB, 0 dB	35.4	169.5	0.37	51.1, 23.5	3.5×10^{-17}
−6 dB, 0 dB	96.9	**171.8**	0.26	31.1, 11.7	3.0×10^{-18}
−6 dB, 4 dB	103.0	132.6	0.15	12.7, 5.9	1.4×10^{-12}

A DFE alone (row 2) opens the eye, but with a closed pre-DFE eye it cannot adapt from start-up, and $p_1/p_0 = 0.54$ invites error propagation. The −6 dB preset with the CTLE near 0 dB gives the best eye, an open pre-DFE eye (97 mV) for the edge samplers, and $p_1/p_0 = 0.26$: a wrong decision costs $2w_1 = 62$ mV of the 86 mV half-eye.

The Open-SERDES partition, and why.

- **TX: 3-tap FFE, mostly de-emphasis** (−3.5/−6 dB presets, pre-tap up to $|c_{-1}| = 0.1$; REF uses −6 dB). It is the only block that removes the pre-cursor without noise, and it lowers the long-run level that would overload the CTLE (Section 7.4). It stays moderate because it cannot adapt without a back-channel and because lost launch swing is lost SNR.
- **RX: CTLE, 0–12 dB.** On REF it sits near 0–4 dB and acts mainly as the front-end gain stage. Its range covers longer channels and board-to-board and temperature variation, keeps the pre-DFE eye open for the CDR, and, unlike the TX presets, adapts at the receiver.
- **RX: 2-tap half-rate DFE, direct feedback.** It removes p_1 and p_2 without noise, with 55–65 ps of tap-1 slack (Worked Example 8.2). Each extra tap adds summer capacitance and slows

tap 1 for about 10 mV more eye.

When we would need more. Beyond about 22 dB of loss, add a second CTLE stage (> 12 dB), 3–5 DFE taps or an IIR tail tap, and a larger pre-tap. Unroll tap 1 if post-layout timing exceeds about 0.7 UI.

▶ What this means for your design

The partition follows the pulse anatomy of Figure 5.4: the TX pre-tap takes the pre-cursor, de-emphasis and the CTLE take the long tail and keep the pre-DFE eye open, and the DFE takes p_1 and p_2 without noise. A noise-free PDA would push all linear work into the TX; noise, crosstalk and adaptation pull some of it back into the receiver. Always report the result in millivolts against the noise budget of Section 4.4.

Exercises

8.1 The DFE of Worked Example 8.1 with one tap only: what is the PDA eye? If $\hat{a}_{n-1}$ is wrong, how much margin is left on bit n?

8.2 Redo Worked Example 8.2 with 120 fF of summer capacitance (more taps). Does direct feedback still fit? What R_L restores it, and at what cost in summer gain?

8.3 Show that the SS-LMS update $E[\mathrm{sgn}(e_n)\hat{a}_{n-1}]$ is zero at $w_1 = Ap_1$ if the remaining ISI plus noise is symmetric. Why does this not stop the wandering in Figure 8.5(b)?

8.4 From the cursors in Section 8.4 ($p_{-1} = 0.0138$, $p_0 = 0.1832$, tail to p_8), compute the PDA eye after 0, 1 and 2 DFE taps. Why is it lower than the PRBS7 numbers of Figure 8.6? Then use Table 8.1: at 25 dB, which buys more, three extra taps or a 19 dB CTLE?

KEY TAKEAWAYS

- A DFE is an IIR equalizer with a slicer in the loop: it cancels post-cursors without amplifying noise, with taps equal to the post-cursors at its input. It cannot touch pre-cursors, so a TX pre-tap plus a DFE beats either alone (0.688 vs 0.519 and 0.65 in Worked Example 8.1).
- Tap 1 must settle within 1 UI even at half rate. At 5 Gb/s direct feedback fits in about 140 ps. Unrolling is the fallback if it does not.
- SS-LMS adapts taps with one error sampler and counters. It wanders if a large pre-cursor is left uncancelled.
- On REF + 8 dB CTLE, DFE taps give $60 \to 111 \to 130 \to 143$ mV for 0–3 taps. Beyond four taps the gain is negligible.
- The Open-SERDES partition is a TX 3-tap FFE (-6 dB on REF), a CTLE with 0–12 dB and a 2-tap half-rate DFE. On REF it gives 172 mV (PRBS7) and 134 mV (PDA).
- Beyond about 22 dB of loss, more CTLE peaking and more DFE taps become necessary.

PART III

Transceiver Circuits

Turning the system model into transistors: the serializer and output driver on the transmit side, and the termination, front-end and samplers on the receive side, all with SG13G2 HBTs and CMOS.

CHAPTER 9

The Transmitter

Sixteen slow bits go in, one fast differential wave comes out: the serializer, the termination, the CML driver with its FFE, and the pad that connects them to the channel.

IN THIS CHAPTER YOU WILL LEARN

- Explain with the FO4 rule why the last serializer stages and their clocks must be half-rate CML in 130 nm.
- Design an HBT CML 2:1 multiplexer and read its timing, including duty-cycle errors.
- Compare series, parallel and double termination, and compute what double termination costs.
- Size the Open-SERDES CML driver (current, swing, power, headroom, BV_{CEO}) and compare it with a voltage-mode driver.
- Build the 3-tap FFE as a current-summing driver and choose its tap resolution.
- Budget the pad, ESD and driver capacitance, and decide when a T-coil is worth its area.

9.1 The transmitter at a glance

The transmitter takes a 16-bit word from the digital core every 3.2 ns (312.5 MHz) and launches it as a 5 Gb/s NRZ wave into the 50 Ω-per-side channel (Figure 9.1). Each 2:1 stage doubles the lane rate and is clocked at *half* its output bit rate. The clocks come from the 2.5 GHz PLL (Chapter 12) through a chain of ÷2 dividers. Static CMOS on 1.2 V handles the slow, wide part. HBT current-mode logic (CML) on the 2.5 V analog supply handles everything at 1.25 GHz and above. The final 2:1 multiplexer is built three times, so that the driver receives D_{n+1}, D_n and D_{n-1} for its three FFE taps.

9.2 The serializer

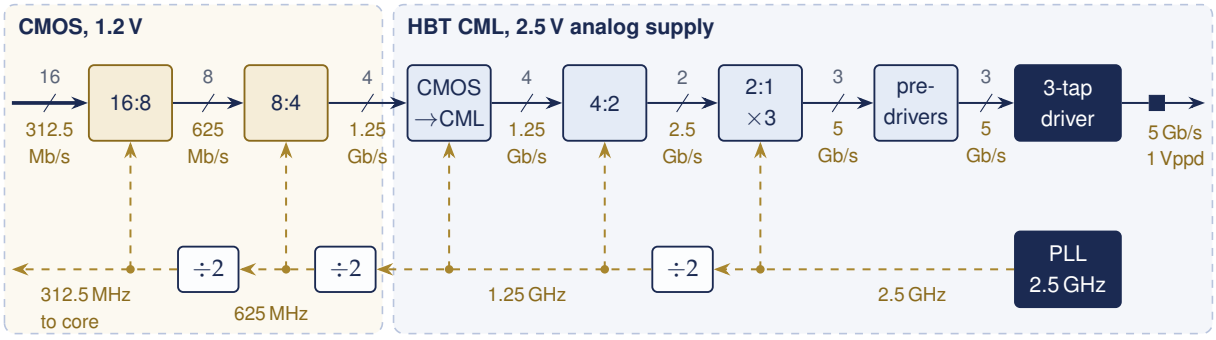


Figure 9.1. The Open-SERDES transmitter is a tree of 2:1 stages, each clocked at half its output rate. Slash numbers give the number of parallel lanes; gold text gives the rate of each lane or clock. CMOS does 16:4. HBT CML does the 4:2 and the final half-rate 2:1, which is built three times to feed the three FFE taps.

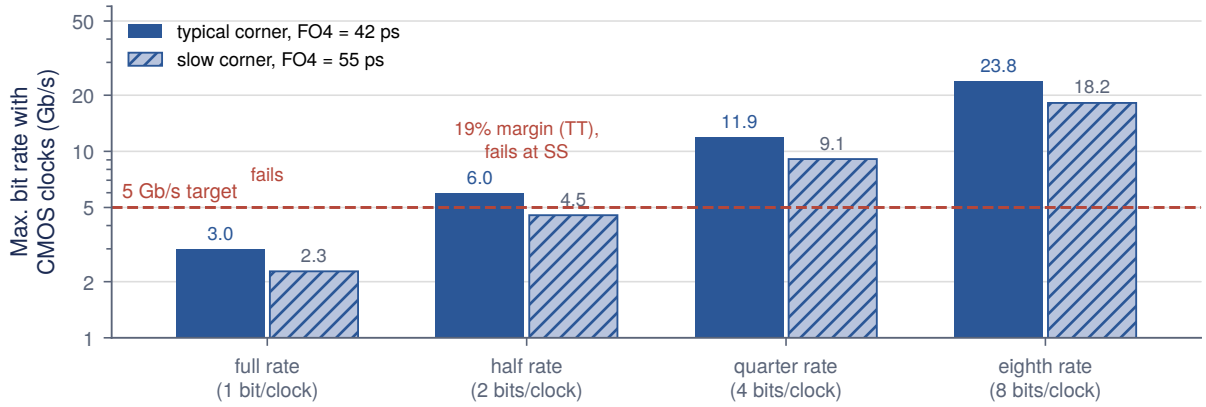


Figure 9.2. The 8-FO4 rule rules out full-rate CMOS and leaves half-rate CMOS without margin. Maximum bit rate from Equation (9.1) at the typical (42 ps) and slow (55 ps) FO4 delays. Only bars above the dashed line can serialize 5 Gb/s. Quarter and eighth rate pass on paper but need multi-phase clocks and high fan-in output muxes.

9.2.1 Clocks, not gates, set the speed

A 130 nm CMOS gate switches in tens of picoseconds, yet a 5 GHz CMOS serializer does not work, because the clock has to be *distributed*. A clock passing through a chain of inverters keeps its shape only if each half-period lets every inverter swing fully. Below about 8 fan-out-of-4 delays per period, the pulses shrink at each stage and vanish [24]. The **FO4 delay** t_{FO4}, the delay of an inverter driving four copies of itself, is the natural unit of CMOS speed.

KEY EQUATION · The clock-distribution limit

$$T_{clk} \geq 8\,t_{FO4} \quad \Longrightarrow \quad f_b \leq \frac{n}{8\,t_{FO4}}, \tag{9.1}$$

where n is the number of bits per clock period: 1 at full rate, 2 at half rate, 4 at quarter rate.

WORKED EXAMPLE 9.1 · Can CMOS clock the last stage?

Given: $t_{FO4} \approx 42$ ps for 1.2 V SG13G2 CMOS at the typical corner and about 55 ps at the slow corner. These are 130 nm estimates: measure yours in **ngspice** with a chain of inverters, each loaded by four copies of itself.

Full rate needs $T_{clk} = 200\,\text{ps} = 4.8\,t_{FO4}$, and Equation (9.1) allows only 3.0 Gb/s. **Half rate** needs $T_{clk} = 400$ ps: $9.5\,t_{FO4}$ typical but $7.3\,t_{FO4}$ slow, a limit of 5.95 and 4.55 Gb/s. CMOS fails at the slow corner (Figure 9.2).

Conclusion: the final 2:1 stage and its clock must be CML, whose speed is set by $R_L C_L$ rather than FO4. Stages at 625 MHz and below ($T_{clk} \geq 29\,t_{FO4}$) stay in CMOS.

9.2.2 The half-rate CML 2:1 multiplexer

A **half-rate** multiplexer has no full-rate flip-flop at its output. The clock *itself* is the select: the even bit goes out while CK is high and the odd bit while CK is low [24]. Figure 9.3(a) shows the HBT version, a stacked (series-gated) CML gate. The clock pair Q_5/Q_6 steers the tail current I_T into one of the two data pairs, and the selected pair steers it into one of the shared loads R_L. The deselected pair carries no current. The data bases sit at CML levels and the clock bases one V_{BE} lower, through emitter followers, so every device keeps its V_{CE} near 0.8 V on the 2.5 V supply.

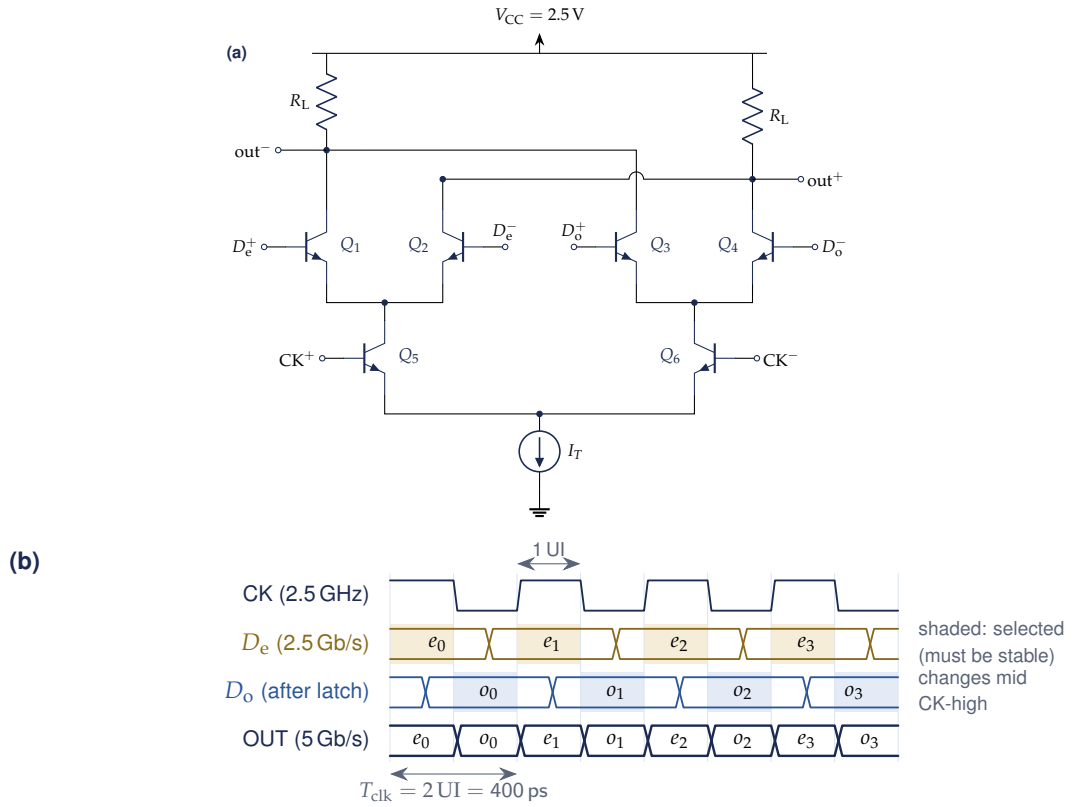


Figure 9.3. A half-rate CML multiplexer: the clock picks which data pair gets the current. (a) When CK^+ is high, Q_5 conducts and the even pair Q_1/Q_2 drives the loads; when it is low, the odd pair Q_3/Q_4 does. The hop marks a crossing without connection. (b) Each input changes only while it is deselected. The even data is selected in the gold windows and changes half a UI after CK falls; an extra latch delays the odd data by half a period so it changes while CK is high. The output UIs follow the clock halves exactly.

KEY EQUATION · CML gate design

$$\Delta V = I_T R_L, \qquad \tau = R_L C_L, \qquad P = V_{CC} I_T = \frac{V_{CC}\,\Delta V\,C_L}{\tau}. \tag{9.2}$$

Choose the swing ΔV first, then R_L from the speed target τ, then I_T.

Glitch-free timing and retiming. The select must never switch while a selected input changes. In Figure 9.3(b) each input changes in the middle of the half where it is deselected: the even data directly, the odd data after one extra latch. Data from the 4:2 stage, which runs on the divided 1.25 GHz clock, is retimed by latches on the 2.5 GHz clock, and transitions placed mid-way through the deselected half leave ±100 ps for divider delay, wiring, setup and hold (800 ps at the CMOS-to-CML boundary). A skew beyond the window does not degrade the eye gracefully: it shifts the bits by one position.

Duty cycle becomes UI width. A clock with duty cycle D produces even UIs of $2DT_b$ and odd UIs of $2(1-D)T_b$. A 45 % duty cycle gives alternating 180 and 220 ps bits, 20 ps of deterministic jitter before the channel acts (Chapter 11). Restore the clock to 50 % with a CML buffer chain or a duty-cycle corrector right before the mux.

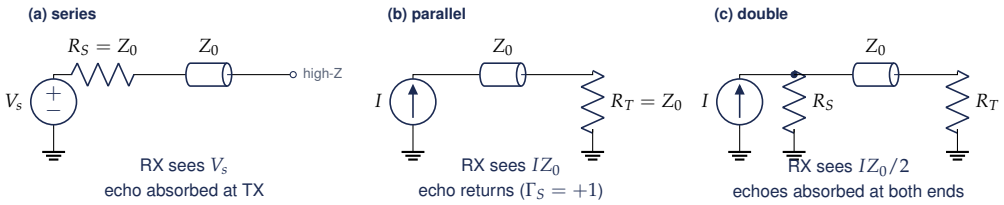


Figure 9.4. Three ways to terminate a line. (a) Series (source) termination launches $V_s/2$, which the open end doubles; it needs a high-impedance RX, which pad capacitance prevents at 5 Gb/s. (b) Parallel (load) termination: an echo from an imperfect RX returns to the high-impedance source, reflects again and arrives a round trip later as ISI. (c) Double termination ($R_S = R_T = Z_0$) costs half the swing but absorbs echoes at both ends. It is the Open-SERDES CML driver; a voltage-mode driver with R_T is also doubly terminated.

▶ What this means for your design

Build the last two serializer stages and the 2.5 GHz clock buffers in HBT CML, and everything from 625 MHz down in 1.2 V CMOS. Half rate removes the 5 GHz clock but turns duty cycle into UI width: budget for a duty-cycle corrector, and simulate every clock-domain boundary at all corners.

9.3 Termination

A wave that meets an impedance step reflects (Section 3.4). A link can absorb it at the source, at the load or at both ends (Figure 9.4).

Why double termination wins. Nothing is perfectly matched: pad capacitance, resistor tolerance and the package always reflect a few percent of the wave at the RX. With $R_S = Z_0$ that echo dies at the TX ($\Gamma_S = 0$) instead of returning as a delayed copy of the bit, which would be ISI at a fixed, unknown cursor where no FFE or DFE tap is waiting. The price is swing: half the source current heats the transmitter's own termination.

On-chip resistors in SG13G2. Place both terminations on the die at the pad. An off-chip resistor leaves bond wire, package and pad as an unterminated stub; on chip, the package becomes part of the line and only the lumped pad capacitance remains (Section 9.6). Use an unsilicided poly resistor: linear and well matched, but spreading by roughly ±20–30 % over process and temperature (check the PDK [1]). ±20 % still gives 19 dB of return loss (Section 3.4). A 3-bit trim in 6–8 % steps is still worth adding, because a CML driver's swing is proportional to its load resistance. Size each TX resistor for the full 20 mA.

9.4 The CML output driver

The Open-SERDES driver (Figure 9.5) is Figure 9.4(c) in differential form. An HBT pair Q_1/Q_2 steers the tail current I_T into one of two loads $R_L = Z_0 = 50\,\Omega$, which are also the TX back-terminations. The RX terminates the line in 100 Ω differential.

Swing. When Q_2 takes all of I_T, the current divides between its own load and the line, so each output sees $R_L \parallel Z_0 = 25\,\Omega$ and swings $I_T \cdot 25\,\Omega$ single-ended. The differential voltage flips sign with the data, so $V_{d,pp} = 2 \cdot I_T \cdot 25\,\Omega = I_T Z_0$.

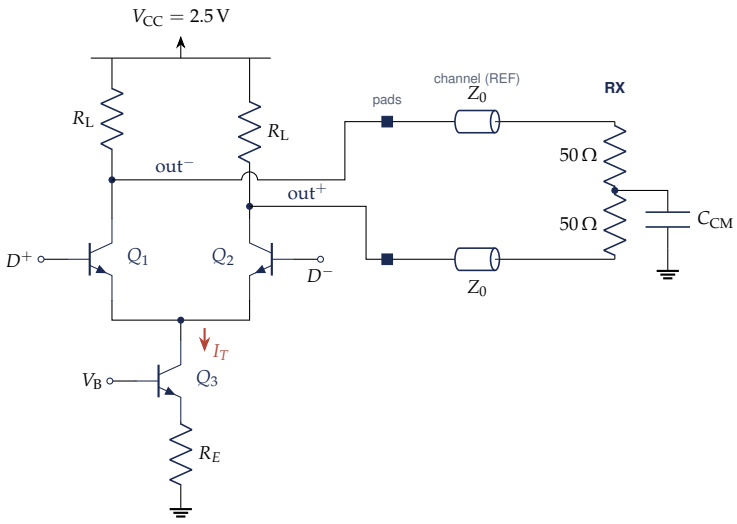


Figure 9.5. The Open-SERDES CML driver. Q_1/Q_2 steer the tail current I_T, set by Q_3, R_E and V_B, into one of the 50 Ω loads R_L. The loads double as TX back-terminations, so each output sees $R_L \parallel Z_0 = 25\,\Omega$. The RX terminates in 2×50 Ω with an AC-grounded center tap. 20 mA gives 1 Vppd.

Voltage mode. The alternative is a **voltage-mode** (VM) driver, usually a source-series-terminated (SST) one: each output is an inverter-like switch pair in series with a resistor, $R_{on} + R_S = Z_0$ [25]. Its current flows only around the loop of TX resistor, line, 100 Ω RX termination, line and TX resistor: $4Z_0 = 200\,\Omega$. A supply V_s drives $I = V_s/4Z_0$, the RX sees $\pm V_s/2$, and $V_{d,pp} = V_s$.

KEY EQUATION · Driver current for a differential swing $V_{d,pp}$

$$I_{CML} = \frac{V_{d,pp}}{Z_0}, \qquad I_{VM,\,SE\ term.} = \frac{V_{d,pp}}{2Z_0}, \qquad I_{VM,\,diff.\ term.} = \frac{V_{d,pp}}{4Z_0}. \tag{9.3}$$

CML needs four times the current of VM into a differential termination: half of it flows in its own termination, and it never reverses through the line.

WORKED EXAMPLE 9.2 · CML versus SST at Open-SERDES swings

CML: 800 mVppd needs $I_T = 0.8/50 = 16\,mA$, 40 mW from 2.5 V. At 1 Vppd (the model value, $A = 0.5\,V$) it needs 20 mA and **50 mW**. The output common mode sits $I_T \times 25\,\Omega = 0.5\,V$ below V_{CC}, and each output moves between 1.75 and 2.25 V.

SST: 800 mVppd needs $I = 0.8/200 = 4\,mA$ from $V_s \approx 0.8\,V$: 3.2 mW, or 4.8 mW through a regulator from 1.2 V (6 mW at 1 Vppd). Keep $R_{on} \leq 10\,\Omega$, so a ±30 % spread moves the output impedance by only ±3 Ω.

Verdict: SST saves about 44 mW (Figure 9.6), but Open-SERDES uses CML: a passive-resistor impedance with no calibration loop, constant supply current, FFE taps that leave the impedance untouched, and fast HBTs instead of CMOS pre-drivers passing 200 ps (4.8 FO4) pulses. SST is the power upgrade for a second tape-out.

Why HBTs. A CML stage must switch its tail current fully. A bipolar pair splits it as $\Delta I = I_T \tanh(V_{id}/2V_T)$, so $V_{id} \approx 4V_T \approx 100\,mV$ steers 96 % of it at any current. A MOS pair needs $\sqrt{2}\,V_{OD}$, typically 200–300 mV. The HBT's $g_m/I_C = 1/V_T = 38.7/V$ is also several times a MOSFET's.

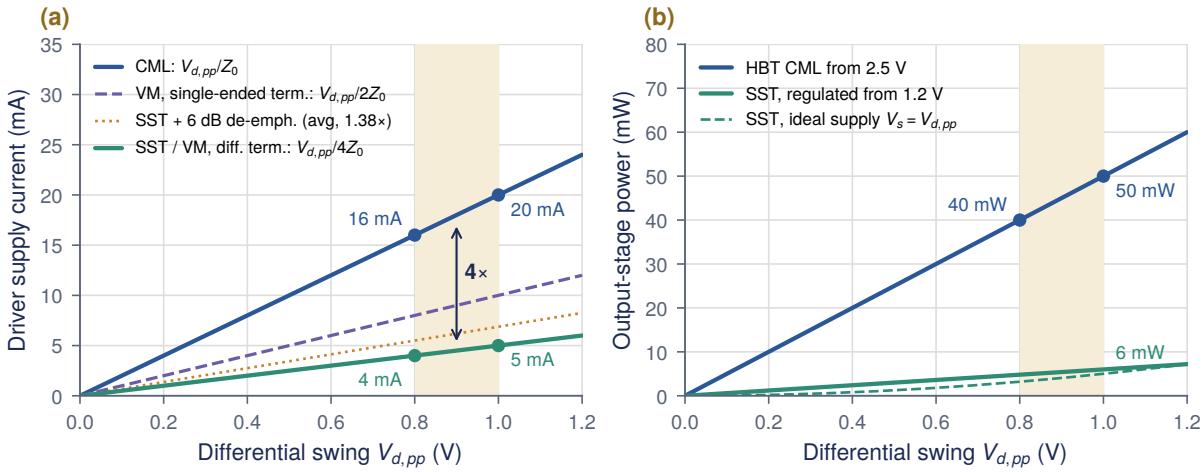


Figure 9.6. A CML driver pays four times the current of a voltage-mode driver for the same swing. (a) Supply current from Equation (9.3); the band is the Open-SERDES range of 0.8–1.0 Vppd. With 6 dB of de-emphasis an SST driver draws 1.38 times more on average, because its main and post slices fight; CML draws the same current at every FFE setting. (b) Output-stage power: 40–50 mW for HBT CML from 2.5 V versus about 6 mW for SST.

WORKED EXAMPLE 9.3 · HBT versus MOS in a 2.5 GHz CML stage

Given: a mux or pre-driver stage loaded by $C_L = 40\,\text{fF}$, with $f_{3dB} \geq 7.5\,\text{GHz}$, so $\tau \leq 21\,\text{ps}$. By Equation (9.2), $R_L = 500\,\Omega$ gives $\tau = 20\,\text{ps}$; $\Delta V = 300\,\text{mV}$ then needs $I_T = 0.6\,\text{mA}$, or 1.5 mW from 2.5 V.

HBT pair: at the crossing each device carries 0.3 mA, $g_m = 11.6\,\text{mS}$, and the gain is $g_m R_L = \Delta V / 2V_T = \mathbf{5.8}$. Full switching needs about 104 mV.

MOS pair at the same current with $V_{OD} = 150\,\text{mV}$: $g_m = 4\,\text{mS}$, gain $\Delta V / V_{OD} = \mathbf{2}$, full switching at 212 mV. More gain needs wider devices, more C_L and more current. A chain of HBT stages can therefore run at smaller swing, with fewer and smaller stages.

! Headroom and the BV_{CEO} limit

SG13G2 HBTs break down at about $BV_{CEO} \approx 1.6\,\text{V}$ (check the PDK). On 2.5 V the *off* driver transistor is at risk: its collector sits at the high output level. Stack up Worked Example 9.2 at 1 Vppd. Outputs: 2.25 V (off) and 1.75 V (on). Bases high: 1.9 V, from a pre-driver whose common-mode resistor pulls its outputs 0.6 V below V_{CC}. Common emitter: $1.9 - V_{BE} \approx 1.0\,\text{V}$. So $V_{CE} = 0.75\,\text{V}$ on and 1.25 V off, with 1.0 V left for the tail. The 0.35 V margin is eaten by $V_{CC} + 5\,\%$, overshoot and reflections; an emitter-follower pre-driver with bases near 1.6 V would leave the off device at 1.5 V nominal. Check every V_{CE} at every corner *with the real RX termination attached*: a far end terminated to ground moves the output common mode.

9.5 The FFE inside the driver

Chapter 6 chose the taps; here we build them. Put one differential pair per tap on the same pair of loads, drive each with its own copy of the data, and set each tail current to the tap weight (Figure 9.7). The loads add the currents, which forms the FIR sum of Section 6.2. Negative side taps cost nothing: swapping a pair's base inputs inverts its contribution.

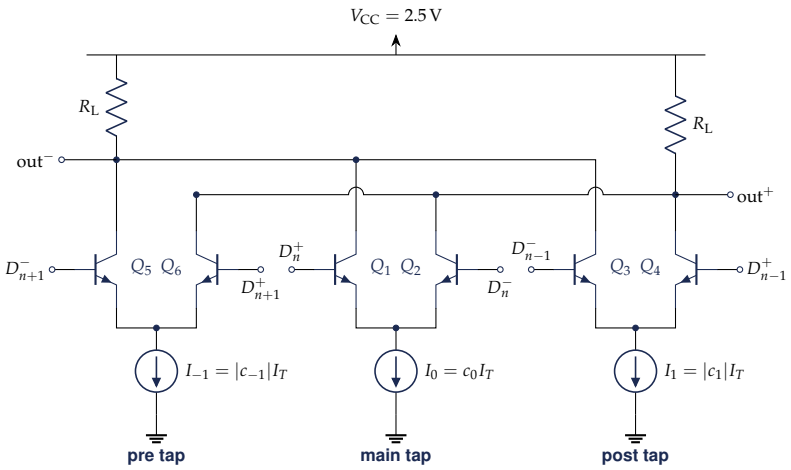


Figure 9.7. A 3-tap FFE is three CML pairs on one pair of loads. Each pair is driven by its own copy of the data and steers a tail current proportional to its tap weight. The pre and post pairs have their inputs swapped, which gives $c_{-1}, c_1 < 0$. The tails are current DACs sharing one reference, with $I_{-1} + I_0 + I_1 = I_T = 20\,\text{mA}$, so the supply current never changes. Hops mark crossings without connection.

KEY EQUATION · Current-summing FFE

$$v_d[n] = \frac{Z_0}{2} I_T \sum_{k=-1}^{1} c_k\, a_{n-k}, \qquad c_k = \pm\frac{I_k}{I_T}, \qquad \sum_k |c_k| = \frac{I_{-1} + I_0 + I_1}{I_T} = 1. \tag{9.4}$$

The peak-swing constraint of Section 6.2 holds by construction: the taps are slices of one fixed tail current.

Making the delayed copies. No 5 GHz clock is needed. Call the half-rate streams E_m (bit $2m$) and O_m (bit $2m+1$). The main mux selects (E_m, O_m). The post-tap mux must emit bit $2m-1$ in the even slot and bit $2m$ in the odd slot, so it selects (O_{m-1}, E_m); the pre-tap mux selects (O_m, E_{m+1}). About four extra latches on the 2.5 GHz clock keep these inputs glitch-free (Figure 9.3b).

WORKED EXAMPLE 9.4 · Tap currents for the -6 dB preset

The de-emphasis of a 2-tap setting is $\text{DE} = 20\log_{10}\left[(|c_0| + |c_1|)/(|c_0| - |c_1|)\right]$, the ratio of transition to repeated-bit swing. For 6 dB that ratio is 2; with $|c_0| + |c_1| = 1$, $|c_0| - |c_1| = 0.5$, so $c_0 = 0.75$ and $c_1 = -0.25$, the PCIe preset of Section 6.2. (The often-quoted $c_1 = -1/3$ is the 9.5 dB setting.)

At 1 Vppd ($I_T = 20\,\text{mA}$): $I_0 = 15\,\text{mA}$, $I_1 = 5\,\text{mA}$. Transition bits swing $(15+5)\,\text{mA} \times Z_0 = 1.0\,V_{ppd}$, repeated bits $(15-5) \times 50 = 0.5\,V_{ppd}$. The supply stays at 50 mW for every preset.

Resolution: build I_T from 32 cells of 0.625 mA (5 bits). The -6 dB preset is exact (24 and 8 cells). The -3.5 dB preset needs 5.3 cells; five give $c_1 = -0.156$ and 3.2 dB, which is harmless because the eye is flat near its optimum (Chapter 6). A pre-tap moves in steps of $1/32 = 0.031$.

The resolution trade-off. Range and resolution cost capacitance, not current. Sizing each pair for its largest current (20 mA main, 8 mA for $|c_1| \le 0.4$, 3 mA for $|c_{-1}| \le 0.15$) loads the output with 31 mA worth of collectors, 1.55 times a plain driver. A **segmented** driver instead uses 32 identical cells, each with a selector in its pre-driver that assigns it to the pre, main or post stream: the output always sees exactly 20 mA of devices, and the cost moves into the pre-drivers.

▶ What this means for your design

Build the FFE from current. The tail current is fixed, so power and output impedance do not depend on the preset, and $\sum|c_k| = 1$ holds automatically. Five bits are enough. Spend layout effort on matching the three tap paths: a 10 ps skew between D_n and D_{n-1} moves the de-emphasis edge by 0.05 UI.

9.6 Pad, ESD and bandwidth

At the pad the driver's collector capacitance, the termination's parasitics, the ESD diodes and the bond pad hang on one node (Figure 9.9a). That node sees $R_L \parallel Z_0 = 25\,\Omega$, so the total capacitance C_p makes a pole at

$$f_{3dB} = \frac{1}{2\pi(Z_0/2)C_p} = \frac{1}{\pi Z_0 C_p} \tag{9.5}$$

and spoils the return loss (Section 3.4). The **ESD** protection cannot be left out: dual diodes from the pad to both rails, plus a supply clamp, steer an electrostatic discharge away from the HBT and thin-oxide junctions. Diodes rated for a few kilovolts of human-body-model discharge are large and often dominate C_p.

WORKED EXAMPLE 9.5 · The Open-SERDES pad capacitance budget

Illustrative values (extract the real ones from layout): bond pad 80 fF, ESD diodes 250 fF, driver collectors and termination parasitics 300 fF: $C_p = 0.63$ pF.

Return loss at f_N: $x = \omega Z_0 C_p = 0.495$ and $|S_{11}| = x/\sqrt{4+x^2} = 0.24$, or -12.4 **dB**. It passes a 10 dB target ($C_p \le 0.85$ pF) but fails 15 dB ($C_p \le 0.46$ pF). **Bandwidth:** $f_{3dB} = 1/(\pi \cdot 50 \cdot 0.63\,\text{pF}) = 10.1$ GHz, four times f_N.

System check: the REF model assumes 300 fF per pad. With `c_pad=0.63e-12` in `channel()` the loss at f_N grows from 17.4 to 17.8 dB and the FFE-only PDA eye drops from 62 to 54 mV. Always rerun the system model with the extracted pad capacitance.

The T-coil. When the budget does not close, a **bridged T-coil** separates the capacitance from the line [26, 27]. Two coupled inductors connect pad and termination, the capacitive load hangs on their center tap, and a small C_B bridges the ends (Figure 9.9b). With the right values the inductive and capacitive impedances cancel exactly [28].

KEY EQUATION · Bridged T-coil, $\zeta = 1/\sqrt{2}$

$$L_1 = L_2 = \frac{R^2C}{2(1+k)}, \quad k = \frac{4\zeta^2-1}{4\zeta^2+1} = \frac{1}{3}, \quad C_B = \frac{C}{16\zeta^2} = \frac{C}{8}, \quad Z_{in} = R \text{ at all } f. \tag{9.6}$$

The total inductance is $L_1 + L_2 + 2M = R^2C$. From a current source, the capacitor node then has 2.83 times the bandwidth of the plain RC. In a doubly terminated pad the plain pole already sees $R/2$, so the bandwidth gain is $\sqrt{2}$; the bigger gain is in return loss.

For $C = 0.63$ pF and $R = 50\,\Omega$, Equation (9.6) gives $L_1 = L_2 = 0.59$ nH, $M = 0.20$ nH and $C_B = 79$ fF (often the coil's own inter-winding capacitance): better than 25 dB of return loss at f_N with 3 Ω per coil (Figure 9.8). In SG13G2 it fits in roughly 60 µm × 60 µm on the thick top metals; take k, C_B and self-resonance from EM simulation (Section 15.4).

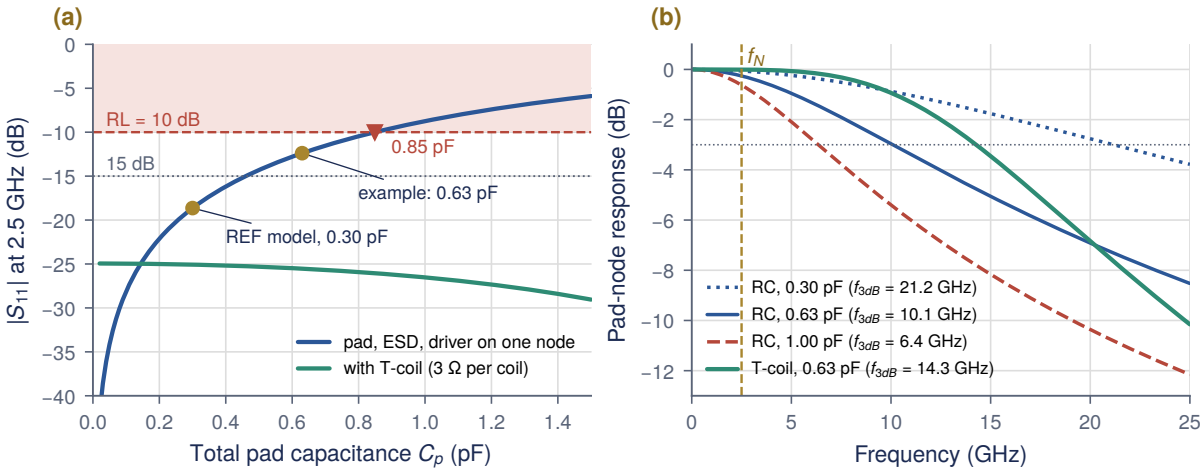


Figure 9.8. Pad capacitance costs return loss long before it costs bandwidth. (a) Return loss at 2.5 GHz versus total pad capacitance. The 0.63 pF example passes the 10 dB line (limit 0.85 pF). A T-coil with 3 Ω of loss per coil holds $|S_{11}|$ near −25 dB. (b) Response at the capacitive node of a doubly terminated pad: the T-coil moves the 0.63 pF pole from 10.1 to 14.3 GHz with a maximally flat response.

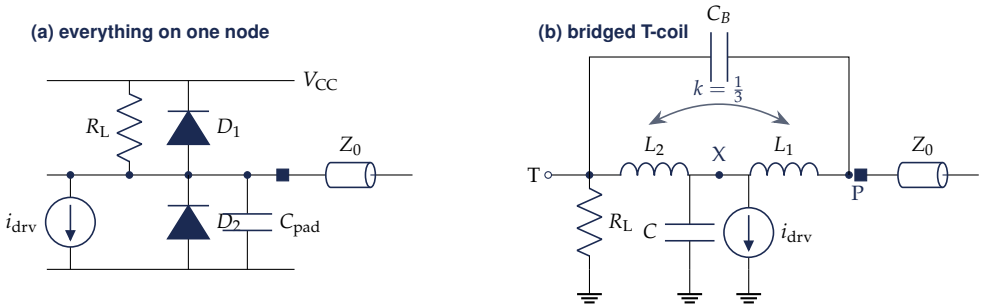


Figure 9.9. A T-coil hides the pad capacitance from the line. (a) Without it, driver, termination R_L, ESD diodes D_1/D_2 and pad share one node, and $C_p = C_{drv} + C_{ESD} + C_{pad}$ sets both the pole of Equation (9.5) and the return loss. (b) With a bridged T-coil the load $C = C_{ESD} + C_{drv}$ sits at the center tap X, and R_L (returned to V_{CC}) at the far end T. The line at P sees R_L at all frequencies.

▶ Design rule of thumb

At 5 Gb/s a T-coil is *optional*. Below about 0.85 pF the pad already meets 10 dB of return loss at f_N, and its pole at 7.5 GHz or above costs little ISI. Add a T-coil if ESD and driver push C_p toward 1 pF, or if the return-loss target is tighter (15 dB at f_N). Either way, give the system model your extracted C_p, not the 300 fF of REF.

9.7 TX design checklist

Table 9.1 collects the checks and a first-cut power estimate of about 100 mW, half of it in the driver (Section 14.6 adds the RX and clocking).

KEY TAKEAWAYS

- A CMOS clock tree needs 8 FO4 per period. With $t_{FO4} \approx 42$ ps (55 ps slow), full-rate CMOS cannot reach 5 Gb/s and half-rate CMOS fails at the slow corner, so the final 2:1 mux and its 2.5 GHz clock are HBT CML.

Table 9.1. TX checklist and first-cut power at 1 Vppd. CML gates as in Worked Example 9.3, about 1.2 mA each with emitter followers. Confirm every line in **ngspice**.

Block	Check before layout	Power
3-tap CML driver	$I_T = 20\,\text{mA}$ (16 mA at 0.8 Vppd); every $V_{CE} < \text{BV}_{CEO}$ at $V_{CC} + 5\,\%$	50 mW
Pre-drivers ($\times 3$)	full switching ($\geq 4V_T$); tap paths matched to a few ps	10 mW
2:1 muxes ($\times 3$), latches ($\times 4$)	glitch-free phases; 5-bit tap DACs, $-3.5/-6$ dB presets	21 mW
Clock buffers, 4:2, $\div 2$	duty cycle $50 \pm 1\,\%$; clock-to-data windows at all corners	20 mW
CMOS 16:4 (1.2 V)	FO4 measured; timing closed at 625 MHz	3 mW
Termination and pad	poly R_L for 20 mA; $C_p < 0.85$ pF or T-coil; RL ≥ 10 dB at f_N	–
Total TX	41 mA from 2.5 V, 2.5 mA from 1.2 V	**≈104 mW**

- A half-rate mux uses the clock as the select. Each input must change only while deselected, and the clock duty cycle becomes the UI width.
- Double termination absorbs echoes at both ends for half the swing. A CML driver needs $I_T = V_{d,\text{pp}}/Z_0$: 20 mA and 50 mW from 2.5 V at 1 Vppd, four times a voltage-mode driver.
- HBTs switch fully with about 100 mV: a 2.5 GHz CML stage has a gain of 5.8 against 2 for MOS at the same current.
- The off driver transistor sees the highest V_{CE}; keep it below $\text{BV}_{CEO} \approx 1.6$ V at every corner.
- The FFE is current-steering pairs on shared loads; -6 dB is $c_0 = 0.75$, $c_1 = -0.25$ (15 and 5 mA). Five bits are enough.
- Keep C_p below 0.85 pF for 10 dB return loss at f_N; a T-coil ($k = 1/3$, $C_B = C/8$) absorbs larger loads.

CHAPTER 10

Receiver Front-End and Samplers

The receiver must turn a 100 mV, 200 ps-wide eye into clean logic levels, 10^{12} times in a row without a mistake. This chapter builds the circuits that do it.

