

# **EE 437/538B: Integrated Systems**

## **Capstone/Design of Analog Integrated Circuits and Systems**

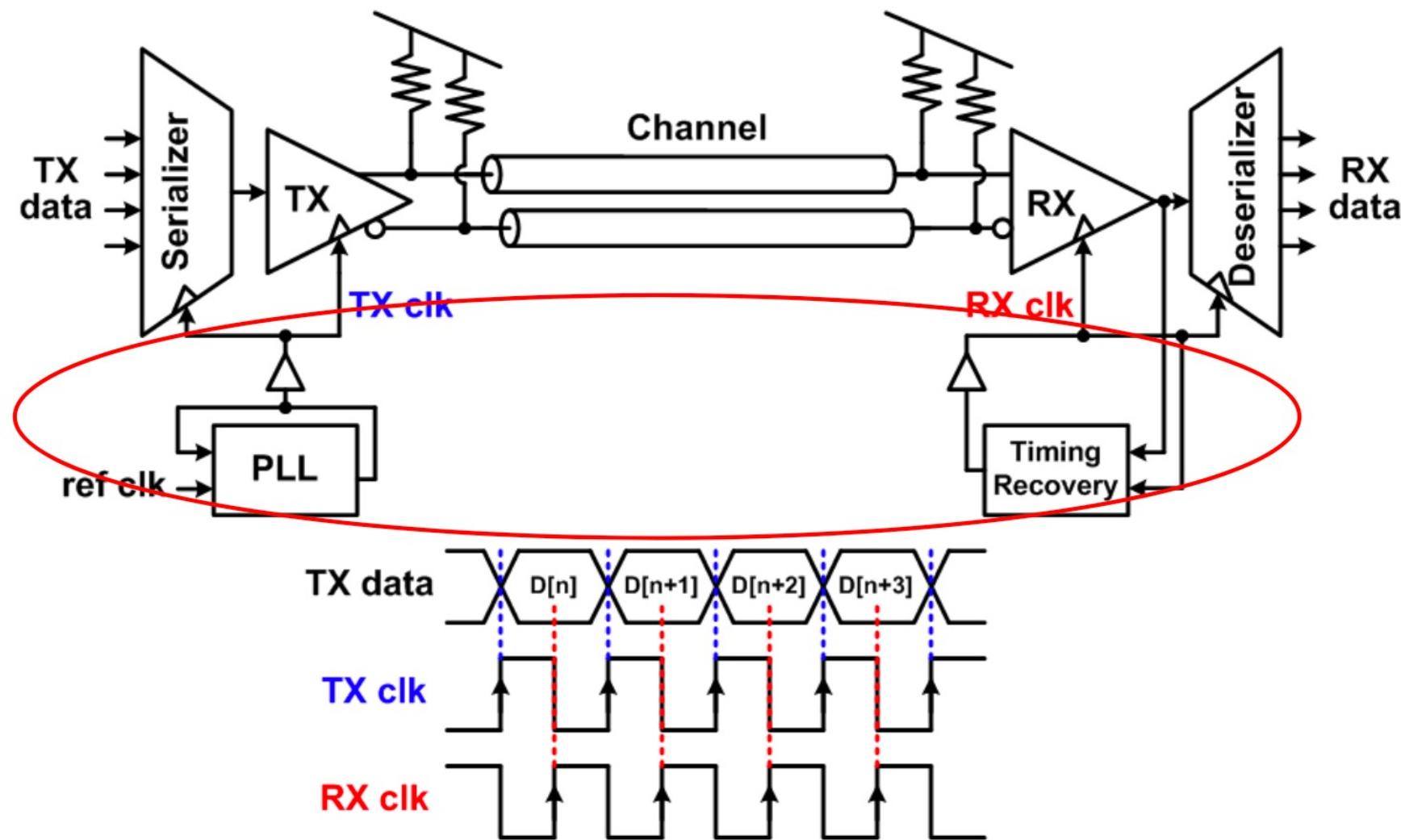
### **Lecture 8: Timing Basics**

Prof. Sajjad Moazeni

[smoazeni@uw.edu](mailto:smoazeni@uw.edu)

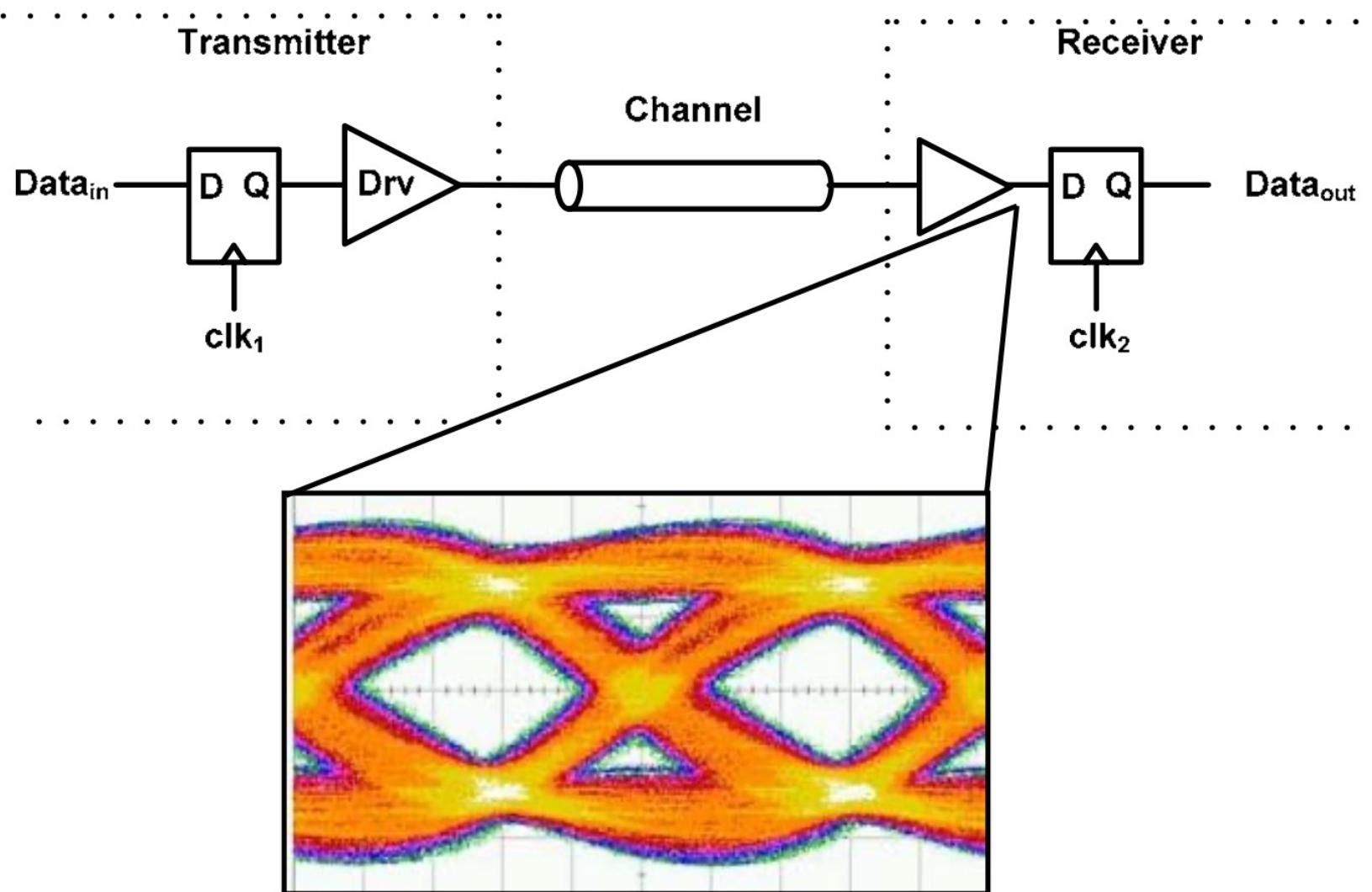
Spring 2022

# High-Speed Electrical Link System



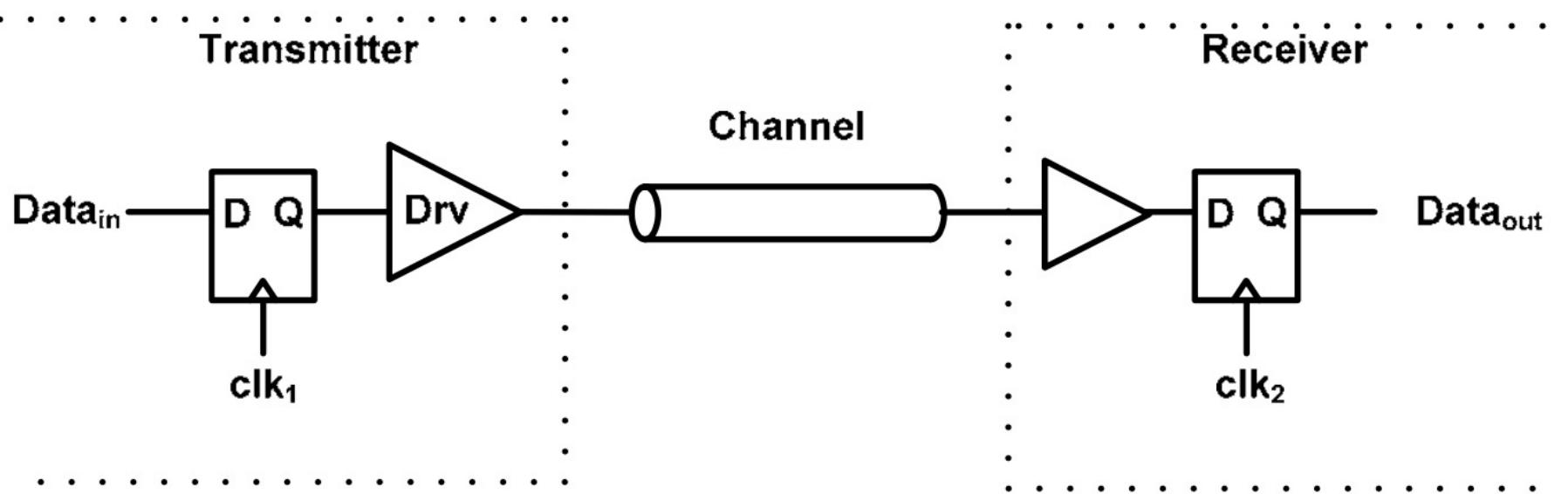
[Sam Palermo]

# Why We Need to Talk About Timing



[Elad Alon]

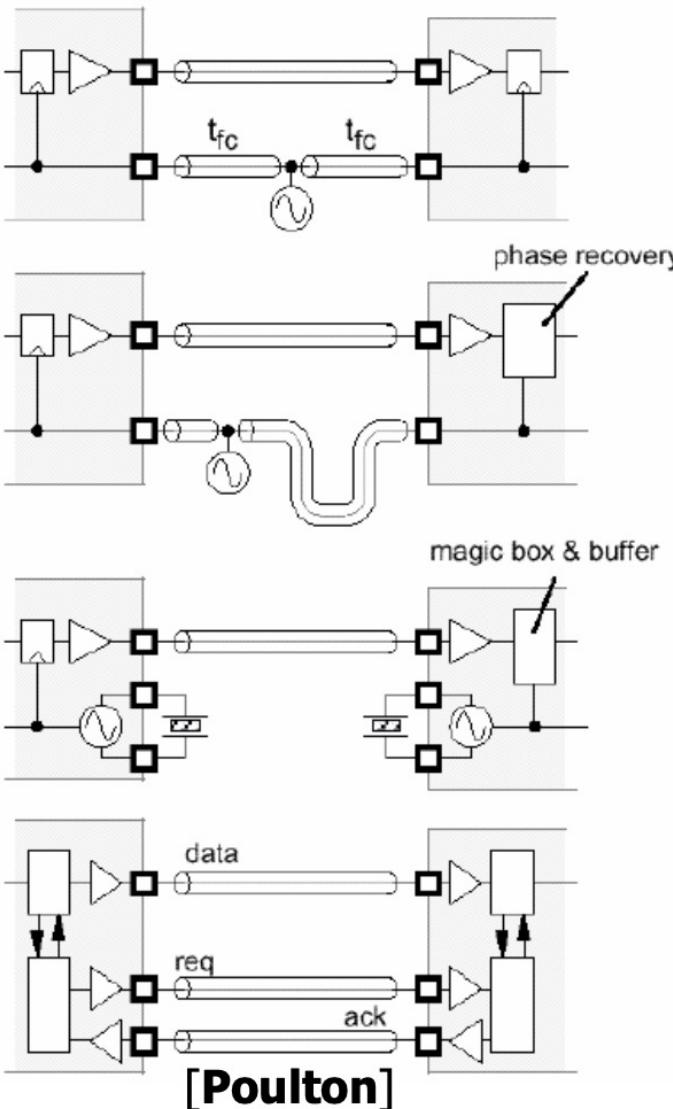
# Clocking Types



- Many different options...
- All boil down to relationship between (or even existence of) clk<sub>1</sub> and clk<sub>2</sub>