IN THIS CHAPTER YOU WILL LEARN

- Draw the Open-SERDES receiver from pad to 16-bit word, and set its termination, common mode and coupling.
- List what a sampler must deliver (sensitivity, offset, aperture, metastability, kick-back, noise) and derive the regeneration time constant of the StrongARM and the HBT CML latch.
- Compute the metastability error probability and explain why edge samplers need extra latch stages.
- Estimate sampler offset from Pelgrom's law and size a 5-bit calibration DAC.
- Build a circuit-level noise and sensitivity budget that supports the system budget of Section 4.4.
- Deserialize half-rate decisions to 16 bits at 312.5 MHz and cross into the core clock domain.

10.1 The receiver front-end

Figure 10.1 shows the receive path. The 100 Ω termination absorbs the wave at the pad. The CTLE (Section 7.4) boosts the high frequencies, and the DFE summer subtracts the post-cursor echoes of earlier decisions (Section 8.2). Four samplers then decide. At half rate, D_0 and D_1 sample the data at 0° and 180° of the 2.5 GHz clock, one bit each per clock period. E_0 and E_1 sample the transitions between them at 90° and 270°. The deserializer turns these four 2.5 Gb/s streams into 16 data bits and 16 edge bits at 312.5 MHz. The CDR logic uses both to steer the phase interpolator (Chapter 13), which shifts the four clock phases until E_0 and E_1 sit on the zero crossings.

Termination and common mode. The RX termination is two 50 Ω poly resistors in series across the inputs, with their center tap AC-grounded by a capacitor C_{CM} (Figure 10.2). Differential signals see 100 Ω. Common-mode noise sees a low impedance and is absorbed too. The termination sits right at the pad, inside the ESD diodes, and shares the pad capacitance budget of Section 9.6. The CTLE input pair needs its own common mode, and the input stage must stay linear for the largest signal: long runs on a short channel arrive at nearly the full ±0.5 V. A switchable input attenuator, or the CTLE's own gain control, handles short channels (Section 7.4).

DC or AC coupling. With **DC coupling** the RX inherits the TX common mode: about 2.0 V from the CML driver of Section 9.4, which suits an HBT CTLE on the same 2.5 V supply. **AC coupling** with series capacitors lets each side set its own common mode, at the price of a high-pass corner. A run of N identical bits then droops by $1 - e^{-NT_b/\tau}$ (**baseline wander**). Off-chip 100 nF capacitors in the 100 Ω loop give $\tau = 10\,\mu s$, and PRBS31's 31-bit run droops by only 0.06 %. On chip, 2 pF into a 10 kΩ bias resistor gives $\tau = 20$ ns and a 27 % droop, which needs DC-balanced (scrambled or 8b/10b) data. Open-SERDES is DC-coupled for near-end loopback and has board footprints for off-chip capacitors.

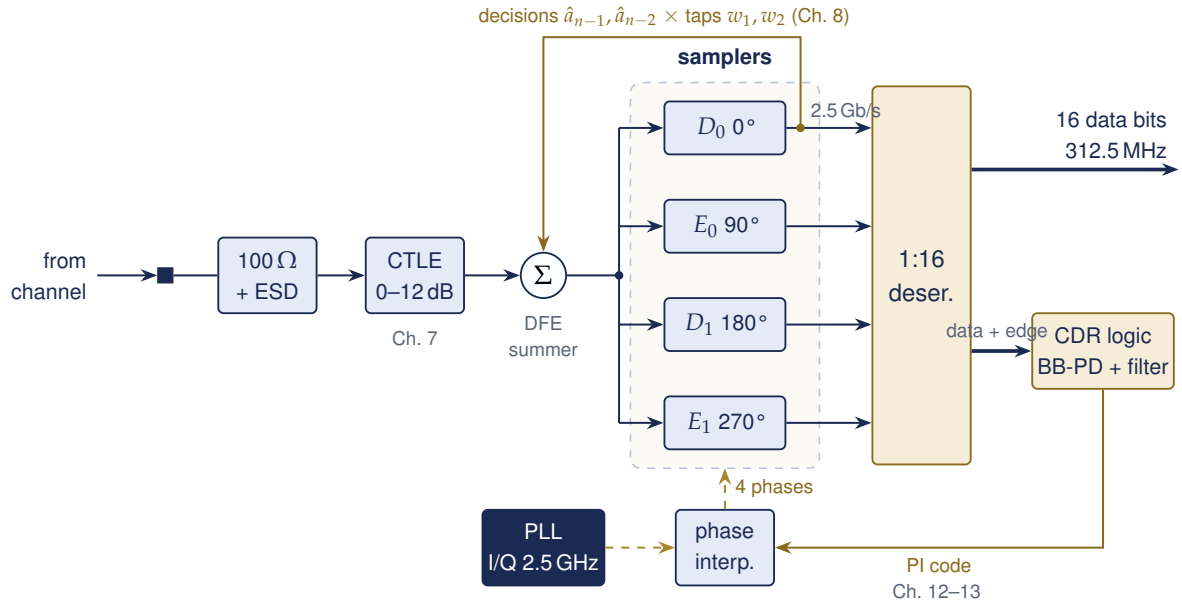


Figure 10.1. The Open-SERDES receiver. After termination, CTLE and DFE summer, four half-rate samplers decide on the four phases of the 2.5 GHz clock: data at 0°/180°, edges at 90°/270°. The decisions feed back to the summer (gold) and, deserialized to 312.5 MHz, drive the CDR logic that steers the phase interpolator. The DFE uses separate even and odd summers in practice (Section 8.2).

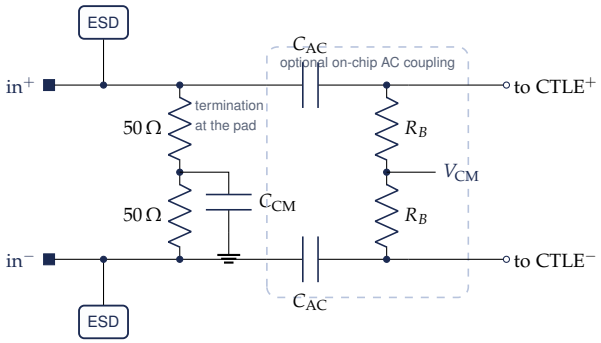


Figure 10.2. RX input network. Two 50 Ω resistors terminate the differential signal at the pad; the AC-grounded center tap also absorbs common-mode noise. An optional AC-coupling stage (C_{AC}, R_B) lets the RX set its own common mode V_{CM}, at the cost of baseline wander for long runs.

10.2 Samplers

A **sampler** (clocked comparator, or slicer) looks at its input at one instant and amplifies the sign of the difference to a full logic level. It is a positive-feedback latch, and everything we want from it follows from that. For Open-SERDES it must deliver:

- **Sensitivity:** a correct decision at 10^{-12} for inputs down to about 20 mVpp, the value budgeted in Section 4.4;
- **Offset** below about 5 mV after calibration (Section 10.4);
- **Aperture** well below 1 UI, so that it samples a point rather than averaging the eye, and a sampling bandwidth well above f_N;
- **Regeneration** fast enough that even tiny inputs resolve in time (Section 10.3), and a clock-to-output delay that fits the DFE loop of Section 8.2;
- **Low kick-back:** the large internal swings must not couple back into the CTLE and disturb

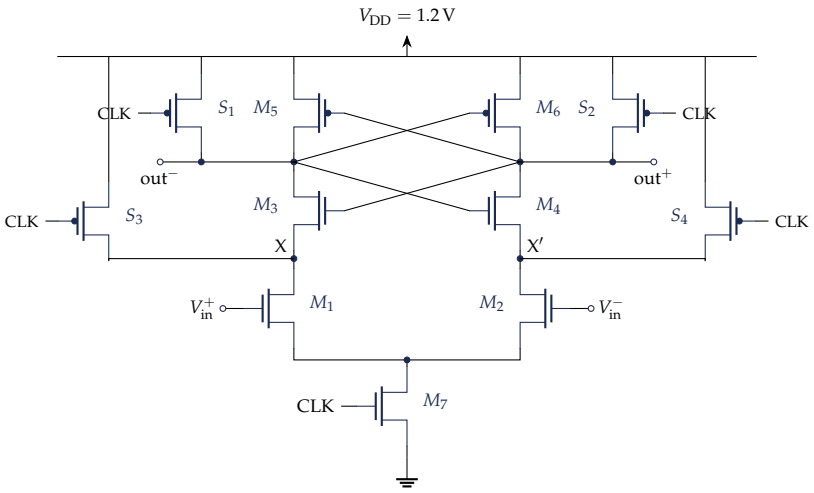


Figure 10.3. The StrongARM latch. The input pair M_1/M_2 integrates the input onto X/X′ and the outputs. The cross-coupled inverters M_3/M_5 and M_4/M_6 (crossing wires) regenerate it, and S_1–S_4 precharge every internal node while CLK is low. The outputs drive an SR latch (not shown). No current flows once a decision is made.

the other three samplers;

- **Low noise:** at least the kT/C noise of its internal nodes, $\sqrt{kT/C} = 0.45\,\text{mV}$ at 20 fF.

10.2.1 The StrongARM latch

The **StrongARM** latch (Figure 10.3) is the standard CMOS comparator [29, 30]. It draws no static current and produces rail-to-rail outputs. Its operation has four phases:

1. **Reset** (CLK low): S_1–S_4 precharge out$^{\pm}$, X and X′ to V_{DD}, and M_7 is off.
2. **Integration** (CLK rises): M_7 turns on. The input pair M_1/M_2 discharges X and X′, then the output nodes, with currents $I_D \pm g_m\, v_{in}/2$. The common mode falls and a differential voltage grows linearly.
3. **Regeneration:** when the outputs have fallen by $|V_{TP}|$, the PMOS M_5/M_6 turn on. The two cross-coupled inverters (M_3/M_5, M_4/M_6) now amplify the difference exponentially. The input pair is in triode and no longer sees the input.
4. **Decision:** one output returns to V_{DD} and the other falls to ground. An SR latch holds the result while the next reset begins [31].

Regeneration time constant

In regeneration each inverter, biased near its switching point, is a transconductance g_m (NMOS plus PMOS) driving its own output capacitance C from the other output: $C\,\dot{v}_1 = -g_m v_2$ and $C\,\dot{v}_2 = -g_m v_1$. Subtracting, the difference $\Delta V = v_1 - v_2$ obeys $C\,\dot{\Delta V} = g_m \Delta V$, a growing exponential. The integration phase sets its starting value: the outputs fall at I_D/C, so regeneration begins after $t_1 = C|V_{TP}|/I_D$, and by then the input pair has built $\Delta V(t_1) = g_{m,in} v_{in} t_1/C = A_{int}\, v_{in}$.

KEY EQUATION · StrongARM regeneration and decision time

$$\Delta V(t) = A_{int}\, v_{in}\, e^{(t-t_1)/\tau}, \qquad \tau = \frac{C}{g_m}, \qquad t_{dec} = t_1 + \tau \ln \frac{V_L}{A_{int}\, v_{in}}. \tag{10.1}$$

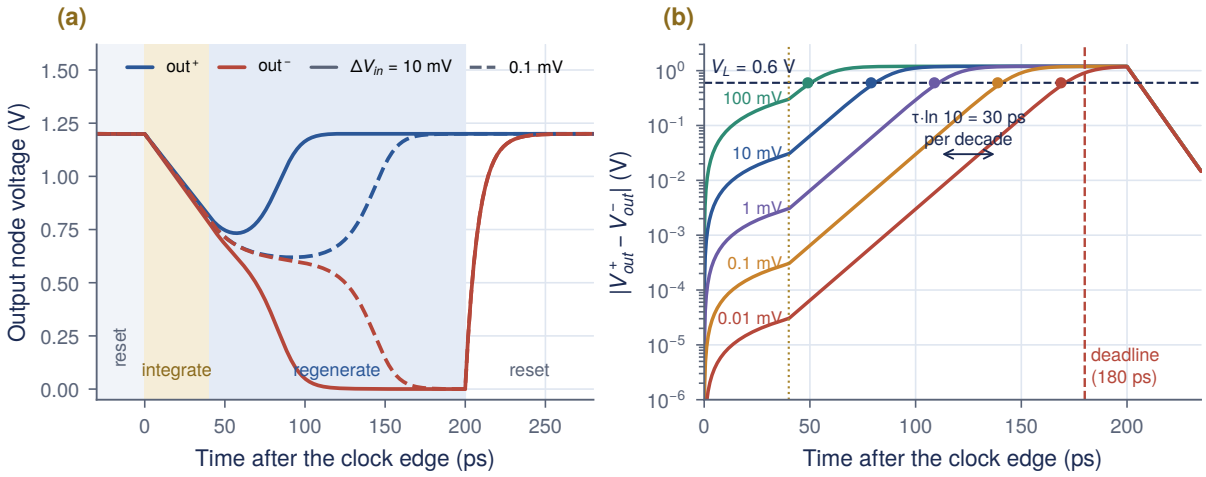


Figure 10.4. A latch amplifies exponentially: halving the input only adds a fixed delay. Behavioral StrongARM model ($t_1 = 40\,\text{ps}$, $A_{\text{int}} = 3$, $\tau = 13\,\text{ps}$). (a) Output nodes for 10 mV and 0.1 mV inputs: both fall together during integration, then split. (b) Output difference on a log scale for inputs from 100 mV to 10 µV. The straight lines have slope $1/\tau$, and each decade of input costs 30 ps. Inputs below about 4 µV miss the 180 ps deadline.

> Each factor of 10 less input adds $\tau \ln 10 = 2.3\tau$ to the decision time. V_L is the swing the next stage needs.

Figure 10.4 shows a behavioral model with the numbers of Worked Example 10.1: $C = 20\,\text{fF}$, $I_D = 200\,\mu\text{A}$ and $|V_{\text{TP}}| = 0.4\,\text{V}$ give $t_1 = 40\,\text{ps}$ and $A_{\text{int}} = 3$; the cross-coupled pair gives $\tau = 20\,\text{fF}/1.5\,\text{mS} \approx 13\,\text{ps}$. A 1 mV input decides in 109 ps, a 10 µV input in 169 ps.

10.2.2 The HBT CML latch

The CML latch (Figure 10.5) is the bipolar alternative [24]. While CK is high, the track pair Q_1/Q_2 is an ordinary CML buffer and the outputs follow the input with time constant $R_{\text{L}}C$. When CK falls, the tail current switches to the cross-coupled pair Q_3/Q_4, which holds and amplifies whatever difference the outputs carried at that instant. Two latches on opposite clock phases form a master–slave flip-flop.

KEY EQUATION · CML latch regeneration

$$C\,\dot{\Delta V} = \left(g_{\text{m}} - \frac{1}{R_{\text{L}}}\right)\Delta V \quad\Longrightarrow\quad \tau = \frac{R_{\text{L}}C}{g_{\text{m}}R_{\text{L}} - 1}, \qquad g_{\text{m}} = \frac{I_T/2}{V_T}. \tag{10.2}$$

The cross-coupled pair adds $g_{\text{m}}\Delta V$ of current and the loads remove $\Delta V/R_{\text{L}}$; regeneration needs a loop gain $g_{\text{m}}R_{\text{L}} > 1$.

Why it is fast in SG13G2. With $I_T = 1\,\text{mA}$, each side carries 0.5 mA and $g_{\text{m}} = 19.3\,\text{mS}$; $R_{\text{L}} = 400\,\Omega$ gives a 400 mV swing and $g_{\text{m}}R_{\text{L}} = 7.7$. With $C = 30\,\text{fF}$, Equation (10.2) gives $\tau = 1.8\,\text{ps}$, and 3–5 ps once base resistance and C_μ are included. That is three to seven times faster than the StrongARM, because the HBT's $g_{\text{m}}/I = 1/V_T$ and its 300 GHz f_T give large g_{m} with little capacitance. The latch also kicks back less, since its current never changes, and its sampling instant does not move with the input common mode. Its cost is static power: 2.5 mW per latch from 2.5 V.

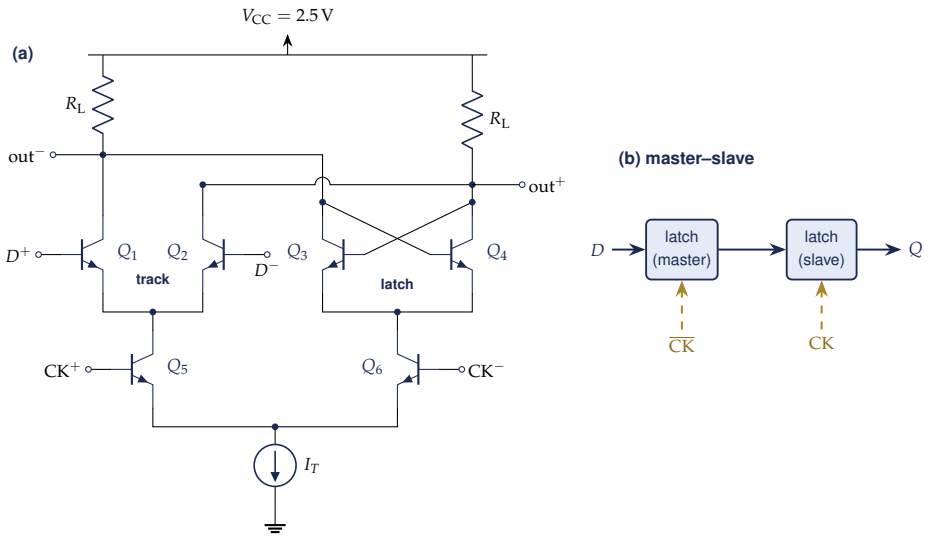


Figure 10.5. The HBT CML latch. (a) With CK high the track pair Q_1/Q_2 drives the loads; with CK low the tail current moves to the cross-coupled pair Q_3/Q_4 (crossing wires), which regenerates the stored difference. The swing is $I_T R_L$ and the supply current never changes. (b) A master latch on $\overline{\text{CK}}$ followed by a slave on CK forms a flip-flop that samples on the rising edge.

The Open-SERDES choice. All four samplers are HBT CML master–slave flip-flops. The data samplers must, because the direct-feedback DFE needs a clock-to-output delay of 25–35 ps (Section 8.2); a 1.2 V StrongARM with its SR latch needs 100 ps or more for small inputs. The edge samplers use the same circuit so that data and edge samplers have the same aperture delay, as the next paragraph shows. A StrongARM remains the right choice where a decision can wait, such as the DFE's error sampler (Section 8.3) or slow monitors.

Aperture. A real sampler does not take an instantaneous sample. Its output is a weighted average $\int v_{\text{in}}(t)h(t)\,dt$ with $\int h = 1$, where $h(t)$ is the **sampling function** [32]. The **aperture** w_{80} is the time that holds the central 80 % of the weight. Its Fourier transform is the sampler's own bandwidth. Figure 10.6 compares the two latches. The StrongARM integrates during the 40 ps after the clock edge; the CML latch remembers the input with its track time constant $R_L C = 12$ ps *before* the edge. Both apertures are about 25–30 ps (0.12–0.14 UI), and both bandwidths, 12–15 GHz, cost less than 0.2 dB at f_N. But their centroids differ by 23 ps. A StrongARM edge sampler next to CML data samplers would make the CDR lock 0.12 UI away from the true eye center. Keep $w_{80} \le 0.2$ UI with sharp clock edges, and use one sampler type for data and edge.

10.3 Metastability

If the input is close enough to zero, the latch has not reached a valid logic level V_L when its time T runs out. It is then **metastable**, and downstream logic may read it either way, or both ways at once. By Equation (10.1) the latch fails only if its starting value is below $V_L e^{-T/\tau}$, that is, if $|v_{\text{in}}| < V_L e^{-T/\tau}/A$, where A is the gain in front of the regenerative node.

KEY EQUATION · Metastability error probability

$$P_{\text{meta}} = \frac{2V_L}{A\,\Delta V}\,e^{-T/\tau}, \qquad \text{and for cascaded stages} \quad P_{\text{meta}} \propto e^{-(T_1/\tau_1 + T_2/\tau_2 + \dots)}. \tag{10.3}$$

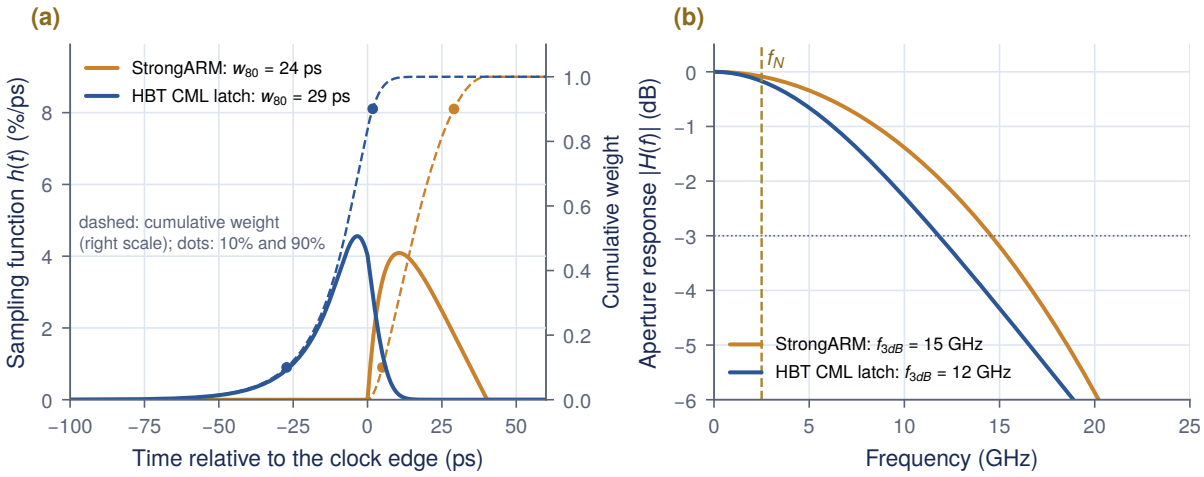


Figure 10.6. A sampler averages its input over its aperture. (a) Sampling functions $h(t)$ of a StrongARM (weight after the clock edge, during integration) and an HBT CML latch (weight before the edge, through the track time constant), with their cumulative weights. Both have $w_{80} \approx$ 25–30 ps, but their centroids are 23 ps apart. (b) The same functions as filters: both pass f_N with less than 0.2 dB of loss.

The input is assumed spread uniformly over a range ΔV; the fraction that lands in the failing window of width $2V_L e^{-T/\tau}/A$ is P_{meta}. Each latch stage behind the first multiplies the probability by its own $e^{-T/\tau}$.

WORKED EXAMPLE 10.1 · StrongARM metastability, data versus edge samplers

Timing: the half-rate clock period is 400 ps, so evaluation lasts 200 ps. Integration takes $t_1 = 40$ ps and the SR latch needs about 20 ps, which leaves $T = 140$ ps of regeneration with $\tau = 13$ ps: a gain of $e^{10.8} = 4.8 \times 10^4$.

Data sampler: reaching $V_L = 0.6$ V needs $0.6/4.8 \times 10^4 = 12$ µV at the latch, or 4 µV at the input with $A_{\text{int}} = 3$. An equalized eye never puts the data sample that close to zero, so data samplers are safe.

Edge sampler: it samples the zero crossing *on purpose*, so its input is often tiny. Take the input spread over ±100 mV ($\Delta V = 0.2$ V) and $A = 3$: $P_{\text{meta}} = (2 \cdot 0.6/0.6)\, e^{-10.8} =$ $\mathbf{4.2 \times 10^{-5}}$ per sample, or 10^5 undefined decisions per second at 2.5 GS/s. That is far too many for a digital CDR loop.

Fix: two more latch stages, each with 200 ps and $\tau \approx 20$ ps, multiply by e^{-20}: $P_{\text{meta}} =$ $\mathbf{8.7 \times 10^{-14}}$, below the BER target. A metastable edge decision then costs one bang-bang vote, which the loop filter averages away.

HBT CML latch: with $\tau \approx 4$ ps and the same 140 ps, $P_{\text{meta}} \approx 4 \times 10^{-16}$ in the master latch alone (with $V_L = 0.2$ V).

▶ Edge samplers need extra stages

A data sampler sees an open eye; an edge sampler is aimed at the zero crossing, so it is metastable thousands of times more often. Always put at least two more latch stages between an edge sampler and the CDR logic, and count their regeneration time. In Open-SERDES the CML slave latch and the first CMOS flip-flops of the deserializer provide

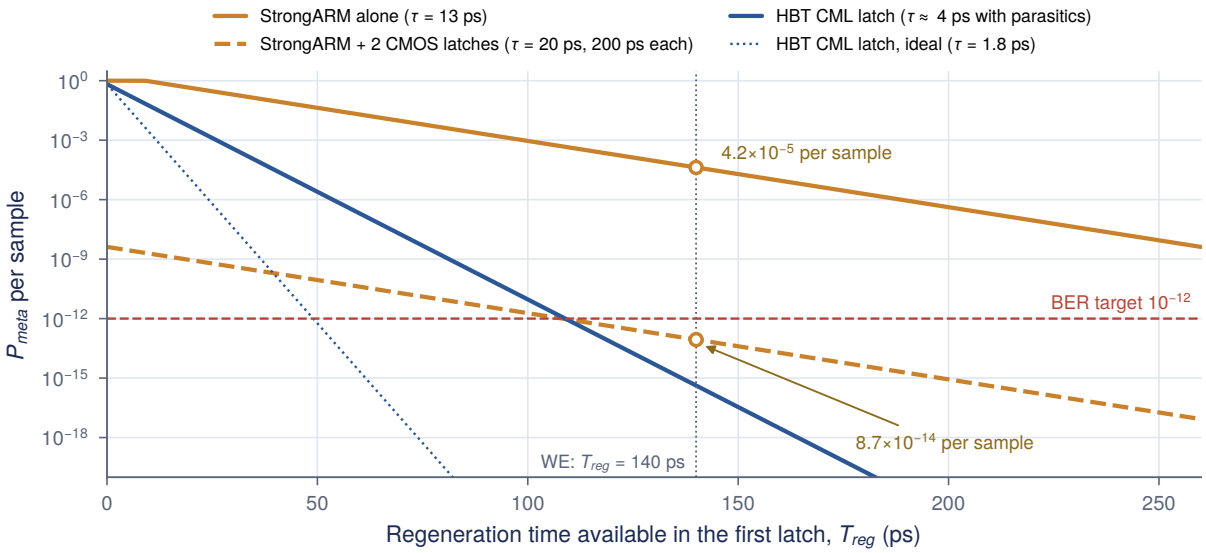


Figure 10.7. Metastability falls exponentially with regeneration time, and every extra latch stage adds its own exponent. Equation (10.3) with $\Delta V = 0.2\,\mathrm{V}$ and $A = 3$. A lone StrongARM leaves 4.2×10^{-5} at the 140 ps of Worked Example 10.1; two extra CMOS latches bring it below 10^{-12}. The HBT CML latch crosses 10^{-12} after about 110 ps.

them.

10.4 Offset and calibration

Two transistors drawn identically are not identical. Random dopant and edge fluctuations give the threshold voltages of a matched pair a random difference whose standard deviation falls with gate area [33]. A sampler with mismatched input devices therefore has an input-referred **offset** V_{os}: it decides as if its threshold were V_{os} instead of zero.

KEY EQUATION · Pelgrom mismatch and calibration sizing

$$\sigma_{V_{os}} \approx \frac{A_{VT}}{\sqrt{WL}}, \qquad \text{range} = \pm 3\sigma_{V_{os}}, \qquad \text{LSB} = \frac{6\sigma_{V_{os}}}{2^{n}}, \qquad |V_{os,res}| \le \frac{\text{LSB}}{2}. \tag{10.4}$$

Halving σ needs four times the area, and so four times the input capacitance. A DAC is cheaper.

WORKED EXAMPLE 10.2 · Sizing the offset DAC

Given: an input pair of $W/L = 4\,\mu\mathrm{m}/0.13\,\mu\mathrm{m}$ ($WL = 0.52\,\mu\mathrm{m}^2$) and $A_{VT} \approx 3.5\,\mathrm{mV}\,\mu\mathrm{m}$. That A_{VT} is an assumption for 130 nm; extract the real value by Monte Carlo if the SG13G2 models include mismatch.

Offset: $\sigma = 3.5/\sqrt{0.52} = 4.9\,\mathrm{mV}$, so $\pm 3\sigma = \pm 14.6\,\mathrm{mV}$. Untrimmed, only 70 % of samplers meet $|V_{os}| < 5\,\mathrm{mV}$, and all four in one receiver do so on only 24 % of chips (Figure 10.8).

DAC: a residual of $\pm 1\,\mathrm{mV}$ needs LSB $\le 2\,\mathrm{mV}$, so $n \ge \log_2(29.2/2) = 3.9$. Choose **5 bits**: LSB = 0.91 mV, residual $\pm 0.46\,\mathrm{mV}$. After trim, 99.8 % of samplers are within 1 mV.

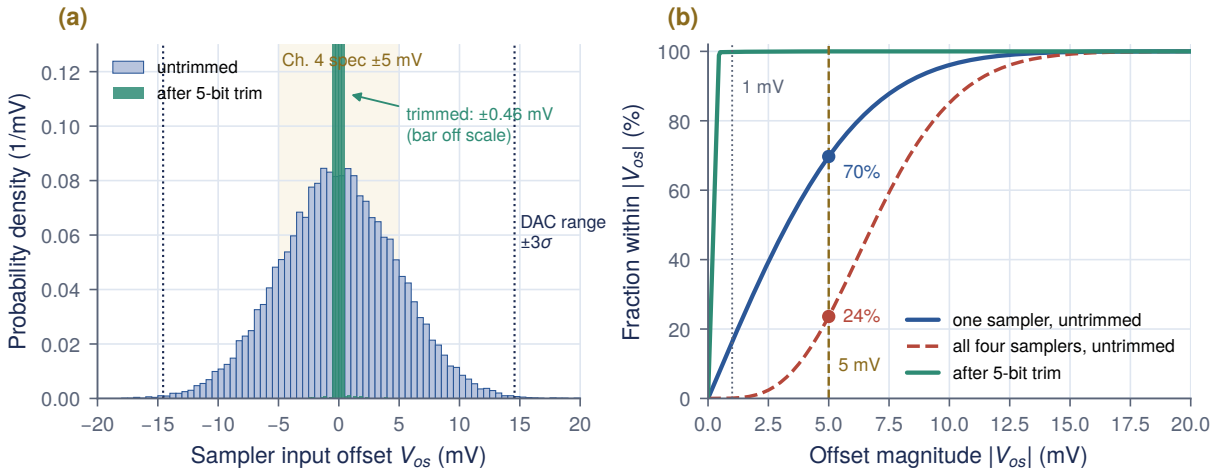


Figure 10.8. Mismatch spreads sampler offsets over tens of millivolts; a 5-bit DAC pulls them inside half a millivolt. Monte Carlo of 20 000 samplers with $\sigma = 4.9\,\text{mV}$ (Worked Example 10.2). (a) Distribution before and after trimming with a $\pm 3\sigma$, 5-bit DAC. (b) Fraction of samplers within a given offset. Untrimmed, 70 % of single samplers but only 24 % of four-sampler receivers meet the 5 mV budget of Section 4.4.

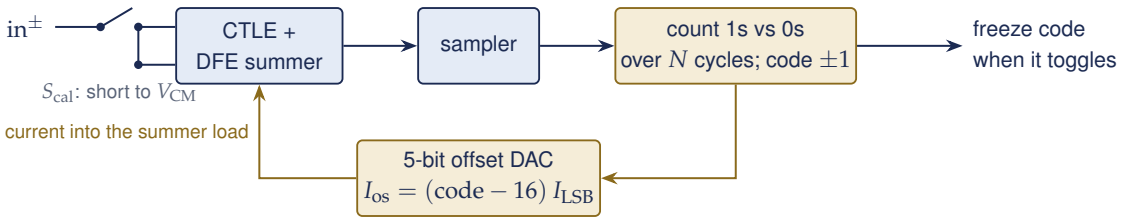


Figure 10.9. Offset calibration is a slow digital loop. With the inputs shorted, a sampler with zero offset outputs ones and zeros equally often. The counter steps a 5-bit current DAC in the summer load until the code toggles by one LSB (0.91 mV, or 3.6 µA into 250 Ω), then freezes it. Running it at power-up and after large temperature changes keeps the residual offset inside the budget.

> **The no-calibration alternative** needs $\sigma = 1\,\text{mV}$, or $WL = 12.2\,\mu\text{m}^2$: 24 times the area, with the capacitance, kick-back and slower τ that come with it.

Calibrating the Open-SERDES receiver. The HBT pairs of the CML samplers match better than MOS pairs, but the CTLE, the summer and the load resistors add offsets of their own. Open-SERDES therefore trims the whole path, one DAC per sampler, by injecting a differential current into the DFE summer load (Figure 10.9). The total current stays constant, so the common mode does not move. Referred to the input with unity front-end gain, one LSB of 0.91 mV across the 250 Ω summer load is 3.6 µA, and the full ±16 LSB range is ±58 µA. The procedure is a short digital loop: short the inputs to the common mode, count ones and zeros for a few thousand cycles, step the code toward 50/50, repeat until it toggles by one LSB, then freeze it. Capacitive trim on the StrongARM's internal nodes is the alternative for CMOS samplers; it costs aperture time.

▶ What this means for your design

Do not buy offset with area. Use minimum-capacitance input devices for speed, calibrate the whole path with a 5-bit DAC sized for $\pm 3\sigma$, and leave the 5 mV of the system budget for drift between calibrations. Simulate offset with Monte Carlo mismatch, never with ideal

devices.

10.5 Noise and the sensitivity budget

Section 4.4 budgeted the receiver with three numbers: $\sigma_n = 1.5\,\text{mV}$ rms input-referred noise, $V_{os} = 5\,\text{mV}$ after calibration, and a sampler sensitivity $V_{sens} = 20\,\text{mV}$. Here we check them against the circuits. The sensitivity of a sampler collects its own Gaussian noise σ_s at the BER multiplier, the minimum overdrive V_{min} that resolves in time (metastability, hysteresis and kick-back memory), and its residual offset [24]:

KEY EQUATION · Sampler sensitivity

$$V_{sens} \approx 2Q\,\sigma_s + V_{min} + 2|V_{os,res}|, \qquad 2Q = 14.07 \text{ at } 10^{-12}. \tag{10.5}$$

WORKED EXAMPLE 10.3 · Circuit-level noise and sensitivity budget

All terms are rms and referred to the RX input, conservatively with unity front-end gain.

Source	σ (mV)	How it was estimated
Termination and line, 50 Ω diff.	0.10	$0.91\,\text{nV}/\sqrt{\text{Hz}}$ over an 11.8 GHz noise bandwidth
CTLE (shot noise, R_E, R_C)	0.25	Section 7.4 densities; confirm with `.noise`
DFE summer loads	0.32	$\sqrt{2kT/C}$, $C = 80\,\text{fF}$
Sampler, input-referred	1.0	kT/C floor 0.45–0.64 mV; 0.7–0.85 mV measured [32]
Total (RSS)	**1.08**	system budget assumes 1.5

Sensitivity: with $\sigma_s = 1\,\text{mV}$, $V_{min} \approx 5\,\text{mV}$ and the 5-bit residual of 0.46 mV, Equation (10.5) gives $14.1 + 5 + 0.9 = \mathbf{20}\,\text{mV}$, the value of Section 4.4.

Margin: with the circuit noise, the linear budget of Section 4.4 becomes $131 - 14.07 \times 1.08 - 3.5 - 10 - 20 = \mathbf{82}\,\text{mV}$ instead of 76 mV. Since V_{sens} already contains the sampler's own noise and residual offset, the system budget counts them twice and is about 15 mV conservative.

The budget follows Palermo's rule for noise sources [24]: classify each one as bounded or Gaussian, and as proportional to swing or independent of it. Thermal noise, offset and sensitivity are independent of swing, so they shrink in relative terms as the eye grows. Residual ISI and crosstalk are proportional to swing, and no TX amplitude can beat them. With 82 mV of margin, the receiver circuits are not the weak link; equalization is.

10.6 The deserializer

The first 1:2 of the deserializer is free: a half-rate receiver already delivers even and odd bits on separate samplers. The rest mirrors the transmitter (Figure 10.10). Each sampler's 2.5 Gb/s output is converted from CML to CMOS levels and enters a 1:8 tree of CMOS 1:2 stages clocked at 1.25 GHz, 625 MHz and 312.5 MHz. That is 400 ps per bit at the first CMOS flip-flop, or 9.5 FO4, which a 1.2 V CMOS flip-flop handles; a CMOS 1:2 demux is also far less sensitive to duty cycle than the TX mux. The two data lanes interleave into the 16-bit word, and the two

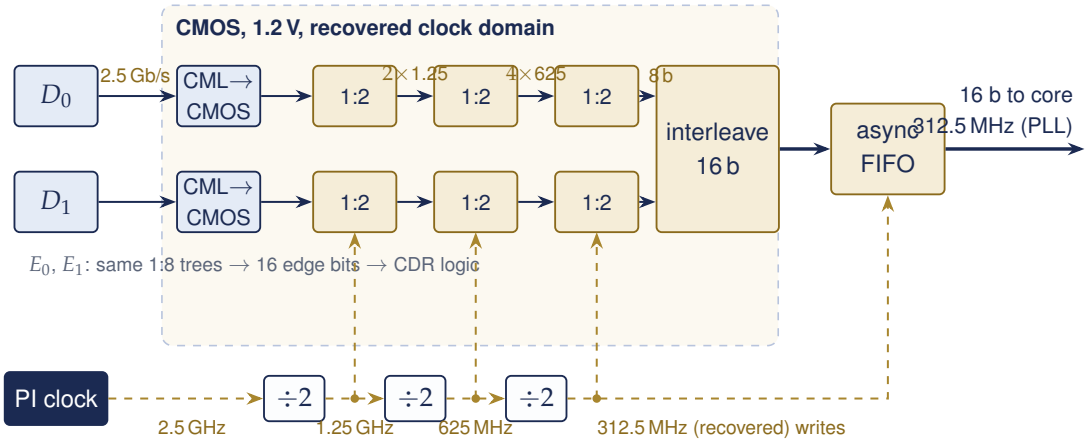


Figure 10.10. The deserializer mirrors the serializer. The half-rate samplers already provide the first 1:2. Each data lane goes from CML to CMOS and through three 1:2 stages to 8 bits at 312.5 MHz; the two lanes interleave into the 16-bit word. The edge lanes do the same for the CDR. All clocks are divided from the recovered (PI) clock, and an asynchronous FIFO hands the word to the core clock.

edge lanes give 16 edge bits for the CDR logic. The clocks come from the recovered 2.5 GHz clock through ÷2 stages, so the whole tree lives in the **recovered clock domain**.

Crossing into the core clock. The recovered 312.5 MHz clock and the core's 312.5 MHz clock, which comes from the local PLL, have the same frequency in near-end loopback, or when both ends share a reference, but an unknown and wandering phase. A small asynchronous FIFO, 4–8 words deep with Gray-coded pointers, absorbs that. If the far end has its own reference (±100 ppm each), the frequencies differ by up to 200 ppm: one word slips every 5000 words, or every 16 µs. That needs an elastic buffer with skip and insert symbols, which is a protocol feature. For bring-up, run the PRBS checker directly in the recovered domain, before the FIFO, so a BER measurement needs no clock crossing at all.

10.7 RX design checklist

Table 10.1 collects the RX checks and a first-cut power estimate of about 70 mW, without the PLL, phase interpolator and CDR logic (Section 14.6).

KEY TAKEAWAYS

- The Open-SERDES receiver is termination, CTLE, DFE summer and four half-rate samplers (data at 0°/180°, edges at 90°/270°), followed by a 1:16 deserializer and the CDR logic.
- A latch amplifies exponentially, $\Delta V \propto e^{t/\tau}$: $\tau = C/g_m \approx 13$ ps for a 130 nm StrongARM and $\tau = R_L C/(g_m R_L - 1) \approx$ 2–5 ps for an SG13G2 HBT CML latch.
- Open-SERDES uses HBT CML master–slave samplers everywhere: the DFE loop needs their 25–35 ps clock-to-output, and one sampler type keeps the data and edge apertures aligned.
- $P_{\text{meta}} = (2V_L/A\Delta V)\, e^{-T/\tau}$. A lone StrongARM edge sampler gives 4×10^{-5}; two extra latch stages bring it below 10^{-13}.
- Pelgrom's law gives $\sigma \approx 4.9$ mV for a small 130 nm pair; a 5-bit DAC over $\pm 3\sigma$ leaves ± 0.46 mV.
- The circuit noise is about 1.1 mV rms and the sensitivity about 20 mV, which supports the system budget with about 80 mV of margin.
- Half rate gives the first 1:2 for free; CMOS does the rest, and a small FIFO crosses from the recovered to

Table 10.1. RX checklist and first-cut power. CML currents from the worked examples here and in Chapters 7 and 8. Confirm every line in **ngspice** with extracted parasitics.

Block	Check before layout	Power
Termination, ESD, pad	100 Ω diff., AC-grounded center tap; $C_p < 0.85$ pF; RL ≥ 10 dB at f_N	–
CTLE and buffer	linear range for the long-run level; 0–12 dB peaking (2 mA + 3 mA)	12.5 mW
DFE summers (×2), taps	settles within the tap-1 budget; offset DAC injection (6 mA)	15 mW
4 CML samplers (M–S)	$\tau \leq 5$ ps; $t_{CK\to Q} \leq 35$ ps; same type for data and edge (12 mA)	30 mW
I/Q clock buffers	sharp edges ($w_{80} \leq 0.2$ UI); phase error $\leq 5°$ (5.6 ps)	10 mW
Offset DACs (×4)	5 bits, $\pm 3\sigma$ range, LSB 0.91 mV	0.6 mW
Deserializer (1.2 V)	two extra latch stages behind E_0/E_1; FIFO or in-domain PRBS checker	5 mW
Total RX	27 mA from 2.5 V, 4 mA from 1.2 V	**≈73 mW**

the core clock.

PART IV

Clocking

Every bit is only as good as the clock edge that samples it. Jitter, clock generation with a phase-locked loop, and clock and data recovery.

CHAPTER 11

Jitter

A 200 ps bit leaves no room for sloppy clock edges: this chapter turns "the clock wobbles" into numbers you can budget, measure and design against.

IN THIS CHAPTER YOU WILL LEARN

- Define time-interval error, period jitter and cycle-to-cycle jitter, and say which one a serial link cares about.
- Convert voltage noise into timing jitter through the slew rate.
- Separate random from deterministic jitter and combine them with the dual-Dirac model, $\mathrm{TJ} = \mathrm{DJ}_{\delta\delta} + 2Q\,\sigma_{\mathrm{RJ}}$.
- Integrate a phase-noise spectrum $\mathcal{L}(f)$ into rms jitter with the right limits.
- Build the Open-SERDES jitter budget and check it against 0.3 UI at 10^{-12}.