[Elad Alon]

# Clocking Terminology



## Synchronous

- Every chip gets same frequency AND phase
- Used in low-speed busses

## Mesochronous

- Same frequency, but unknown phase
- Requires phase recovery circuitry
  - Can do with or without full CDR
- Used in fast memories, internal system interfaces, MAC/Packet interfaces

## Plesiochronous

- Almost the same frequency, resulting in slowly drifting phase
- Requires CDR
- Widely used in high-speed links

## Asynchronous

- No clocks at all
- Request/acknowledge handshake procedure
- Used in embedded systems, Unix, Linux

[Sam Palermo]

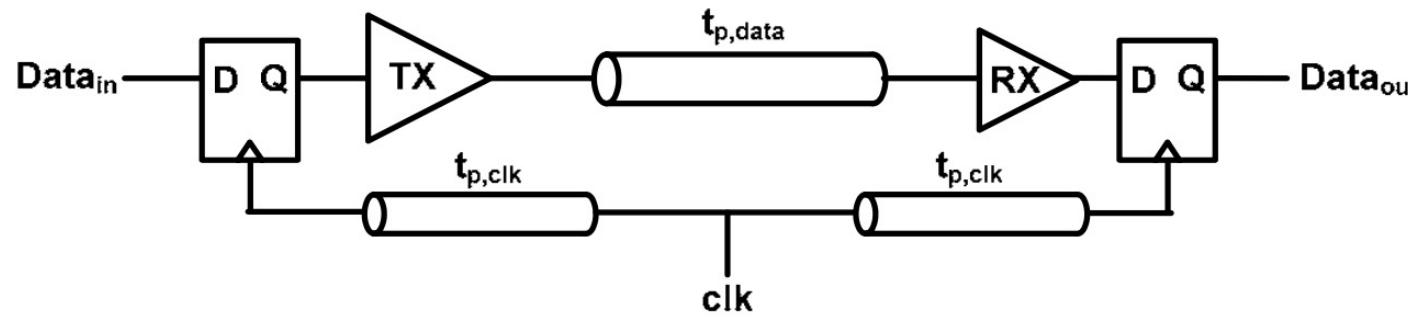
# I/O Clocking Architectures

---

- Three basic I/O architectures
  - Common Clock (Synchronous)
  - Forward Clock (Source Synchronous)
  - Embedded Clock (Clock Recovery)
- These I/O architectures are used for varying applications that require different levels of I/O bandwidth
- A processor may have one or all of these I/O types
- Often the same circuitry can be used to emulate different I/O schemes for design reuse

[Sam Palermo]

# “Simple” Synchronous System

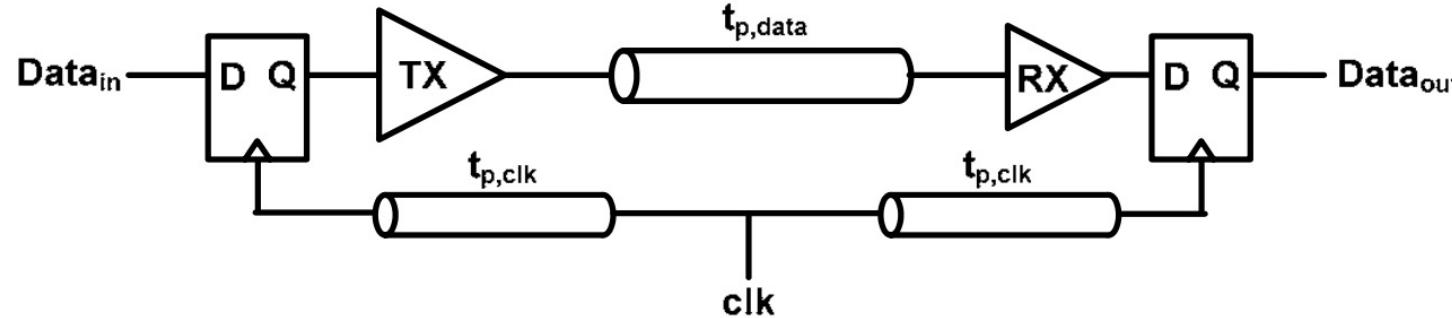


- Under what conditions will this work?
- “EE141” answer:

[Elad Alon]

# An Example

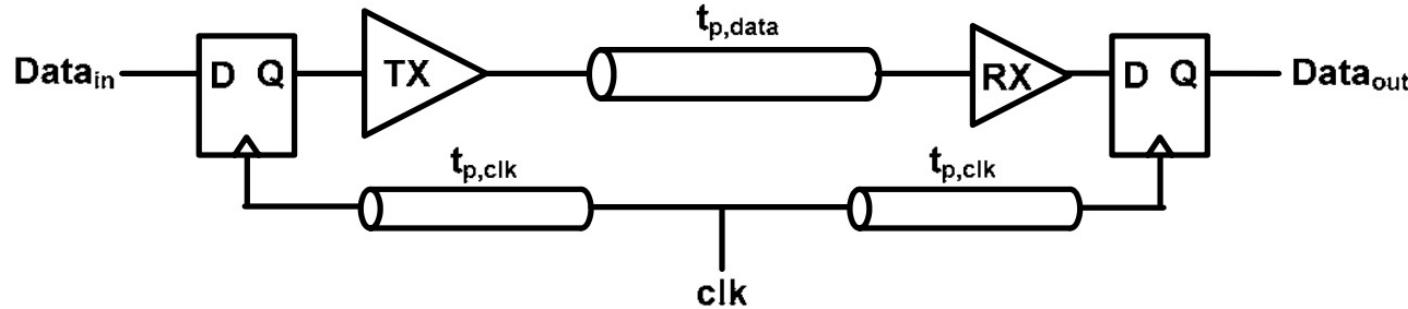
---



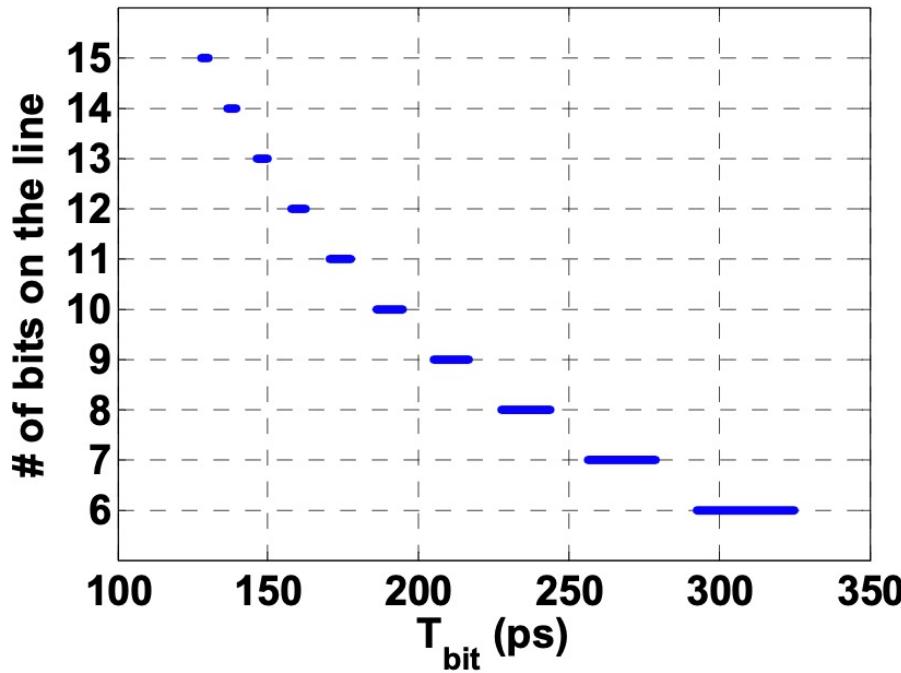
- $t_{p,data} = 2\text{ns}$ ,  $T_{bit} = ?$
- What else do you need to know?

[Elad Alon]

# An Example

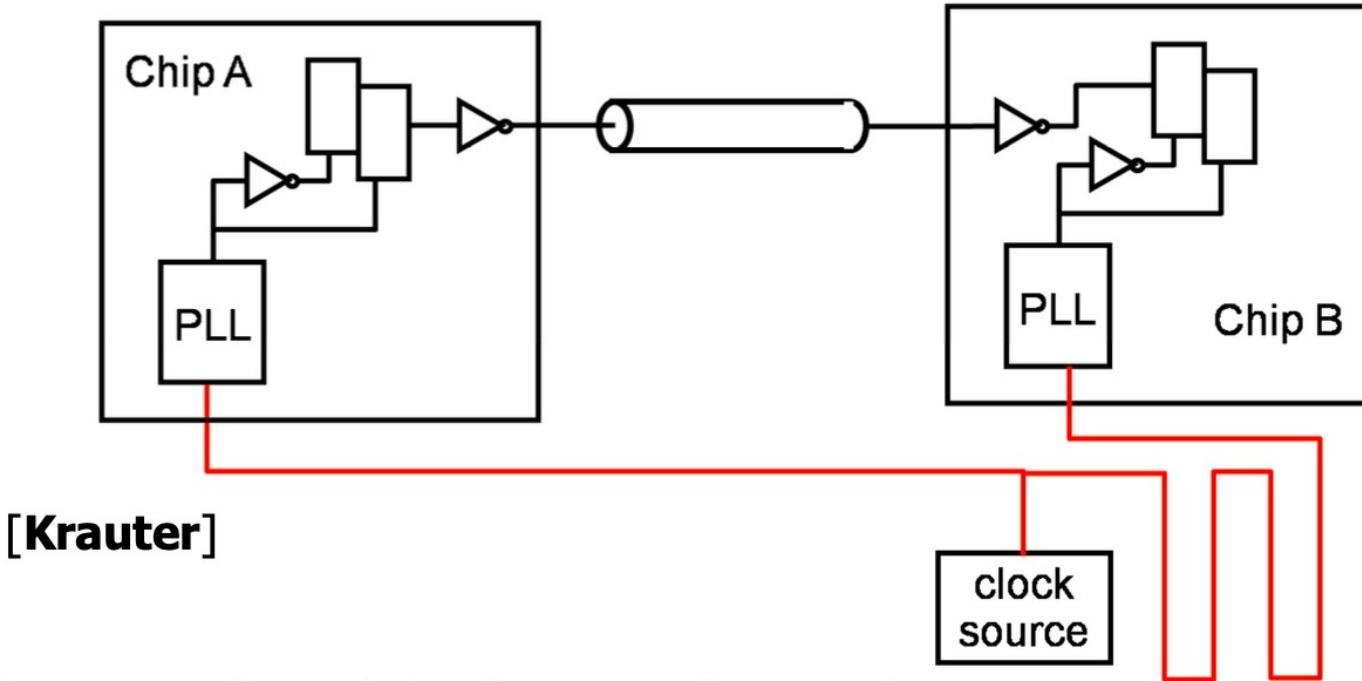


- $t_{p,data} = 2\text{ns}$ ,  $t_{sk+jitt} = +/-50\text{ps}$
- Get “bands” of functionality:



[Elad Alon]