11.1 What jitter is

An ideal clock has edges at exactly $t = nT$; a real clock has edges at t_n, each a little early or late. **Jitter** is that deviation. Three measures are in common use (Figure 11.1a):

$$\mathrm{TIE}_n = t_n - nT, \qquad J_{\mathrm{per},n} = T_n - T = \mathrm{TIE}_{n+1} - \mathrm{TIE}_n, \qquad J_{\mathrm{cc},n} = T_{n+1} - T_n, \tag{11.1}$$

where $T_n = t_{n+1} - t_n$ is the n-th period. The **time-interval error** (TIE, also called absolute jitter) is the one a link cares about, because the sampler compares a clock edge with a data edge, not one clock period with the next. **Period jitter** and **cycle-to-cycle jitter** matter for digital setup and hold, but they can be tiny while the TIE is large, because small period errors add up.

Figure 11.1(b) shows the difference. A free-running oscillator has no memory of where its edges should be, so its TIE random-walks. A phase-locked loop (Chapter 12) compares its edges with

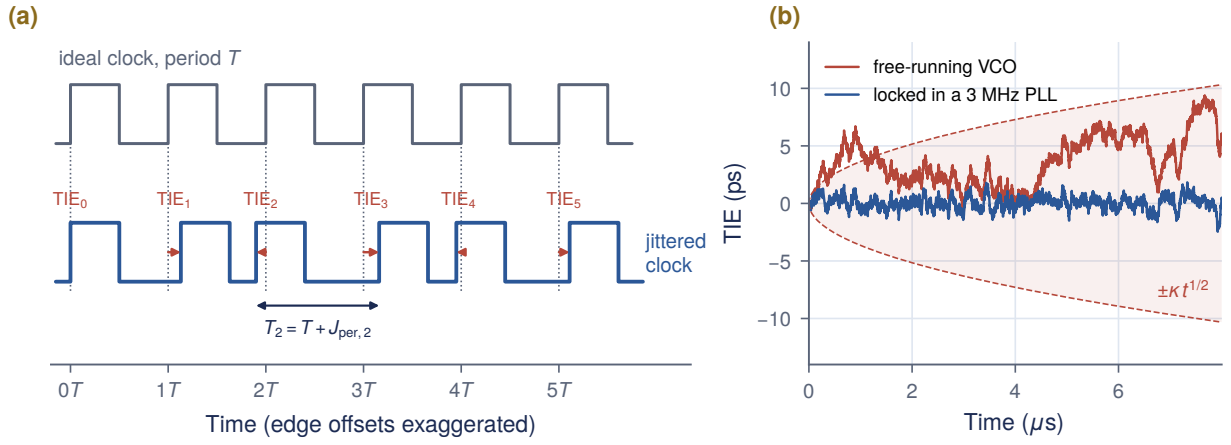


Figure 11.1. Jitter is measured against an ideal clock, and it accumulates. (a) TIE_n is the offset of each edge from nT; the period T_n differs from T by the period jitter. (b) TIE of the team's 2.5 GHz LC VCO (73 fs rms period jitter, Worked Example 11.2). Free-running, the TIE is a random walk that grows as $\kappa\sqrt{t}$ and passes 10 ps within 8 µs. Locked in a 3 MHz PLL it stays near 0.6 ps rms.

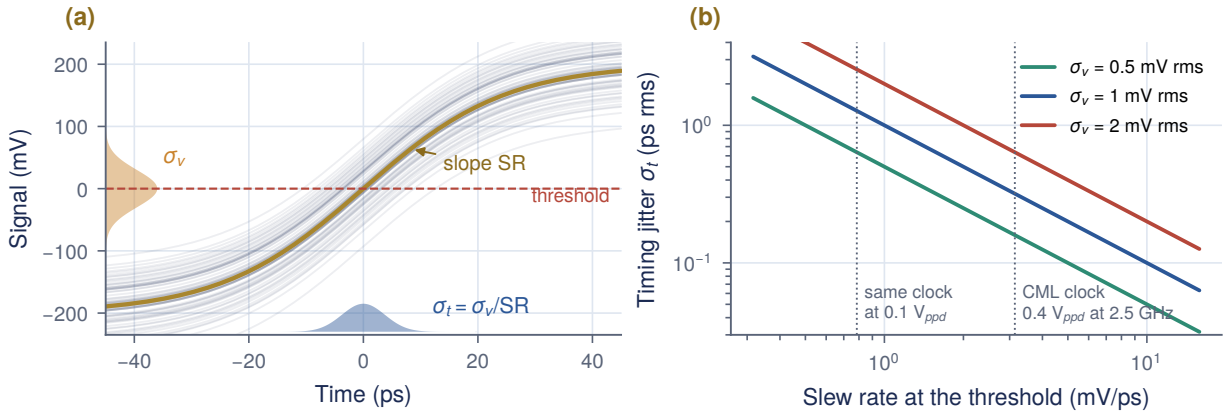


Figure 11.2. Timing jitter is voltage noise divided by the slope. (a) Noisy edges (noise exaggerated to 30 mV rms) cross the threshold at spread-out times, with spread σ_v/SR. (b) σ_t against slew rate. Attenuating a 2.5 GHz clock from 0.4 to 0.1 V_{ppd} makes it four times more jitter-sensitive.

a clean reference and keeps the TIE bounded. The CDR (Chapter 13) does the same for the data: it tracks slow jitter and leaves only the fast part to close the eye.

WORKED EXAMPLE 11.1 · Three jitter numbers from six edges

Given: $T = 400$ ps and edges at 0, 401.0, 799.5, 1201.5, 1600.0 and 2000.8 ps. **TIE:** 0, 1.0, −0.5, 1.5, 0, 0.8 ps: 2.0 ps peak-to-peak, 0.69 ps rms about the mean. **Period jitter:** 1.0, −1.5, 2.0, −1.5, 0.8 ps: 1.42 ps rms. **Cycle-to-cycle:** −2.5, 3.5, −3.5, 2.3 ps: 3.0 ps rms.

Lesson: $J_{cc} > J_{per} >$ TIE is typical of a locked clock: a bounded TIE means a long period is usually followed by a short one. Always say which measure a number is.

Voltage noise becomes timing noise. Near a threshold crossing, a noise voltage moves the crossing by the noise divided by the slope of the edge (Figure 11.2a):

KEY EQUATION · Noise-to-jitter conversion

$$\sigma_t = \frac{\sigma_v}{\mathrm{SR}}, \qquad \mathrm{SR} = \left.\frac{dv}{dt}\right|_{\text{crossing}} = 2\pi f_0 A_c \text{ for a sine of amplitude } A_c. \tag{11.2}$$

A 2.5 GHz CML clock of 0.4 V_{ppd} ($A_c = 0.2$ V) slews at 3.14 mV/ps, so 1 mV rms of noise costs 0.32 ps rms. Attenuated to 0.1 V_{ppd} by a lossy route, the same clock turns the same noise into 1.27 ps (Figure 11.2b). The relation also works in reverse: Section 2.6 showed that sampling a few picoseconds off the eye centre costs millivolts.

▶ Design rule of thumb

Keep clock swings large and edges fast wherever a clock is buffered or sampled: halving the slew rate doubles the jitter from the same noise. This is why Open-SERDES restores its 2.5 GHz clocks with CML limiting buffers (Section 12.5) instead of letting them decay along long wires.

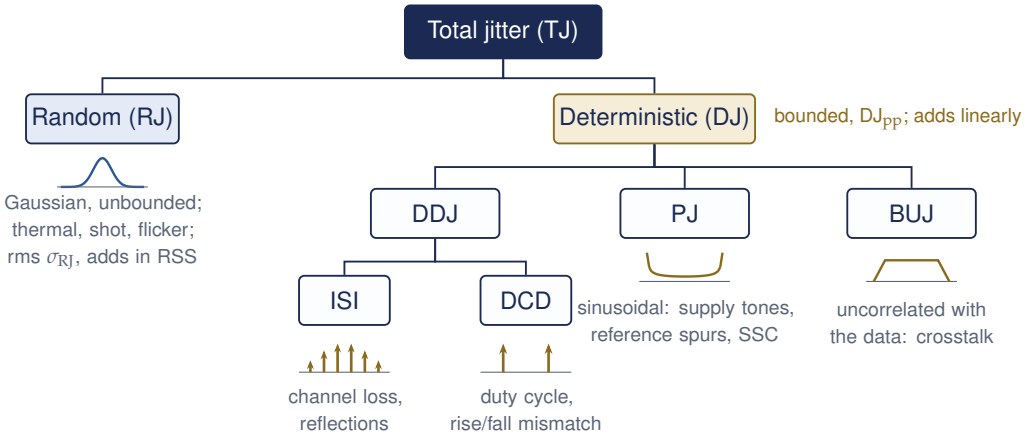


Figure 11.3. Jitter splits into an unbounded random part and bounded deterministic parts. Each leaf shows the shape of its probability density. RJ terms are independent Gaussians and add in root-sum-square; DJ terms are added linearly, which is conservative.

After Palermo [34].

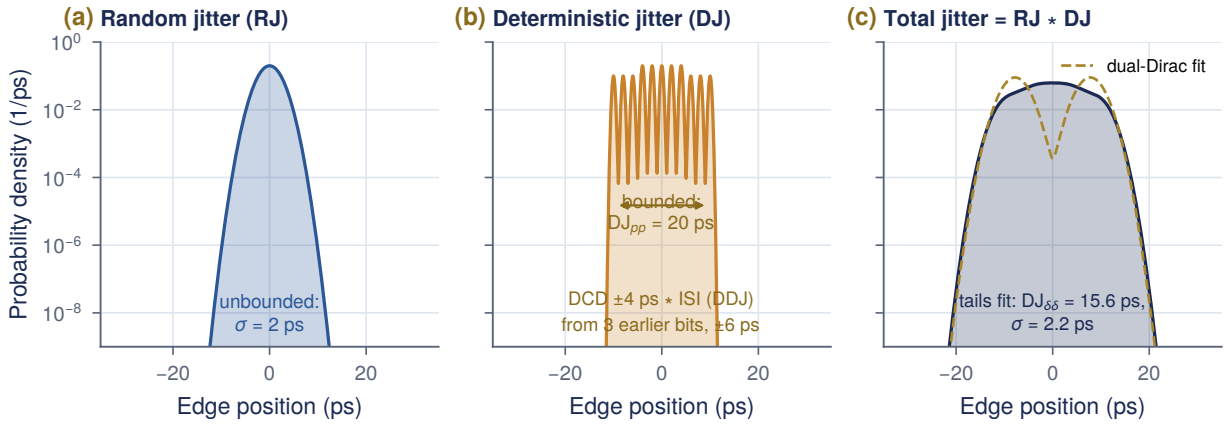


Figure 11.4. Total jitter is the convolution of RJ and DJ, and its tails are Gaussian (log scale). (a) RJ never ends. (b) DJ, here DCD of ± 4 ps convolved with ISI from three earlier bits, stops at ± 10 ps. (c) The convolution. The dashed dual-Dirac model is fitted to the tails between BER 10^{-4} and 10^{-12}; it ignores the centre and returns $\mathrm{DJ}_{\delta\delta} = 15.6$ ps, less than the true 20 ps peak-to-peak.

11.2 Random and deterministic jitter

Jitter sources fall into two families with different statistics (Figure 11.3). **Random jitter** (RJ) comes from thermal, shot and flicker noise. It is Gaussian and *unbounded*, and is described by its rms σ_{RJ}; a peak-to-peak RJ is meaningless without a BER. **Deterministic jitter** (DJ) is *bounded*. Its parts are **data-dependent jitter** (DDJ), mainly ISI from the channel plus **duty-cycle distortion** (DCD); **periodic jitter** (PJ) from supply tones, reference spurs or spread-spectrum clocking, whose arcsine PDF piles up at its peaks; and **bounded uncorrelated jitter** (BUJ), mostly crosstalk.

When independent sources act on the same edge, the PDF of the total is the *convolution* of their PDFs, the rule Section 4.3 used for ISI. In Figure 11.4 a 2 ps RJ convolved with 20 $\mathrm{ps}_{\mathrm{pp}}$ of DJ gives a total whose centre looks like the DJ and whose tails are Gaussian.

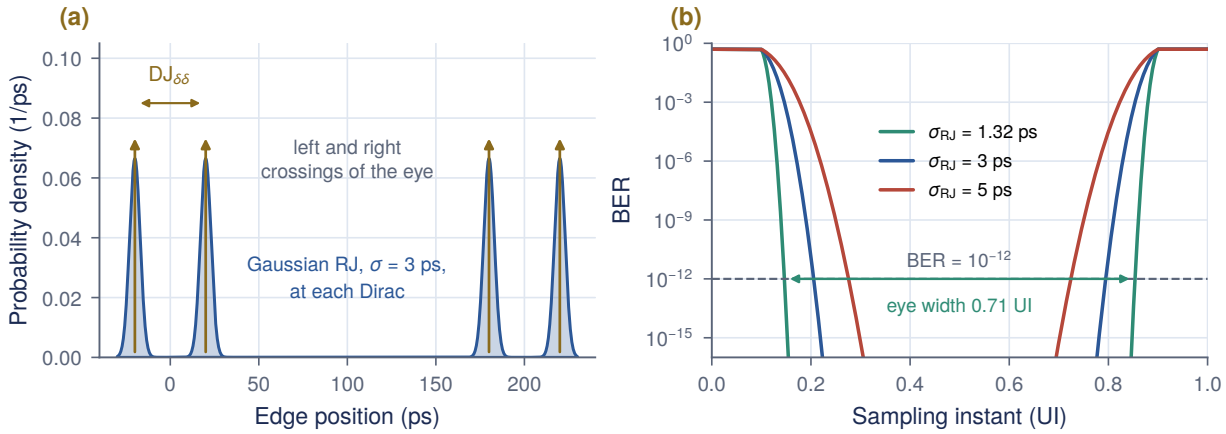


Figure 11.5. The bathtub: BER against sampling instant. (a) Dual-Dirac PDFs at both crossings of the eye, $\mathrm{DJ}_{\delta\delta}$ = 40 ps and σ_{RJ} = 3 ps. (b) Bathtubs for $\mathrm{DJ}_{\delta\delta}$ = 40 ps. At 10^{-12} the eye is 0.71 UI wide with the Open-SERDES budget ($\sigma = 1.32$ ps), 0.59 UI with 3 ps and 0.45 UI with 5 ps.

11.3 The dual-Dirac model and total jitter

At 10^{-12} only the far tails of the jitter PDF matter. The **dual-Dirac model** [21] keeps the tails and simplifies the rest: all DJ becomes two impulses at $\pm\mathrm{DJ}_{\delta\delta}/2$, each carrying a Gaussian of rms σ_{RJ} (Figure 11.5a).

From two Gaussians to total jitter

Measure the sampling instant x from the left crossing of the eye. Far out, only the Gaussian centred at $+\mathrm{DJ}_{\delta\delta}/2$ can put an edge later than x, with probability $Q\big((x - \mathrm{DJ}_{\delta\delta}/2)/\sigma_{\mathrm{RJ}}\big)$. Adding the mirror image from the right crossing gives the **bathtub curve**

$$\mathrm{BER}(x) \approx Q\left(\frac{x - \mathrm{DJ}_{\delta\delta}/2}{\sigma_{\mathrm{RJ}}}\right) + Q\left(\frac{\mathrm{UI} - x - \mathrm{DJ}_{\delta\delta}/2}{\sigma_{\mathrm{RJ}}}\right).$$

The target BER is reached where $x - \mathrm{DJ}_{\delta\delta}/2 = Q_{\mathrm{BER}}\,\sigma_{\mathrm{RJ}}$ on each side, with $Q(Q_{\mathrm{BER}}) = \mathrm{BER}$. The eye that remains is $\mathrm{UI} - \mathrm{DJ}_{\delta\delta} - 2Q_{\mathrm{BER}}\,\sigma_{\mathrm{RJ}}$; the rest is *total jitter*.

KEY EQUATION · Total jitter at a BER (dual-Dirac)

$$\mathrm{TJ}(\mathrm{BER}) = \mathrm{DJ}_{\delta\delta} + 2\,Q_{\mathrm{BER}}\,\sigma_{\mathrm{RJ}}, \qquad \text{eye width} = 1\,\mathrm{UI} - \mathrm{TJ}, \qquad 2Q_{10^{-12}} = 14.07. \tag{11.3}$$

Q_{BER} is the Gaussian quantile of Equation (2.9): 6.00 at 10^{-9}, 7.03 at 10^{-12}, 7.94 at 10^{-15}. The derivation drops two factors of $\frac{1}{2}$ (each impulse carries half the edges, and about half the bits have no transition); keeping them gives $Q = 6.84$. The industry keeps 7.03, which is 3% conservative, and so do we.

Figure 11.5(b) completes the teaser of Section 2.7. Note the factor 14.07 in Equation (11.3): a picosecond of RJ costs 14 ps of eye at 10^{-12}, while a picosecond of DJ costs one.

! $\mathrm{DJ}_{\delta\delta}$ is not $\mathrm{DJ}_{\mathrm{PP}}$

$\mathrm{DJ}_{\delta\delta}$ is a model parameter fitted to the tails. It is smaller than the true peak-to-peak DJ unless the DJ is two-valued, such as pure DCD (15.6 versus 20 ps in Figure 11.4). Use $\mathrm{DJ}_{\delta\delta}$ in Equation (11.3), and fit the tails at deep BER (Q of 4 or more): a fit at 10^{-3} overestimates

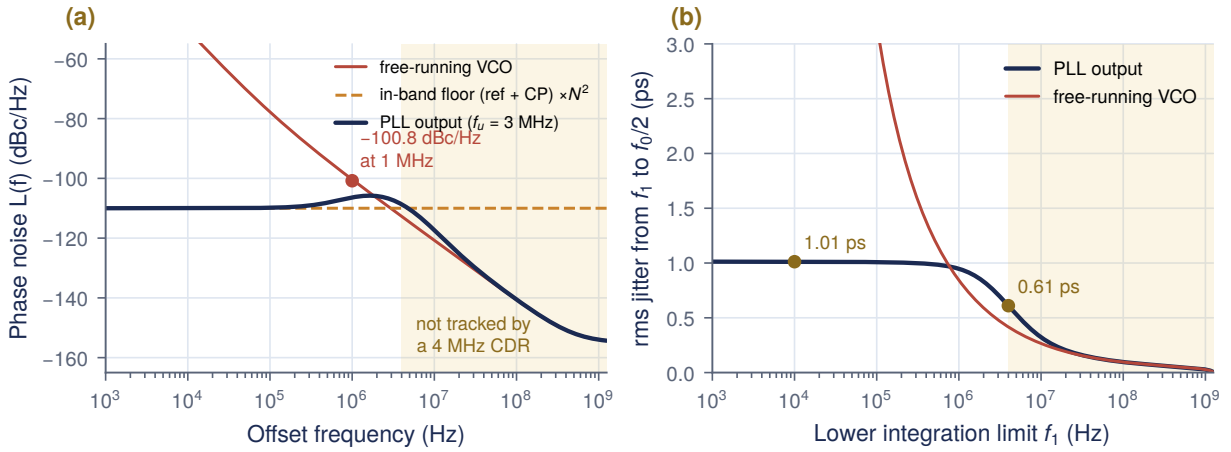


Figure 11.6. Integrating phase noise gives jitter, and the lower limit decides the answer. (a) The free-running VCO (-100.8 dBc/Hz at 1 MHz), the PLL's in-band floor, and the output of the 3 MHz PLL of Section 12.2: the output follows the floor inside the loop bandwidth and the VCO outside it. (b) rms jitter from f_1 to 1.25 GHz. The PLL output has 1.01 ps from 10 kHz, but only 0.61 ps above a 4 MHz CDR bandwidth. The free-running VCO diverges as $f_1 \to 0$, which is the random walk of Figure 11.1(b).

> σ_{RJ}. A direct TJ(10^{-12}) measurement needs about 10^{13} bits, half an hour per point at 5 Gb/s.

11.4 From phase noise to jitter

Oscillators are specified in the frequency domain. The **single-sideband phase noise** $\mathcal{L}(f)$ is the noise power in 1 Hz at offset f from the carrier, relative to the carrier, in dBc/Hz. For small noise it is half the one-sided phase spectrum $S_\phi(f)$, so integrating it gives the phase variance, and dividing by $2\pi f_0$ turns phase into time:

KEY EQUATION · Phase noise to rms jitter

$$\sigma_t = \frac{\sigma_\phi}{2\pi f_0} = \frac{1}{2\pi f_0}\sqrt{2\int_{f_1}^{f_2}\mathcal{L}(f)\,df}, \qquad \mathcal{L} \text{ in linear units } \left(10^{\mathcal{L}_{\mathrm{dB}}/10}\right). \tag{11.4}$$

The limits matter. The upper limit f_2 is about $f_0/2$. The lower limit f_1 is set by what the receiver tracks: jitter far below the CDR bandwidth moves the clock and the data together and does not close the eye. Integrating from 10 kHz gives the clock's absolute jitter; integrating from the CDR bandwidth gives the jitter that reaches the sampler (Figure 11.6).

A useful shortcut. In the $1/f^2$ region of an LC oscillator, $\mathcal{L}(f) = \mathcal{L}(\Delta f)(\Delta f/f)^2$, and the free-running TIE grows as $\kappa\sqrt{t}$ with $\kappa = (\Delta f/f_0)\sqrt{\mathcal{L}(\Delta f)}$. If a loop tracks this noise below a corner f_c (a first-order high-pass), Equation (11.4) integrates in closed form to

$$\sigma_t \approx \kappa/\sqrt{4\pi f_c}, \tag{11.5}$$

so jitter falls only as $1/\sqrt{f_c}$: quadrupling the bandwidth halves the VCO's share.

WORKED EXAMPLE 11.2 · Phase noise of the team PLL as jitter

Given: $\mathcal{L} = -100.8\,\text{dBc/Hz}$ at 1 MHz [9], applied at $f_0 = 2.5\,\text{GHz}$. **Assumptions** (they change the answer): the value describes the *free-running* VCO with a $1/f^2$ slope and a 100 kHz $1/f^3$ corner; the in-band floor from reference and charge pump is $-110\,\text{dBc/Hz}$; the PLL is the 3 MHz design of Section 12.2.

Free-running: $\mathcal{L} = 8.3 \times 10^{-11}$, so $\kappa = 4 \times 10^{-4}\sqrt{8.3 \times 10^{-11}} = 3.65 \times 10^{-9}\ \sqrt{\text{s}}$. Period jitter $\kappa\sqrt{400\,\text{ps}} = 73\,\text{fs}$; TIE after 1 µs = 3.65 ps. **Locked, shortcut:** $3.65 \times 10^{-9}/\sqrt{4\pi \times 3\,\text{MHz}} = 0.59\,\text{ps}$ from the VCO. **Locked, full model** (Figure 11.6): **1.01 ps** rms from 10 kHz, **0.61 ps** above 4 MHz.

Check the assumption: Section 12.4 shows that a $Q = 16.8$ tank should be about 20 dB quieter than $-100.8\,\text{dBc/Hz}$, so the number may be a closed-loop value. If so, every VCO term here is pessimistic. Confirm with the team before freezing the PLL bandwidth.

11.5 The Open-SERDES jitter budget

A **jitter budget** divides the UI among the blocks that move edges, so each designer gets a number to meet. Add $\text{DJ}_{\delta\delta}$ linearly, add RJ in RSS, and combine with Equation (11.3). The Open-SERDES target is **TJ ≤ 0.3 UI = 60 ps at** 10^{-12}, measured at the sampler; the remaining 0.7 UI covers the sampler aperture and setup time (Section 10.2), static phase offsets and margin. Only jitter the CDR does *not* track is counted: the clock RJ rows use the phase noise above the CDR bandwidth (0.61 ps from Worked Example 11.2, rounded up to cover the buffers), and the CDR row is the CDR's own tracking error (Table 11.1, Figure 11.7).

Table 11.1. The Open-SERDES jitter budget at 10^{-12}. DJ terms are dual-Dirac values; RJ terms are rms.

Contributor	$\text{DJ}_{\delta\delta}$ (ps)	RJ (ps rms)	Owner
TX: half-rate mux DCD, driver data-dependent jitter	8	–	Chapter 9
Channel: residual ISI after equalization (DDJ)	6	–	Chapter 8
RX clock path: I/Q error, duty cycle, PI nonlinearity	6	–	Section 12.5
CDR: PI quantization and bang-bang dither	20	–	Section 13.6
TX clock: PLL plus buffers, not tracked by the CDR	–	1.0	Chapter 12
RX clock: PLL, PI and buffers, not tracked	–	0.7	Chapter 12
Samplers: aperture and noise	–	0.5	Section 10.2
Total	**40**	**1.32** (RSS)	

WORKED EXAMPLE 11.3 · Does the budget close?

DJ: $8 + 6 + 6 + 20 = 40\,\text{ps}$. **RJ:** $\sigma = \sqrt{1.0^2 + 0.7^2 + 0.5^2} = 1.32\,\text{ps}$, so $2Q\sigma = 14.07 \times 1.32 = 18.6\,\text{ps}$. **TJ:** $40 + 18.6 = \mathbf{58.6}\,\text{ps} = \mathbf{0.293\,UI}$, inside 0.3 UI with 1.4 ps to spare; the eye at 10^{-12} is 141 ps wide.

Turn it around: with DJ at 40 ps, RJ may be at most $(60 - 40)/14.07 = 1.42\,\text{ps}$ rms; with RJ fixed, DJ may grow by only 1.4 ps. Tight budgets are normal. **Consistency:** Section 4.3 used 2 ps RJ and 20 ps DJ for the sampling clock; the CDR row here is that 20 ps, and 2 ps conservatively covers the RX share of the RJ.

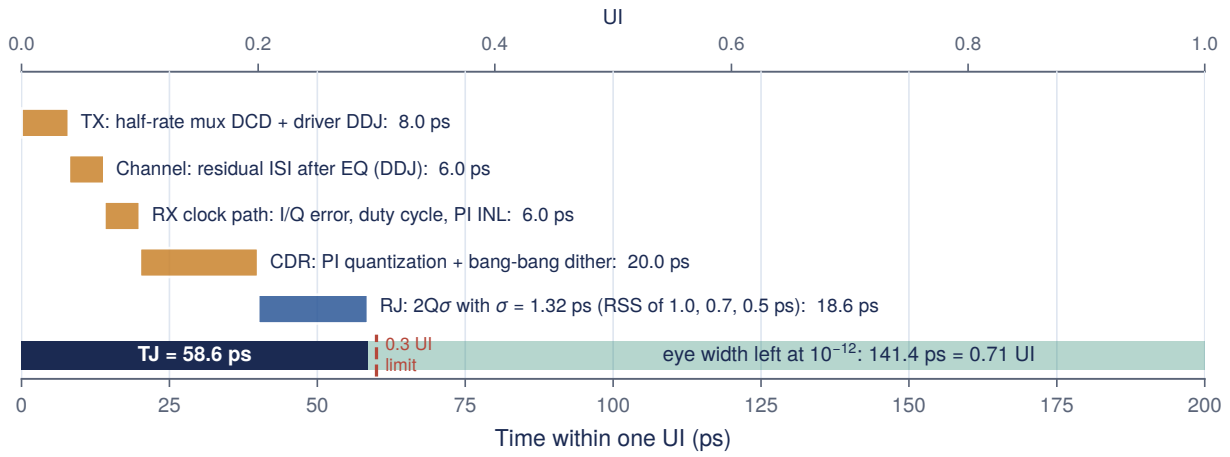


Figure 11.7. The jitter budget fills 29% of the UI. DJ terms (gold) stack linearly; the RJ terms are combined in RSS and multiplied by $2Q = 14.07$ (blue). The total, 58.6 ps, sits just inside the 0.3 UI line and leaves a 0.71 UI eye.

▶ What this means for your design

The largest line is the CDR's own dither, and the most expensive per picosecond is RJ, because of the factor 14. So: design the CDR loop for low dither (Section 13.6); keep the PLL's untracked jitter below 1 ps rms (Chapter 12); and keep duty-cycle and quadrature errors small, because they become DJ at the half-rate samplers (Section 12.5).

11.6 Jitter transfer, generation and tolerance

Three more terms describe how a block treats jitter versus jitter frequency. **Jitter transfer** is the ratio of output to input jitter: a PLL is low-pass to its reference, a CDR low-pass to the data. **Jitter generation** is the jitter a block adds with a clean input. **Jitter tolerance** (JTOL) is the largest sinusoidal jitter a receiver can take at a given frequency and still meet the BER. Because the CDR tracks slow jitter, a receiver tolerates many UI at low frequency but only its eye margin at high frequency; Section 13.5 checks the Open-SERDES CDR against a mask.

>_ Lab 6: jitter and the bathtub

Section 16.7 adds RJ and DJ to the REF link and plots the bathtub. Reproduce Figure 11.5(b), then change σ_{RJ} by 0.5 ps and watch the eye width move by 7 ps.

KEY TAKEAWAYS

- A link cares about time-interval error, and only the part the CDR does not track.
- Voltage noise becomes jitter through the slope, $\sigma_t = \sigma_v/\mathrm{SR}$: keep clock swings large and edges fast.
- RJ is Gaussian and unbounded (rms, adds in RSS); DJ is bounded (DDJ, DCD, PJ, BUJ; adds linearly). Their PDFs convolve.
- Dual-Dirac: $\mathrm{TJ} = \mathrm{DJ}_{\delta\delta} + 2Q\sigma_{\mathrm{RJ}}$ with $2Q = 14.07$ at 10^{-12}; one picosecond of RJ costs 14 ps of eye.
- $\sigma_t = \sqrt{2\int \mathcal{L}\,df}/(2\pi f_0)$, integrated from the CDR bandwidth. The team PLL model gives 1.01 ps from 10 kHz and 0.61 ps above 4 MHz.
- The Open-SERDES budget (40 ps DJ, 1.32 ps RJ) gives TJ = 58.6 ps = 0.293 UI against 0.3 UI; the CDR's 20 ps is the largest line.

CHAPTER 12

Clock Generation: The Phase-Locked Loop

A crystal gives a clean 50 MHz; the link needs a clean 2.5 GHz in four phases. The charge-pump PLL with an LC VCO bridges the two.

IN THIS CHAPTER YOU WILL LEARN

- Explain how a phase-frequency detector and charge pump turn a timing error into charge, and why the reset delay removes the dead zone.
- Derive the type-II loop's ω_n and ζ, and design the Open-SERDES loop filter for a given bandwidth and phase margin.
- Use noise transfer functions to choose the loop bandwidth that minimizes output jitter.
- Size an LC tank, state the oscillation condition, and use Leeson's model to sanity-check a phase-noise number.
- Generate and distribute quadrature (I/Q) clocks, and budget their phase and duty-cycle errors.

12.1 The charge-pump PLL

A free-running VCO drifts: its TIE random-walks (Figure 11.1b) and its frequency wanders with temperature and supply. A **phase-locked loop** (PLL) fixes both by comparing the divided VCO output with a crystal reference and steering the VCO until the two are aligned. Gardner's analysis of the charge-pump PLL [35] underlies this chapter; Razavi [36], Gardner's book [37] and ECEN 720 [34] treat it in more depth. Locked, the output frequency is exactly Nf_{ref}. Open-SERDES uses an integer-N charge-pump PLL (Figure 12.1) with $f_{\text{ref}} = 50\,\text{MHz}$ and $N = 50$, reusing the team's LC-VCO PLL design [9] retuned from 2.4 to 2.5 GHz.

Phase-frequency detector and charge pump. The **PFD** is two flip-flops with their D inputs tied high (Figure 12.2). A rising reference edge sets UP, a rising divider edge sets DN, and when both are high an AND gate resets them after a short delay T_{rst}. The width of UP minus the

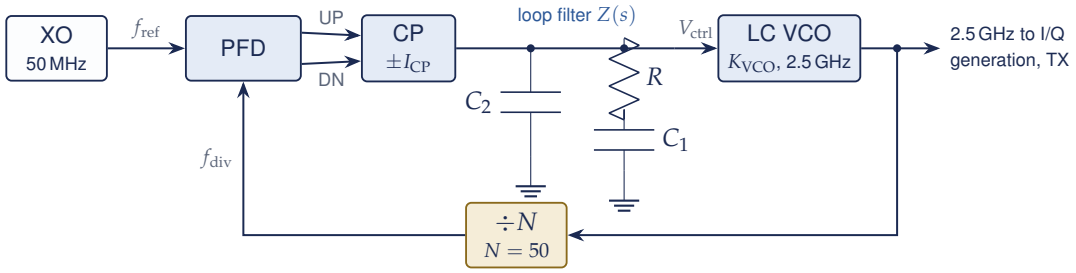


Figure 12.1. The Open-SERDES integer-N charge-pump PLL. The phase-frequency detector (PFD) compares the 50 MHz reference with the VCO divided by 50; the charge pump (CP) turns the timing difference into charge; the loop filter turns charge into the control voltage. At lock $f_{\text{out}} = Nf_{\text{ref}} = 2.5\,\text{GHz}$.

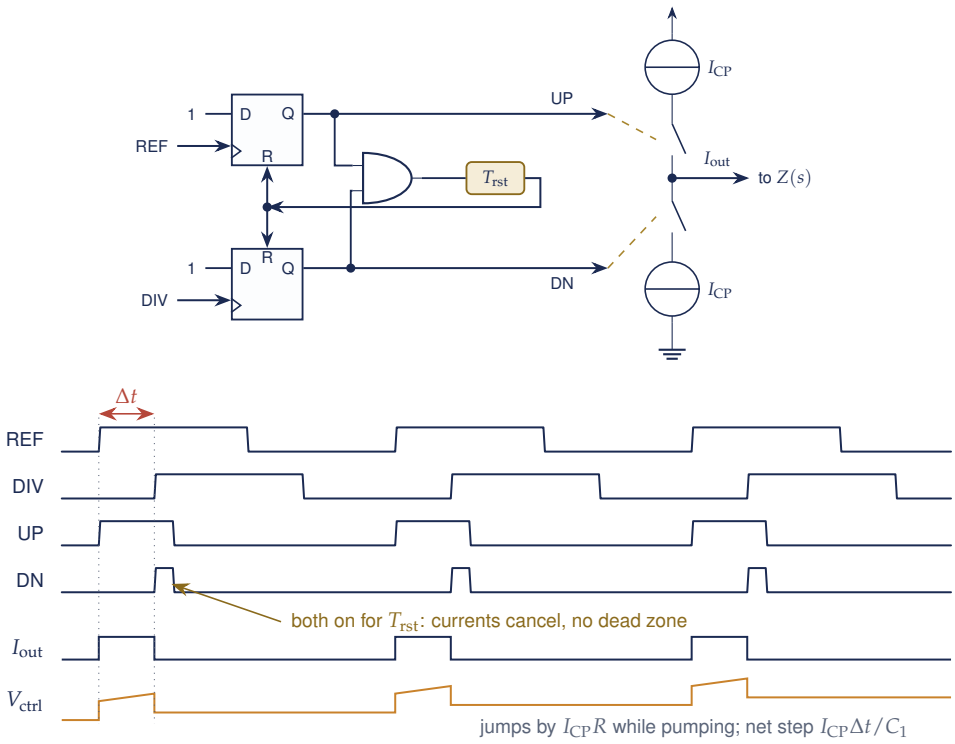


Figure 12.2. The PFD measures the edge offset as a pulse width; the charge pump turns it into charge. Top: two reset flip-flops, an AND gate with reset delay T_{rst}, and the up/down current sources. Bottom: REF leads DIV by Δt, so UP is wider than DN by Δt; both are on together for T_{rst} and cancel. Each cycle the control voltage jumps by $I_{CP}R$ while current flows and keeps a net step of $I_{CP}\Delta t/C_1$.

width of DN equals the timing difference Δt between the edges. If the frequencies differ, one output dominates for many cycles, so the PFD also pulls the frequency in: it is a phase *and* frequency detector. The **charge pump** switches a current I_{CP} into the loop filter during UP and out during DN. The average current is $I_{CP}\,\Delta t/T_{ref} = I_{CP}\,\Delta\phi/2\pi$, so the PFD-plus-CP gain is $I_{CP}/2\pi$ amperes per radian of reference phase.

Dead zone and mismatch. Without the reset delay, a tiny Δt would give UP and DN pulses too short to switch the charge pump on, so near lock the loop would get no correction: a **dead zone** in which the VCO drifts. With T_{rst} of 100–200 ps both pulses always last at least T_{rst}, and the net charge stays proportional to Δt down to zero. The price is mismatch: if the currents differ by ΔI, the loop locks at a static offset $\Delta t = (\Delta I/I_{CP})\,T_{rst}$ (7.5 ps for 5% and 150 ps), and the current pulses every 20 ns create a **reference spur**, a periodic jitter that C_2 and the loop must filter.

12.2 Loop dynamics

Near lock every block is linear in *phase*, which gives the model of Figure 12.3. The PFD and CP give $I_{CP}/2\pi$ (A/rad); the loop filter impedance $Z(s)$ turns current into V_{ctrl}; the VCO integrates frequency into phase, K_{VCO}/s (rad/s/V divided by s); and the divider scales phase by $1/N$. The same figure marks where each noise source enters.

Type-II second-order loop. Ignore C_2 for now, so $Z(s) = R + 1/(sC_1)$. The open-loop gain is

$$LG(s) = \frac{I_{CP}}{2\pi}\,Z(s)\,\frac{K_{VCO}}{s}\,\frac{1}{N} = \frac{I_{CP}K_{VCO}}{2\pi N C_1}\cdot\frac{1+sRC_1}{s^2}.$$

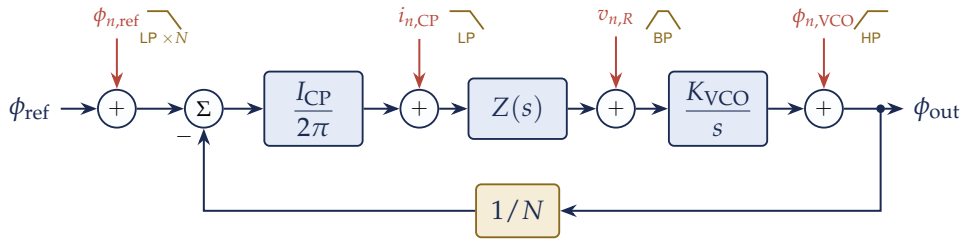


Figure 12.3. Linear phase model of the CP-PLL, with its noise sources. Each source reaches the output through a different filter: reference and charge-pump noise are low-pass filtered (and multiplied by N), loop-filter resistor noise is band-pass, and VCO noise is high-pass. The loop bandwidth sets the crossover between "trust the reference" and "trust the VCO".

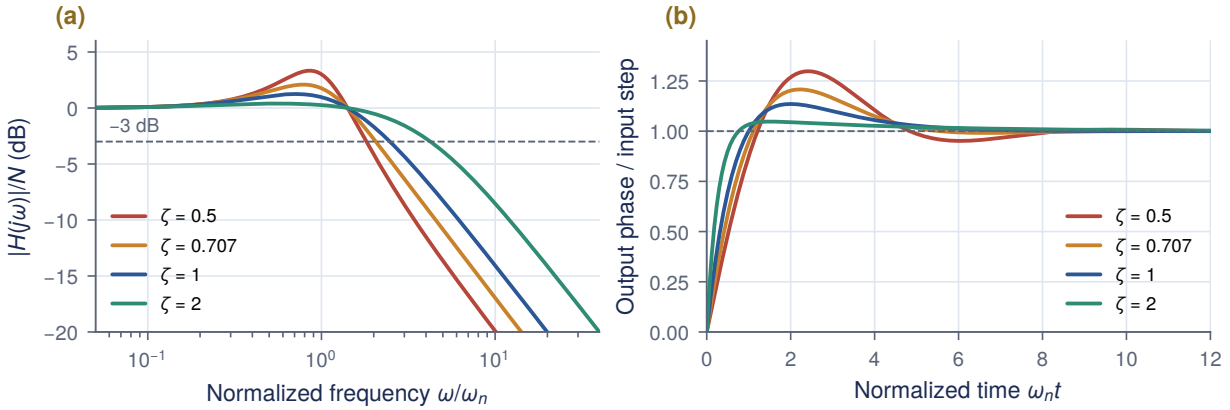


Figure 12.4. Damping trades peaking and overshoot for bandwidth. (a) Closed-loop $|H|/N$: low damping gives a sharp, peaky response; high damping a wide, flat one. (b) Output phase after a reference phase step: $\zeta = 0.5$ overshoots by 30%, $\zeta = 1$ by 13%. Transmit PLLs use $\zeta \approx 0.7$–1; CDRs, which must not amplify jitter, use $\zeta \gg 1$ (Chapter 13).

It has two integrators (the VCO and C_1), so it is a **type-II** loop: it settles to zero phase error even after a frequency step. The closed-loop transfer $\phi_{\mathrm{out}}/\phi_{\mathrm{ref}} = N\,LG/(1+LG)$ becomes, after multiplying through by s^2,

KEY EQUATION · CP-PLL natural frequency and damping

$$\frac{\phi_{\mathrm{out}}}{\phi_{\mathrm{ref}}} = N\,\frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \qquad \omega_n = \sqrt{\frac{I_{\mathrm{CP}}K_{\mathrm{VCO}}}{2\pi N C_1}}, \qquad \zeta = \frac{R}{2}\sqrt{\frac{I_{\mathrm{CP}}K_{\mathrm{VCO}}C_1}{2\pi N}}. \tag{12.1}$$

Matching coefficients gives $\omega_n^2 = I_{\mathrm{CP}}K_{\mathrm{VCO}}/(2\pi N C_1)$ and $2\zeta\omega_n = I_{\mathrm{CP}}K_{\mathrm{VCO}}R/(2\pi N)$. Raising R raises the damping without changing ω_n; raising C_1 lowers ω_n and raises ζ. The zero in the numerator causes some **jitter peaking** at every damping (Figure 12.4, Table 12.1): the loop amplifies reference jitter near ω_n. The -3 dB bandwidth is $\omega_{-3\mathrm{dB}} = \omega_n\left[1 + 2\zeta^2 + \sqrt{(1+2\zeta^2)^2+1}\right]^{1/2}$, which tends to $2\zeta\omega_n$ for large ζ.

Table 12.1. Bandwidth and peaking of the type-II loop of Equation (12.1) (computed).