# Common Clock I/O Architecture

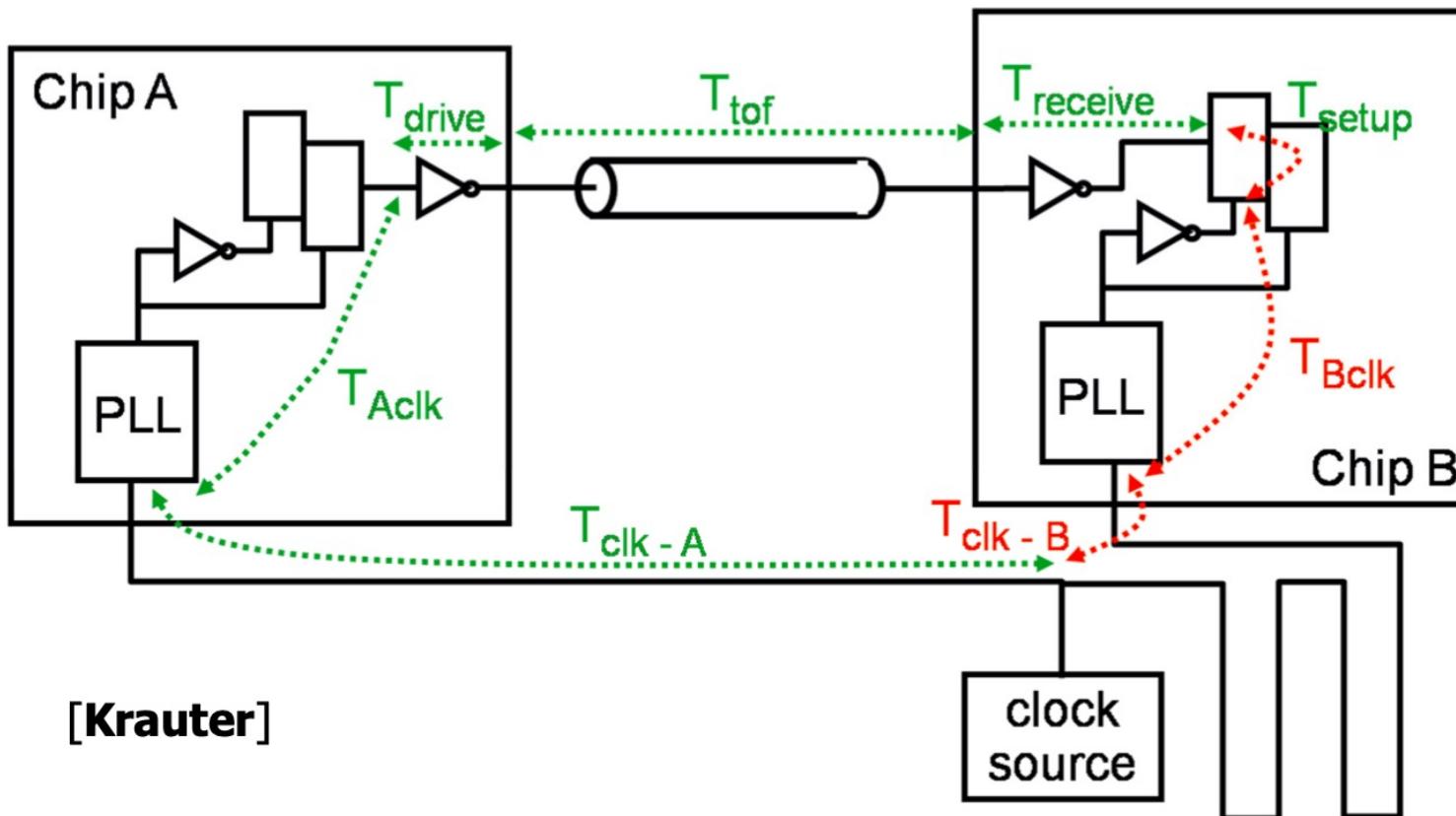


- Common in original computer systems
- Synchronous system by design (no active deskew)
- Common bus clock controls chip-to-chip transfers
- Requires equal length routes to chips to minimize clock skew
- Data rates typically limited to ~100Mb/s

# Common Clock I/O Cycle Time

Cycle time to meet setup time

$$\max(T_{\text{clk-A}} + T_{\text{Aclk}} + T_{\text{drive}} + T_{\text{tof}} + T_{\text{receive}} + T_{\text{setup}}) - \min(T_{\text{Bclk}} - T_{\text{clk-B}}) < T_{\text{cycle}}$$



9

[Sam Palermo]

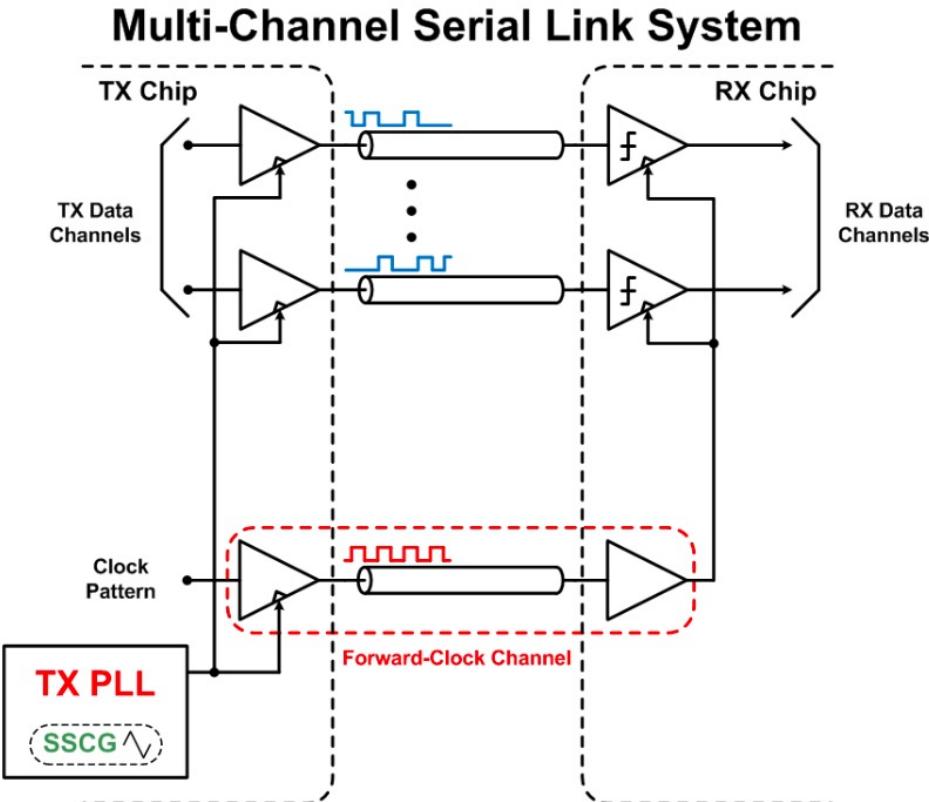
# Common Clock I/O Limitations

---

- Difficult to control clock skew and propagation delay
- Need to have tight control of absolute delay to meet a given cycle time
- Sensitive to delay variations in on-chip circuits and board routes
- Hard to compensate for delay variations due to low correlation between on-chip and off-chip delays
- While commonly used in on-chip communication, offers limited speed in I/O applications