ζ	0.5	0.707	1	2	4	4.5
$\omega_{-3\mathrm{dB}}/\omega_n$	1.82	2.06	2.48	4.25	8.13	9.11
Peaking (dB)	3.3	2.1	1.25	0.40	0.12	0.09

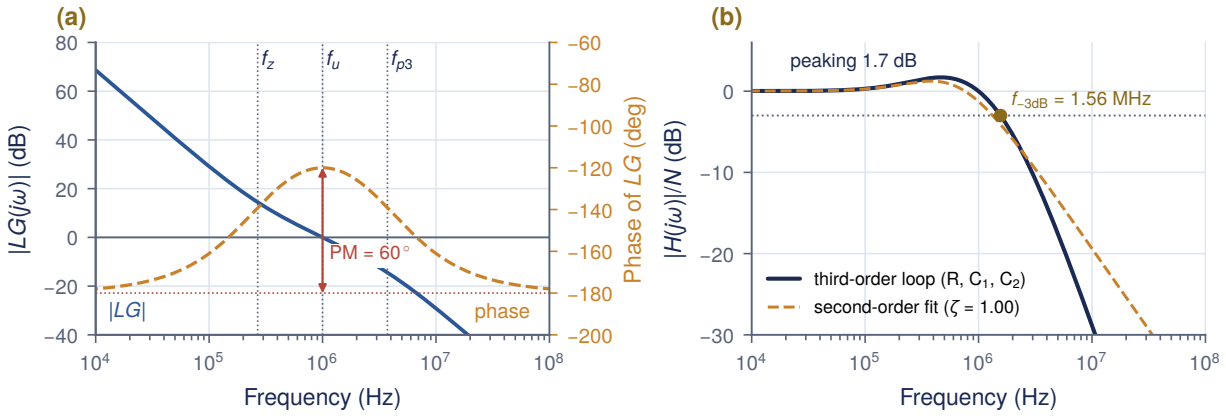


Figure 12.5. The designed loop has 60° of phase margin at 1 MHz. (a) Open-loop gain and phase of Worked Example 12.1: the phase is closest to $-180°$ furthest from it at f_u, midway between f_z and f_{p3} on a log scale. (b) The third-order closed loop (navy) against the second-order fit with $\zeta = 1$: they agree up to the bandwidth, and C_2 adds extra roll-off above it.

The third pole and phase margin. The capacitor C_2 in parallel smooths the voltage step $I_{CP}R$ seen in Figure 12.2, which would otherwise modulate the VCO every reference cycle. It adds a pole at $\omega_{p3} = 1/(RC_s)$ with $C_s = C_1C_2/(C_1 + C_2)$, and the loop becomes third order. The usual design places the unity-gain frequency ω_u at the geometric mean of the zero $\omega_z = 1/(RC_1)$ and ω_{p3}, where the phase margin Φ_m is largest. With $K_C = C_1/C_2$:

$$K_C = 2\tan^2\Phi_m + 2\tan\Phi_m\sqrt{\tan^2\Phi_m + 1}, \qquad \omega_z = \frac{\omega_u}{\sqrt{1+K_C}}, \qquad \omega_{p3} = \omega_u\sqrt{1+K_C},$$
$$C_2 = \frac{I_{CP}K_{VCO}}{2\pi N\omega_u^2}\sqrt{\frac{\omega_u^2+\omega_z^2}{\omega_u^2+\omega_{p3}^2}}, \qquad C_1 = K_CC_2, \qquad R = \frac{1}{\omega_zC_1}. \tag{12.2}$$

For $\Phi_m = 60°$, $K_C = 12.9$. Keep $f_u \le f_{ref}/10$, or the sampled nature of the PFD makes the continuous-time model wrong.

WORKED EXAMPLE 12.1 · Loop filter for the Open-SERDES PLL

Given: $f_{ref} = 50\,\text{MHz}$, $N = 50$, $K_{VCO} = 2\pi \times 150\,\text{MHz/V}$, $I_{CP} = 200\,\mu\text{A}$, $f_u = 1\,\text{MHz}$, $\Phi_m = 60°$.

Pole and zero: $K_C = 12.93$, so $f_z = 1\,\text{MHz}/\sqrt{13.93} = 268\,\text{kHz}$ and $f_{p3} = 3.73\,\text{MHz}$. **Components** from Equation (12.2): $C_2 = \mathbf{4.07}\,\text{pF}$, $C_1 = K_CC_2 = \mathbf{52.6}\,\text{pF}$, $R = 1/(\omega_zC_1) = \mathbf{11.3}\,\text{k}\Omega$. **Check** with Equation (12.1): $f_n = 537\,\text{kHz}$ and $\zeta = 1.00$. The full third-order loop, simulated in Python (Figure 12.5), crosses unity gain at 1.00 MHz with 60.0° margin and has $f_{-3dB} = 1.56\,\text{MHz}$ with 1.7 dB peaking. **Practical:** $f_{ref}/f_u = 50$, comfortably above 10. C_1 is $0.035\,\text{mm}^2$ of MIM at about $1.5\,\text{fF}/\mu\text{m}^2$ (check the PDK value), the largest item in the PLL layout. R adds $\sqrt{4kTR} = 13.7\,\text{nV}/\sqrt{\text{Hz}}$ of band-pass noise.

12.3 PLL noise and the optimum bandwidth

From Figure 12.3, each noise source sees its own transfer function:

$$\text{NTF}_{ref} = N\frac{LG}{1+LG}, \quad \text{NTF}_{CP} = \frac{2\pi}{I_{CP}}N\frac{LG}{1+LG}, \quad \text{NTF}_{VCO} = \frac{1}{1+LG}, \quad \text{NTF}_R = \frac{K_{VCO}/s}{1+LG}. \tag{12.3}$$

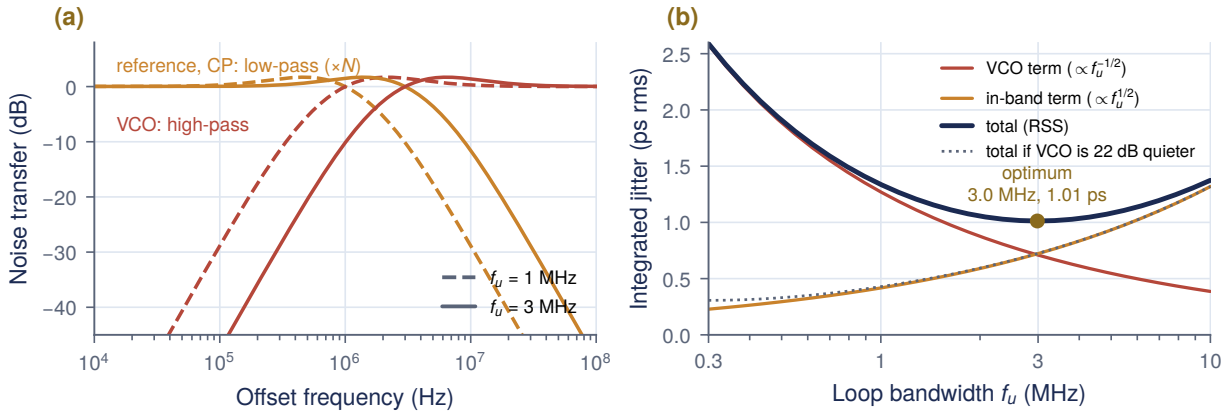


Figure 12.6. The loop bandwidth trades reference noise against VCO noise. (a) Noise transfer for f_u = 1 and 3 MHz: the reference and CP path is low-pass, the VCO path high-pass. (b) Integrated output jitter (10 kHz to 1.25 GHz) against f_u, for the VCO of Worked Example 11.2 and a −110 dBc/Hz in-band floor. The total is flat near its 3 MHz minimum of 1.01 ps. If the VCO is 22 dB quieter (Section 12.4), the optimum moves below 0.3 MHz (dotted).

Reference and charge-pump noise pass at low frequency and are multiplied by N (+34 dB for $N = 50$); VCO noise is suppressed at low frequency and passes above the bandwidth (Figure 12.6a). Widening the loop therefore lets in more reference noise and keeps out more VCO noise. The best bandwidth is roughly where the free-running VCO noise crosses the in-band floor.

WORKED EXAMPLE 12.2 · Choosing the PLL bandwidth

Given: the VCO noise of Worked Example 11.2 (−100.8 dBc/Hz at 1 MHz), and an in-band floor of −110 dBc/Hz at the output. That floor corresponds to a normalized PLL figure of merit of $-110 - 10\log_{10}(50\,\text{MHz}) - 20\log_{10}50 = -221$ dBc/Hz, typical of a good CP-PLL.

Sweep (60° loops of Equation (12.2), Figure 12.6b): f_u = 0.5, 1, 2, 3, 6 MHz gives 1.91, 1.34, 1.06, **1.01**, 1.14 ps rms. At 3 MHz the VCO and in-band terms are equal (0.71 and 0.72 ps).

Redesign at 3 MHz: capacitors scale as I_{CP}/f_u^2 and R as f_u/I_{CP}. Tripling I_{CP} to 600 µA keeps $R = 11.3\,\text{k}\Omega$ and gives $C_1 = 17.5$ pF, $C_2 = 1.36$ pF (a third of the area), $f_{-3\text{dB}} = 4.7$ MHz and $f_{\text{ref}}/f_u = 17$. The resistor noise adds only 0.11 ps in RSS. (Keeping 200 µA would need $R = 34\,\text{k}\Omega$ and a 0.45 pF C_2 that parasitics would swamp.)

▶ What this means for your design

Choose $f_u \approx 3$ MHz with $I_{CP} = 600$ µA. The minimum is shallow (within 15% from 2 to 6 MHz), so bandwidth can be traded for area or settling time. But it depends on the VCO's free-running noise, which is uncertain (Section 12.4): pin that down before freezing f_u.

12.4 The LC VCO

An LC tank stores energy and loses a little of it each cycle in the tank resistance. The **cross-coupled pair** of Figure 12.7 replaces that loss: looking into the two drains, the pair presents a negative resistance $-2/g_m$. The tank's loss is a parallel resistance $R_p = Q\omega_0 L$ (differential), so

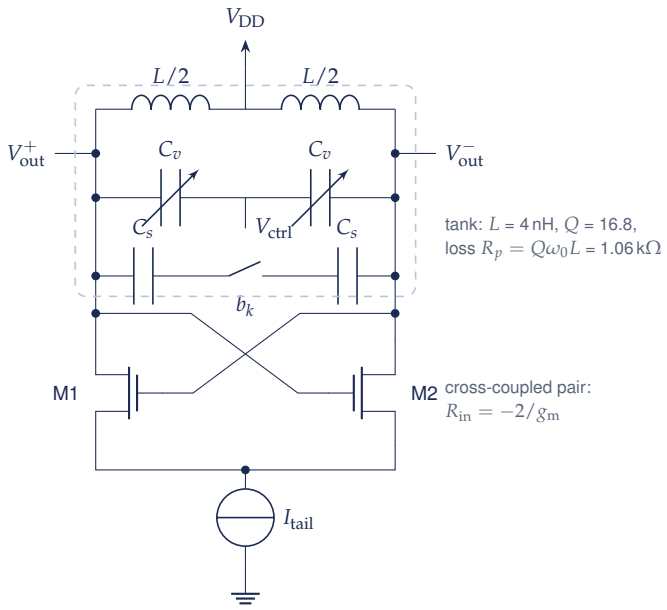


Figure 12.7. Cross-coupled LC VCO. A center-tapped inductor and the capacitors set f_0; varactors C_v give fine tuning through V_{ctrl}, and switched capacitors C_s (bits b_k) select coarse bands. The cross-coupled pair cancels the tank loss when $g_m R_p/2 \geq 1$, and the tail current sets the amplitude. Drawn with MOSFETs; npn13G2 HBTs work the same way, with capacitive cross-coupling to keep V_{CE} below BV_{CEO}.

KEY EQUATION · LC oscillator: frequency and start-up

$$f_0 = \frac{1}{2\pi\sqrt{L\,C_{tot}}}, \qquad R_p = Q\,\omega_0 L, \qquad \text{oscillation if } \frac{g_m R_p}{2} \geq 1 \ \text{(design for 2–3)}. \tag{12.4}$$

The amplitude grows until the pair limits, at about $(4/\pi)\,I_{tail}R_p/2$ (differential). C_{tot} sums the varactors (fine tuning, which sets K_{VCO}), a switched-capacitor bank (coarse bands for process corners) and all parasitics.

WORKED EXAMPLE 12.3 · The 2.5 GHz tank

Given: the team's 4 nH differential inductor with $Q = 16.8$ [9], $f_0 = 2.5$ GHz. **Capacitance:** $C_{tot} = 1/(\omega_0^2 L) = 1/\left((2\pi \cdot 2.5\times10^9)^2 \cdot 4\times10^{-9}\right) = \mathbf{1.01}$ pF, including all parasitics. **Loss:** $R_p = 16.8 \times 2\pi \cdot 2.5\times10^9 \times 4\times10^{-9} = \mathbf{1.06}$ kΩ. **Start-up:** $g_m \geq 2/R_p = 1.89$ mS; with a 3× margin, 5.7 mS per device (about 0.15 mA of collector current for an HBT, more for a MOSFET). **Tuning:** ±10% in frequency needs $C_{max}/C_{min} = (1.1/0.9)^2 = 1.49$ across varactors and bank together, with the fixed parasitics counted in both.

Phase noise: Leeson's model. Leeson's empirical model [38] captures the three regions of an oscillator's phase noise (Figure 12.8a):

$$\mathcal{L}(\Delta f) = 10\log_{10}\left[\frac{2FkT}{P_{sig}}\left(1+\left(\frac{f_0}{2Q\,\Delta f}\right)^2\right)\left(1+\frac{f_{1/f^3}}{\Delta f}\right)\right], \tag{12.5}$$

where $P_{sig} = V_{pk}^2/(2R_p)$ is the power in the tank and F an excess-noise factor of the active devices. Thermal noise near the carrier is shaped by the tank into a $1/\Delta f^2$ slope, flicker noise adds $1/\Delta f^3$ close in, and far out the noise is flat. For a fixed inductor, $\mathcal{L} \propto 1/(Q\,V_{pk}^2)$: a higher Q and a larger swing both help (Figure 12.8b).

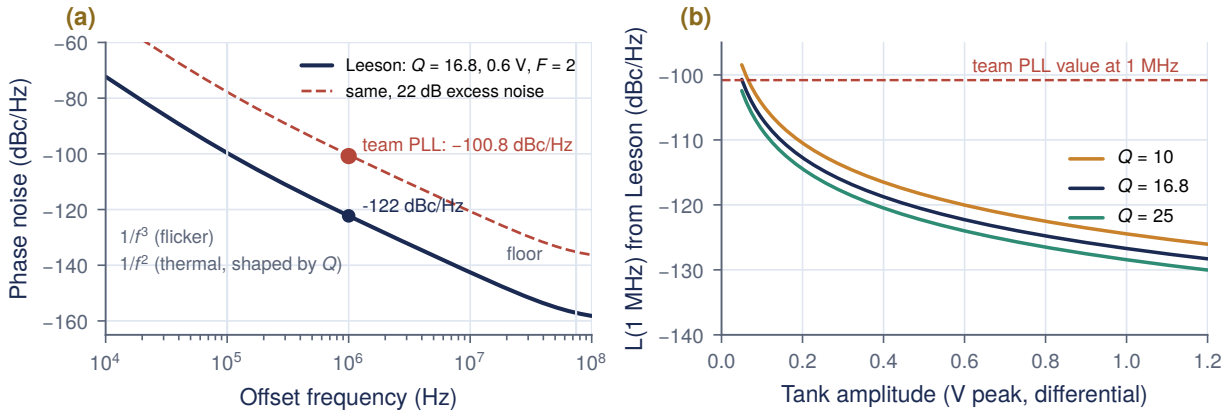


Figure 12.8. Leeson's model says the team's tank should be about 21 dB quieter than the quoted number. (a) Leeson's model for the 2.5 GHz tank ($Q = 16.8$, 0.6 V peak, $F = 2$): -122 dBc/Hz at 1 MHz. Matching -100.8 dBc/Hz would need 22 dB of excess noise (dashed). (b) $\mathcal{L}(1\,\text{MHz})$ against tank amplitude: even 50 mV of swing would beat the quoted value.

! Is -100.8 dBc/Hz the VCO, or the PLL?

With $Q = 16.8$, $R_p = 1.06\,\text{k}\Omega$, 0.6 V of swing and $F = 2$, Equation (12.5) predicts -122 dBc/Hz at 1 MHz. The common oscillator figure of merit,

$$\text{FoM} = -\mathcal{L}(\Delta f) + 20\log_{10}(f_0/\Delta f) - 10\log_{10}(P/1\,\text{mW}),$$

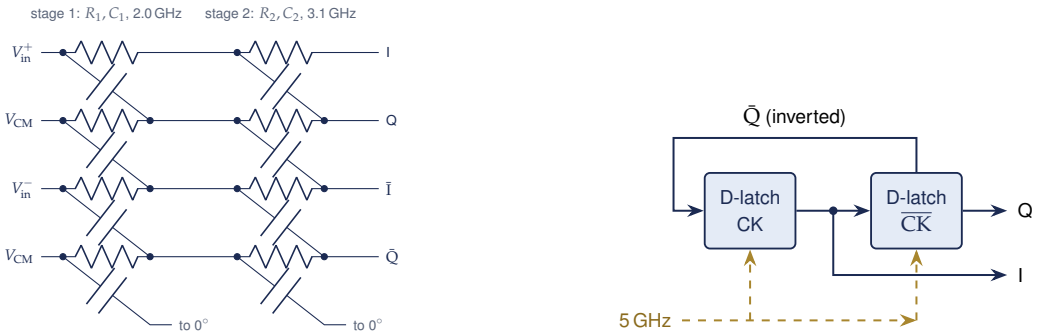
would be only 162 dB for -100.8 dBc/Hz at 5 mW of VCO power, against 180–190 dB for typical LC VCOs. The quoted value is probably the *closed-loop* PLL output near the loop bandwidth, or the VCO has excess noise (tail noise, low swing). This book uses it as free-running VCO noise, the pessimistic reading; resolving it is a Phase 0 task, because it moves the optimum of Worked Example 12.2.

▶ Retuning the team's 2.4 GHz design to 2.5 GHz

Keep the 4 nH inductor: its Q and layout are proven. The tank capacitance must fall from 1.10 to 1.01 pF, by 86 fF or 8%: remove one switched-capacitor unit or shift the band code, and re-check that V_{ctrl} sits mid-range at all corners. R_p rises by 4% with frequency, so the start-up margin improves slightly. Replace the fractional-N divider with an integer ÷50 and redesign the loop filter for $f_u \approx 3$ MHz (Worked Example 12.2). Target $K_{\text{VCO}} \approx 2\pi \times 150$ MHz/V, so that supply and control noise stay small, and cover PVT with a 3–5 bit capacitor bank.

12.5 Quadrature clocks and distribution

The half-rate receiver samples data on the 0° and 180° edges of the 2.5 GHz clock and the data transitions on the 90° and 270° edges, with four samplers D0, E0, D1, E1 (Section 10.2). It needs four phases, I, Q, $\bar{\text{I}}$ and $\bar{\text{Q}}$, spaced by exactly 100 ps. Open-SERDES derives them from the 2.5 GHz VCO with an **RC polyphase filter** (PPF); the alternative is a 5 GHz VCO followed by a ÷2 (Figure 12.9).



(a) Two-stage RC polyphase filter **(b)** Alternative: ÷2 from a 5 GHz VCO

Figure 12.9. Two ways to make quadrature. (a) Each PPF stage has an R from each input phase and a C from the previous phase (the last capacitor wraps around to the 0° output). Driven differentially, one stage gives outputs exactly 90° apart, with equal amplitude only at $f = 1/(2\pi RC)$; a second stage with a staggered pole keeps them equal across process. (b) A master-slave ÷2 gives I and Q at 2.5 GHz from both edges of a 5 GHz clock; its accuracy depends on the 5 GHz duty cycle, and it needs a VCO at twice the frequency.

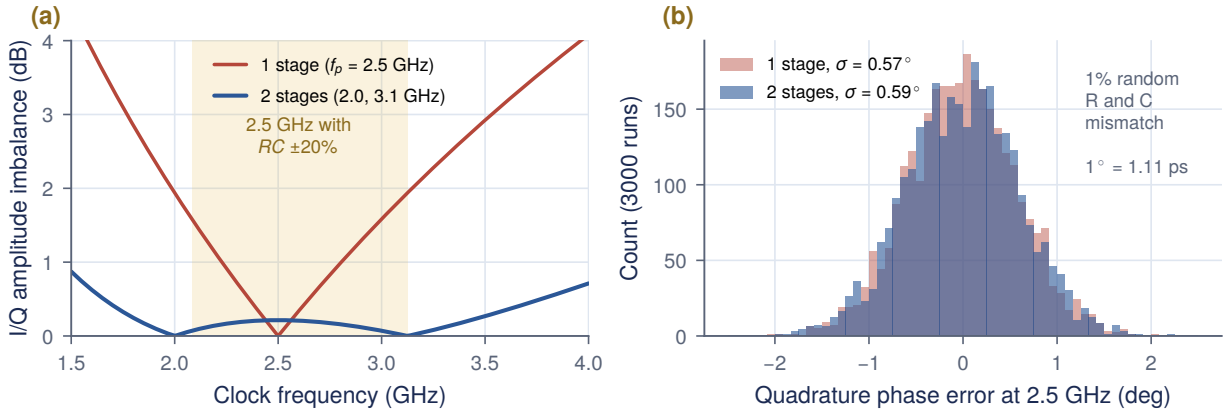


Figure 12.10. Two PPF stages fix the amplitude; matching sets the phase. (a) I/Q amplitude imbalance against frequency. One stage is balanced only at its pole; two staggered stages stay within 0.07 dB over the whole ±20% RC range (shaded) at 2.5 GHz. (b) Quadrature phase error over 3000 Monte Carlo runs with 1% R and C mismatch: about 0.6° rms (0.66 ps) either way.

How accurate is the polyphase filter?. For a single stage driven differentially, the outputs are $V/(1+j\omega RC)$ and $j\omega RC\,V/(1+j\omega RC)$: always 90° apart, with amplitude ratio ωRC. A ±20% process shift of RC therefore leaves 1.6–1.9 dB of amplitude imbalance, while two stages with poles at 2.0 and 3.1 GHz keep it below 0.07 dB (Figure 12.10a). Phase errors come from random mismatch instead: with 1% mismatch on every R and C, Monte Carlo gives 0.57° rms for one stage and 0.59° for two (Figure 12.10b). The last stage sets the phase error, so it needs large, well-matched resistors and MIM capacitors in a common-centroid layout. The PPF is passive and lossy, −5.2 dB for two unloaded stages, so CML buffers must restore the swing.

What the errors do to the half-rate samplers. At 2.5 GHz one degree is 1.11 ps. The CDR locks the edge samplers (Q, $\bar{Q}$) onto the data transitions, so an I/Q error δ moves the data samplers $\delta \times 1.11$ ps away from the eye centre: a static offset that eats timing margin. A duty-cycle error is worse, because the D1 and E1 samplers fire on the falling edges: a 1% duty error shifts them by 4 ps relative to D0 and E0, a two-valued DCD seen by every other bit.

Distribution. CML limiting buffers after the PPF restore the swing, remove the residual imbalance and keep the slew rate high (Equation (11.2)). Give them 2–3$\times f_{\mathrm{clk}}$ of bandwidth (shunt peaking or tuned loads): a buffer near f_{clk} amplifies jitter and duty-cycle error. A duty-cycle correction (DCC) loop that servoes the clock's average to zero holds the duty cycle within about $\pm 0.5\%$.

WORKED EXAMPLE 12.4 · The RX clock-path line of the jitter budget

I/Q error: two-stage PPF, 1% mismatch: $3\sigma = 3 \times 0.59^\circ = 1.8^\circ \rightarrow$ **2.0 ps**. **Duty cycle:** $\pm 0.5\%$ after DCC: $0.5\% \times 400\,\mathrm{ps} \rightarrow$ **2.0 ps**. **PI nonlinearity:** ± 0.6 ps with corrected weights (Section 13.6) $\rightarrow$ **1.2 ps**. **Total** 5.2 ps, inside the 6 ps allocated in Table 11.1, with 0.8 ps of margin. Without DCC, a 52% duty cycle alone would cost 8 ps.

Open-SERDES clock generation targets

Integer-N CP-PLL, $f_{\mathrm{ref}} = 50\,\mathrm{MHz}$, $N = 50$, $K_{\mathrm{VCO}} \approx 2\pi \times 150\,\mathrm{MHz/V}$, $I_{\mathrm{CP}} = 600\,\mu\mathrm{A}$, $f_u \approx 3\,\mathrm{MHz}$ ($f_{-3\mathrm{dB}} \approx 4.7\,\mathrm{MHz}$), 60° phase margin. Jitter ≤ 1.0 ps rms from 10 kHz and ≤ 0.6 ps above 4 MHz. Quadrature from a two-stage PPF and CML buffers: I/Q error $\leq 2^\circ$ (3σ), duty cycle $50 \pm 0.5\%$. Power estimate: PLL about 13 mW (the team's design draws 12.73 mW); PPF buffers and distribution 10–15 mW (four CML stages of about 1.5 mA at 2.5 V).

KEY TAKEAWAYS

- The PFD turns the edge offset into an UP/DN width difference; the charge pump gives $I_{\mathrm{CP}}/2\pi$ A/rad. The reset delay removes the dead zone; current mismatch causes a static offset and reference spurs.
- Type-II loop: $\omega_n = \sqrt{I_{\mathrm{CP}} K_{\mathrm{VCO}}/(2\pi N C_1)}$, $\zeta = (R/2)\sqrt{I_{\mathrm{CP}} K_{\mathrm{VCO}} C_1/(2\pi N)}$; peaking is 2.1 dB at $\zeta = 0.707$ and 0.4 dB at $\zeta = 2$.
- For 60° margin at 1 MHz: C_2 = 4.07 pF, C_1 = 52.6 pF, R = 11.3 kΩ ($f_{-3\mathrm{dB}}$ = 1.56 MHz); at the 3 MHz optimum, I_{CP} = 600 µA, C_1 = 17.5 pF.
- Reference noise is low-pass and VCO noise high-pass; the bandwidth that equalizes them, about 3 MHz here, minimizes jitter at 1.0 ps rms.
- The 4 nH, $Q = 16.8$ tank needs 1.01 pF at 2.5 GHz, has R_p = 1.06 kΩ, and oscillates if $g_{\mathrm{m}} R_p/2 \geq 1$. Leeson suggests the quoted -100.8 dBc/Hz is closed-loop: check it.
- A two-stage PPF plus CML buffers gives I/Q within about 2°; every degree is 1.11 ps and every 1% of duty cycle is 4 ps at the half-rate samplers.

CHAPTER 13

Clock and Data Recovery

The receiver has no clock wire: it must find the centre of every bit from the data itself, and keep finding it while the two crystals drift apart.

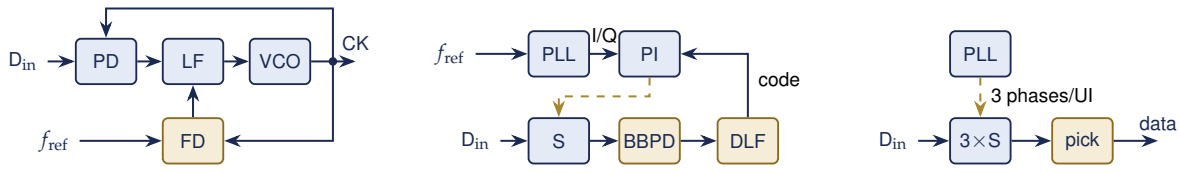
IN THIS CHAPTER YOU WILL LEARN

- Explain why a receiver needs a CDR and compare PLL-based, PI-based and oversampling architectures.
- Analyze the Hogge (linear) and Alexander (bang-bang) phase detectors, including the half-rate version used by Open-SERDES.
- Model a linear CDR and a bang-bang CDR: gain K_{BB}, slewing bandwidth and dither.
- Read a jitter-tolerance mask and size the loop to pass it.
- Design the Open-SERDES phase-interpolator CDR: PI linearity, step size, frequency-offset range and a behavioural simulation.

13.1 Why the receiver needs a CDR

The TX and RX of a serial link run from different crystals, each accurate to about ± 100 ppm. Their bit rates can differ by 200 ppm, so the data phase drifts past a fixed RX clock by 2×10^{-4} UI every UI: one whole UI every 5000 bits, or every microsecond at 5 Gb/s. A fixed-phase sampler would walk out of the eye within half a microsecond, so the receiver must extract phase and frequency from the data transitions: **clock and data recovery** (CDR). Line coding or scrambling guarantees the transitions, and the CDR tracks slow jitter as a bonus (Section 11.6).

Figure 13.1 shows the three common architectures. A **PLL-based CDR** has a VCO in the loop: lowest dither, but a VCO per lane and a frequency detector to pull in. A **PI-based dual-loop CDR** shares one clean PLL and rotates its phase per lane with a digitally controlled phase interpolator: synthesizable, free of false lock and robust over PVT, at the cost of quantization and latency. An **oversampling** CDR samples three times per UI and picks the best sample, with no loop but three times the samplers. Open-SERDES uses the PI dual loop because it reuses the team's PLL (Chapter 12).



(a) PLL-based **(b)** PI-based dual loop (Open-SERDES) **(c)** Oversampling

Figure 13.1. Three ways to recover the clock. (a) A VCO inside the loop, with a frequency detector (FD) to pull in. (b) A global PLL plus a per-lane phase interpolator steered by a bang-bang phase detector and a digital loop filter (DLF). (c) Blind oversampling: three samples per UI and a decision on which to keep.

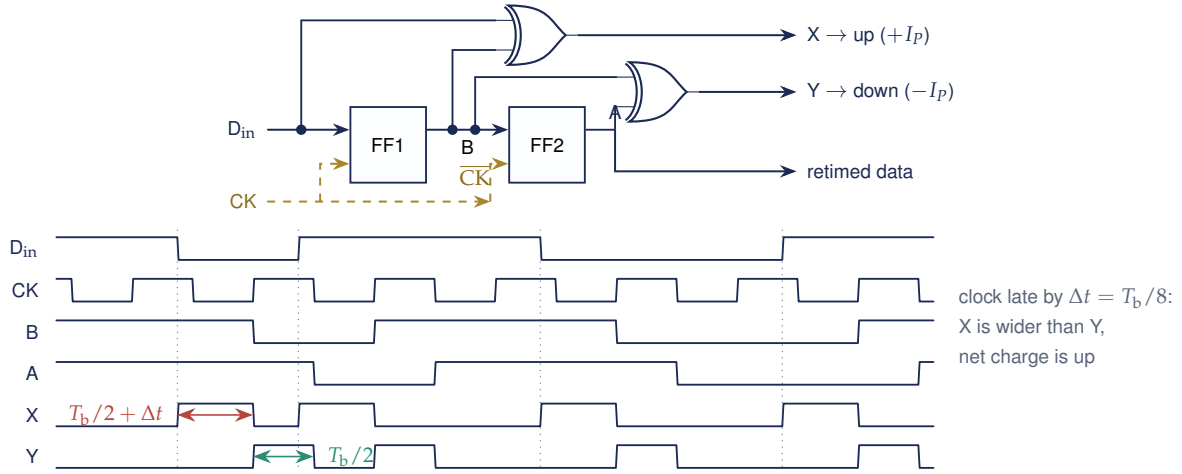


Figure 13.2. The Hogge detector measures phase as a pulse-width difference. FF1 retimes the data (B), FF2 delays it by half a period (A). X spans from each data transition to the next clock edge, $T_b/2 + \Delta t$; Y is always $T_b/2$. With no transition, both stay low: the detector tri-states.

After Razavi [40].

13.2 Phase detectors

A PLL's PFD does not work on data: random data skips edges, and a missing edge looks like a frequency error. A CDR phase detector must compare edges only where the data has them, and output *nothing* during runs of identical bits (**tri-state**). Two classic detectors do this.

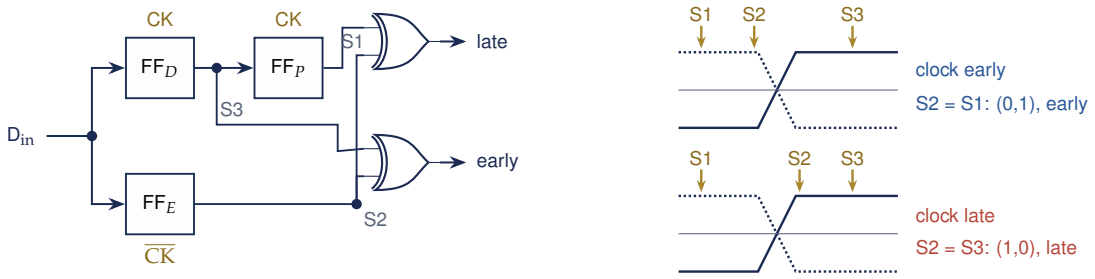
The Hogge linear detector. Hogge's detector [39] uses two flip-flops and two XORs (Figure 13.2). FF1 retimes the data (B), FF2 delays B by half a cycle (A). $X = D_{in} \oplus B$ lasts from each data transition to the next rising clock edge, $T_b/2 + \Delta t$ when the clock is Δt late; $Y = B \oplus A$ always lasts $T_b/2$. X pumps up and Y pumps down, so the net charge per transition is proportional to Δt: a linear detector of gain $TD \cdot I_P/2\pi$, where the **transition density** TD is about 0.5 for random data. It tri-states and retimes the data. Its weakness is speed: at 5 Gb/s X and Y are 100 ps pulses from gates with 20–30 ps edges, and the clock-to-Q delay leaves a static error unless matched; at 10 Gb/s the pulses shrink to 50 ps, close to the practical limit.

The Alexander bang-bang detector. The Alexander detector [41] (Figure 13.3) takes two samples per bit: a data sample at the bit centre and an **edge sample** half a bit earlier, at the expected transition. Three consecutive samples, S1 (previous data), S2 (edge) and S3 (current data), tell everything. If S1 = S3 there was no transition and the detector says nothing. Otherwise S2 equals whichever data bit it was closer to: if S2 = S1 the edge sample was taken *before* the transition and the clock is early; if S2 = S3 it was taken after, and the clock is late. With $X = S1 \oplus S2$ and $Y = S2 \oplus S3$:

(X, Y)	(0,0)	(0,1)	(1,0)
Meaning	no transition	clock early: delay it	clock late: advance it

The output is only the *sign* of the error, hence "bang-bang" (BB). It reuses the data samplers, retimes, tri-states and runs at the samplers' full speed: the most common CDR detector.

Half-rate in Open-SERDES. At half rate the four samplers D0, E0, D1, E1 fire on I, Q, $\bar{I}$, $\bar{Q}$ of the 2.5 GHz clock (Sections 10.2 and 12.5). Each clock cycle yields two early/late decisions, from



(a) Three flip-flops and two XORs **(b)** Early and late, for a 0→1 (solid) or 1→0 transition

Figure 13.3. The Alexander detector compares the edge sample with its neighbours. (a) FF$_D$ samples the data (S3), FF$_E$ samples the edge half a period earlier (S2), FF$_P$ holds the previous bit (S1); in practice a fourth flip-flop retimes S2 to CK. (b) If the edge sample lands before the transition it equals S1 (clock early); after it, S3 (clock late).

(D0, E0, D1) and (D1, E1, D0′), so every transition is used. The cost is that I/Q and duty-cycle errors now appear as phase offsets between the two halves (Worked Example 12.4).

Mueller-Müller. This **baud-rate** detector [42] uses one data sample and one error sample per UI and no edge samplers. Its output $z_k = e_k d_{k-1} - e_{k-1} d_k$ averages to $h_1 - h_{-1}$, so it locks where the first pre- and post-cursors are equal, a channel-dependent point. It suits ADC-based receivers; Open-SERDES has edge samplers and uses Alexander.

13.3 The linear CDR model and jitter transfer

With a linear detector, a CDR is a PLL with $N = 1$ and a detector gain $TD \cdot I_P/2\pi$. With proportional resistor R_p and integrating capacitor C_p, Equation (12.1) gives

KEY EQUATION · Linear CDR

$$H(s) = \frac{\phi_{\text{out}}}{\phi_{\text{in}}} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \qquad \omega_n = \sqrt{\frac{K_{\text{VCO}} I_P}{2\pi C_p}}, \quad \zeta = \frac{R_p}{2}\sqrt{\frac{K_{\text{VCO}} I_P C_p}{2\pi}}, \quad \omega_{-3\text{dB}} \approx 2\zeta\omega_n = \frac{K_{\text{VCO}} I_P R_p}{2\pi}. \tag{13.1}$$

with TD absorbed into I_P. $|H|$ is the **jitter transfer**: the jitter the recovered clock follows. $|1 - H|$ is what the clock does *not* follow, and therefore what reaches the sampler as eye closure (Figure 13.4a). Wireline CDRs use $\zeta \gg 1$, typically 4–5, because any peaking would amplify jitter in a chain of repeaters; with $\zeta = 4.5$ the peaking is 0.09 dB (Table 12.1).

13.4 Bang-bang CDR dynamics

A BB detector has infinite gain at zero error, but jitter rescues the linear model: with Gaussian edge jitter of rms σ_j, the *average* output is the sign function smoothed by the jitter PDF:

$$\overline{\text{PD}}(\Delta\phi) = TD \cdot E\big[\,\text{sgn}(\Delta\phi + j)\big] = TD \cdot \text{erf}\Big(\frac{\Delta\phi}{\sqrt{2}\,\sigma_j}\Big),$$

whose slope at zero, $TD \cdot 2/(\sqrt{2\pi}\sigma_j)$, is the linearized gain (Figure 13.5a):

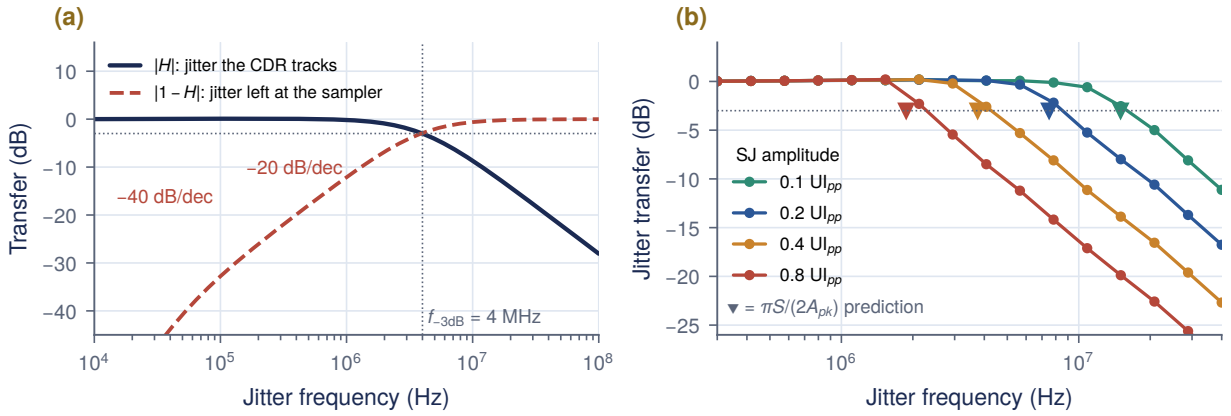


Figure 13.4. A CDR is a low-pass filter for jitter, and a bang-bang CDR's corner moves with the jitter amplitude. (a) Linear CDR with $f_{-3\mathrm{dB}}$ = 4 MHz and ζ = 4.5: the clock tracks slow jitter ($|H|$), and the error $|1-H|$ rises at 40 then 20 dB/decade. (b) Simulated jitter transfer of the analog bang-bang CDR of Worked Example 13.1 (R = 200 Ω, $S = 3\times10^6$ UI/s) for four sinusoidal-jitter amplitudes. The corner halves each time the amplitude doubles; the triangles mark the slewing prediction $\pi S/(2A_{\mathrm{pk}})$, within 7–22% of simulation.

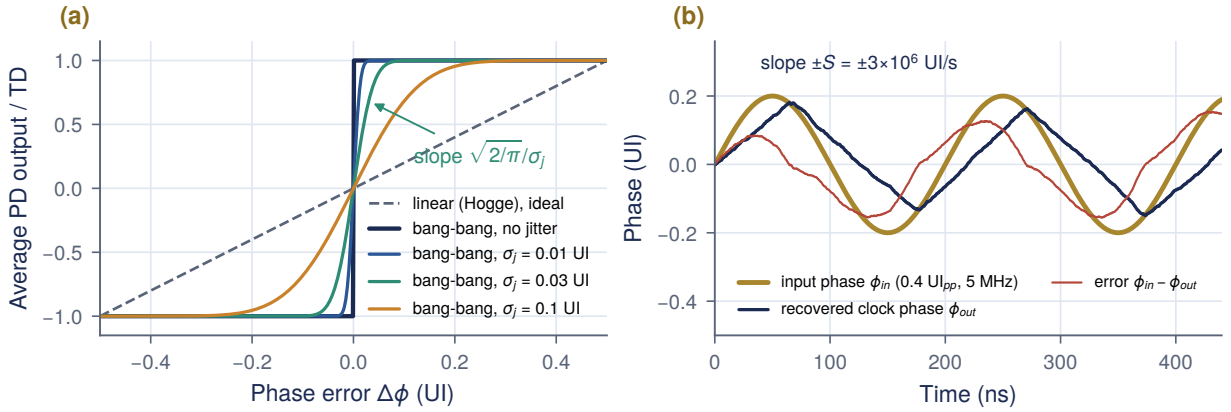


Figure 13.5. Jitter linearizes the bang-bang detector; large jitter makes the loop slew. (a) Average detector output against phase error: the ideal sign function smoothed by Gaussian jitter, with slope $\sqrt{2/\pi}/\sigma_j$ at zero, against an ideal linear detector. (b) Analog BB-CDR (proportional path only) with 0.4 $\mathrm{UI_{pp}}$ of 5 MHz input jitter. The input slope, $\omega A_{\mathrm{pk}} = 6.3\times10^6$ UI/s, exceeds the loop's slew rate $S = 3\times10^6$ UI/s, so the recovered phase becomes a triangle that lags the input.

KEY EQUATION · Bang-bang detector gain, linearized by jitter

$$K_{\mathrm{BB}} = TD\sqrt{\frac{2}{\pi}}\frac{1}{\sigma_j} \quad \text{(per UI of phase error).} \tag{13.2}$$

The loop gain, and so the bandwidth, is inversely proportional to the input jitter: a clean input makes a fast loop, a noisy input a slow one. Always quote a BB-CDR's bandwidth together with the jitter it was measured with.

Slewing. For large jitter the detector saturates and the recovered phase moves at its maximum rate, the **slew rate** S. In a VCO-based loop each decision shifts the VCO frequency by $\Delta f_P = K_{\mathrm{VCO}} I_P R_p$ for one bit, moving the phase by $\Delta\phi_P = \Delta f_P\, T_{\mathrm{b}}$ clock cycles, and decisions arrive at rate TD/T_{b}; so $S = TD\,\Delta\phi_P/T_{\mathrm{b}}$. A sinusoid $A_{\mathrm{pk}}\sin\omega t$ is tracked as long as its peak slope ωA_{pk} is below S. Beyond that the output is a triangle of peak $S\,T/4$ (Figure 13.5b), so its gain is $\pi S/(2A_{\mathrm{pk}}\omega)$ and

KEY EQUATION · Slew-limited BB-CDR bandwidth

$$S = TD\,\frac{\Delta\phi_P}{T_\mathrm{b}}, \qquad \omega_{-3\mathrm{dB}} = \frac{\pi S}{2A_\mathrm{pk}}, \qquad \mathrm{dither}_\mathrm{pp} \approx 2(D+1)\,\Delta\phi_P. \tag{13.3}$$

The last term is the **dither**: with no jitter at all, a BB loop hunts around the lock point, and a loop latency of D extra decision periods lets it overshoot by D more steps each way. Large S helps jitter tolerance but costs dither; the integral path must be weak compared with the proportional path, $K_P/K_I \geq 16$ (for an analog loop, $R_pC_p/T_\mathrm{b} \gg 1$), or the loop rings.

Table 13.1. The slewing model against a measured 90 nm, 10 Gb/s BB-CDR: R_p = 10 Ω, C_p = 100 nF, K_VCO = 1 GHz/V, I_P = 300 µA, so Δf_P = 3 MHz and, at full rate with TD = 0.5, $S = 1.5 \times 10^6$ UI/s.

Jitter ($\mathrm{UI_{pp}}$)	0.125	0.25	0.5	R_p (Ω) / C_p (nF)	5 / 100	10 / 100	20 / 100
Predicted $f_{-3\mathrm{dB}}$ (MHz)	6.0	3.0	1.5	Measured (MHz)	1.0	1.5	2.7
Measured (MHz)	4.1	2.8	1.45	C_p = 10 / 32 / 100 nF		2.8 / 2.55 / 2.6	

Table 13.1 checks Equation (13.3) against measurements. The prediction matches at 0.25 and 0.5 $\mathrm{UI_{pp}}$; at 0.125 UI the loop is no longer slewing (the input slope is below S) and the measured corner saturates. The bandwidth scales with R_p, the proportional path, and hardly depends on C_p: a BB-CDR is designed through its proportional step.

WORKED EXAMPLE 13.1 · An analog BB-CDR that fails, and the fix

Given: a half-rate VCO-based BB-CDR at 5 Gb/s, K_VCO = 150 MHz/V, I_P = 100 µA, R_p = 100 Ω, C_p = 1 nF, TD = 0.5, and a stressed input that leaves a timing margin TM = 0.4 UI.

Step: Δf_P = 1.5 MHz; one cycle of the 2.5 GHz clock is 2 UI, so $\Delta\phi_P = 1.5 \times 10^6 \times 200\,\mathrm{ps} \times 2 = 6 \times 10^{-4}$ UI per bit and $S = 0.5 \times 6 \times 10^{-4}/200\,\mathrm{ps} = 1.5 \times 10^6$ UI/s. **JTOL** (Section 13.5): $\mathrm{TM} + 2S/\omega = 0.4 + 0.48 = 0.88\,\mathrm{UI_{pp}}$ at 1 MHz, below the 1 UI mask point: **fails**. Simulation gives 0.95 UI at 1 MHz and 0.35 UI at 2 MHz (mask 0.5). **Fix:** R_p = 200 Ω doubles S to 3×10^6 UI/s: $0.4 + 0.95 = 1.35$ UI at 1 MHz (simulated 1.36) and 0.61 UI at 2 MHz: **passes**. **Costs:** dither with a 3-bit latency is $2 \times 4 \times 1.2 \times 10^{-3} = 0.01$ UI, negligible. $R_pC_p/T_\mathrm{b} = 1000$, so the loop is well damped. The proportional path alone pulls in only $|\Delta f/f| < TD\,\Delta\phi_P \approx 600$ ppm, so a frequency detector or reference loop must bring the VCO that close first.

13.5 Jitter tolerance

Jitter tolerance (JTOL) is the largest sinusoidal jitter (SJ), versus frequency, that the receiver accepts while meeting the BER, measured with a stressed input that already carries the specified RJ and DJ. Let TM be the timing margin left by that stress. A linear CDR leaves the error $|1 - H|$ times the input, so

$$\mathrm{JTOL}(\omega) = \frac{\mathrm{TM}}{|1 - H(j\omega)|}, \qquad \mathrm{JTOL_{BB}}(\omega) \approx \mathrm{TM} + \frac{2S}{\omega}. \tag{13.4}$$

Far below the bandwidth the CDR tracks almost everything and JTOL is many UI; far above, JTOL falls to TM. A BB-CDR in its slew-limited region follows $2S/\omega$, a −20 dB/decade line. Standards define a **JTOL mask** that the curve must stay above. We use a USB 3 Gen 1-style mask: 2 $\mathrm{UI_{pp}}$ at 0.5 MHz falling at −20 dB/decade to 0.2 UI at 4.9 MHz, flat to 50 MHz (check

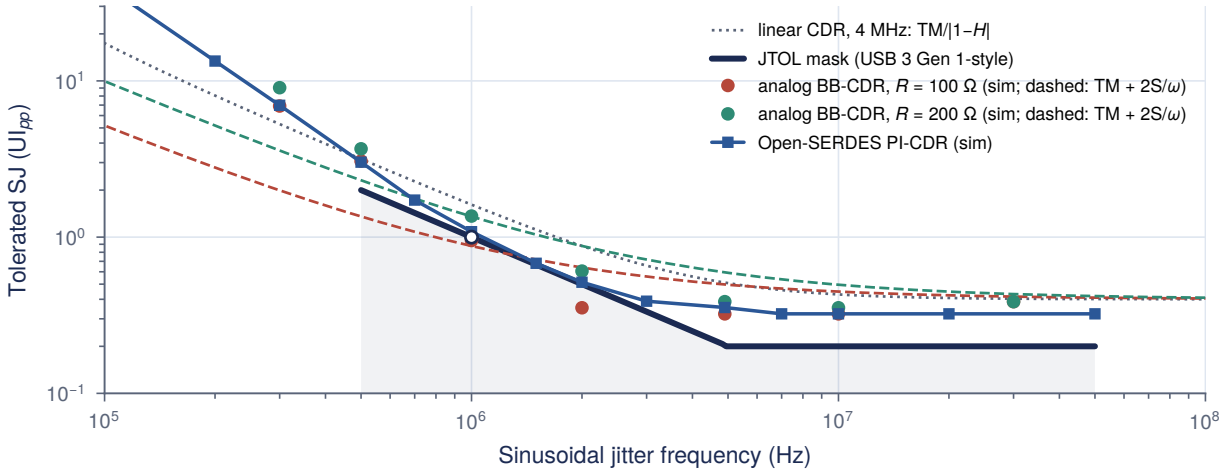


Figure 13.6. Jitter tolerance against the mask. Navy: the mask. Dotted: a linear CDR with 4 MHz bandwidth. Red and teal: the analog BB-CDR of Worked Example 13.1; dots are simulations, dashed lines Equation (13.4). With R_p = 100 Ω the curve dips under the mask at 1–2 MHz; 200 Ω passes. Blue: the simulated Open-SERDES PI-CDR, which passes with a thin margin at 1–2 MHz. At high frequency every BB curve flattens near 0.3 UI, TM minus the loop's own dither.

the exact points against the standard you target). Passing the −20 dB/decade part needs $S \gtrsim \pi f(A_{\text{mask}} - \text{TM})$, about 2.5×10^6 UI/s at 0.5 MHz.

13.6 The Open-SERDES PI-based CDR

Figure 13.7 shows the Open-SERDES CDR. The PLL supplies I and Q at 2.5 GHz. Two **phase interpolators** (PIs) rotate them: the data PI (code k) clocks D0 and D1, the edge PI (code $k + 16$, a quarter period later) clocks E0 and E1. A 1:8 demultiplexer feeds the digital loop at 625 MHz (T_{upd} = 1.6 ns, 8 UI), where up to eight Alexander early/late decisions per update are reduced by a majority vote to $v \in \{-1, 0, +1\}$. The loop filter is proportional plus integral:

$$f \leftarrow f + K_I\, v, \qquad \text{acc} \leftarrow \text{acc} + K_P\, v + f, \qquad k = \text{round}(\text{acc}) \bmod 64. \tag{13.5}$$

The **frequency register** f learns the TX-RX frequency offset, and the **phase accumulator** has fractional bits below the PI's 6.25 ps LSB, so the proportional step K_P can be a fraction of a PI step. The baseline uses $K_P = 1/8$ LSB (0.78 ps), $K_P/K_I = 128$ and a latency of two updates ($D = 1$).

The phase interpolator. A PI forms $(1-\alpha)\,\text{I} + \alpha\,\text{Q}$ with two current-steered weights. For sinusoidal I and Q the output phase is $\theta = \arctan[\alpha/(1-\alpha)]$, not $90°\alpha$: linear weights trace the chord between the two phasors, not the arc (Figure 13.8a). With 16 codes per quadrant, the integral nonlinearity (INL) reaches ±4.1° (±4.5 ps) at $\alpha \approx 0.25$ and 0.75, and the amplitude dips to 0.71 at $\alpha = 0.5$. Weights taken from a sine/cosine table and quantized to 6 bits cut the INL to ±0.55° (0.6 ps) (Figure 13.8b); a limiting buffer after the PI removes the amplitude dip. The law assumes near-sinusoidal inputs, with edges about as slow as the 100 ps phase spacing, which the PPF and CML buffers of Section 12.5 provide; square-wave inputs make the PI jump between phases.

Frequency-offset range. A ±300 ppm offset (two ±100 ppm crystals plus margin) drifts the data by $300 \times 10^{-6} \times 1.6\,\text{ns} = 0.48$ ps per update, 0.077 PI step; the integral register holds this rate

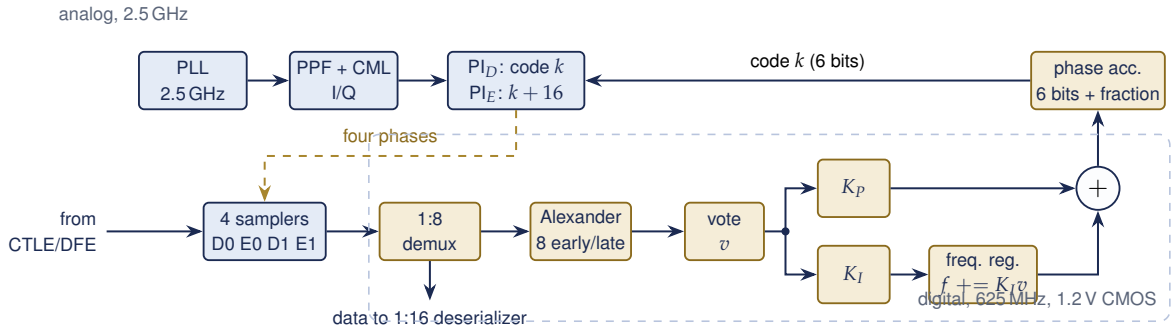
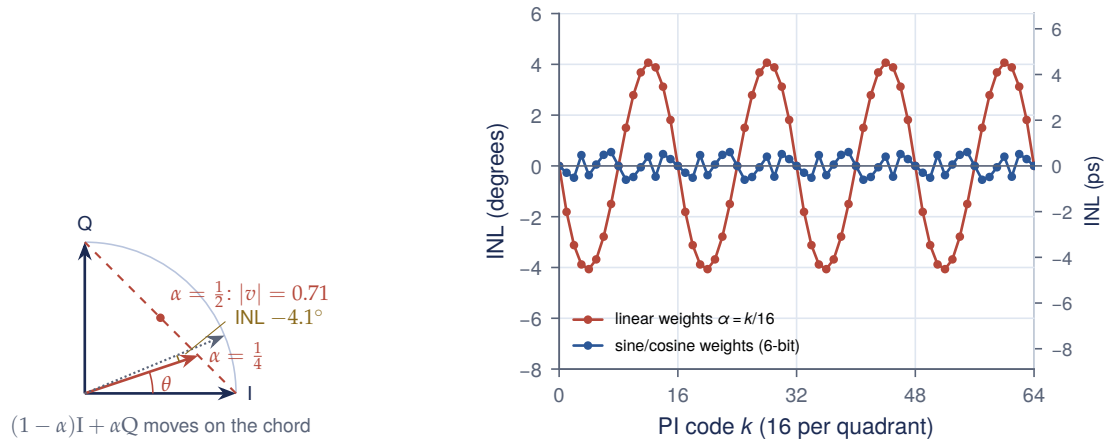


Figure 13.7. The Open-SERDES PI-based dual-loop CDR. The PLL and polyphase filter make I/Q; two PIs rotate them under a 6-bit code; four half-rate samplers feed an Alexander detector through a 1:8 demux; a majority vote drives a proportional path (K_P) and an integral path whose register holds the frequency offset. The phase accumulator keeps fractional bits below the PI step.



(a) Phasors **(b)** INL over the 64 codes

Figure 13.8. Linear weights make the PI nonlinear. (a) Interpolating between I and Q with linear weights moves along the chord (dashed), so at $\alpha = 1/4$ the phase is 18.4° instead of 22.5° and the amplitude drops. (b) INL for linear weights ($\pm 4.1°$ = ± 4.5 ps, repeating every quadrant) and for sine/cosine weights quantized to 6 bits ($\pm 0.55°$).

and the PI rotates continuously, wrapping every 833 updates (1.33 µs). The proportional path alone could follow 0.78 ps/1.6 ns = 488 ppm, so ±300 ppm would use 61% of its slew: hence the integral path. The PI's ceiling, one step per update, is 6.25 ps/1.6 ns = 3906 ppm. The rotating code sweeps the INL pattern at four times the 750 kHz rotation rate, a 3 MHz periodic jitter. Spread-spectrum clocking (5000 ppm) is out of range.

WORKED EXAMPLE 13.2 · PI step size: 64 or 128 steps?

Question: does doubling the PI resolution reduce the CDR's own jitter? **Method:** the behavioural model of `ch13_sim.py`: +200 ppm offset, input RJ 1.32 ps and ±10 ps DDJ, K_P chosen so that every design has the same slew $S = 2.44 \times 10^6$ UI/s (the JTOL requirement), $D = 1$. "Wander" is the recovered-phase error; the peak-to-peak value is taken at the 10^{-5} quantiles of 1.2×10^5 updates.

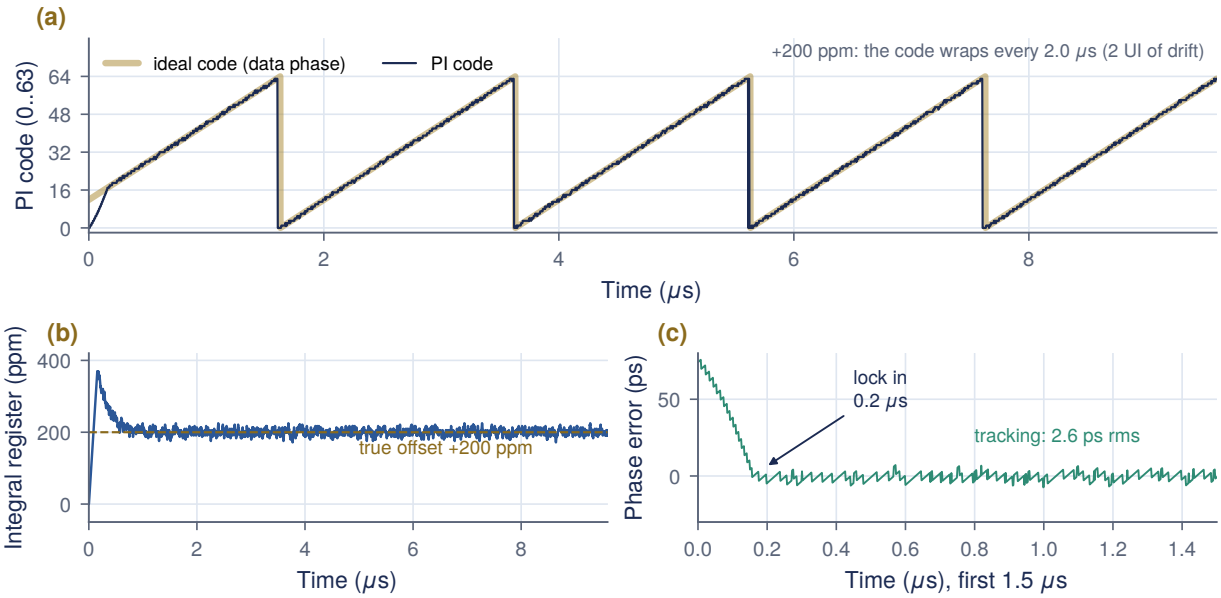


Figure 13.9. The Open-SERDES PI-CDR locks to a +200 ppm input in 0.2 µs and then tracks it. Behavioural simulation of the baseline (64 steps, 625 MHz loop, K_P = 1/8 LSB, K_P/K_I = 128). (a) The PI code follows the drifting data phase and wraps every 2.0 µs, one full rotation of 2 UI. (b) The integral register learns the offset and settles at +200 ppm. (c) Phase error: 74 ps at start, locked after 0.19 µs, then 2.6 ps rms tracking error.

PI steps	loop clock	K_P	wander rms	wander pp	JTOL vs mask
64	312.5 MHz	1 LSB (integer)	10.7 ps	58.1 ps	2.6×
64	312.5 MHz	1/4 LSB	3.3 ps	24.3 ps	pass
128	312.5 MHz	1/2 LSB	2.9 ps	23.0 ps	pass
64	**625 MHz**	**1/8 LSB**	**2.5 ps**	**19.2 ps**	pass
128	625 MHz	1/4 LSB	1.9 ps	15.6 ps	pass

Answer: at a fixed loop clock, 128 steps saves only 1.3 ps. What matters is the proportional step in picoseconds (fixed by JTOL) and the latency in nanoseconds, since dither $\approx 2S\,T_{\text{lat}}$; an integer-step loop is far too noisy. A 625 MHz loop halves the latency and brings 64 steps inside the 20 ps allocation of Table 11.1. Keep 64 steps; use 128 only if the 625 MHz loop fails timing.

Behavioural check. In Figure 13.9 the baseline starts 0.37 UI off with +200 ppm offset: it locks in 0.19 µs, the integral register settles at 200 ppm within about 1 µs, and the tracking error is 2.6 ps rms. The same model gives the JTOL curve in Figure 13.6: 3.0 UI_{pp} at 0.5 MHz, 1.08 at 1 MHz, 0.51 at 2 MHz (mask 0.5: marginal, within the sweep's 10% resolution) and 0.32 at high frequency. Its jitter transfer has a −3 dB corner near 9 MHz for 0.2 UI_{pp} input and 16 MHz for 0.05 UI_{pp}, as Equation (13.3) predicts for a BB loop; Chapter 11 conservatively assumed 4 MHz when counting the PLL's untracked jitter.

▶ What this means for your design

Size a BB-CDR in this order: S from the JTOL mask; $K_P \cdot \text{LSB} = S\,T_{\text{upd}}$; the shortest latency the logic allows (dither $\propto S\,T_{\text{lat}}$); then $K_I \le K_P/16$ and an integral register wide enough

for the ppm range.

>_ Lab 7: CDR

Section 16.8 builds a fuller version of `ch13_sim.py`. Reproduce Figure 13.9, then sweep K_P and the latency D and plot wander against JTOL margin: you will find the trade-off of Worked Example 13.2.

Open-SERDES CDR targets

Half-rate Alexander BBPD; two 64-step PIs (6.25 ps) with sine/cosine weights (INL $\leq 1°$); P+I loop at 625 MHz, $K_P = 1/8$ LSB, $K_P/K_I = 128$, latency ≤ 2 updates; ± 300 ppm; lock < 1 µs; wander ≤ 20 ps pp; JTOL above the mask with TM = 0.4 UI. Power estimate: PIs and buffers 6–10 mW, digital loop about 2 mW.

13.7 Exercises

13.1 TX and RX crystals are at $+100$ and -80 ppm. How long does the data take to drift one UI at 5 Gb/s, and how many PI codes per microsecond must the CDR rotate?

13.2 In the Hogge detector at 5 Gb/s, the clock is 20 ps early. Give the widths of X and Y, and the average current for $I_P = 100$ µA and $TD = 0.5$.

13.3 A BB-CDR sees $\sigma_j = 0.03$ UI. Compute K_{BB} for $TD = 0.5$. By what factor does the small-signal bandwidth change if the channel doubles the jitter?

13.4 For a 625 MHz loop, find the smallest K_P (in PI LSBs) that passes the mask point 2 $\mathrm{UI_{pp}}$ at 0.5 MHz with TM = 0.4 UI. What dither would $D = 3$ give?

KEY TAKEAWAYS

- 200 ppm drifts one UI per microsecond, so the receiver recovers phase and frequency from the data; Open-SERDES uses a PI dual loop that reuses the PLL.
- Hogge is linear but needs 100 ps pulses; Alexander reuses the samplers: (0,1) early, (1,0) late, (0,0) no transition.
- Linear CDR: a PLL with $N = 1$, $\omega_{-3\mathrm{dB}} \approx K_{\mathrm{VCO}} I_P R_p / 2\pi$; only $|1 - H|$ of the jitter reaches the sampler.
- Bang-bang: $K_{\mathrm{BB}} = TD\sqrt{2/\pi}/\sigma_j$; slewing gives $\omega_{-3\mathrm{dB}} = \pi S/(2A_{\mathrm{pk}})$; JTOL $\approx$ TM + $2S/\omega$.
- The Open-SERDES PI-CDR locks to +200 ppm in 0.2 µs, wanders 2.6 ps rms and passes the mask.

PART V

Putting It Together

The complete Open-SERDES architecture and its budgets, the open-source design and verification flow, and hands-on labs that let you reproduce every key plot in this tutorial.

CHAPTER 14

The Open-SERDES Architecture

Every decision of the last thirteen chapters on a few pages: what the project delivers, the blocks and clocks, the number each block must meet at 5 and at 10 Gb/s, the test chip around them, and the plan to get there.

IN THIS CHAPTER YOU WILL LEARN

- Say what the project delivers, and which chapter feeds each deliverable.
- Read the complete Open-SERDES block diagram and follow a bit from the 16-bit TX word to the 16-bit RX word.
- Trace every clock in the chip from the 50 MHz reference to the last divider, with its frequency and phase.
- Use the block specification table to find the 5 and 10 Gb/s targets, and the chapter that derives them, for any block you design.
- Explain why the same architecture reaches 10 Gb/s, and list the six things that must change.
- Build a first-cut power budget from tail currents and convert it to energy per bit.
- Place the transceiver on a test chip: its test logic, its floorplan and its package.
- Defend the ten main architecture decisions, and map the five project phases to what each must prove.

14.1 What Open-SERDES delivers

Before the block diagram, the goal. Open-SERDES is not only a chip: it is a chip, the flow that produced it and the data that proves it, all in one public repository. The project's own summary has five highlights:

Project highlights

1. The first open multi-Gb/s mixed-signal SerDes on an open PDK.
2. A 10 Gb/s-capable NRZ transceiver, with TX, RX, PLL and CDR, in SG13G2, guaranteed at 5 Gb/s (Section 14.4).
3. One reproducible flow from the channel model to GDS (Chapter 15).
4. A link model calibrated against post-layout simulation and, later, silicon (Section 15.2).
5. Everything public: design files, models and lab data.

Table 14.1 turns the highlights into six things anyone will be able to download, and points to the chapters that produce the numbers inside each one.

▶ The test of success

Clone the repository, run `make all`, and reproduce the chip, the models and the measurements. Every design choice in this chapter should make that sentence easier to keep, not harder.

Table 14.1. The six Open-SERDES deliverables. The phase is the roadmap phase of Section 14.8 in which the deliverable is first released.

Deliverable	What it is	Phase	Built in
GDSII	the tape-out-ready 5/10 Gb/s transceiver test chip of Section 14.5	2	Chapters 9, 10, 12, 13 and 15
RTL + testbenches	serializer, deserializer, PRBS generator and checker, CDR loop logic, and their benches	1–2	Sections 9.2, 10.6, 13.6 and 15.3
Link model	the Python link simulator, statistical eye and GUI, driven by one spec file	0	Chapters 4 and 16
Cell library	reusable HBT/CML drivers, latches, muxes and CTLE cells for SG13G2	1	Chapters 7, 9 and 10
Documentation	the GitHub repository and this tutorial	all	every chapter
Paper + data	a journal or conference paper (IEEE JSSC or TCAS, SMACD 2027) with open measurement data	3	Section 15.5

14.2 The complete transceiver

Figure 14.1 puts the whole chip on one page. It is a single-lane transceiver: one transmitter, one receiver and one clock generator on the same SG13G2 die, connected to each other through the off-chip REF channel (Section 3.7) or through an on-chip loopback.

Transmit path. A 16-bit word arrives every 3.2 ns (312.5 MHz), from the user or from the on-chip PRBS7/31 generator (Section 2.2). CMOS on 1.2 V performs the first 16:4 serialization; HBT CML on the 2.5 V analog supply performs the 4:2 and the final half-rate 2:1, built three times so the driver receives D_{n+1}, D_n and D_{n-1} for its three FFE taps (Sections 9.2 and 9.5). The CML driver launches 1 Vppd into 50 Ω per side (Section 9.4).

Receive path. Behind the ESD diodes and the 100 Ω termination (Section 9.6), a loopback multiplexer selects the pad or the near-end loopback. The CTLE adds 0–12 dB of peaking (Chapter 7), the DFE summer subtracts w_1 and w_2 times the previous decisions (Section 8.2), and four samplers decide: data D0, D1 and edges E0, E1 (Section 10.2). The deserializer delivers 16 data and 16 edge bits at 312.5 MHz (Section 10.6); the PRBS checker counts errors.

Clocking and control. An integer-N PLL with an LC VCO multiplies the 50 MHz reference by $N = 50$ to 2.5 GHz (Chapter 12). The TX gets it through buffers and ÷2 dividers; the RX gets I and Q from a polyphase filter (Section 12.5), rotated by two phase interpolators (PIs) under the digital CDR loop filter (Section 13.6). An SPI port writes every setting.

14.2.1 The clocking plan

Figure 14.2 follows each clock from the reference to the last flip-flop. All of the frequencies come from one relation between the bit rate, the half-rate clock and the word rate.

KEY EQUATION · The Open-SERDES clock frequencies

$$f_{\mathrm{clk}} = \frac{f_{\mathrm{b}}}{2} = N f_{\mathrm{ref}} = 2.5\,\mathrm{GHz}, \qquad f_{\mathrm{word}} = \frac{f_{\mathrm{b}}}{16} = 312.5\,\mathrm{MHz}, \qquad \Delta t_{\mathrm{PI}} = \frac{1}{64\, f_{\mathrm{clk}}} = 6.25\,\mathrm{ps} \tag{14.1}$$

The TX needs one clock phase: the final 2:1 mux uses both edges of the 2.5 GHz clock as its

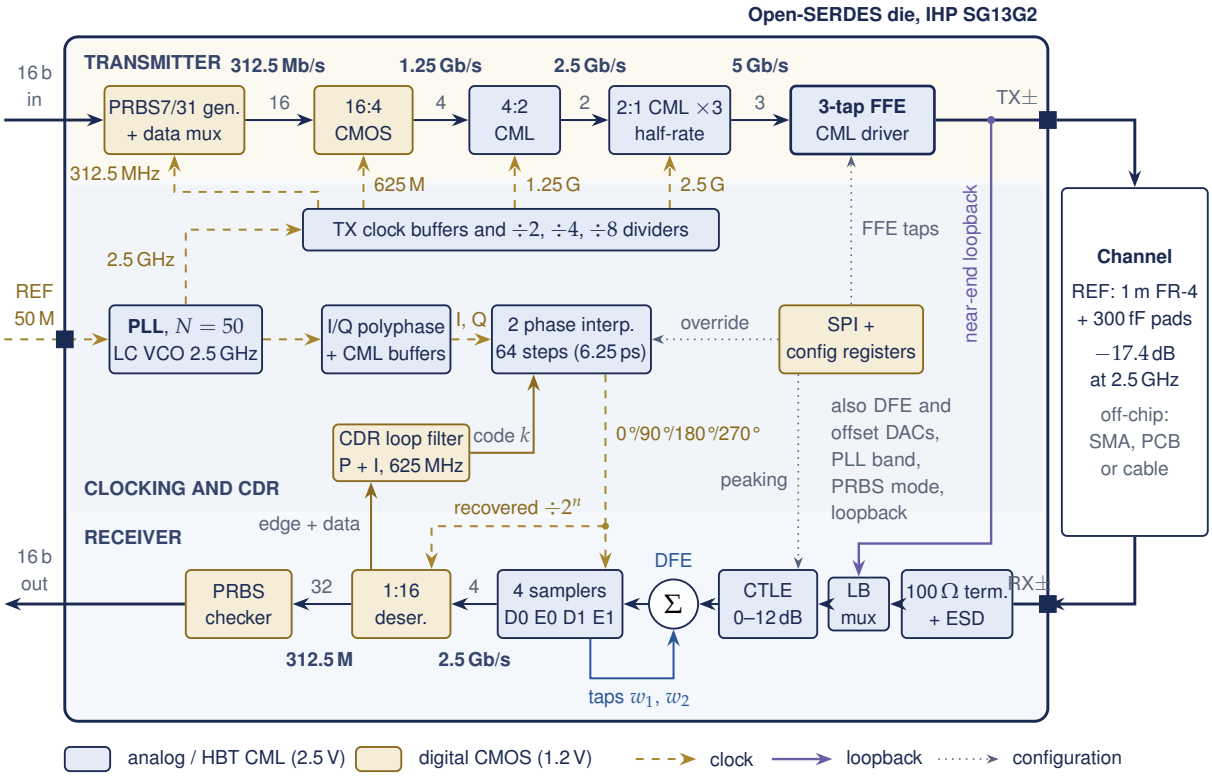


Figure 14.1. The complete Open-SERDES transceiver: a CMOS core at 312.5 MHz, wrapped in HBT CML wherever a signal or clock runs at 1.25 GHz or faster. Read the TX band left to right and the RX band right to left; bold numbers above the TX row are lane rates, and slash numbers count parallel lanes. Notice three things. The 2.5 GHz PLL clock is used twice: divided down for the TX (fixed phase), and split into I/Q and rotated by the PIs for the RX (recovered phase). The CDR is a digital loop: edge and data bits leave the deserializer, the loop filter computes the PI code, and the samplers move. The near-end loopback (violet) takes the driver output straight to the RX input, so the whole link can be tested without a board channel.

select (Section 9.2). The RX needs four phases half a UI apart: data samplers on the edges of I (eye centers), edge samplers on the edges of Q (crossings), all moved together by the CDR in 6.25 ps steps. The RX dividers run from the *recovered* clock, so the deserializer and the CDR logic sit in a clock domain that is not phase-locked to the TX word clock; a small FIFO crosses back to the core (Section 10.6). In near-end loopback both ends share the PLL, so the frequency offset is zero. Through a board to a second chip, the CDR must absorb up to ±300 ppm (Section 13.6). Section 14.4 keeps this plan for a 10 Gb/s target with the clocks doubled, and takes I and Q from a ÷2 instead of the polyphase filter.

▶ What this means for your design

Three boundaries organize the chip, and most integration bugs live on them: the **speed boundary** between CMOS and CML (1.25 Gb/s in the TX, 2.5 Gb/s in the RX), which needs level converters and half a slow-clock cycle of timing margin; the **supply boundary** between 1.2 and 2.5 V, where every V_{CE} must stay below BV_{CEO}; and the **clock-domain boundary** between the TX clock and the recovered RX clock. Draw all three on the floorplan first.

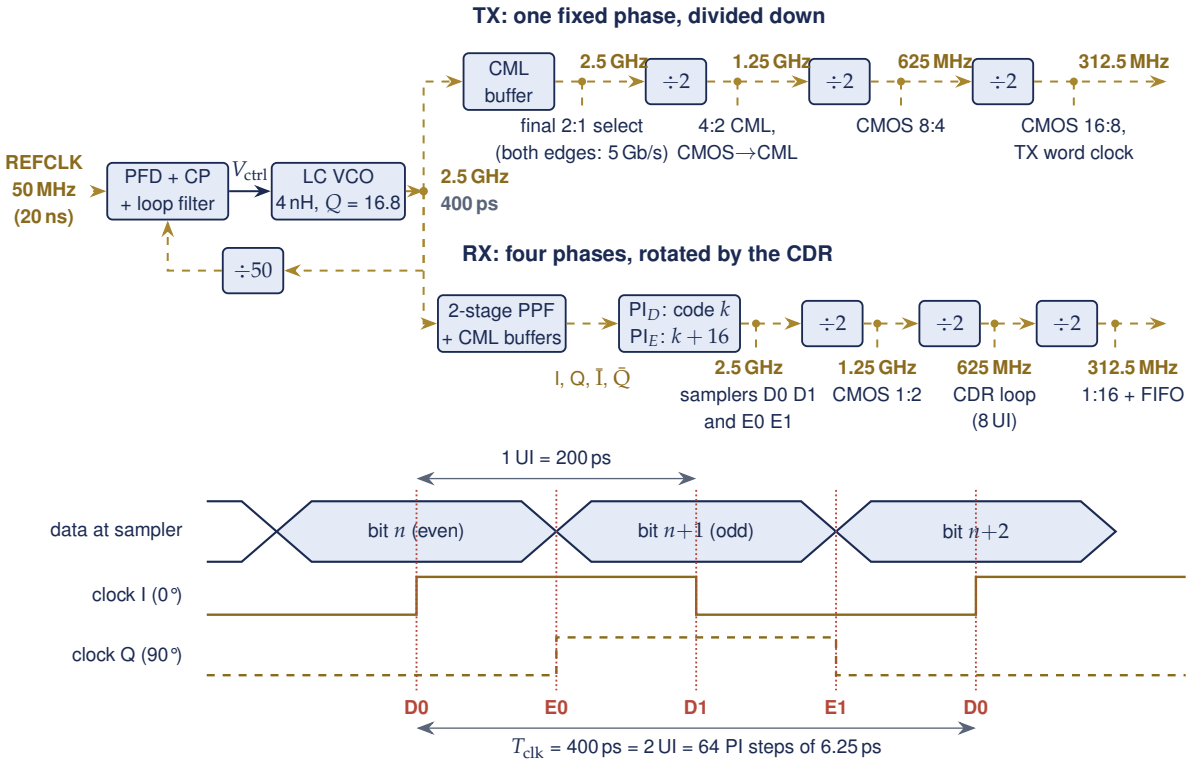


Figure 14.2. One 2.5 GHz PLL serves the whole chip; the TX divides it, the RX rotates it. Top: the frequency at every node. The TX chain divides by 2, 4 and 8 down to the 312.5 MHz word clock. The RX chain first makes four phases with the polyphase filter, rotates them with the two PIs, and divides the recovered clock for the deserializer and for the CDR loop, which updates every 8 UI at 625 MHz. Bottom: what the four phases do. D0 and D1 sample the eye centers on the two edges of I; E0 and E1 sample the crossings on the edges of Q. One 400 ps clock period carries two bits and contains 64 PI steps.

14.3 Block specifications

Table 14.2 is the contract between the system model and the circuit designers: one row per block, the numbers that block must meet, and the chapter where each number is derived. It has two target columns. The **5 Gb/s** column is the baseline that the rest of this book derives, and the pass criterion of first silicon. The **10 Gb/s** column is the design target of Section 14.4: every high-speed block is sized for it, and "same" means the 5 Gb/s value already covers it. Values marked † are first estimates from hand analysis or behavioral models. They must be confirmed by simulation in SG13G2, and some of them will move.

The link budget in one paragraph. Both budgets close (Figure 14.3). In voltage, REF with only the 2-tap TX de-emphasis leaves a 131 mV statistical eye at 10^{-12}; noise ($2Q\sigma_n = 21$ mV), jitter (3.5 mV), offset (10 mV) and sensitivity (20 mV) leave a **76 mV** margin by linear addition, 89 mV statistically (Section 4.4), and the full partition (−6 dB preset + 2-tap DFE, 172 mV noise-free PRBS7 eye, Section 8.5) only adds to it. In time, 40 ps of DJ plus $2Q \times 1.32$ ps of RJ give TJ = **58.6 ps = 0.293 UI**, just inside 0.3 UI (Section 11.5). Timing is the tighter budget, and its largest line is the CDR's own dither.

Table 14.2. Open-SERDES block specifications: 5 Gb/s guaranteed, 10 Gb/s target. † First estimate, not yet confirmed by circuit simulation in SG13G2. REF-10 is the REF line shortened to 0.5 m (Section 14.4).

Block	Key specification	5 Gb/s: guaranteed	10 Gb/s: target	Derived in
Link	rate, BER, channel	5 Gb/s NRZ; BER $\leq 10^{-12}$ over REF (-17.4 dB at 2.5 GHz); margin up to 20 dB	10 Gb/s NRZ; BER $\leq 10^{-12}$ over REF-10 (-15.3 dB at 5 GHz)	Sections 1.4, 3.7 and 14.4
Serializer	ratio, clocks, jitter	16:1 from 16 b at 312.5 MHz; CMOS 16:4, CML 4:2, half-rate CML 2:1; DCD + DDJ ≤ 8 ps†	16 b at 625 MHz; CMOS 16:8, CML 8:4, 4:2, half-rate 2:1; DCD + DDJ ≤ 4 ps†	Section 9.2
TX driver	topology, swing	HBT CML, 50 Ω back-termination per side; 0.8–1.0 Vppd (I_T = 16–20 mA); every $V_{CE} < BV_{CEO} \approx 1.6$ V at all corners	same; 20–80 % edges ≤ 30 ps†	Section 9.4
TX FFE	taps, range, resolution	c_{-1}, c_0, c_1 with $\sum\lvert c_k\rvert = 1$; c_1 to -0.25 (6 dB), c_{-1} to -0.1; 5-bit current cells (steps of 1/32)	same	Sections 6.2 and 9.5
Termination, pad	impedance, C_p, return loss	100 Ω diff., ±10 % after 3–4-bit trim; $C_p \leq 0.85$ pF (0.63 pF estimated†); RL ≥ 10 dB at f_N	RL ≥ 10 dB at 5 GHz: $C_p \leq 0.42$ pF, so the T-coil is required	Sections 9.3 and 9.6
CTLE	peaking, poles, noise	0–12 dB programmable; $f_{p1} \approx 2.5$ GHz, $f_{p2} \geq 7.5$ GHz; noise ≤ 0.25 mV rms†	same peaking; $f_{p1} \approx 5$ GHz, $f_{p2} \geq 15$ GHz	Chapter 7
DFE	taps, loop timing	half-rate, 2 taps, direct feedback; w_1 up to 0.5 p_0, w_2 to ±0.15 p_0; tap-1 loop ≈140 ps† < 1 UI; loop unrolling as fallback	tap 1 unrolled (the ≈140 ps loop exceeds the 100 ps UI); tap 2 direct	Section 8.2
Samplers	count, sensitivity, offset	4 HBT CML master–slave (D0 E0 D1 E1); $V_{sens} \approx 20$ mV; offset $\sigma \approx 4.9$ mV raw†, ±0.46 mV after 5-bit trim; $\tau \leq 5$ ps; aperture ≤ 0.2 UI	6 (two per data phase for the unrolled tap, E0, E1); $\tau \leq 3$ ps†; aperture ≤ 20 ps	Sections 10.2 and 10.4
Deserializer	ratio, output	1:16 CMOS; 16 data + 16 edge bits at 312.5 MHz; two extra latches on the edge path; FIFO to the core	one CML 1:2 first, then CMOS to 16 + 16 bits at 625 MHz	Sections 10.3 and 10.6
PLL	architecture, jitter	integer-N CP-PLL, $f_{ref} = 50$ MHz, $N = 50$; LC VCO 2.5 GHz (4 nH, $Q = 16.8$); $K_{VCO} \approx 2\pi \times 150$ MHz/V; $f_u \approx 3$ MHz; RJ ≤ 1.0 ps rms†	LC VCO at 10 GHz, $N = 200$; RJ ≤ 0.5 ps rms†	Chapter 12
Quadrature	accuracy	2-stage polyphase + CML buffers; I/Q error $\leq 2°$ (3σ); duty cycle 50 ± 0.5 % with DCC	CML ÷2 from 10 GHz gives 5 GHz I/Q; a second ÷2 gives the 2.5 GHz I/Q of the 5 Gb/s mode; no polyphase	Section 12.5
PI	resolution, linearity	2 PIs, 64 steps per 400 ps (6.25 ps = 0.031 UI); INL $\leq 1°$	64 steps per 200 ps (3.125 ps = 0.031 UI); same INL	Section 13.6
CDR	loop, tracking	half-rate Alexander BBPD; P+I at 625 MHz, $K_P = 1/8$ LSB, $K_P/K_I = 128$; ±300 ppm; lock < 1 µs; wander ≤ 20 ps pp†; JTOL above the USB-style mask	same logic at 625 MHz, now every 16 UI; dither ≤ 8 ps pp†	Sections 13.4 and 13.5
Timing budget	total jitter	TJ ≤ 0.3 UI (60 ps) at 10^{-12}; estimated 58.6 ps†	TJ ≤ 0.3 UI (30 ps): DJ ≤ 18 ps, RJ ≤ 0.85 ps rms; the 5 Gb/s circuits give 0.46 UI†	Section 11.5 and Worked Example 14.1
Voltage budget	margin at the sampler	≥ 50 mV after noise, offset and sensitivity; 76 mV on REF with 2-tap de-emphasis†	≥ 50 mV; worst-case eye 193 mV on REF-10 (134 mV at 5 Gb/s on REF)†	Sections 4.4 and 14.4
Test	on-chip	PRBS7/31 generator and checker with error counter; near-end loopback; SPI registers	same, plus a rate-select register and an external clock input	Sections 14.5 and 15.5
Die, package	size, pinout	die about 2 × 2 mm² (to be sized); QFN-48, 7 × 7 mm; high-speed bond wires ≤ 1 mm (concept)	high-speed bond wires ≤ 0.5 mm	Section 14.5
Power	total, per bit	≈216 mW, 43 pJ/b†	≈280 mW, ≈28 pJ/b†	Sections 14.4 and 14.6

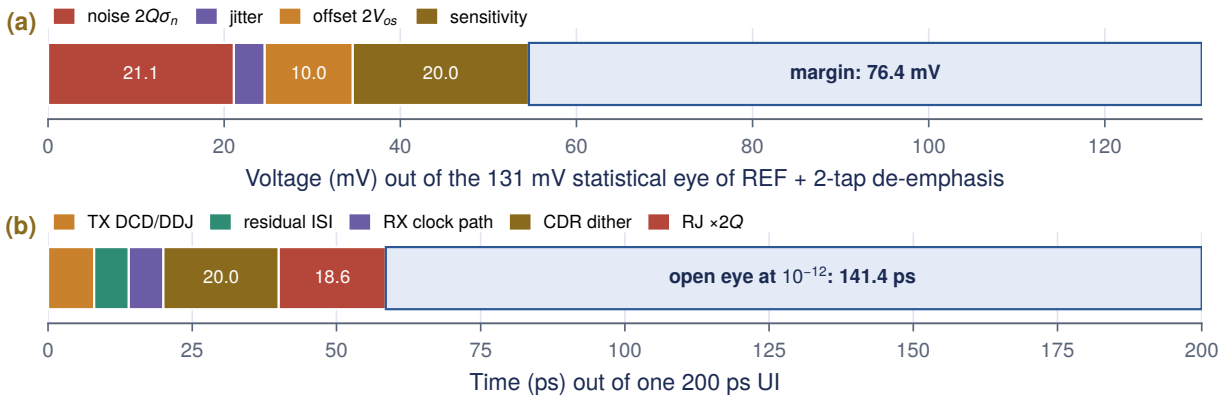


Figure 14.3. Both budgets close, but timing is the tight one. (a) The 131 mV statistical eye of REF with 2-tap de-emphasis, minus the terms of Section 4.4: 76 mV of margin remains, more than half the eye. (b) One 200 ps UI minus the deterministic jitter (linear sum) and $2Q$ times the random jitter (RSS): 141 ps remains, only 1.4 ps more than the 0.7 UI target allows.