[Sam Palermo]

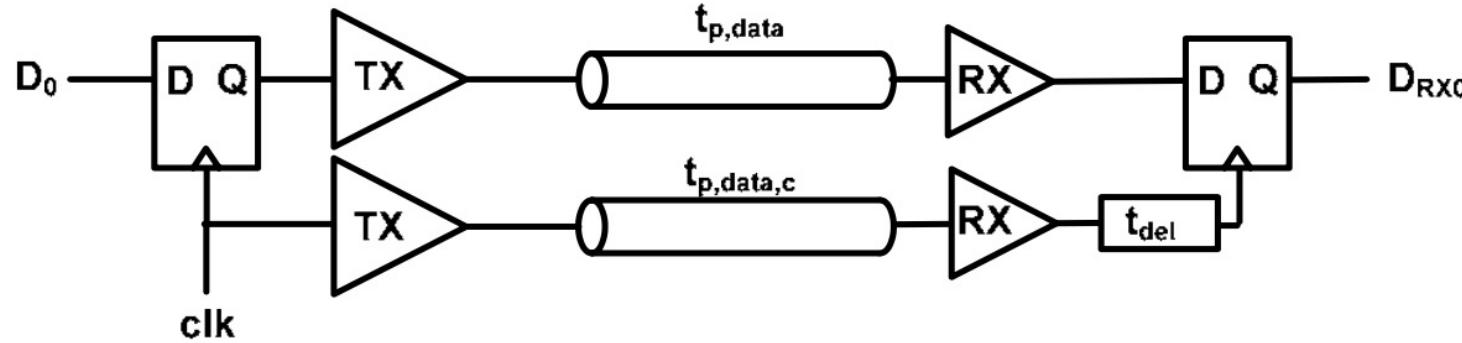
# Forward Clock I/O Architecture



- Common high-speed reference clock is forwarded from TX chip to RX chip
  - Mesochronous system
- Used in processor-memory interfaces and multi-processor communication
  - Intel QPI
  - Hypertransport
- Requires one extra clock channel
- “Coherent” clocking allows low-to-high frequency jitter tracking
- Need good clock receive amplifier as the forwarded clock is attenuated by the channel

[Sam Palermo]

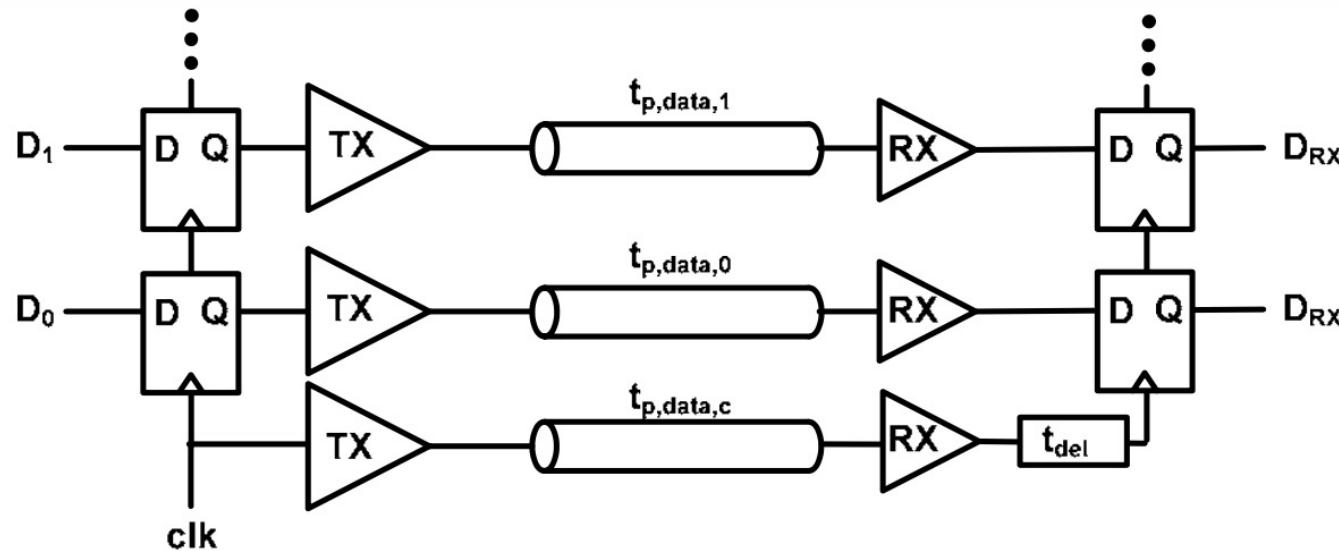
# Source Synchronous Clocking



- **Key idea: match clock and data paths**
  - Link ideally works from DC up to timing uncertainty-limited frequency
- **What is the “right”  $t_{del}$ ?**

[Elad Alon]