14.4 Design for 10, guarantee 5

Everything so far describes the 5 Gb/s baseline that the rest of this book derives. Can the same architecture go faster? The transistors are not what stops it: a 100 ps UI still holds about fifty SG13G2 CML gate delays (Figure 1.6), while 130 nm CMOS ran out of margin at 5 Gb/s. The limits are the channel, the clocks and the timing budget. This section shows that 10 Gb/s NRZ fits the same architecture with six changes, and sets the project target accordingly: **design every high-speed block for 10 Gb/s, and keep 5 Gb/s over REF as the pass criterion of first silicon.** The 10 Gb/s column of Table 14.2 collects the resulting numbers.

The equalizer sees decibels, not bits per second. Figure 14.4(a) repeats the partition study of Section 8.5 at 5, 8 and 10 Gb/s NRZ and at 10 Gb/s PAM4, with the CTLE poles moved to each Nyquist frequency. Over REF, 10 Gb/s NRZ meets −30.1 dB at 5 GHz and the worst-case eye closes to 18 mV. Over half the line, 0.5 m with the same pads, it opens to **193 mV**, more than the 134 mV of the 5 Gb/s baseline over REF. We call this channel **REF-10**: −15.3 dB at 5 GHz. The rule behind it is simple. The equalizer only sees the loss at Nyquist; two links with the same loss need the same equalizer, whatever their bit rates. PAM4 at 10 Gb/s keeps the 2.5 GHz Nyquist frequency of REF but gives up two-thirds of each sub-eye: 14 mV over REF and 97 mV over REF-10, below NRZ at both lengths and with three slicers per sample (Section 4.5).

Six changes. The block diagram of Figure 14.1 stays: half-rate clocking, the CMOS/CML split, FFE + CTLE + DFE, a PI-based bang-bang CDR. Six things change.

1. **Clocks at 5 GHz.** The half-rate clock doubles. A 10 GHz LC VCO followed by a CML ÷2 gives I and Q at 5 GHz, with the quadrature set by the divider rather than by matching (Section 12.5). A second ÷2 gives the 2.5 GHz I and Q of the 5 Gb/s mode, so one register selects the rate and the polyphase filter goes away. The cost: the team's 2.4 GHz PLL [9] is reused for its loop and its flow, not for its tank.
2. **The timing budget.** Jitter set in picoseconds does not shrink with the UI. Worked Example 14.1 shows that the budget closes only if those terms roughly halve; this is the main extra work.
3. **One more CML stage on each side.** CMOS stays at 1.25 Gb/s per lane or less (Section 9.2). The TX becomes CMOS 16:8 on a 625 MHz word clock, then CML 8:4, 4:2 and the half-rate

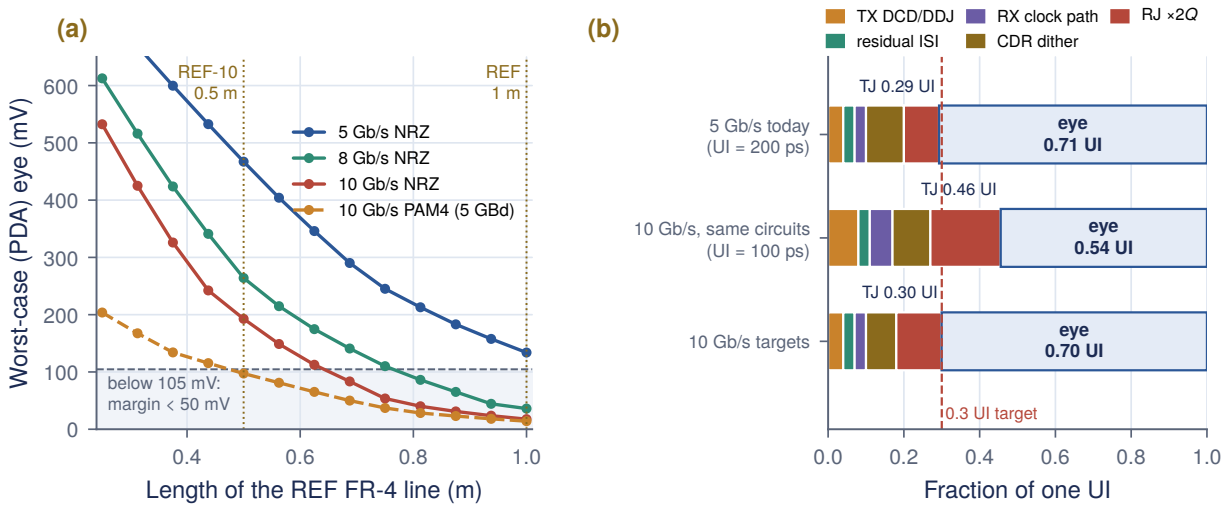


Figure 14.4. The same equalizer carries 10 Gb/s over half the FR-4; it is the timing budget that must improve. (a) Worst-case (PDA) eye after the Open-SERDES partition (3-tap TX FFE within its limits, CTLE 0–12 dB with poles at f_N and $4f_N$, 2-tap DFE) against the length of the REF line, for 5, 8 and 10 Gb/s NRZ and 10 Gb/s PAM4. The shaded band is the eye needed for a 50 mV margin after the noise, jitter, offset and sensitivity terms of Figure 14.3. (b) The timing budget of Figure 14.3 as a fraction of the UI: 5 Gb/s today, the same circuits at 10 Gb/s, and the 10 Gb/s targets of Table 14.2.

2:1; the RX adds one CML 1:2 in front of the CMOS deserializer.

4. **DFE tap 1 is unrolled.** The direct tap-1 loop needs about 140 ps (Section 8.2), more than the 100 ps UI, so tap 1 becomes speculative, the fallback already listed in Table 14.2. Each data phase gets two samplers.
5. **Pad and package.** For 10 dB of return loss at 5 GHz the pad may carry only half the capacitance, 0.42 pF, and a bond wire only half the inductance, about 0.5 mm (Worked Example 14.2). The T-coil of Section 9.6 stops being a fallback and becomes part of the pad.
6. **Power.** CML draws its tails whatever the rate, so only the additions cost power: the extra CML stages, faster clock buffers, two more samplers and the dividers add about 65 mW. That gives ≈280 mW, or about **28 pJ/b** at 10 Gb/s against 43 pJ/b at 5 Gb/s.

WORKED EXAMPLE 14.1 · Does the timing budget close at 10 Gb/s?

Given: the 5 Gb/s budget of Section 11.5: TX DCD/DDJ 8 ps, residual ISI 6 ps, RX clock path 6 ps, CDR dither 20 ps; RJ σ = 1.32 ps (RSS of 1.0, 0.7 and 0.5 ps); $2Q$ = 14.07 at 10^{-12}; target TJ $\leq$ 0.3 UI.

Same circuits at 10 Gb/s: residual ISI and CDR dither scale with the UI (3 and 10 ps); duty-cycle error, clock-path jitter and RJ are set in picoseconds and stay. DJ = 8 + 3 + 6 + 10 = 27 ps; $2Q\sigma$ = 18.6 ps; TJ = 45.6 ps = 0.46 UI. The budget fails.

Targets: duty-cycle correction to DCD + DDJ $\leq$ 4 ps, clock path $\leq$ 3 ps, CDR dither $\leq$ 8 ps give DJ = 18 ps. Then $2Q\sigma \leq 30 - 18 = 12$ ps, so $\sigma \leq 0.85$ ps rms, for example 0.5 ps from the PLL, 0.5 ps from the clock path and 0.45 ps from the TX (RSS 0.84 ps).

Result: at 10 Gb/s every jitter source specified in picoseconds must roughly halve. That is why the PLL row of Table 14.2 asks for 0.5 ps rms at 10 GHz, and why Figure 14.4(b) shows timing, not the eye height, as the 10 Gb/s problem.

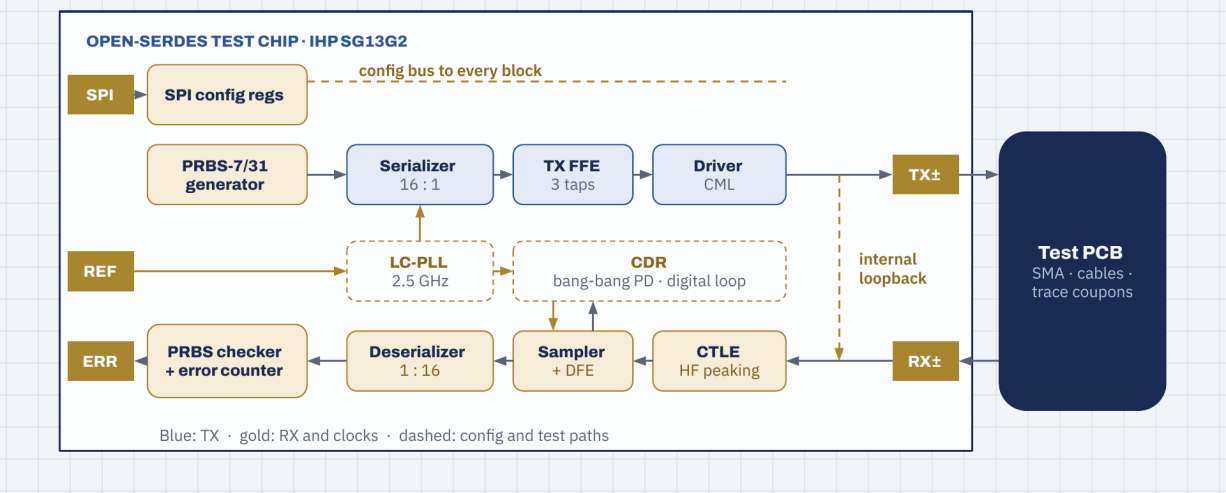


Figure 14.5. The Open-SERDES test chip (concept): the transceiver of Figure 14.1 plus everything needed to test it without a BERT. Blue blocks are TX, gold blocks are RX and clocking, and dashed lines are configuration and test paths. Only five signals leave the chip besides supplies: TX±, RX±, the reference clock, the SPI bus and an error flag. From the Open-SERDES introduction slides.

▶ Why not 25 Gb/s?

SiGe BiCMOS of this class has been used for clock recovery at far higher rates. But past about 12 Gb/s over these channels the architecture changes: a quarter-rate or all-CML datapath with eight calibrated clock phases, more DFE taps with adaptation, clocks below 300 fs rms, flip-chip or a better package, and channels under about 25 cm. That is a different chip and a larger team, so it belongs to Phase 4 of Section 14.8.

14.5 From transceiver to test chip

Figure 14.1 is the transceiver. To measure it, the transceiver must sit on a die with test logic around it, a floorplan that respects its three boundaries, and a package that does not ruin its 2.5 GHz signals. This section shows the concept for each. The details will change as the blocks are designed, but the rules behind them will not.

14.5.1 The test chip

Figure 14.5 is the chip-level view of Figure 14.1, with the test logic made explicit. Digital blocks sit on the left: SPI configuration registers that reach every block, a PRBS7/31 generator for the TX, and a PRBS checker with an error counter for the RX. The TX row runs across the top (serializer, 3-tap FFE, CML driver), and the RX row runs back along the bottom (CTLE, sampler with DFE, deserializer). The LC-PLL takes an external reference and clocks the TX; the CDR steers the sampling phase from the data itself. Two paths close the loop. The internal loopback sends the driver output straight to the RX input, which tests the whole chip with no board at all. The external path goes through a test PCB, SMA cables and trace coupons, which adds a real channel.

The error counter is what makes a cheap lab possible. With PRBS31 running and the counter at zero, a BER below 10^{-12} is shown at 95 % confidence after $3/\mathrm{BER} = 3 \times 10^{12}$ bits, which takes

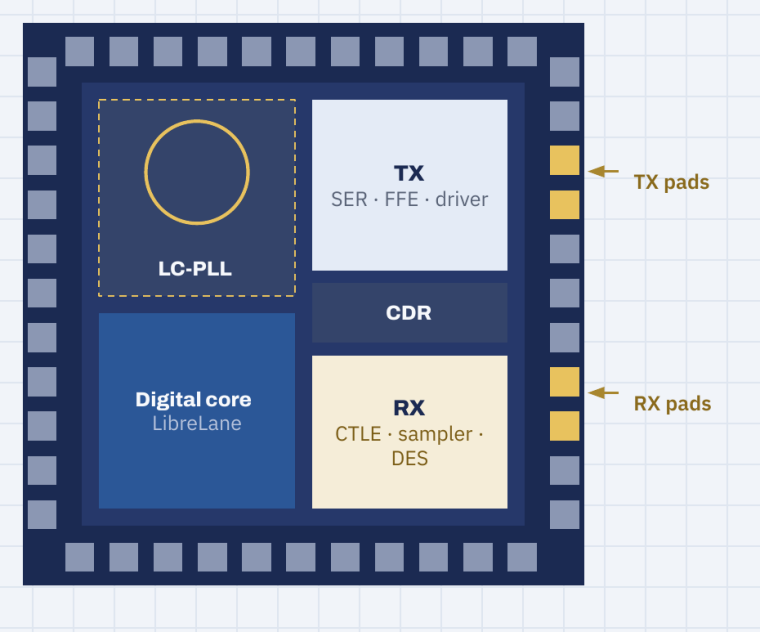


Figure 14.6. Floorplan concept: the high-speed blocks hug one edge, the PLL gets a corner to itself. TX (top right) and RX (bottom right) sit beside their pads (gold); the CDR sits between them, so the four sampling clocks travel short and equal distances. The LC-PLL (dashed, with its guard ring) and the LibreLane digital core fill the left half. Sizes and positions are placeholders. From the Open-SERDES introduction slides.

10 minutes at 5 Gb/s (Section 15.5). An oscilloscope then only has to show the eye, not count errors.

14.5.2 Floorplan

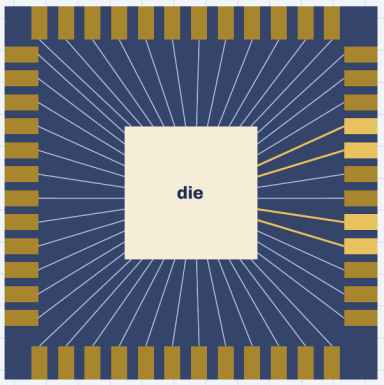
Figure 14.6 is a first floorplan. Each placement follows from a number in an earlier chapter:

- **TX and RX pads on one edge,** so the high-speed bond wires are the shortest on the die (Worked Example 14.2).
- **The LC-PLL in its own corner,** with a guard ring and its own supply, away from the switching digital core. The 4 nH inductor needs keep-out space on TopMetal2 (Section 15.4).
- **TX and RX next to their pads, with the CDR between them,** because the I/Q clocks and the PI outputs must reach four samplers with matched delays (Section 12.5).
- **Three supply domains:** AVDD at about 2.5 V for the HBT CML, DVDD at 1.2 V for CMOS, and VDDIO at 3.3 V for the pads. This is the supply boundary of Section 14.2, drawn on silicon.
- **A digital core hardened with LibreLane,** and the seal ring, fill and logo taken from the JKU template flow [5].

The die is about $2 \times 2\,\mathrm{mm}^2$ for now. The PLL alone is $0.347\,\mathrm{mm}^2$ [9], and the final size waits for the block layouts.

14.5.3 Package and pinout

A QFN package is cheap and easy to solder to a test PCB, but every bond wire is a series inductor of about 1 nH per millimeter (Worked Example 3.4). Figure 14.7 shows a QFN-48



Group	Pins	Notes
High-speed	TXP, TXN, RXP, RXN	GSSG, shortest bond wires
Clock	REFCLK P / N	external reference
Control	SCK, MOSI, MISO, CS	SPI config + BER readout
Supplies	AVDD, DVDD, VDDIO, VSS	separate domains
Status	ERR, LOCK, DBG	debug and monitoring

Bonding plan generated from a config file, as in the JKU template flow.

Figure 14.7. Package and pinout concept: QFN-48, 7 mm × 7 mm. Left: the shortest bond wires go to TX± and RX±. Right: the pin groups, with high-speed pins in a GSSG arrangement, a differential reference clock, SPI for configuration and BER readout, separate supply domains and three status pins. From the Open-SERDES introduction slides.

concept: the four high-speed pins on one side, with ground pins between them (GSSG), and the clock, SPI, supply and status pins grouped around the rest. As in the JKU template, the bonding plan is generated from a configuration file, so a pin change is one edit.

WORKED EXAMPLE 14.2 · How long may a high-speed bond wire be?

Given: a bond wire of $L \approx 1\,\text{nH}$ per mm in series between two $50\,\Omega$ environments; a return-loss goal of 10 dB at $f_N = 2.5\,\text{GHz}$, the same goal as for the pad in Table 14.2.

Return loss of a series inductor: with $x = \omega L/(2Z_0)$, $|S_{11}| = x/\sqrt{1+x^2}$ (Worked Example 3.4).

- 1 mm, 1 nH: $x = 2\pi \cdot 2.5\,\text{GHz} \cdot 1\,\text{nH}/100\,\Omega = 0.157$, so $|S_{11}| = 0.155$ and RL = 16.2 dB.
- 2.5 mm, 2.5 nH: $x = 0.393$, so RL = 8.7 dB. This already misses the goal, before the pad capacitance is added.
- 3 mm, 3 nH (a corner pad of a large QFN): RL = 7.4 dB.
- At 5 GHz, the Nyquist frequency of the 10 Gb/s target (Section 14.4), 1 mm gives $x = 0.314$ and RL = 10.5 dB, with nothing left for the pad; 0.5 mm gives 16.2 dB again.

Result: keep the high-speed bond wires at or below about 1 mm, and about 0.5 mm for the 10 Gb/s target. That rule puts the TX and RX pads on the die edge closest to their package pins, and puts those pins in the middle of one package side. If the wires cannot be that short, part of the inductance can be absorbed into the T-coil of Section 9.6.

14.6 Power budget

A CML gate draws its tail current whatever the data, so a CML-heavy chip has a power budget you can write down before any layout: list the tails, multiply by the supply. CMOS logic burns $\alpha C V^2 f$ instead.

Table 14.3. First-cut Open-SERDES power budget. Currents from the checklists and target boxes of Chapters 9, 10, 12 and 13; every line is an estimate to be replaced by simulation.

	Blocks	Assumption	mW
TX	3-tap CML driver	I_T = 20 mA from 2.5 V (40 mW at 0.8 Vppd)	50.0
	CML 4:2, 2:1 (×3), latches, pre-drivers, clock buffers, ÷2	about 17 CML gates of 1.2 mA	51.0
	CMOS 16:4	1.2 V, 625 MHz	3.0
RX	CTLE + buffer, DFE summers and tap DACs	5 mA + 6 mA	27.5
	4 CML samplers + their clock buffers	12 mA + 4 mA	40.0
	CMOS 1:16 deserializer, offset DACs	1.2 V; 5 bits ×4	5.6
Clocking	PLL (team design)	measured, 130 nm BiCMOS [9]	12.7
	polyphase + I/Q buffers, distribution, 2 PIs	6 mA + 8 mW (Section 13.6)	23.0
Digital	CDR loop filter, PRBS gen. + checker, SPI	CMOS at 625 and 312.5 MHz	3.2
Total		TX 104, RX 73, clocking 36, digital 3	**216**

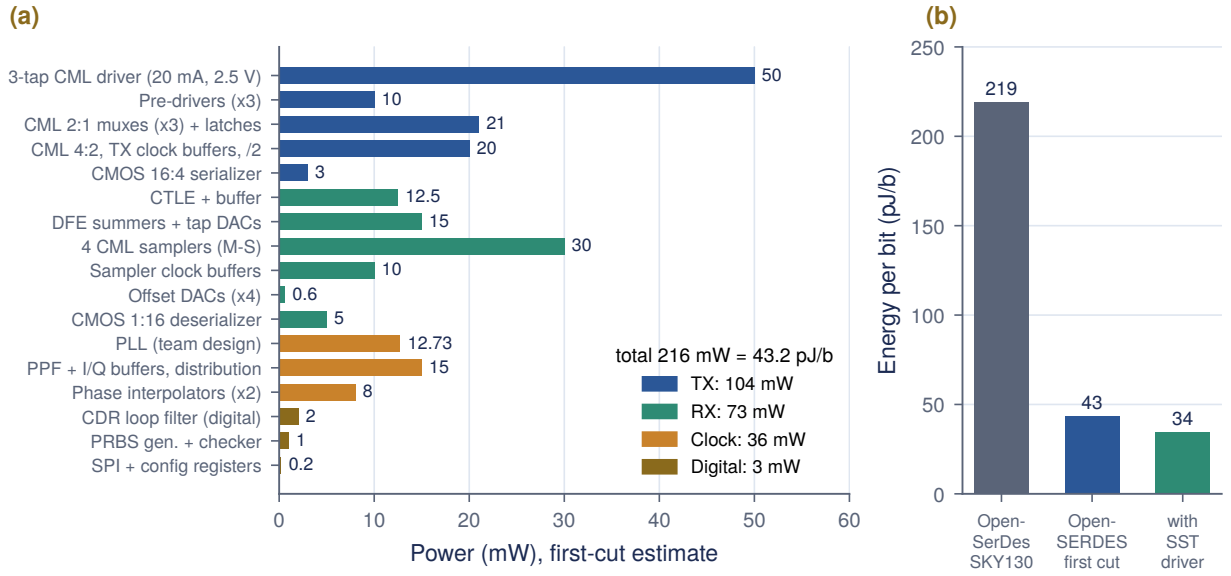


Figure 14.8. The CML driver alone is almost a quarter of the chip's power; CMOS logic is a rounding error. (a) The first-cut budget of Table 14.3, colored by section. The TX takes 48 %, mostly static CML current; the four samplers are the largest RX item. (b) Energy per bit against the Purdue OpenSerDes, an all-digital 2 Gb/s link in SKY130 at about 438 mW [11], and against Open-SERDES with the voltage-mode (SST) driver of Section 9.4.

KEY EQUATION · CML power and energy per bit

$$P_{\text{CML}} = V_{\text{CC}} \sum_i I_{T,i}, \qquad P_{\text{CMOS}} = \alpha\, C_{\text{sw}} V_{\text{DD}}^2 f, \qquad E_b = \frac{P_{\text{total}}}{f_{\text{b}}} \tag{14.2}$$

Table 14.3 collects the estimates of Chapters 9, 10, 12 and 13 and Figure 14.8 plots them. Assumptions: 1 Vppd swing (20 mA driver tail); CML gates as in Worked Example 9.3, about 1.2 mA each with emitter followers; CML on 2.5 V, CMOS on 1.2 V; the team's measured PLL power of 12.73 mW [9]. The polyphase buffers, the PIs and the sampler clock buffers are separate circuits and are counted separately.

WORKED EXAMPLE 14.3 · Energy per bit, and where to save

Total: the CML tails add up to 77 mA from 2.5 V (192 mW); the PLL, the CMOS blocks and the DACs add 25 mW: $P \approx 216\,\text{mW}$. By Equation (14.2), $E_b = 216\,\text{mW}/5\,\text{Gb/s} = \mathbf{43.2}\,\text{pJ/b}$.

Comparison: the Purdue OpenSerDes spends $438\,\text{mW}/2\,\text{Gb/s} = 219\,\text{pJ/b}$ [11]. Open-SERDES runs 2.5 times faster for about half the power, five times less energy per bit, because the HBTs do in a few CML gates what an all-digital 130 nm CMOS design must do with oversampling and many more gates.

Savings, largest first: (1) the SST driver of Section 9.4 saves about 44 mW, giving 172 mW and 34.4 pJ/b; (2) a swing of 0.8 Vppd saves 10 mW; (3) each CML gate removed from the clock paths saves about 3 mW. Lowering the CML supply would save the most of all, but every V_{CE} must still fit below BV_{CEO} with the swing on top (Section 9.4), so this waits for silicon data.

14.7 Design decisions

Table 14.4 lists the ten choices that shape Open-SERDES. Each row says what was chosen, what was rejected and why, and where the argument is made in full. Use it in design reviews: if someone proposes to change a row, the chapter in the last column says which numbers must be recomputed.

14.8 Roadmap: from blocks to silicon

Open-SERDES is built in five phases (Figure 1.5). Each phase ends with a tagged release of design files, models and data, and each must prove something specific before the next one starts. Table 14.5 states what that is.

▶ **Why Phase 1 comes before the full chip**

Every † in Table 14.2 is a risk until it is measured. A small shuttle with only the driver, the CTLE, the samplers and the 10 GHz VCO turns the riskiest estimates (pad capacitance, CTLE peaking, sampler offset, phase noise at 10 GHz) into data a year before the full chip, while the PLL loop and the CDR logic can rely on the team's existing PLL and on simulation.

14.9 Risks and open questions

The architecture closes on paper. It can still fail in these places, roughly in order of the cost of a late fix:

- **ESD and pad capacitance.** ESD diodes can take most of the 0.85 pF budget alone; get the SG13G2 ESD cell capacitance early, and keep the T-coil of Section 9.6 as the fallback.
- BV_{CEO} **headroom.** The off driver transistor sits at 1.25–1.5 V against about 1.6 V; simulate every V_{CE} at every corner with the real termination.
- **PLL phase noise.** The quoted −100.8 dBc/Hz at 1 MHz is far worse than Leeson predicts (Section 12.4); the 1 ps RJ line depends on what it really is.

Table 14.4. The ten main Open-SERDES architecture decisions.

Decision	Chosen	Alternatives	Why	Ch.
Modulation	NRZ	PAM4	on REF NRZ wins by 2.6 dB; PAM4 pays only above about 24 dB and needs 3 slicers and linear circuits; at 10 Gb/s NRZ wins at every length	Sections 4.5 and 14.4
Clock rate	half rate: 2.5 GHz at 5 Gb/s, 5 GHz at 10 Gb/s	full rate, quarter rate	full rate doubles every clock; quarter rate needs 8 calibrated phases	Sections 9.2 and 12.5
Fast logic	SiGe HBT CML	CMOS, MOS CML	full switching at 100 mV, gain 5.8 vs. 2; CMOS misses the 8 FO4 rule at 2.5 GHz, slow corner	Section 9.2
TX driver	CML	SST (voltage mode)	resistor-set impedance, constant current, taps leave the impedance alone; SST (−44 mW) is the upgrade	Section 9.4
TX FFE	3 taps	2 taps, 4+ taps	post-tap does most of the work; pre-tap for p_{-1} on longer channels (on REF, 2 taps win: Lab 3)	Chapter 6 and Section 16.4
RX equalizer	CTLE 0–12 dB + 2-tap DFE	CTLE only, more DFE taps, RX FFE	DFE removes p_1, p_2 without noise; CTLE adds gain, adaptivity and reach above 22 dB; tap 1 unrolled at 10 Gb/s	Sections 8.2 and 8.5
VCO	LC: 2.5 GHz baseline, 10 GHz target	ring VCO	15–30 dB lower phase noise; the team's proven 2.4 GHz design sets the topology and the flow	Sections 12.4 and 14.4
CDR	PI dual loop, digital BB	VCO-based BB, linear (Hogge)	reuses the clean PLL; synthesizable; PVT-independent gain; no false lock	Section 13.6
Quadrature	baseline: 2-stage polyphase; target: 10 GHz VCO + CML ÷2	quadrature VCO	polyphase keeps the 2.5 GHz VCO; ÷2 gives exact I/Q at both rates and a second ÷2 selects the rate	Sections 12.5 and 14.4
On-chip test	PRBS7/31 + checker, loopback	external BERT only	scope and PC suffice; 10^{-12} in 10 minutes; loopback isolates the chip	Section 15.5

- **Inductor EM modeling.** The 4 nH tank and any T-coil must be EM-simulated on TopMetal2 (Section 15.4) and matched to the team's measured Q.
- **Timing loops.** DFE tap 1 uses about 140 of its 200 ps and the CDR dither 19 of its 20 ps; both need post-layout confirmation.
- **The 10 GHz clock and the 10 Gb/s timing budget.** The VCO is a new design, and the budget needs every picosecond term roughly halved (Worked Example 14.1). Put the VCO and its ÷2 on the Phase-1 shuttle, and give the test chip an external clock input so that both modes can be measured without the on-chip PLL.
- **Packaging and measurement.** Bond wires longer than about 1 mm (0.5 mm at 10 Gb/s) cost return loss (Worked Example 14.2); the test PCB, a clean reference clock and access to a sampling scope or BERT must be ready before first silicon.
- **Area, pads and the channel.** The PLL is 0.347 mm^2 [9], the MIM capacitors add tenths of a mm^2, and the high-speed pads need SMA-grade access; REF must be replaced by measured S-parameters (Section 16.9).

Table 14.5. The five Open-SERDES phases and what each must prove. Exit criteria use the numbers of Table 14.2; the dates are still to be agreed.

Phase	Work	Exit criterion and release	Chapters
0 · Feasibility	channel and link model; pick the architecture; freeze block specs	the link, TX and RX rows of Table 14.2 confirmed in simulation; REF replaced by a standard channel; release: model + spec file	Chapters 3, 4 and 8
1 · Building blocks	HBT/CML cell library sized for 10 Gb/s; driver, CTLE, sampler, and the 10 GHz VCO with its ÷2; a test-chip shuttle (Tiny Tapeout [43] IHP or an IHP open MPW)	measured CTLE $H(f)$, driver eye, sampler offset and sensitivity, and VCO phase noise back in the model; release: cell library + block GDS	Chapters 7, 9, 10 and 12
2 · 5/10 Gb/s transceiver	TX + RX + PLL + CDR with a rate select; on-chip PRBS loopback; full-chip tape-out	post-layout statistical eye meets the pass criteria at 5 Gb/s over REF (required) and at 10 Gb/s over REF-10 (target); full-chip DRC/LVS clean; release: GDSII + RTL	Chapters 12, 13 and 15 and Section 14.4
3 · Measure and publish	test PCB; eye and BER in both modes; open measurement data; paper (IEEE JSSC or TCAS, SMACD 2027)	BER $\leq 10^{-12}$ at 5 Gb/s through a real channel and at 10 Gb/s through a short one; bathtub and JTOL measured; release: data + paper	Section 15.5 and Chapter 16
4 · Beyond	16–25 Gb/s (quarter-rate or all-CML datapath); adaptive DFE; PAM4 for lossy channels; multi-lane and chiplet links	a new spec table, starting from the measured Phase-3 data	Sections 4.5, 8.3 and 14.4

KEY TAKEAWAYS

- Open-SERDES delivers six things: the GDSII, the RTL and benches, the link model and GUI, an HBT/CML cell library, documentation, and a paper with open data.
- Open-SERDES is a single-lane transceiver: CMOS at 312.5 MHz in the core, HBT CML at 1.25 GHz and above, joined by level converters, two supplies and two clock domains.
- One 2.5 GHz LC PLL feeds both sides: the TX divides it; the RX makes four phases and rotates them in 6.25 ps steps under digital CDR control.
- The block specification table is the contract: every block has a 5 Gb/s and a 10 Gb/s number and a chapter; † marks estimates awaiting simulation.
- Design for 10, guarantee 5: the same architecture carries 10 Gb/s NRZ over 0.5 m of FR-4 (193 mV worst-case eye) because the equalizer only sees the loss at Nyquist. The work is in the clocks and the timing budget (RJ ≤ 0.85 ps rms), not in the transistors.
- The voltage budget leaves about 76 mV; the timing budget leaves a 141 ps eye with 1.4 ps to spare.
- The test chip adds SPI, a PRBS generator and checker with an error counter, and an internal loopback; TX and RX pads share one die edge, and high-speed bond wires stay at or below about 1 mm.
- Five phases, each with an exit criterion and a tagged release: feasibility, building blocks, the 5/10 Gb/s transceiver, measurement, and beyond.
- First-cut power is about 216 mW (43 pJ/b), five times less energy per bit than OpenSerDes, dominated by static CML current; the SST driver is the first saving.

CHAPTER 15

Design and Verification Flow with Open Tools

From a Python model to a GDS file and a test bench, with nothing but open-source tools: which tool does what, how the model and the circuits keep each other honest, and how to prove the chip works.

IN THIS CHAPTER YOU WILL LEARN

- Name the open tool used at each stage of the Open-SERDES flow and say what it checks.
- Close the loop between the serdespy model and the transistor-level circuits by back-annotating simulated responses.
- Choose a testbench, a measurement and a pass/fail criterion for every block, including corners and Monte Carlo.
- Apply the layout rules that matter at 5 Gb/s: symmetry, current density, metal width, inductors, extraction and decoupling.
- Plan the lab bring-up: test board, loopback, on-chip PRBS checking and the measurements that compare silicon with the model.

The flow in Figure 15.1 runs left to right, but it is not a pipeline. Every stage feeds numbers back to the system model, and the chip is ready for tape-out only when the model, rerun with those numbers, still shows margin at every corner.

15.1 The open toolchain

Every tool in this chapter ships in one Docker image, **IIC-OSIC-TOOLS** [2], together with the IHP SG13G2 PDK [1]; pinning one image tag gives the whole team identical tools. The Open-SERDES repository builds on the JKU **analog mixed-signal chip template** [5] (it needs image 2026.06 or later). Its recursive Makefiles make one `make all` run every testbench including PVT and Monte Carlo, run DRC, LVS and extraction on the analog macros, harden the digital logic and assemble the chip digital-on-top with seal ring, pad frame and fill, run a full-chip LVS without black boxes, and draw the bonding diagram. GitHub Actions rerun it on every commit, so a broken block is noticed the day it breaks.

Figure 15.2 maps the tools onto the flow, and Table 15.1 says what each one is for.

15.2 The model–circuit loop

The labs of Chapter 16 describe each block by an ideal transfer function. The circuits will not match those ideals, and the only way to know whether the difference matters is to put the circuit's real response back into the model (Figure 15.3). Because the link is linear up to the sampler, each simulated block becomes one more factor in the cascade:

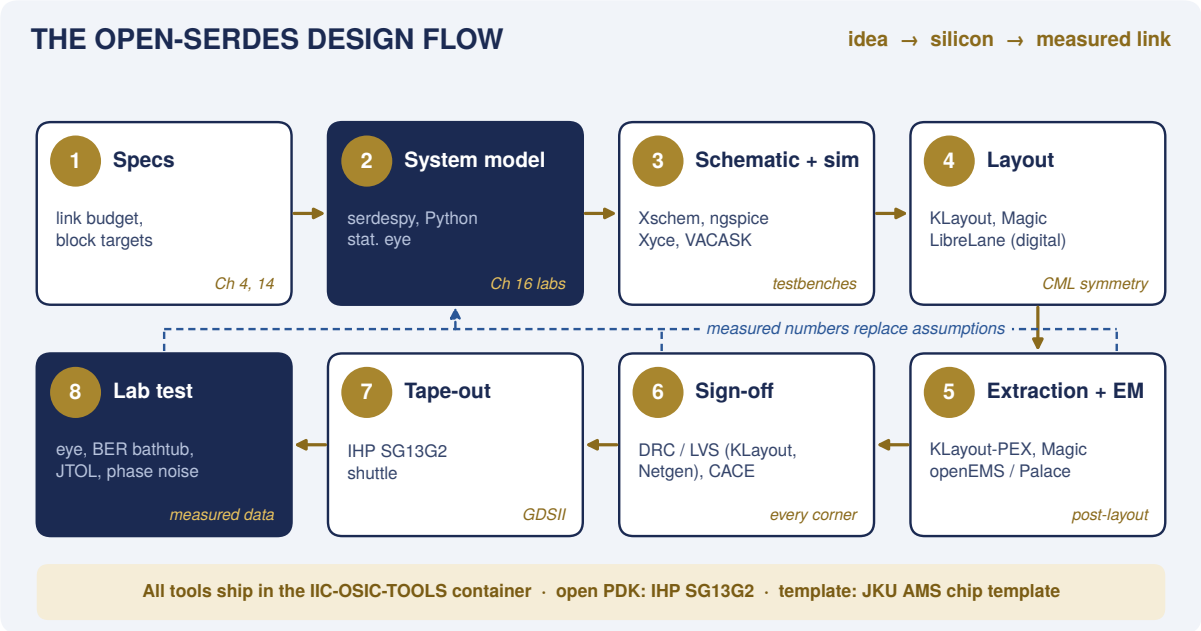


Figure 15.1. The Open-SERDES design flow, end to end. Specifications and the serdespy system model come first; schematics in **Xschem** with **ngspice**, layout in **KLayout**, extraction and EM simulation, and sign-off (DRC, LVS, **CACE** datasheets) follow; then tape-out and lab test. The arrows back to the model are the important ones: each stage returns measured numbers that replace assumptions.

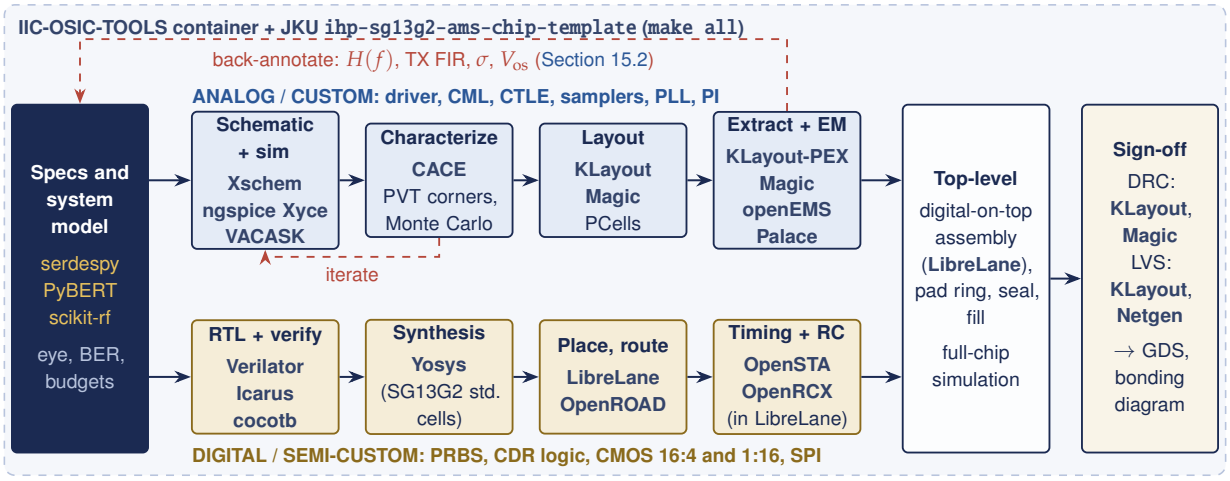


Figure 15.2. Two lanes, one container. The analog lane (HBT CML driver, CTLE, samplers, PLL, PIs) is hand-designed and verified block by block; the digital lane (PRBS, CDR loop filter, CMOS serializer and deserializer stages, SPI) is synthesized by **LibreLane**. Both merge in a digital-on-top assembly and one sign-off. The red dashed arrows are the feedback paths: corners and Monte Carlo send the designer back to the schematic, and extracted responses go back to the system model.

Table 15.1. Open tools in the Open-SERDES flow. All are in the IIC-OSIC-TOOLS image except serdespy, PyBERT and scikit-rf, which install with `pip`.

Tool	Purpose in Open-SERDES	Stage
serdespy [3], **PyBERT** [44]	link models, eyes, BER (Chapter 16); PyBERT adds a GUI and IBIS-AMI	system model
scikit-rf [18]	Touchstone files, mixed-mode S-parameters, de-embedding	model, EM, lab
Xschem [45]	schematics, symbols, testbench netlists for every simulator	schematic
ngspice [46]	`.op`, `.ac`, `.noise`, `.tran`, transient noise, Verilog-A via OSDI	simulation
Xyce [47], **VACASK** [48]	parallel SPICE for large post-layout runs; Verilog-A simulator with PSS and HB	simulation
CACE [49]	YAML datasheet: runs testbenches over corners, checks min/typ/max limits	characterization
KLayout [50], **Magic** [51]	layout editing; IHP DRC and LVS decks; RC extraction (**Magic**)	layout, sign-off
Netgen [52], **KLayout-PEX** [53]	LVS comparison; parasitic extraction for SG13G2	sign-off, extraction
openEMS [54], **Palace** [55]	FDTD and FEM field solvers for inductors, pads and package	EM
Verilator [56], **Icarus** [57], **cocotb** [58]	Verilog simulation and lint; Python testbenches for PRBS, CDR, SPI	RTL
Yosys [59], **LibreLane** [60]	synthesis to SG13G2 cells; RTL-to-GDS flow (successor of OpenLane 2 [61])	digital
OpenROAD [62], **OpenSTA** [63]	placement, clock tree, routing; static timing (inside **LibreLane**)	digital

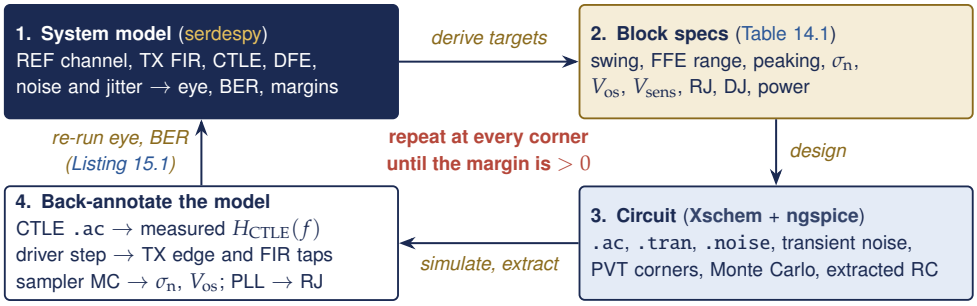


Figure 15.3. The model sets the specs; the circuits return their real responses; the model decides. Each pass replaces one ideal block with a simulated one. The loop ends when the margin is positive at every corner with every block back-annotated.