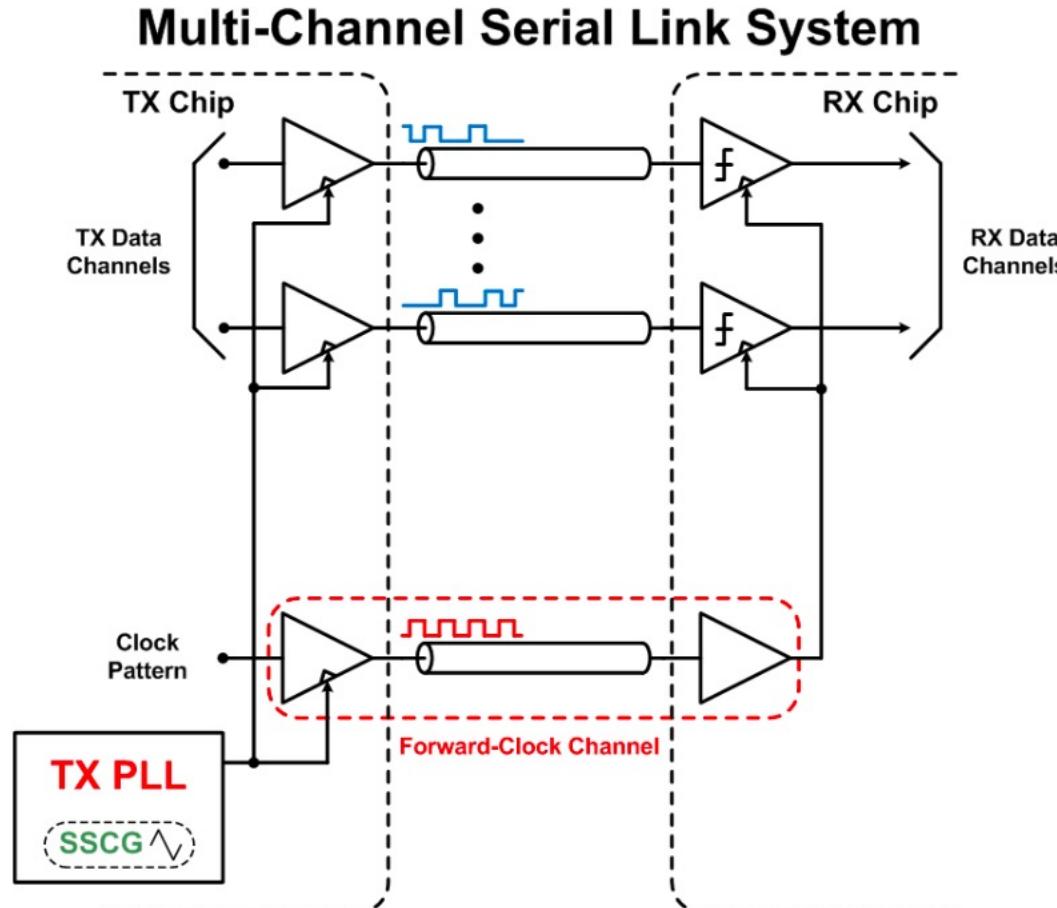
# Source Synchronous Clocking



- Want one clock “link” for multiple data links
  - Reduce overhead
- What if data lines don’t match each other?
  - Or don’t match clock line
  - Or  $t_{del}$  isn’t quite right (depends on  $T_{bit}$ , PVT, etc.)

[Elad Alon]

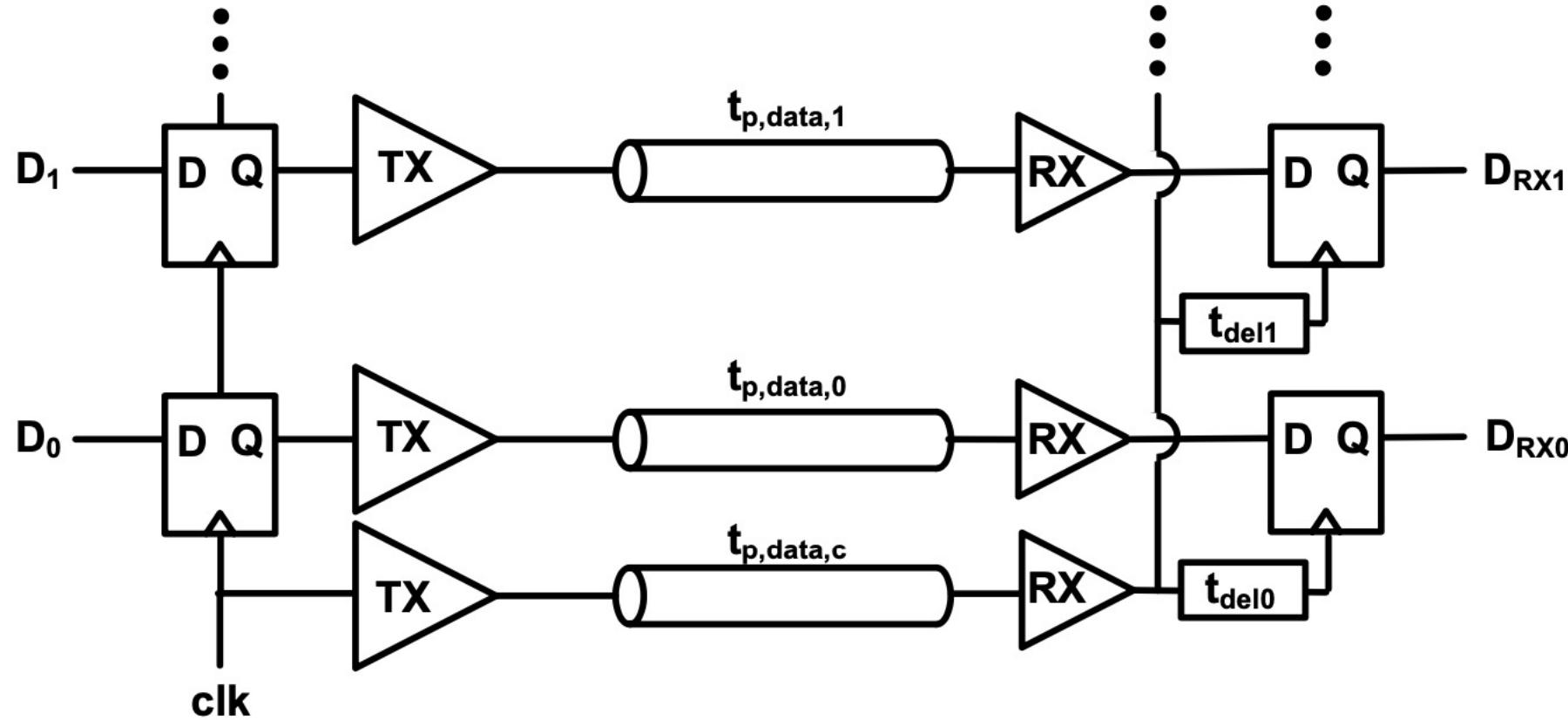
# Forward Clock I/O Limitations



- Clock skew can limit forward clock I/O performance
  - Driver strength and loading mismatches
  - Interconnect length mismatches
- Low pass channel causes jitter amplification
- Duty-Cycle variations of forwarded clock