KEY EQUATION · Back-annotating the link model

$$H_{\text{link}}(f) = H_{\text{TX}}(f)\, H_{\text{ch}}(f)\, H_{\text{CTLE,ckt}}(f), \qquad p_{\text{TX}}(t) = s(t) - s(t - T_{\text{b}}), \qquad \sigma_{\text{n}}^2 = \sum_i \sigma_{n,i}^2. \tag{15.1}$$

Here $s(t)$ is the simulated step response of the driver into its load, so $p_{\text{TX}}(t)$ is its real 1-UI pulse, with finite edges and any ringing; sampling it at T_{b} spacing gives the effective FIR taps. The CTLE enters through its `.ac` response, and each noise source through its rms value from `.noise` or Monte Carlo.

Listing 15.1 loads an **ngspice** AC sweep into the Lab 4 model. In **ngspice**, `wrdata ctle_ac.txt v(outp,outn)` after an `ac dec 100 1meg 100g` writes frequency, real and imaginary parts in three columns. The sweep is resampled onto the model's 10 MHz grid, the end values are held outside it, and the DC point is forced real, as `sdp.freq2impulse` requires.

```
from common import *                    # F grid, channel(), impulse(), ...
```

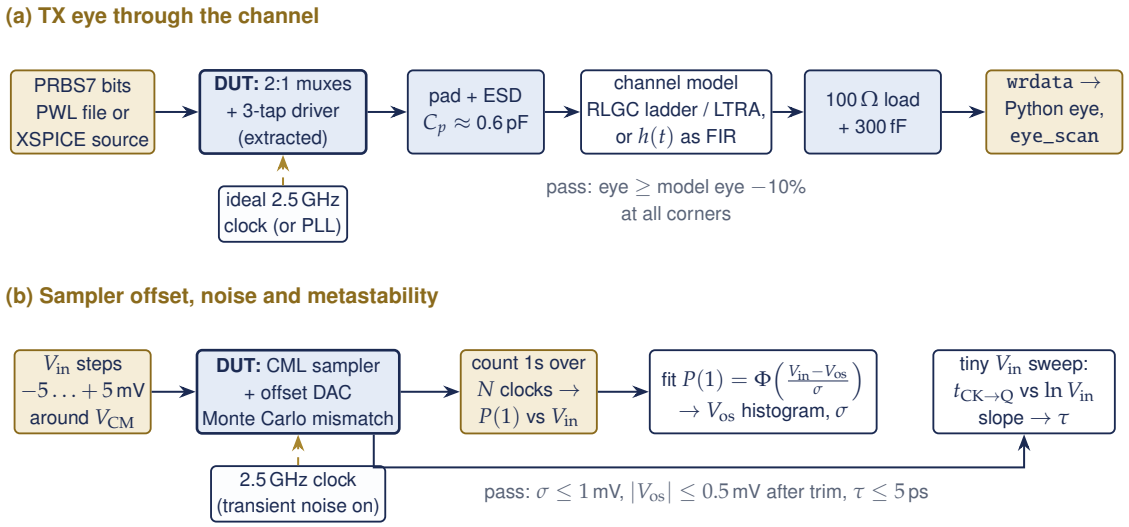


Figure 15.4. Two testbenches that decide whether the link works. (a) The TX eye: PRBS bits drive the extracted serializer and driver into the pad and a channel model; the waveform goes back to Python for the eye measurement. (b) The sampler: a slow input step sweep under Monte Carlo gives $P(1)$ curves whose center is the offset and whose width is the noise, and a sweep of tiny inputs gives the regeneration time constant τ from the slope of $t_{CK\to Q}$ against $\ln V_{in}$ (Section 10.3).

```
f_sp, re, im = np.loadtxt("ctle_ac.txt", usecols=(0, 1, 2), unpack=True)
H_sp = re + 1j * im                          # wrdata ctle_ac.txt v(outp,outn)
mag = np.interp(F, f_sp, np.abs(H_sp))       # resample onto the model grid;
phs = np.interp(F, f_sp, np.unwrap(np.angle(H_sp)))  # holds end values
H_ckt = mag * np.exp(1j * phs)
H_ckt[0] = mag[0]                            # DC must be real
h_link, t = impulse(channel() * H_ckt)       # REF x circuit CTLE
bits = np.tile(sdp.prbs7(1), 40)
y = through(tx_wave(bits), h_link)
report_eye("REF + ngspice CTLE", y, bits)
```

Listing 15.1. Back-annotating a circuit CTLE (tested with a synthetic `wrdata` file).

WORKED EXAMPLE 15.1 · What a slow second pole costs

Check: feed Listing 15.1 the ideal `ctle(8)` response sampled from 1 MHz to 100 GHz. It returns 60.0 mV and 0.56 UI, exactly the Lab 5 pre-DFE eye, so the resampling loses nothing.

A realistic circuit: suppose the extracted CTLE has its output pole at $f_{p2} = 6$ GHz instead of the 10 GHz of the model, because the four samplers load it more than planned. The 12 dB eye drops from 104.0 to 98.0 mV and the 14 dB eye from 113.3 to 107.4 mV, about 6 %. At 5 GHz the loss is 9 %. That is affordable against 76 mV of margin, but it is exactly the kind of number that only the loop reveals.

15.3 Testbenches

Every specification in Table 14.2 needs a testbench that measures it and a pass/fail limit. Table 15.2 lists the key ones and Figure 15.4 sketches two of them.

The channel in SPICE. The REF channel has skin-effect loss that grows as $\sqrt{f}$ and dielectric loss that grows as f (Section 3.3). Neither open simulator models that directly in a transient. **ngspice**

Table 15.2. Key Open-SERDES testbenches. Limits from Table 14.2; every analog row runs at all PVT corners, and the mismatch-sensitive rows also run Monte Carlo.

Block	Testbench	Measure	Pass
TX serializer + driver	`.tran`, PRBS7 into 100 Ω + C_p; then through the channel model	swing, edges, DCD, eye, every V_{CE}	0.8–1.0 Vppd; DCD + DDJ ≤ 8 ps; $V_{CE} < \mathrm{BV_{CEO}}$
Termination, pad	`.ac` or `.sp` with ESD and pad parasitics	S_{11}, C_p	RL ≥ 10 dB at 2.5 GHz
CTLE	`.ac` at every code; `.noise`; large-signal `.tran`	peaking, f_z, f_{p2}, noise, linearity	0–12 dB; $f_{p2} \geq 7.5$ GHz; ≤ 0.25 mV
DFE summer	`.tran` with tap 1 toggling every UI	settling inside the tap-1 loop	$t_{\mathrm{CK}\to\mathrm{Q}} + t_{\mathrm{settle}} + t_{\mathrm{setup}} \leq 1$ UI
Sampler	DC-step sweep counting $P(1)$, Monte Carlo; tiny-input sweep; transient noise	V_{os}, σ, τ	$\sigma \leq 1$ mV, $\tau \leq 5$ ps, trim range ≥ $\pm 3\sigma$
PLL	lock transient; VCO phase noise (transient noise, or PSS/pnoise in **VACASK**); Python loop model	lock time, $\mathcal{L}(f)$, RJ	RJ ≤ 1 ps rms
PPF + PI	`.tran` at all 64 codes	phase vs. code, amplitude	I/Q ≤ 2°, INL ≤ 1°
CDR logic	**cocotb** with a Python phase model driving the early/late inputs	lock, ppm range, dither	±300 ppm; ≤ 20 ps pp
PRBS, SPI	**cocotb** on **Verilator** or **Icarus**; error injection	error count, resync	exact counts

offers the lossless line (`T`), the SPICE3 lossy line LTRA (`O`), and the single and coupled lossy lines TXL and CPL; all of them take constant per-unit-length R, L, G, C. It has no HSPICE-style W-element with frequency-dependent tables and no element that reads a Touchstone file; its `.sp` analysis *computes* S-parameters of a circuit, it does not import them. **Xyce** has TRA, LTRA and a lumped line (YTRANSLINE), and a linear YLIN device that reads Touchstone data, documented mainly for harmonic-balance and other frequency-domain analyses. So there are three practical options: (i) a long ladder of RLGC sections, (ii) a behavioral source that convolves with $h(t)$, or (iii) splitting the problem at the channel.

KEY EQUATION · Sections needed for a lumped line model

$$\Delta\ell \leq \frac{\lambda_{\min}}{10} = \frac{v}{10\, f_{\max}} \quad \Rightarrow \quad N \geq \frac{10\,\ell\, f_{\max}}{v} \tag{15.2}$$

For REF, $\ell = 1$ m, $v = c/\sqrt{4} = 1.5 \times 10^8$ m/s and $f_{\max} = 10$ GHz give $N \geq 667$ sections, each needing extra R–L branches to mimic the skin effect. That is slow and hard to fit. Option (iii) is exact for a matched link and is what Open-SERDES uses: simulate the TX into a 50 Ω-per-side load in SPICE, export the differential waveform, apply the channel in Python with `through(x, h)`, and feed the result back into the RX testbench as a PWL source behind 50 Ω per side.

Corners, Monte Carlo and CACE. The SG13G2 model libraries have typical, best-case and worst-case sections for MOS devices, HBTs and resistors; combine them with supply ±5 % and 0–85 °C. Mismatch-sensitive specs (sampler offset, I/Q phase, mirror ratios) need Monte Carlo, and the

number of runs sets how well you know the answer.

KEY EQUATION · How many Monte Carlo runs

$$\frac{\Delta\hat{\sigma}}{\sigma} \approx \frac{1}{\sqrt{2N}} \tag{15.3}$$

Fifty runs know σ to about 10 %, 200 runs to 5 % (for the sampler, 4.9 ± 0.25 mV). **CACE** turns all of this into a datasheet: a YAML file per block lists each parameter with its minimum, typical and maximum limits, the testbench that measures it and the conditions to sweep; **CACE** runs every combination and reports pass or fail, so Table 14.2 becomes an executable checklist.

15.4 Layout for high speed

At 5 Gb/s a layout can lose the margin that the schematic won. Six habits keep it.

Symmetry. Draw every differential circuit as two mirror-image halves and route each pair as a pair. Place HBT pairs and mirror devices common-centroid (ABBA) with dummies at the ends. A 10 ps skew between FFE tap paths moves the de-emphasis edge by 0.05 UI (Section 9.5); 1 % of clock duty-cycle error costs 4 ps (Section 12.5).

Current density. An HBT reaches its peak f_T at one collector current density. Size each emitter so that the switching transistors sit near that point at their tail current, from the npn13G2 f_T-versus-J_C curves of the PDK; past the peak, f_T collapses.

Metal width. Every DC path needs $w \geq I/J_{\max}$, with the electromigration limit $J_{\max}$ of that layer and temperature from the SG13G2 rules. Route the 20 mA driver tail, its loads and the 2.5 V straps on TopMetal1 and TopMetal2; a few ohms of strap under 77 mA of CML current shift every common-mode level in the chip.

Inductors and EM. Put the 4 nH VCO inductor (and any T-coil) on TopMetal2, away from other metal. IHP's `gds2openEMS` and `gds2palace` scripts take the GDS and a stackup file, run **openEMS** (FDTD) or **Palace** (FEM), and return S-parameters for a π-model or a wideband fit [1]. Match L and Q to the team's measured inductor before trusting the flow.

Extraction and isolation. Extract every analog block with **KLayout-PEX** or **Magic** and resimulate; clock-to-data coupling and the wiring capacitance on the CTLE output are the usual surprises. Surround the CML blocks, the VCO and the samplers with substrate guard rings, away from the switching CMOS logic.

Decoupling. CML draws constant current; CMOS draws its charge in spikes at the clock edges, and that charge must come from on-chip capacitors.

WORKED EXAMPLE 15.2 · Decoupling the deserializer

Switched capacitance: the 1:16 deserializer burns 5 mW at 1.2 V and 312.5 MHz, so $\alpha C_{sw} = P/(V_{DD}^2 f) = 5\,\text{mW}/(1.44 \times 312.5\,\text{MHz}) = 11$ pF is switched every cycle.

Worst case: if all of it switches on one edge, before the package can refill the rail, the droop is $\Delta V/V \approx C_{sw}/(C_{sw} + C_{dec})$. For 5 % droop, $C_{dec} \geq 19\,C_{sw} = 210$ pF: about 0.14 mm^2 of MIM capacitor at roughly 1.5 fF/µm^2 (check the PDK value). Inside the synthesized digital

> macros, **LibreLane** adds decap cells [5]; budget the rest as MIM next to the macro, and count it in the die area.

The digital flow. The PRBS logic, the CDR loop filter, the SPI registers and the slow CMOS serializer and deserializer stages are Verilog, verified with **cocotb** and hardened by **LibreLane**. Synthesize what runs at 625 MHz or slower; place the 1.25 GHz deserializer stage and the CMOS-to-CML converters by hand from standard cells.

Sign-off. DRC with the IHP **KLayout** decks (**Magic** as a second opinion), LVS with **KLayout** or **Netgen**, then antenna and density checks after fill, and the template's full-chip LVS without black boxes [5]: run it whenever a macro changes.

15.4.1 Repository structure

The Open-SERDES repository keeps the recursive layout of the JKU template: every analog block is a macro with its own schematics, testbenches, layout, verification and Makefile, so a block can be built, simulated and verified alone. The listing shows these folders for `tx_serializer`; every other macro has the same ones. The digital half (PRBS, the CMOS serializer and deserializer stages, the CDR loop filter and SPI) is not a macro: it lives in `rtl/` and is hardened with the chip, digital-on-top. Open-SERDES adds four folders of its own: `spec/`, the one specification file that the link model and every testbench read; `model/`; `channels/`; and `lab/`, for the test board and the measured data of Section 15.5. The others (`flow/`, `layout/`, `netlist/`, `ip/`, `packaging/`, `release/` and `doc/`) come with the template.

```
open-serdes/                  # from ihp-sg13g2-ams-chip-template
|-- Makefile                  # make model && make all
|-- spec/                     # one spec file: model and testbenches
|-- model/                    # link model, stat. eye, GUI, labs (Ch. 16)
|-- channels/                 # S-parameters: REF, REF-10, 802.3, boards
|-- macros/                   # analog macros, one per block
|   |-- tx_serializer/        # CML 4:2 and 2:1, CMOS-to-CML
|   |   |-- schematic/  testbenches/  layout/  netlist/
|   |   |-- verification/     # drc/  lvs/  cace/ (limits from spec/)
|   |   |-- scripts/
|   |   |-- final/            # GDS, LEF, lib for the chip assembly
|   |   `-- Makefile          # make all: build, sim, DRC, LVS
|   |-- tx_ffe_driver/        # 3-tap CML driver, termination, ESD, pad
|   |-- rx_ctle/              # termination, ESD, loopback mux, CTLE
|   |-- rx_sampler_dfe/       # 2-tap DFE summer, four samplers
|   |-- pll_lcvco/            # integer-N PLL, LC VCO, dividers
|   |-- cdr_pi/               # I/Q generation, two phase interpolators
|   `-- cml_lib/              # shared HBT CML cells
|-- rtl/                      # PRBS, CMOS SER/DES, CDR loop filter, SPI
|-- flow/                     # LibreLane: rtl/ hardened digital-on-top
|-- verification/             # link regression, eye checks, full-chip DRC/LVS
|-- layout/  netlist/  ip/  packaging/  release/
|-- lab/                      # test board, bring-up scripts, data
`-- doc/                      # documentation (from the template)
```

The template's `make macro FROM=inverter NAME=rx_ctle` creates a new macro from its example. The macro's CACE datasheet reads every limit from `spec/`, so a number changes in one place only.

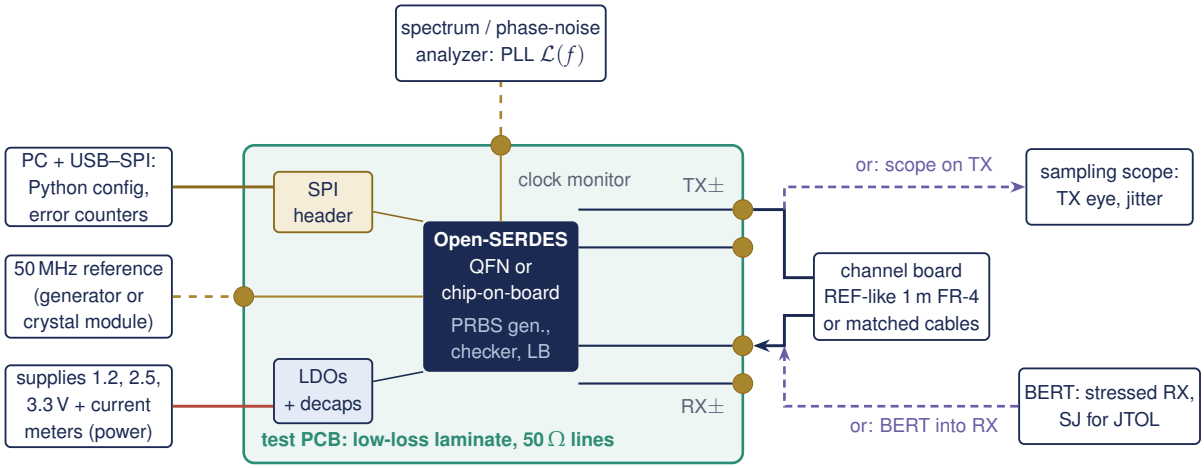


Figure 15.5. The chip tests itself; the instruments check that it tells the truth. The baseline loop needs only a reference generator, supplies, a PC and a channel board: the on-chip PRBS generator drives TX, the channel board brings the signal back to RX, and the on-chip checker counts errors. A sampling scope on TX (eye and jitter), a BERT into RX (stressed eye, sinusoidal jitter for JTOL) and a phase-noise analyzer on the clock monitor are added when available.

15.5 Testing the chip in the lab

Figure 15.5 shows the bring-up bench. The test board carries the chip, SMA connectors for TX±, RX±, the reference clock and a divided clock-monitor output, local regulators and decoupling, and an SPI header. Keep the 50 Ω traces from the pads to the SMAs short and identical within each pair, and de-embed them with **scikit-rf** from a measured test structure.

Bring-up order. (1) Supplies and currents against Table 14.3. (2) SPI read-back of every register. (3) PLL lock, seen on the clock monitor. (4) TX eye with PRBS7 on a scope. (5) Near-end loopback with the on-chip checker: this tests the chip without the board. (6) The external loop through the channel board, sweeping FFE presets and CTLE codes. (7) Freeze the CDR, step the PI code through the override register and count errors at each step: an on-chip bathtub, with the offset DACs giving the vertical scan.

KEY EQUATION · Bits needed to claim a BER

$$N_{\text{bits}} \geq \frac{-\ln(1-\text{CL})}{\text{BER}} \approx \frac{3}{\text{BER}} \quad \text{(zero errors, CL = 95 \%)} \tag{15.4}$$

Claiming 10^{-12} at 95 % confidence takes 3×10^{12} error-free bits, or 600 s at 5 Gb/s: ten minutes per setting, which is why the checker is on chip.

Table 15.3 is the measurement plan. Each row ends in a comparison with a model number, which closes the last loop of the flow: when silicon and model disagree, the model is updated first, and the next design starts from it.

KEY TAKEAWAYS

- One container, IIC-OSIC-TOOLS, holds every tool and the SG13G2 PDK; the JKU template turns the whole flow into `make all` with regression on every commit.
- The system model is not finished when the circuits start: back-annotate the CTLE `.ac` response, the driver step and the noise and offset of every block, and rerun the eye at each corner.

Table 15.3. Measurement plan: each silicon measurement and the model result it is compared with.

Measurement	Setup	Target or comparison
TX eye and jitter	PRBS7, sampling scope at the TX SMAs	0.8–1.0 Vppd; edges, DCD vs. simulation
BER and bathtub	loopback or channel board; on-chip checker; PI-code sweep	BER $\leq 10^{-12}$ (10 min); bathtub vs. Lab 6
Equalizer sweep	FFE presets $\times$ CTLE codes $\times$ DFE on/off	best setting vs. Section 8.5
Jitter tolerance	BERT with sinusoidal jitter into RX	above the mask of Section 13.5
PLL phase noise	clock monitor into a phase-noise analyzer	$\mathcal{L}(1\,\text{MHz})$; RJ ≤ 1 ps rms
Return loss	VNA at the RX SMAs, de-embedded	RL ≥ 10 dB at 2.5 GHz
Power	current per supply rail in each mode	vs. Table 14.3

- Open SPICE engines have only constant-RLGC lossy lines; for REF, simulate the TX into a matched load and apply the channel in Python.
- Run every analog testbench at PVT corners; N Monte Carlo runs know σ to $1/\sqrt{2N}$; let CACE check every limit of the specification table.
- Layout for 5 Gb/s means mirror symmetry, HBTs at their peak-f_T current density, metal sized for the DC current, EM-simulated inductors on TopMetal2, extraction, guard rings and real decoupling area.
- The chip tests itself: PRBS generator, checker and loopback verify 10^{-12} in ten minutes; instruments confirm the eye, jitter tolerance, phase noise and power against the model.

CHAPTER 16

Hands-on Labs with serdespy

Seven short Python scripts reproduce the key numbers of this tutorial, from the closed eye of the bare channel to a bang-bang CDR locking onto real data.

IN THIS CHAPTER YOU WILL LEARN

- Set up serdespy on a current Python and run every lab from `tutorial/code/`.
- Turn the REF channel into a pulse response, cursors and eye, and see why it must be equalized.
- Design a TX FFE, a CTLE and a DFE in the model and measure what each buys.
- Count bit errors with jitter and noise, and compare them with the semi-analytic BER and the bathtub.
- Build a behavioral half-rate bang-bang PI-CDR and watch it track a frequency offset.

Every lab has the same shape: a goal, the few lines of code that matter, the figure and numbers you should get, and questions that push past the script. The figures here are drawn by `figsrc/ch16_labN.py` with exactly the computations of the labs, so your printed numbers must match to the last digit; each lab also writes its own PNGs.

16.1 Lab 0: set-up

Goal. A working serdespy environment, and a map of the labs.

```
python3 -m venv ~/serdes-venv          # tested with Python 3.13
source ~/serdes-venv/bin/activate
pip install serdespy                    # pulls numpy, scipy, matplotlib, scikit-rf
cd tutorial/code
python lab1_channel.py                  # PNGs appear in this folder
```

serdespy 1.0 still uses `np.complex_`, which NumPy 2 removed; `common.py`, imported by every lab, restores it with `np.complex_ = np.complex128` before serdespy is imported. It also defines the simulation grid (64 samples per UI, $\Delta t = 3.125$ ps, 0–160 GHz in 10 MHz steps) and the shared functions: `channel()` for REF, `ctle()`, `impulse()`, `pulse()`, `tx_wave()`, `through()`, the eye measurement `eye_scan()` (serdespy has none) and the Gaussian BER estimate `ber_q()`. Table 16.1 maps the labs to the chapters.

! serdespy quirks that cost an afternoon each

- `sdp.Transmitter(data, levels, f)`: `f` is the *Nyquist* frequency (pass 2.5e9 for 5 Gb/s), whatever the docstring says.
- Call `Transmitter.FIR()` before `oversample()`, with taps `[pre, main, post]` or `[main, post]`.
- `sdp.channel_coefficients()` always plots, at 1200 dpi by default: pass `res=150`.
- `sdp.forcing_ffe()` sets the main tap to 1: rescale to $\sum|c_k| = 1$ yourself.
- `Receiver.noise()` discards any FFE or DFE already applied: add noise first, then equal-

Table 16.1. The labs. Run times on a laptop; every script is self-contained apart from `common.py`.

Lab	Script	What it shows (key number)	Theory	Time
1	`lab1_channel.py`	REF S_{21}, $h(t)$, $p(t)$, cursors (-17.38 dB; PDA -0.161 V)	Sections 2.4 and 3.7	3 s
2	`lab2_prbs_eye.py`	PRBS7 eye before and after REF (-77.8 mV)	Section 2.6	2 s
3	`lab3_tx_ffe.py`	zero-forcing TX FFE (2-tap 162.7 mV, 3-tap 144.8 mV)	Chapter 6	3 s
4	`lab4_ctle.py`	CTLE peaking sweep (113.3 mV at 14 dB)	Chapter 7	4 s
5	`lab5_dfe.py`	DFE taps and BER (3.2×10^{-14} at 10 mV)	Chapter 8	3 s
6	`lab6_jitter_ber.py`	counted vs. estimated BER, bathtub (0.45 UI)	Chapter 11	21 s
7	`lab7_bbcdr.py`	bang-bang PI-CDR (lock in 0.13 µs, 0 errors)	Chapter 13	7 s

ize. Use `Receiver.FFE()`, not `FFE_BR()`, which misaligns the data.

- serdespy has no CDR and no MLSE, whatever the README lists; Lab 7 builds the CDR.

16.2 Lab 1: the channel

Goal. Build REF from serdespy's ABCD-matrix functions, plot $|S_{21}|$, and turn it into the impulse response, the pulse response and its cursors (Sections 2.4 and 3.7).

```
from common import *

H = channel()                                   # complex S21 on the grid F
h, t = impulse(H)                               # h is per-sample: sum(h) = H(0)
p = pulse(h)
n_pre, n_post = 2, 10
cur = sdp.channel_coefficients(p, t, SPS, n_pre, n_post, res=150,
                               title="Pulse response and cursors")
isi = np.sum(np.abs(cur)) - cur[n_pre]
print(f"Peak-distortion eye (1 V swing, {n_pre}+{n_post} cursors) = "
      f"{cur[n_pre] - isi:+.4f} V  (negative = closed)")
```

What you should see. Figure 16.1. The loss is -0.59 dB at DC, -10.40 dB at 1.25 GHz, -17.38 **dB at 2.5 GHz** and -30.07 dB at 5 GHz. The cursors are $p_{-1} = 0.0486$, $p_0 = 0.3396$, $p_1 = 0.1817$, $p_2 = 0.0909$, $p_3 = 0.0537$ V, with $p_1/p_0 = 0.535$. The ISI magnitudes add to 0.5008 V, more than the main cursor, so the peak-distortion eye is -0.161 **V**: closed.

Questions and extensions. (1) Call `channel(LOSS_FAMILY[10])` and `channel(LOSS_FAMILY[25])` (0.572 and 1.441 m). How do p_1/p_0 and the PDA eye change with loss? (2) Set `c_pad=1e-12`. Where in the tail do the pad reflections appear, and why can a short DFE not cancel them?

16.3 Lab 2: a PRBS eye through the channel

Goal. Send PRBS7 NRZ through REF and measure the eye at both ends (Sections 2.2 and 2.6).

```
prbs = sdp.prbs7(1)                         # one period: 127 bits, seed = 1
bits = np.tile(prbs, 40)                    # 5080 bits = 1.016 us of data
tx = sdp.Transmitter(bits, LEVELS, F_NYQ)  # NB: 3rd argument = Nyquist frequency
```

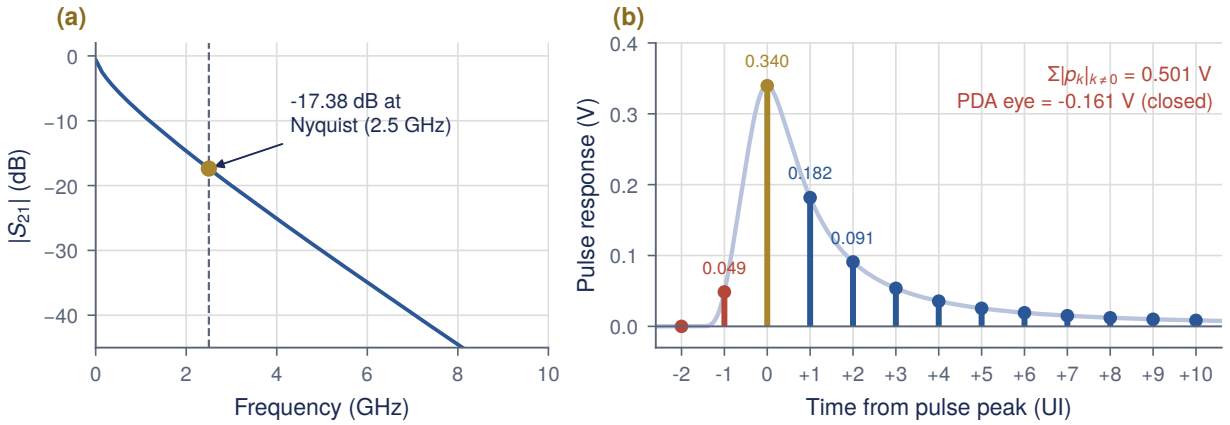


Figure 16.1. Lab 1: a third of the pulse arrives in its own UI, and the tail outweighs it. (a) $|S_{21}|$ of REF; the dot is the 2.5 GHz Nyquist loss. (b) The 1-UI, 1 V pulse response (pale) and its cursors at 1 UI spacing from the peak: pre-cursor red, main gold, post-cursors blue.

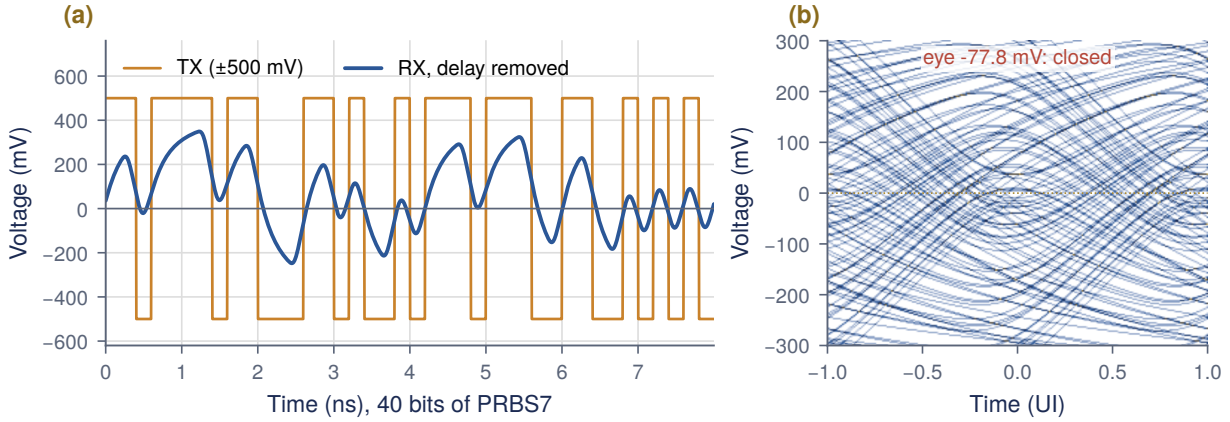


Figure 16.2. Lab 2: the levels survive, the eye does not. (a) 40 bits of PRBS7 at the TX (amber) and after REF with the 6.7 ns delay removed (blue): single bits barely cross zero. (b) Density eye at the channel output, centerd on the best phase: the inner eye is closed by 77.8 mV.

```
tx.oversample(SPS)
x = tx.signal_ideal
H = channel()
h, t = impulse(H)
y = through(x, h)
hts_tx, ph_tx, _, _ = report_eye("TX output", x, bits)
hts_rx, ph_rx, S, A = report_eye("Channel output", y, bits)
save_eye(y, ph_rx, "Eye after 1 m channel (17.4 dB @ 2.5 GHz)", "lab2_eye_rx.png")
```

What you should see. Figure 16.2. The TX eye is 1000 mV and 1.00 UI. After REF the eye is -77.8 **mV** and 0 UI wide: closed at every phase. The mean levels are $+169.8$ and -165.3 mV, at a best phase of 6/64 UI, and the channel delay is 6.722 ns.

Questions and extensions. (1) Repeat with `sdp.prbs13(1)` (tile it 2 times). Section 4.2 quotes -181 mV: why is PRBS7 optimistic? (2) Count the longest run of identical bits in PRBS7 and in PRBS13. Which cursor sum does each run length expose?

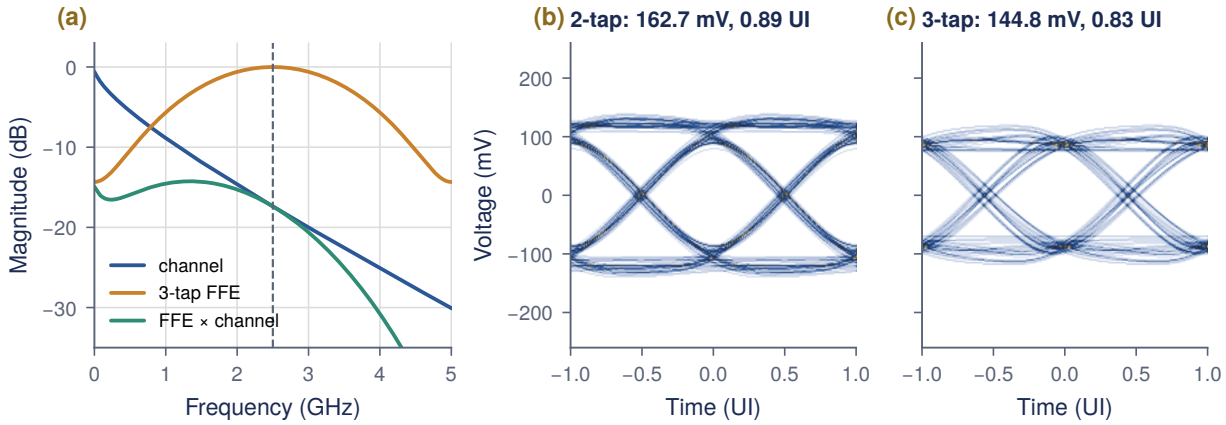


Figure 16.3. Lab 3: zero-forcing flattens the channel to Nyquist, but the pre-tap costs more than it gives. (a) REF (blue), the 3-tap FFE (amber, 0 dB at Nyquist and −14.3 dB at DC) and their product (teal), flat to about 2.5 GHz. (b), (c) PRBS7 eyes with the 2-tap and the 3-tap FFE.

16.4 Lab 3: TX FFE

Goal. Compute zero-forcing 2-tap and 3-tap TX FFEs from the cursors, impose the peak-swing constraint, and measure the eyes (Chapter 6).

```
cur = sdp.channel_coefficients(p, t, SPS, 1, 1, res=150)
c3 = sdp.forcing_ffe(1, cur)              # [c-1, c0, c+1], c0 = 1
c3 = c3 / np.sum(np.abs(c3))              # peak-swing constraint sum|c| = 1
c2 = sdp.forcing_ffe(0, cur[1:])          # [c0, c+1]: classic de-emphasis
c2 = c2 / np.sum(np.abs(c2))
# ...
bits = np.tile(sdp.prbs7(1), 40)
for name, c, png in ((None, None, None), ("2-tap", c2, None),
                     ("3-tap", c3, "lab3_eye_ffe3.png")):
    y = through(tx_wave(bits, fir=c), h)
    hts, ph, S, A = report_eye(f"Channel + {name or 'no'} FFE", y, bits)
```

What you should see. Figure 16.3. The 3-tap taps are $[-0.0853, 0.5959, -0.3188]$ (DC gain 0.192, 14.34 dB of Nyquist boost), the 2-tap taps $[0.6515, -0.3485]$ (10.37 dB). The main cursor drops from 0.3396 to 0.1749 V, the price of equalizing at the TX. The eyes are **162.7 mV, 0.89 UI with 2 taps** and **144.8 mV, 0.83 UI with 3 taps**.

Questions and extensions. (1) Zero-forcing does not maximize the eye. Hold c_1 and sweep c_{-1} from 0 to −0.1, renormalizing each time: where is the best eye, and why does the pre-tap lose under $\sum|c_k| = 1$? (2) Quantize the taps to multiples of 1/32 (the 5-bit driver of Section 9.5). How much eye is lost?

16.5 Lab 4: CTLE

Goal. Sweep the peaking of a one-zero, two-pole CTLE, find the best eye, and translate it into a degenerated differential pair (Section 7.2).

```
FP1, FP2 = 2.5e9, 10e9                    # pole 1 at Nyquist, output pole 10 GHz
H = channel()
bits = np.tile(sdp.prbs7(1), 40)
```

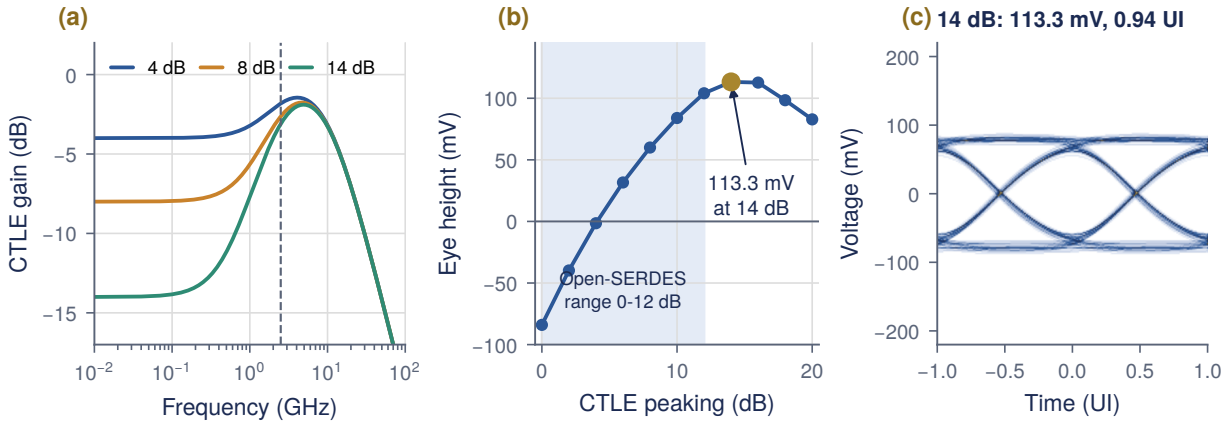


Figure 16.4. Lab 4: more peaking opens the eye until the CTLE starts making ISI of its own. (a) CTLE responses for 4, 8 and 14 dB, all 0 dB at high frequency. (b) Eye height against peaking; the shaded band is the 0–12 dB range of the Open-SERDES CTLE. (c) The eye at the 14 dB optimum.

```
x = tx_wave(bits)
peaks, eh, ew = np.arange(0, 21, 2), [], []
for pk in peaks:
    h_link, t = impulse(H * ctle(pk, FP1, FP2))   # channel x CTLE
    hts, ph, S, A = eye_scan(through(x, h_link), bits)
    eh.append(hts[ph]); ew.append(np.sum(hts > 0) / SPS)
best = peaks[int(np.argmax(eh))]
# ...
gm, a = 20e-3, 10**(best / 20)            # example: gm = 20 mS per side
RS = 2 * (a - 1) / gm                     # A = 1 + gm*RS/2
CS = 1 / (RS * 2 * np.pi * FP1 / a)       # wz = 1/(RS*CS)
RD, CL = 1 / gm, 1 / (2 * np.pi * FP2 / gm)   # gm*RD = 1, wp2 = 1/(RD*CL)
```

What you should see. Figure 16.4. The eye opens at 6 dB (31.6 mV), reaches 104.0 mV at 12 dB, peaks at **113.3 mV, 0.94 UI at 14 dB** and falls again beyond. With 14 dB nominal, the real peak-to-DC gain is only 12.11 dB because the second pole eats some of it; the CTLE is -14.00 dB at DC and -3.10 dB at Nyquist. The circuit values are $R_S = 401\,\Omega$, $C_S = 0.795$ pF, $R_D = 50\,\Omega$ and $C_L = 318$ fF.

Questions and extensions. (1) The optimum lies outside the Open-SERDES range. What does the DFE of Lab 5 do with the 8 dB eye that the CTLE alone cannot? (2) Move `FP2` to 6 GHz and re-sweep; compare with Worked Example 15.1.

16.6 Lab 5: DFE

Goal. Take the post-cursors left by a moderate 8 dB CTLE as DFE taps, measure the eye against the number of taps, and estimate the BER (Chapter 8).

```
PEAK = 8                                  # CTLE peaking [dB] (lab 4 model)
h_link, t = impulse(H * ctle(PEAK))       # channel x CTLE
y = through(tx_wave(bits), h_link)        # CTLE output waveform
cur = sdp.channel_coefficients(pulse(h_link), t, SPS, 1, 8, res=150)
h_pre, h0, post = cur[0], cur[1], cur[2:]
for n in (0, 1, 2, 3, 4, 5, 8):
    rx = sdp.Receiver(y, SPS, F_NYQ, LEVELS, shift=True)  # centre eye at 0
    if n:
```

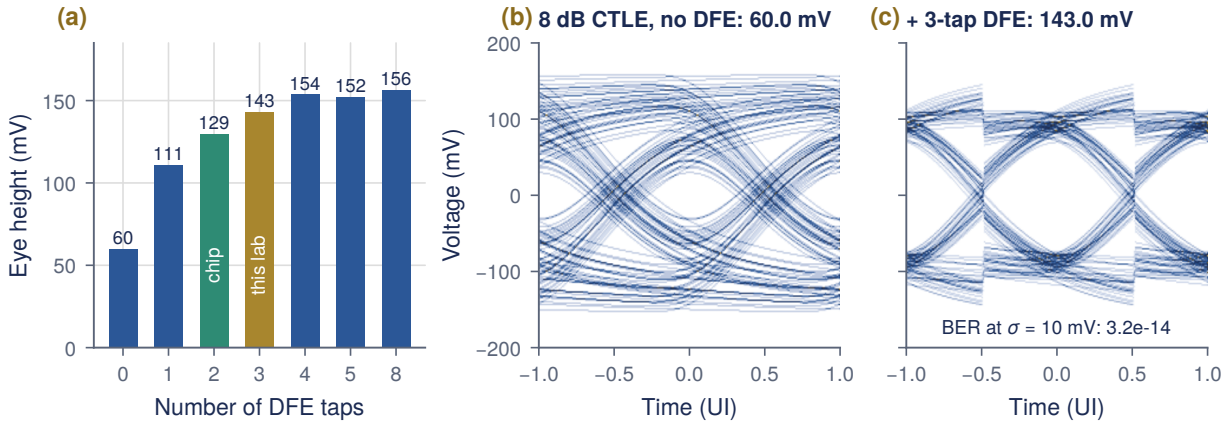


Figure 16.5. Lab 5: the first DFE tap is worth more than the next five together. (a) Eye height against the number of DFE taps after an 8 dB CTLE; teal is the Open-SERDES choice, gold the lab's. (b), (c) The eye before and after a 3-tap DFE. The steps at ± 0.5 UI in (c) are an artifact of serdespy's per-UI feedback; only the sampling instant is meaningful.

```
        rx.nrz_DFE(post[:n])
    report_eye(f"CTLE {PEAK} dB + {n}-tap DFE", rx.signal, bits)
# ... final design: 3 taps, then ber_q(S[ph], A[ph], sigma) for 5..20 mV
```

What you should see. Figure 16.5. With the 8 dB CTLE, $p_{-1} = 0.0138$, $p_0 = 0.1832$ and p_1–$p_3 = 0.0552, 0.0235, 0.0160$ V. The eye grows from 60.0 mV with no DFE to 110.7, 129.5 and **143.0 mV** with 1, 2 and 3 taps, then saturates near 155 mV. With 3 taps the feedback is ± 27.6, ± 11.8 and ± 8.0 mV, and the estimated BER is 5.9×10^{-46}, 3.2×10^{-14}, 6.0×10^{-8} and 1.8×10^{-5} at $\sigma = 5$, 10, 15 and 20 mV. A noise-free run through `sdp.prbs_checker` gives 0 errors in 5024 bits. The lab uses 3 taps; the chip has 2, which gives 129.5 mV here.

Questions and extensions. (1) Reproduce the Open-SERDES operating point of Section 8.5: TX taps $[0.75, -0.25]$ (-6 dB), a 0 dB CTLE and 2 DFE taps. You should get 171.8 mV. (2) The worst-case Q at 10 mV is $143.0/2/10 = 7.15$, yet the BER is 3.2×10^{-14}, not $Q(7.15)$. Why is `ber_q` more optimistic than the worst case?

16.7 Lab 6: jitter, noise and BER

Goal. Add 5 ps rms TX random jitter and RX noise to the Lab 5 link, count real bit errors, and compare them with the estimate and the bathtub (Section 11.3).

```
np.random.seed(2026)                        # repeatable jitter and noise
PEAK, N_TAPS, RJ_UI = 8, 3, 0.025           # RJ = 2.5 % UI = 5 ps rms
bits = np.tile(sdp.prbs7(1), 800)           # 101 600 bits
y = through(tx_wave(bits, jitter_ui=RJ_UI), h_link)   # TX RJ -> channel -> CTLE
rx = sdp.Receiver(y, SPS, F_NYQ, LEVELS, shift=True)  # sampling phase -> index 0
rx.nrz_DFE(taps)
hts, ph, S, A = report_eye("With 5 ps TX RJ, no noise", rx.signal, bits)
for sigma in (10e-3, 15e-3, 20e-3, 25e-3):
    rx.noise(sigma)                         # resets to pre-DFE signal + AWGN
    rx.nrz_DFE(taps)                        # real decisions: error propagation
    d = sdp.nrz_a2d(rx.signal, SPS, 0.0)[skip:-5]
    n_err = sdp.prbs_checker(7, sdp.prbs7(1), d)[0]
    est = ber_q(S[0], A[0], sigma)
```

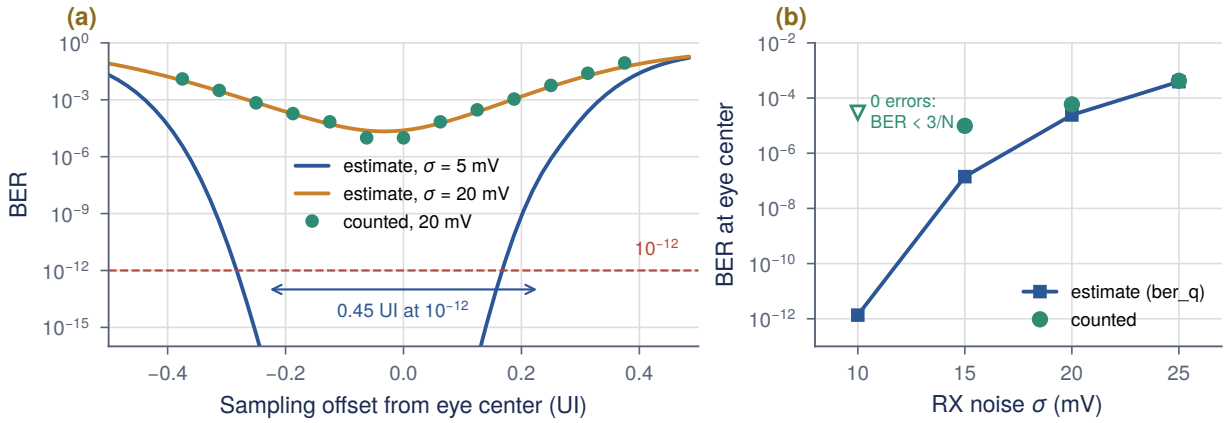


Figure 16.6. Lab 6: counted errors confirm the estimate where counting is possible; the estimate takes over below 10^{-5}. (a) Bathtubs from `ber_q` for 5 and 20 mV of noise, with error counts at 20 mV (dots) at offsets up to ± 0.375 UI. (b) BER at the eye center against noise. The open triangle at 10 mV is the 95 % bound $3/N$ of Equation (15.4) for zero errors in N bits.

```
# ... bathtub: ber_q(S[o % SPS], A[o % SPS], sigma) at every phase offset o
```

What you should see. Figure 16.6. With jitter but no noise the eye is 115.6 mV and 0.73 UI. Counting over 101 544 bits:

σ (mV)	10	15	20	25
errors counted	0	1	6	43
counted BER	$< 3 \times 10^{-5}$	9.9×10^{-6}	5.9×10^{-5}	4.2×10^{-4}
estimated BER	1.4×10^{-12}	1.4×10^{-7}	2.4×10^{-5}	3.9×10^{-4}

At 25 mV count and estimate agree within 10 %; the single 15 mV error is a true rare event (a 63.6 mV margin hit by 4.25σ of noise). With $\sigma = 5$ mV the bathtub is open **0.45 UI at** 10^{-12} (0.66 UI at 10^{-5}); at 20 mV it never reaches 10^{-5}, and the counted points fall on the estimate.

Questions and extensions. (1) How many bits would you have to simulate to see one error at $\sigma = 10$ mV? How long would that take? (2) Raise the RJ to 6 ps rms (`RJ_UI = 0.03`). How much does the 10^{-12} opening at 5 mV shrink? Compare with the 14σ rule of Equation (11.3).

16.8 Lab 7: a bang-bang PI-CDR

Goal. serdespy has no CDR, so `lab7_bbcdr.py` builds one on the Lab 4 link (REF + 14 dB CTLE, 5 ps RJ, 5 mV noise): a half-rate Alexander detector, a P+I loop filter at 625 MHz and a 64-step PI, with the RX clock 200 ppm fast and starting 74 ps off the eye center, as in Figure 13.9. It samples the real waveform where the PI says: per UI one data sample D and one edge sample E half a UI earlier, as the four half-rate samplers deliver. A transition with E equal to the previous D means the clock is early (the loop body below is excerpted; the script also counts errors and has a fixed-clock mode).

```
for k in range(n_upd):
    n = k * UPD + np.arange(UPD)                    # RX bit indices of this update
    # ... phi_d = data phase: phi0 + ppm offset + optional sinusoidal jitter
```

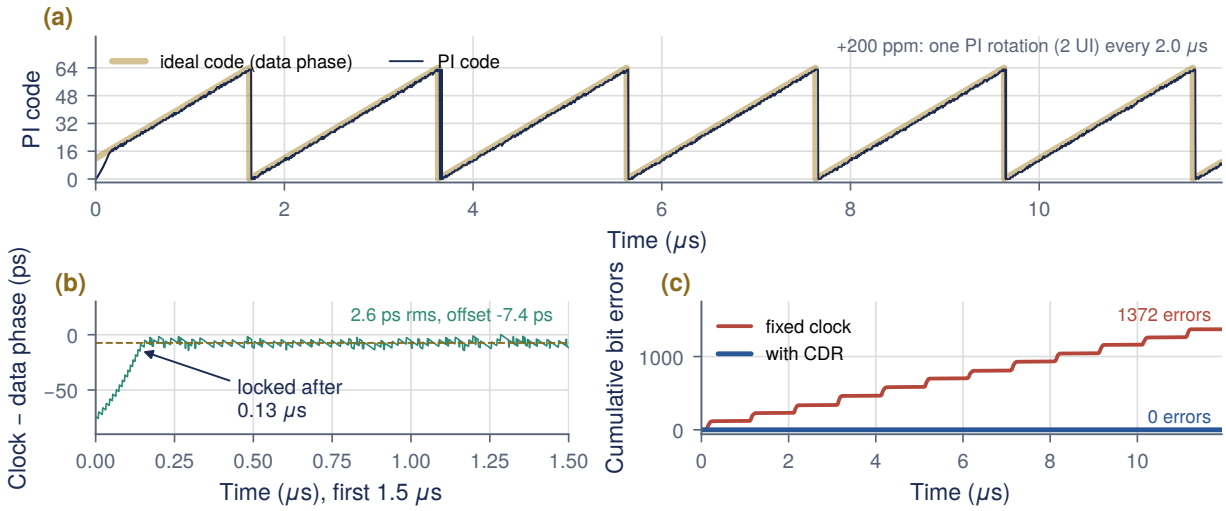


Figure 16.7. Lab 7: the PI rotates continuously to follow a 200 ppm offset, and the errors stop. (a) PI code (navy) over the code the data phase requires (gold); it wraps every 2.0 µs. (b) Clock-minus-data phase: the 74 ps initial error is pulled in by 0.13 µs, then the loop dithers around a −7.4 ps lock point. (c) Cumulative errors: fixed clock (red, one burst per UI of drift) and CDR (blue, none).

```
    phi_c = np.round(acc) * LSB                          # PI phase (UI)
    t_d = t0 + SPS * (n - phi_d + phi_c)                 # data-sampler instants
    d = sample(y, t_d) + sigma * rng.standard_normal(UPD) > 0
    e = sample(y, t_d - SPS / 2) + sigma * rng.standard_normal(UPD) > 0
    prev = np.concatenate(([d[0] if d_prev is None else d_prev], d[:-1]))
    trans = prev != d                                    # a transition before each bit
    votes = np.where(e == prev, 1, -1) * trans           # +1: clock early -> later
    pipe.append(np.sign(votes.sum()))                    # majority vote of the block
    v = pipe.pop(0)                                      # ... seen d_lat updates later
    if cdr:
        freg += ki * v                                   # integral path (frequency)
        acc += kp * v + freg                             # proportional path + phase
```

What you should see. Figure 16.7, with the baseline of Section 13.6 ($K_P = 1/8$ LSB, $K_P/K_I = 128$, one update of extra latency). The loop locks in **0.13 µs** (the phase-only model of Figure 13.9, with a stricter lock test, reports 0.19 µs); the integral register settles at 199 ppm against the true 200; the tracking error is **2.57 ps rms** and 17.3 ps peak-to-peak, around a static offset of −7.4 ps from the maximum-height eye center. There are **0 bit errors** in 59 680 bits with the CDR, against **1372** (BER 2.3×10^{-2}) with a fixed clock, which slides across an eye edge once per microsecond. Raising K_P to 1/2 and 1 LSB raises the wander to 5.13 and 9.90 ps rms (31.1 and 57.5 ps pp): the trade-off of Worked Example 13.2.

Questions and extensions. (1) Sweep K_P from 1/16 to 1 LSB and the latency `d_lat` from 0 to 4. Plot the rms and peak-to-peak wander and compare with dither $\approx 2S\,T_{\text{lat}}$ (Section 13.4). (2) Set `ki=0` and raise `ppm` to 300, 450 and 550. Where does the proportional-only loop lose lock, and does it match the 488 ppm of Section 13.6? Watch the static phase offset grow with the frequency offset before it fails. (3) Add sinusoidal jitter with `sj_pp` and `sj_f`. Find the largest amplitude at 1 and 4.9 MHz with zero errors, and plot your JTOL points on Figure 13.6. Why does the loop lock 7 ps before the eye center?

16.9 Where to go next

Build the GUI demo. The proposal lists a GUI front-end for the link model among its deliverables, in the spirit of the Julia SerDes framework (JLSD) [64] it cites. The labs are the engine: put sliders for loss, taps, peaking and noise in front of `channel()`, the FFE, `ctle()`, the DFE and `lab7_bbcdr.simulate()`, and redraw the eye and bathtub on every change. Cross-check the eyes against PyBERT [44], a mature open link simulator with IBIS-AMI support.

Use a measured channel. Load the test board's 4-port Touchstone file. `sdp.four_port_to_diff` assumes ports 1→2 and 3→4 and returns $V_{\text{out}}/V_{\text{s}}$, half of S_{21} for a matched link; the data must start near DC (**scikit-rf** can extrapolate).

```
import skrf as rf
ntw = rf.Network("board.s4p")                # measured channel, ports 1->2, 3->4
Hd, fd, hd, td = sdp.four_port_to_diff(ntw, None, 50, 50, t_d=DT)
h = 2 * hd                                   # S21-like: same scale as impulse(channel())
```

On a synthetic 4-port built from REF this returns exactly -17.38 dB and $p_0 = 0.3396$ V; Labs 2–7 then run unchanged. Next: crosstalk, adaptation with `sdp.lms_equalizer`, and the statistical eye of Section 4.3.

KEY TAKEAWAYS

- serdespy 1.0 runs on NumPy 2 with one shim; remember its quirks: Nyquist frequency in `Transmitter`, FIR before oversampling, `res=150`, renormalized taps, noise before equalization.
- REF loses 17.38 dB at Nyquist; its PDA eye is -0.161 V and its PRBS7 eye -77.8 mV.
- Under the peak-swing constraint the 2-tap FFE (162.7 mV) beats the zero-forcing 3-tap (144.8 mV); a CTLE alone peaks at 113.3 mV at 14 dB; 8 dB plus a 2- or 3-tap DFE gives 129.5 or 143.0 mV.
- Counted errors confirm `ber_q` where counting is possible; below 10^{-5} only the estimate speaks.
- A bang-bang PI-CDR locks to 200 ppm in 0.13 µs with 2.6 ps rms wander and no errors (a fixed clock: 1372).

APPENDIX A

Formula Sheet

The equations of this tutorial on two pages, in the notation of Chapter 1: $T_\mathrm{b} = 1\,\mathrm{UI} = 200\,\mathrm{ps}$, $f_\mathrm{N} = 2.5\,\mathrm{GHz}$, symbols $a_k = \pm 1$, TX levels $\pm A$ with $A = 0.5\,\mathrm{V}$, cursors p_k of a 1 V, 1-UI pulse.

Formula	Meaning
Signals and BER (Chapters 2 and 4)	
$f_\mathrm{b} = 1/T_\mathrm{b}, \quad f_\mathrm{N} = 1/(2T_\mathrm{b})$	bit rate and Nyquist frequency: 5 Gb/s, 2.5 GHz
$y(t) = A\sum_n a_n\, p(t - nT_\mathrm{b}) + n(t)$	received signal: superposition of shifted pulse responses
$p_k = p(t_\mathrm{s} + kT_\mathrm{b})$	cursors: p_0 main, $k<0$ pre-, $k>0$ post-cursors
$\mathrm{EH} = 2A\left(p_0 - \sum_{k\neq 0}\lvert p_k\rvert\right)$	peak-distortion (worst-case) inner eye height
$\mathrm{BER} = Q(\mathrm{EH}/2\sigma_\mathrm{n}),\ Q(x) = \frac{1}{2}\,\mathrm{erfc}(x/\sqrt{2})$	BER for Gaussian noise of rms σ_n at the slicer
$\mathrm{EH} \geq 2Q\sigma_\mathrm{n} = 14.07\,\sigma_\mathrm{n}$	eye needed for 10^{-12} ($Q = 7.03$)
$S(f) = A^2 T_\mathrm{b}\,\mathrm{sinc}^2(fT_\mathrm{b})$	NRZ power spectral density; first null at f_b
$M = \mathrm{EH}_\mathrm{ISI} - 2Q\sigma_\mathrm{n} - \Delta V_\mathrm{jit} - 2V_\mathrm{os} - V_\mathrm{sens}$	linear voltage margin at the sampler
Channel (Chapter 3)	
$v = c/\sqrt{\varepsilon_r}, \quad t_d = \ell/v$	propagation speed and delay; FR-4: 6.67 ps/mm
$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0}, \quad \mathrm{RL} = -20\log_{10}\lvert S_{11}\rvert$	reflection coefficient and return loss
$\alpha_D \approx \pi f\sqrt{\varepsilon_r}\tan\delta/c$	dielectric loss (Np/m), grows as f; skin loss grows as $\sqrt{f}$
$\lvert S_{11}\rvert = x/\sqrt{4+x^2},\ x = \omega Z_0 C_p$	pad capacitance on a doubly terminated node
$f_\mathrm{3dB} = 1/(\pi Z_0 C_p)$	pad pole (node sees $Z_0/2$)
$\mathrm{IL}(f_\mathrm{N}) - \mathrm{IL}(f_\mathrm{N}/2) > 9.54\,\mathrm{dB}$	PAM4 can win over NRZ only above this octave loss
Equalization (Chapters 6 to 8)	
$y_n = \sum_k c_k x_{n-k}, \quad \sum_k \lvert c_k\rvert = 1$	TX FFE under the peak-swing constraint
$\mathrm{DE} = 20\log_{10}\frac{\lvert c_0\rvert+\lvert c_1\rvert}{\lvert c_0\rvert-\lvert c_1\rvert}$	de-emphasis of a 2-tap FFE ($c_1 = -0.25$: 6 dB)
$H_\mathrm{CTLE}(s) = \frac{1}{A}\frac{1+s/\omega_z}{(1+s/\omega_{p1})(1+s/\omega_{p2})}$	CTLE model, peaking $A = \omega_{p1}/\omega_z$
$A = 1 + g_\mathrm{m}R_S/2,\ \omega_z = \frac{1}{R_S C_S},\ \omega_{p2} = \frac{1}{R_D C_\mathrm{L}}$	degenerated differential pair
$z_n = y_n - \sum_k w_k \hat{d}_{n-k}, \quad w_k = A\,p_k$	DFE; zero-forcing taps equal the post-cursors
$t_{\mathrm{CK}\to\mathrm{Q}} + t_\mathrm{settle} + t_\mathrm{setup} \leq T_\mathrm{b}$	DFE tap-1 timing (direct feedback)
$w_k \leftarrow w_k + \mu\ \mathrm{sgn}(e_n)\ \mathrm{sgn}(\hat{d}_{n-k})$	sign-sign LMS adaptation

Formula	Meaning
Circuits (Chapters 9 and 10)	
$\Delta V = I_T R_\mathrm{L},\ \tau = R_\mathrm{L} C_\mathrm{L},\ P = V_\mathrm{CC} I_T$	CML gate: swing, time constant, static power
$V_{d,\mathrm{pp}} = I_T Z_0$ (CML), $I = V_{d,\mathrm{pp}}/4Z_0$ (SST)	driver current, doubly terminated
$g_\mathrm{m} = I_C/V_T$; full switching at $\approx 4V_T$	HBT pair: 38.7 mS/mA, about 100 mV
$\tau = C/g_\mathrm{m}$ (StrongARM), $\tau = \frac{R_\mathrm{L} C}{g_\mathrm{m} R_\mathrm{L} - 1}$ (CML)	latch regeneration time constant
$P_\mathrm{meta} = \frac{2V_L}{A\,\Delta V}\, e^{-T/\tau}$	metastability probability per decision
$\sigma_{\Delta V_t} = A_{VT}/\sqrt{WL}$	Pelgrom mismatch (offset)
$V_\mathrm{sens} \approx 2Q\sigma_s + V_\mathrm{min} + 2\lvert V_\mathrm{os,res}\rvert$	sampler sensitivity
$\overline{v_n^2} = 4kTR\,\Delta f,\quad v_{n,\mathrm{rms}} = \sqrt{kT/C}$	resistor and sampled (kT/C) noise
Jitter, PLL and CDR (Chapters 11 to 13)	
$\mathrm{TJ} = \mathrm{DJ}_{\delta\delta} + 2Q\,\sigma_\mathrm{RJ}$	total jitter at a BER (dual-Dirac)
$\sigma_\phi^2 = 2\int \mathcal{L}(f)\,df,\quad \sigma_t = \sigma_\phi/(2\pi f_0)$	phase noise to rms jitter
$\omega_n = \sqrt{\frac{I_\mathrm{CP} K_\mathrm{VCO}}{2\pi N C_1}},\ \zeta = \frac{R}{2}\sqrt{\frac{I_\mathrm{CP} K_\mathrm{VCO} C_1}{2\pi N}}$	CP-PLL natural frequency and damping
$f_0 = \frac{1}{2\pi\sqrt{LC}},\ R_p = Q\omega_0 L,\ g_\mathrm{m} R_p/2 \geq 1$	LC tank and start-up
$K_\mathrm{BB} = \mathrm{TD}\sqrt{2/\pi}/\sigma_\mathrm{j}$	bang-bang gain, linearized by jitter
$S = K_P \cdot \mathrm{LSB}/T_\mathrm{upd},\ \mathrm{JTOL} \approx \mathrm{TM} + 2S/\omega$	BB-CDR slew rate and slew-limited JTOL
$\Delta t_\mathrm{PI} = T_\mathrm{clk}/N_\mathrm{steps},\ \mathrm{dither} \approx 2S\,T_\mathrm{lat}$	PI step (6.25 ps) and limit-cycle dither
Flow and test (Chapters 14 and 15)	
$E_b = P/f_\mathrm{b}$	energy per bit (216 mW → 43 pJ/b)
$N_\mathrm{bits} \geq 3/\mathrm{BER}$	error-free bits for a BER claim at 95 % confidence
$N \geq 10\,\ell f_\mathrm{max}/v,\quad \Delta\hat{\sigma}/\sigma \approx 1/\sqrt{2N}$	lumped-line sections; Monte Carlo accuracy

APPENDIX B

Tables and Constants

Table B.1 converts a target BER into the Gaussian multiplier Q, and Table B.2 converts loss in decibels into voltage and power ratios. Table B.3 collects the constants used in the examples.

Table B.1. BER against Q, with BER $= Q(Q)$.

BER	Q	$2Q$ (eye/σ)
10^{-3}	3.090	6.18
10^{-4}	3.719	7.44
10^{-5}	4.265	8.53
10^{-6}	4.753	9.51
10^{-7}	5.199	10.40
10^{-8}	5.612	11.22
10^{-9}	5.998	12.00
10^{-10}	6.361	12.72
10^{-11}	6.706	13.41
10^{-12}	**7.034**	**14.07**
10^{-13}	7.349	14.70
10^{-14}	7.651	15.30
10^{-15}	7.941	15.88

Table B.2. Decibels to ratios (loss).

Loss (dB)	Voltage ratio	Power ratio
0.5	0.944	0.891
1	0.891	0.794
3	0.708	0.501
6	0.501	0.251
10	0.316	0.100
12	0.251	0.0631
14	0.200	0.0398
17.4 (REF)	0.135	0.0182
20	0.100	0.0100
30	0.0316	0.00100
40	0.0100	0.000100

Table B.3. Handy constants and Open-SERDES conversions (at $T = 300\,\mathrm{K}$ where it matters).

Quantity	Value	Use
Boltzmann constant k	$1.381 \times 10^{-23}\,\mathrm{J/K}$	noise
Elementary charge q	$1.602 \times 10^{-19}\,\mathrm{C}$	shot noise, V_T
Thermal voltage kT/q	25.85 mV	HBT $g_\mathrm{m} = I_C/V_T$: 38.7 mS per mA
kT	$4.14 \times 10^{-21}\,\mathrm{J} = -173.8\,\mathrm{dBm/Hz}$	noise floors
$\sqrt{4kTR}$, $R = 50\,\Omega$	$0.91\,\mathrm{nV}/\sqrt{\mathrm{Hz}}$	termination noise
$\sqrt{kT/C}$, $C = 50\,\mathrm{fF}$	0.29 mV	sampled noise
Vacuum permittivity ε_0	$8.854 \times 10^{-12}\,\mathrm{F/m}$	capacitance
Speed of light c	$2.998 \times 10^{8}\,\mathrm{m/s}$	line delay
FR-4 ($\varepsilon_r = 4$): $v = c/2$	$1.50 \times 10^{8}\,\mathrm{m/s} = 6.67\,\mathrm{ps/mm} = 169\,\mathrm{ps/in}$	reflections, flight time
1 UI at 5 Gb/s	200 ps	everything
1 degree of the 2.5 GHz clock	1.11 ps	I/Q and PI errors
1 PI step	6.25 ps = 0.031 UI	CDR

APPENDIX C

Design-Review Checklists

Use these lists in every design review. Each item points to the section that explains it; the numbers are the targets of Table 14.2.

Transmitter

- FO4 measured in the PDK; CMOS stages close timing at the slow corner (Section 9.2).
- Half-rate mux: glitch-free phases, duty cycle $50 \pm 1\,\%$.
- Driver I_T for 0.8–1.0 Vppd; every $V_{CE} < \mathrm{BV}_{CEO}$ at $V_{CC} + 5\,\%$, with the real far-end termination.
- FFE presets $-3.5/-6$ dB exact within one LSB; tap paths matched to a few ps.
- Termination trimmed to $\pm 10\,\%$; $C_p \leq 0.85$ pF; RL ≥ 10 dB at f_N.
- Post-layout eye through REF within 10 % of the model.

Channel and board

- Loss at f_N (not f_b) known; REF replaced by measured S-parameters.
- Via stubs short or back-drilled; first resonance above 5 GHz.
- Pairs length-matched; TX and RX pairs separated (NEXT).
- SMA launches and bond wires modeled; reflections in the pulse tail checked.

Receiver

- CTLE: 0–12 dB at every corner; f_{p2} with the real sampler load.
- DFE tap-1 loop ≤ 1 UI post-layout; unrolling ready if not.
- Sampler $\sigma \leq 1$ mV; offset DAC $\geq \pm 3\sigma$, 5 bits; $\tau \leq 5$ ps.
- Two extra latches behind the edge samplers (metastability).
- All four samplers simulated together (kickback).
- Deserializer FIFO and recovered-clock domain crossing checked.

PLL and clocks

- VCO tunes 2.5 GHz with V_{ctrl} mid-range at all corners; band code range covers PVT.
- Loop: 60° phase margin, $f_u \approx 3$ MHz, $f_{ref}/f_u \geq 10$.
- VCO phase noise simulated open-loop; RJ ≤ 1 ps rms integrated.
- Loop-filter capacitors MIM, not leaky MOS.
- I/Q error $\leq 2°$ (3σ), duty cycle $50 \pm 0.5\,\%$ after DCC.
- Clock buffers have 2–3$\times f_{clk}$ bandwidth.

CDR

- Phase-detector truth table verified for both half-rate phases.
- K_P from the JTOL slew requirement; $K_P/K_I \geq 16$ (baseline 128).
- Latency counted in updates; wander ≤ 20 ps pp in simulation.
- Integral register covers ± 300 ppm; lock < 1 µs.
- PI INL $\leq 1°$ at all 64 codes; code wrap glitch-free.
- Lab 7 model rerun with the real PI and latency.

Tape-out

- DRC clean (KLayout decks), antenna and density after fill.
- LVS clean per macro and full chip without black boxes.
- Post-layout corners and Monte Carlo pass in **CACE**.
- Model back-annotated with extracted responses; margin > 0.
- ESD on every pad; supply domains, decaps and guard rings reviewed.
- Test access: PRBS, checker, loopback, PI override, clock monitor.
- Bonding diagram, pinout and test-board plan signed off.

APPENDIX D

Glossary

Alexander PD	bang-bang phase detector that compares a data, an edge and the next data sample
Aperture	time window over which a sampler effectively averages its input
Bathtub curve	BER against sampling position across the UI
BER	bit-error rate: errors divided by bits; Open-SERDES targets 10^{-12}
BiCMOS	process with both bipolar (HBT) and CMOS transistors, such as SG13G2
BV_{CEO}	collector-emitter breakdown voltage of an HBT with the base open
CDR	clock and data recovery: extracts the sampling phase from the data transitions
CML	current-mode logic: differential pairs steering a constant tail current into resistors
Common centroid	layout that interleaves matched devices so gradients cancel
CTLE	continuous-time linear equalizer: an analog high-frequency boost
Cursor	sample of the pulse response at an integer number of UIs from the main sample
DCD	duty-cycle distortion: unequal high and low times, a two-valued DJ
De-emphasis	TX FFE setting that lowers repeated bits instead of raising transitions
DFE	decision-feedback equalizer: subtracts ISI predicted from past decisions
DJ, RJ	deterministic (bounded) and random (Gaussian) jitter
Dual-Dirac model	jitter model with DJ as two impulses convolved with Gaussian RJ
ESD	electrostatic discharge, and the diodes that protect the pads from it
Eye diagram	overlay of the received waveform folded at one or two UIs
FFE	feed-forward equalizer: an FIR filter, here in the TX driver
HBT	heterojunction bipolar transistor; SG13G2 npn13G2 with $f_T \approx 300\,\text{GHz}$
Insertion loss	attenuation through the channel, $-20\log_{10}\lvert S_{21}\rvert$
ISI	inter-symbol interference: energy of a bit that spills into its neighbors
JTOL	jitter tolerance: largest sinusoidal jitter, against frequency, that still meets the BER
Loopback	internal path from the TX output to the RX input for self-test
LVS, DRC	layout-versus-schematic and design-rule checks before tape-out
Metastability	a latch input so small that the decision is late or undefined
Monte Carlo	simulation repeated with random device mismatch
NRZ, PAM4	two-level and four-level pulse-amplitude modulation
Nyquist frequency	half the symbol rate: 2.5 GHz for 5 Gb/s NRZ
PDA	peak-distortion analysis: worst-case eye from the sum of absolute cursors
Phase interpolator	circuit that mixes two clock phases to make any phase between them
Phase noise	single-sideband noise power near a clock, $\mathcal{L}(f)$, in dBc/Hz
PLL	phase-locked loop: multiplies a clean reference up to the VCO frequency
Polyphase filter	RC network that splits a differential clock into I and Q
PRBS	pseudo-random binary sequence from a linear-feedback shift register
Pulse response	channel output for one isolated 1-UI pulse
Return loss	reflection at a port, $-20\log_{10}\lvert S_{11}\rvert$
S-parameters	frequency-domain description of a network by incident and reflected waves
Sampler	clocked comparator (latch) that decides each bit
Serializer	multiplexer tree that turns a parallel word into a serial stream
SST driver	source-series-terminated (voltage-mode) driver
Statistical eye	eye at a target BER computed from the ISI and noise distributions
T-coil	coupled-inductor network that hides pad capacitance from the line
UI	unit interval: one bit time, 200 ps
Zero-forcing	equalizer that forces selected cursors to zero, ignoring noise

APPENDIX E

Further Reading

Start here. Palermo's ECEN 720 course [4] is the backbone of this tutorial; its lectures on channels and link analysis [13], equalization [22], circuits [24] and clocking [34] go deeper on every chapter. The forthcoming book by Shekhar, Palermo and Hanumolu [65] covers the same ground in textbook form.

Signal integrity and systems. Dally and Poulton [14] is the classic on signaling, noise budgets and timing in digital systems. Hall and Heck [16] treat transmission lines, losses, S-parameters and channel modeling in the detail you need to replace REF with a real board. Stojanović *et al.* [23] show a complete adaptive PAM2/4 transceiver and why the equalization partition matters. The IEEE 802.3 backplane clauses [7] define the channels, masks and test methods of real 2.5 and 5 Gb/s Ethernet links.

Circuits. Razavi's analog text [66] is the reference for differential pairs, noise and mismatch. His optical-communications book [40] covers CML, limiting amplifiers, CDRs and multiplexers at the transistor level, much as Open-SERDES builds them.

Clocking. Razavi's PLL book [36] takes the charge-pump PLL from the loop equations to the VCO and divider circuits. Gardner [37] is the classic on loop dynamics and the source of most PLL design formulas. The team's 2.4 GHz LC-VCO PLL paper [9] is the starting point for the Open-SERDES PLL.

Open-source design. The IHP SG13G2 open PDK [1], the IIC-OSIC-TOOLS image [2] and the JKU chip template [5] are everything you need to tape out; read the template tutorial before your first `make all`. The Purdue OpenSerDes paper [11] is the closest open-source prior art, an all-digital 2 Gb/s link in SKY130. For modeling, serdespy [3] powers the labs of Chapter 16, and PyBERT [44] is the natural next step, with a GUI and IBIS-AMI support.

References

[1] IHP GmbH. IHP Open Source PDK for the SG13G2 130 nm SiGe:C BiCMOS Technology. https://github.com/IHP-GmbH/IHP-Open-PDK, 2025.

[2] Institute for Integrated Circuits, JKU Linz. IIC-OSIC-TOOLS: All-in-One Docker Image for Open-Source IC Design. https://github.com/iic-jku/IIC-OSIC-TOOLS, 2025.

[3] Richard Barrie and Katherine Liang. serdespy: A Python Library for System-Level SerDes Modelling and Simulation. University of Toronto ESC499 thesis project, https://github.com/richard259/serdespy, 2022.

[4] Sam Palermo. ECEN 720: High-Speed Links Circuits and Systems. Course lecture notes, Texas A&M University, https://people.engr.tamu.edu/spalermo/ecen720.html, 2024.

[5] Simon Dorrer and Harald Pretl. From Idea to Tapeout: An Open-Source Analog Mixed-Signal Design Flow with the IHP Open-PDK. FSiC 2026; template at https://github.com/iic-jku/ihp-sg13g2-ams-chip-template, 2026.

[6] PCI-SIG. PCI Express Base Specification, revisions 1.0a–7.0. https://pcisig.com/specifications, 2025.

[7] IEEE. IEEE Std 802.3 – ethernet (including 802.3cb 2.5gbase-kx and 5gbase-kr), 2022.

[8] USB Implementers Forum. Universal Serial Bus 4 (USB4) Specification, version 2.0, 2022.

[9] M. Thiriloganathan et al. A 2.4 GHz LC-VCO fractional-N phase locked loop open-source design in 130-nm BiCMOS. In *Int. Conf. on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2026. Accepted.

[10] D. M. Serasinghe, M. D. Sudara, J. G. D. A. Thilakaratne, V. S. Wijesinghe, and S. Thayaparan. 5 Gbps SerDes IP Development for 100 Gbps Ethernet Applications. Undergraduate project report, 2016.

[11] Gautam Kumar K, Baibhab Chatterjee, and Shreyas Sen. OpenSerDes: An open source process-portable all-digital serial link. In *Design, Automation and Test in Europe (DATE)*, 2021. arXiv:2105.13256.

[12] B. Heinemann, R. Barth, D. Bolze, J. Drews, G. G. Fischer, A. Fox, O. Fursenko, T. Grabolla, U. Haak, D. Knoll, R. Kurps, M. Lisker, S. Marschmeyer, H. Rücker, D. Schmidt, J. Schmidt, M. A. Schubert, B. Tillack, C. Wipf, D. Wolansky, and Y. Yamamoto. SiGe HBT technology with f_t/f_{max} of 300 GHz/500 GHz and 2.0 ps CML gate delay. In *IEEE International Electron Devices Meeting (IEDM)*, 2010.

[13] Sam Palermo. ECEN 720 Lectures 3–4: Channel Modeling, Modulation and Link Analysis. Texas A&M University course notes, 2024.

[14] William J. Dally and John W. Poulton. *Digital Systems Engineering*. Cambridge University Press, 1998.

[15] Howard W. Johnson and Martin Graham. *High-Speed Digital Design: A Handbook of Black Magic*. Prentice Hall, 1993.

[16] Stephen H. Hall and Howard L. Heck. *Advanced Signal Integrity for High-Speed Digital Designs*. Wiley-IEEE Press, 2009.

[17] Antonije R. Djordjević, Radivoje M. Biljic, Vladan D. Likar-Smiljanic, and Tapan K. Sarkar. Wideband frequency-domain characterization of FR-4 and time-domain causality. *IEEE Transactions on Electromagnetic Compatibility*, 43(4):662–667, 2001.

[18] Alexander Arsenovic et al. scikit-rf: An open source Python package for microwave network creation, analysis, and calibration. *IEEE Microwave Magazine*, 23(1):98–105, 2022.

[19] Bryan K. Casper, Matthew Haycock, and Randy Mooney. An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pages 54–57, 2002.

[20] Anthony Sanders, Mike Resso, and John D'Ambrosia. Channel compliance testing utilizing novel statistical eye methodology. In *DesignCon*, 2004.

[21] Agilent Technologies. Jitter Analysis: The dual-Dirac Model, RJ/DJ, and Q-Scale. White paper, 2004.

[22] Sam Palermo. ECEN 720 Lecture 7: Equalization Techniques. Texas A&M University course notes, 2024.

[23] Vladimir Stojanović, Mark Horowitz, et al. Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery. *IEEE Journal of Solid-State Circuits*, 40(4):1012–1026, 2005.

[24] Sam Palermo. ECEN 720 Lectures 5, 6 and 9: Termination and TX Drivers, RX Circuits, Noise. Texas A&M University course notes, 2024.

[25] Koon-Lun Jackie Wong, Hamid Hatamkhani, Mozhgan Mansuri, and Chih-Kong Ken Yang. A 27-mW 3.6-Gb/s I/O transceiver. *IEEE Journal of Solid-State Circuits*, 39(4):602–612, 2004.

[26] Sherif Galal and Behzad Razavi. Broadband ESD protection circuits in CMOS technology. *IEEE Journal of Solid-State Circuits*, 38(12):2334–2340, 2003.

[27] Marcel Kossel, Christian Menolfi, Jonas Weiss, Peter Buchmann, George von Büren, Lucio Rodoni, Thomas Morf, Thomas Toifl, and Martin Schmatz. A T-coil-enhanced 8.5 Gb/s high-swing SST transmitter in 65 nm bulk CMOS with < -16 dB return loss over 10 GHz bandwidth. *IEEE Journal of Solid-State Circuits*, 43(12):2905–2920, 2008.

[28] Behzad Razavi. The bridged T-coil [a circuit for all seasons]. *IEEE Solid-State Circuits Magazine*, 7(4):9–13, 2015.

[29] Tsuguo Kobayashi, Kazutaka Nogami, Tsukasa Shirotori, and Yukihiro Fujimoto. A current-controlled latch sense amplifier for static memories. *IEEE Journal of Solid-State Circuits*, 28 (5):523–527, 1993.

[30] Behzad Razavi. The StrongARM latch [a circuit for all seasons]. *IEEE Solid-State Circuits Magazine*, 7(2):12–17, 2015.

[31] Borivoje Nikolić, Vojin G. Oklobdžija, Vladimir Stojanović, Wenyan Jia, James Kar-Shing Chiu, and Michael Ming-Tak Leung. Improved sense-amplifier-based flip-flop: Design and measurements. *IEEE Journal of Solid-State Circuits*, 35(6):876–884, 2000.

[32] Jaeha Kim, Brian S. Leibowitz, Jihong Ren, and Chris J. Madden. Simulation and analysis of random decision errors in clocked comparators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(8):1844–1857, 2009.

[33] Marcel J. M. Pelgrom, Aad C. J. Duinmaijer, and Anton P. G. Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, 24(5):1433–1439, 1989.

[34] Sam Palermo. ECEN 720 Lectures 10–13: Jitter, Clocking, PLLs and Clock and Data Recovery. Texas A&M University course notes, 2024.

[35] Floyd M. Gardner. Charge-pump phase-lock loops. *IEEE Transactions on Communications*, 28 (11):1849–1858, 1980.

[36] Behzad Razavi. *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge University Press, 2020.

[37] Floyd M. Gardner. *Phaselock Techniques*. Wiley, 3rd edition, 2005.

[38] D. B. Leeson. A simple model of feedback oscillator noise spectrum. *Proceedings of the IEEE*, 54(2):329–330, 1966.

[39] Charles R. Hogge. A self correcting clock recovery circuit. *Journal of Lightwave Technology*, 3 (6):1312–1314, 1985.

[40] Behzad Razavi. *Design of Integrated Circuits for Optical Communications*. Wiley, 2nd edition, 2012.

[41] J. D. H. Alexander. Clock recovery from random binary signals. *Electronics Letters*, 11(22): 541–542, 1975.

[42] Kurt H. Mueller and Markus Müller. Timing recovery in digital synchronous data receivers. *IEEE Transactions on Communications*, 24(5):516–531, 1976.

[43] Tiny Tapeout. Tiny Tapeout: Shared multi-project tapeouts, including IHP SG13G2. https://tinytapeout.com, 2025.

[44] David Banas. PyBERT: Serial communication link bit error rate tester simulator. https://github.com/capn-freako/PyBERT, 2024.

[45] Stefan Schippers. Xschem: Schematic capture and netlisting EDA tool. https://xschem.sourceforge.io, 2025.

[46] The ngspice team. ngspice: Open-source mixed-signal circuit simulator. https://ngspice.sourceforge.io, 2025.

[47] Sandia National Laboratories. Xyce parallel electronic simulator. https://xyce.sandia.gov, 2025.

[48] Árpád Bűrmen. VACASK: Verilog-A circuit analysis kernel. https://codeberg.org/arpadbuermen/VACASK, 2025.

[49] FOSSi Foundation. CACE: Circuit automatic characterization engine. https://github.com/fossi-foundation/cace, 2025.

[50] Matthias Köfferlein. KLayout: IC layout viewer and editor. https://www.klayout.de, 2025.

[51] R. Timothy Edwards. Magic VLSI layout tool. http://opencircuitdesign.com/magic/, 2025.

[52] R. Timothy Edwards. Netgen: Netlist comparison (LVS). http://opencircuitdesign.com/netgen/, 2025.

[53] Martin Köhler. KLayout-PEX: Parasitic extraction for KLayout. https://github.com/martinjankoehler/klayout-pex, 2025.

[54] Thorsten Liebig. openEMS: Free and open electromagnetic field solver. https://www.openems.de, 2025.

[55] Amazon Web Services. Palace: 3D finite element solver for computational electromagnetics. https://github.com/awslabs/palace, 2025.

[56] Wilson Snyder. Verilator. https://www.veripool.org/verilator/, 2025.

[57] Stephen Williams. Icarus Verilog. https://steveicarus.github.io/iverilog/, 2025.

[58] The cocotb contributors. cocotb: Coroutine-based cosimulation testbench environment. https://www.cocotb.org, 2025.

[59] YosysHQ. Yosys open SYnthesis suite. https://yosyshq.net/yosys/, 2025.

[60] The LibreLane contributors. LibreLane: RTL-to-GDS flow. https://github.com/librelane/librelane, 2025.

[61] Efabless Corporation. OpenLane 2. https://github.com/efabless/openlane2, 2024.

[62] The OpenROAD Project. OpenROAD: Open-source RTL-to-GDS tools. https://theopenroadproject.org, 2025.

[63] The OpenROAD Project. OpenSTA: Static timing analyzer. https://github.com/The-OpenROAD-Project/OpenSTA, 2025.

[64] Kevin Zheng. Julia SerDes (JLSD), version 1.0. Zenodo, doi:10.5281/zenodo.12520404, https://github.com/kevjzheng/JLSD, 2024.

[65] Sudip Shekhar, Sam Palermo, and Pavan Kumar Hanumolu. *High-Speed Data Links*. to be published, 2026.

[66] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2nd edition, 2017.