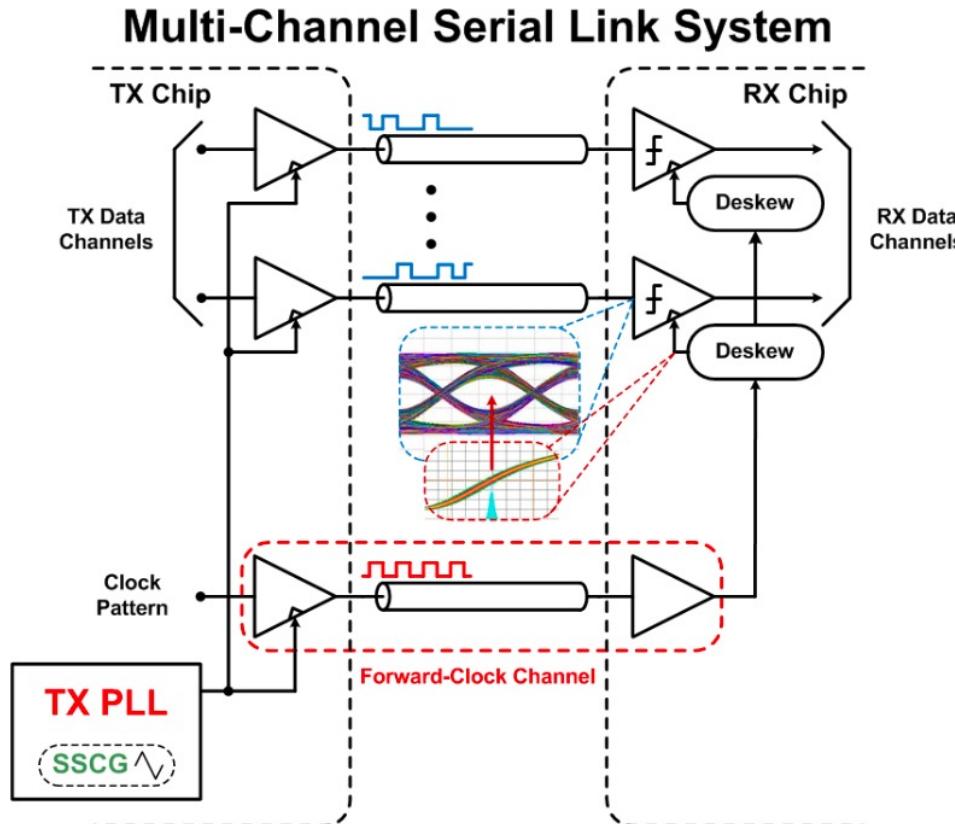
[Sam Palermo]

# Realistic Source Synchronous System



[Elad Alon]

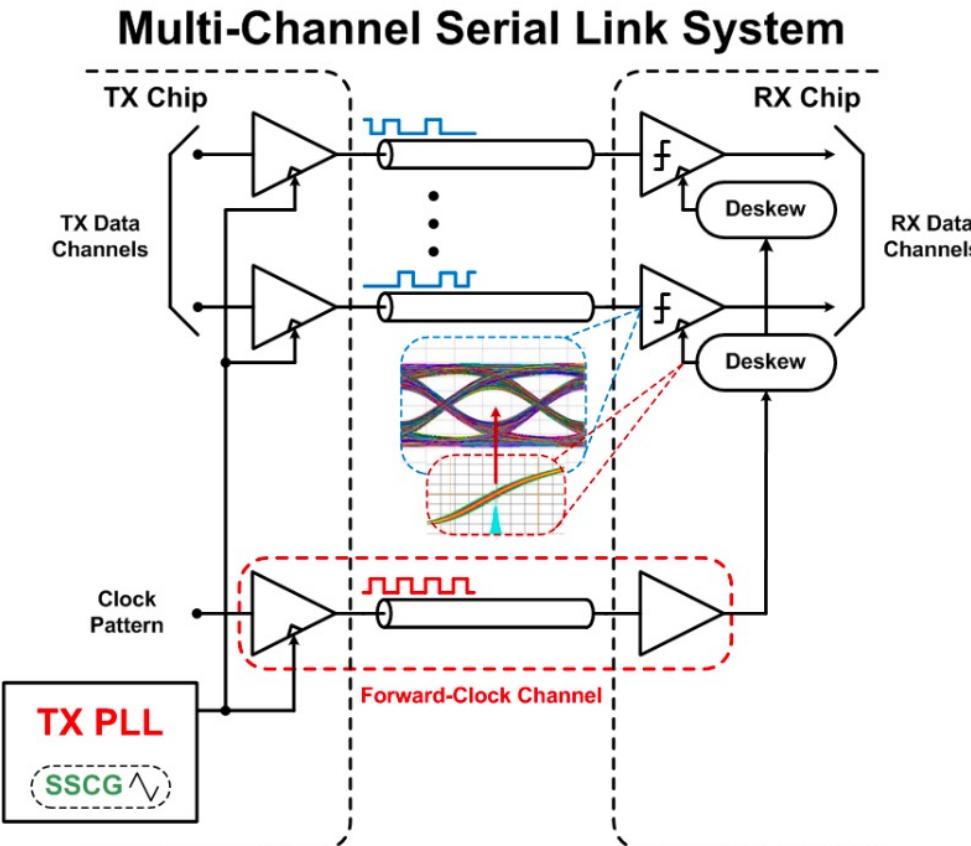
# Forward Clock I/O De-Skew



- Per-channel de-skew allows for significant data rate increases
- Sample clock adjusted to center clock on the incoming data eye
- Implementations
  - Delay-Locked Loop and Phase Interpolators
  - Injection-Locked Oscillators
- Phase Acquisition can be
  - BER based – no additional input phase samplers
  - Phase detector based implemented with additional input phase samplers periodically powered on

[Sam Palermo]

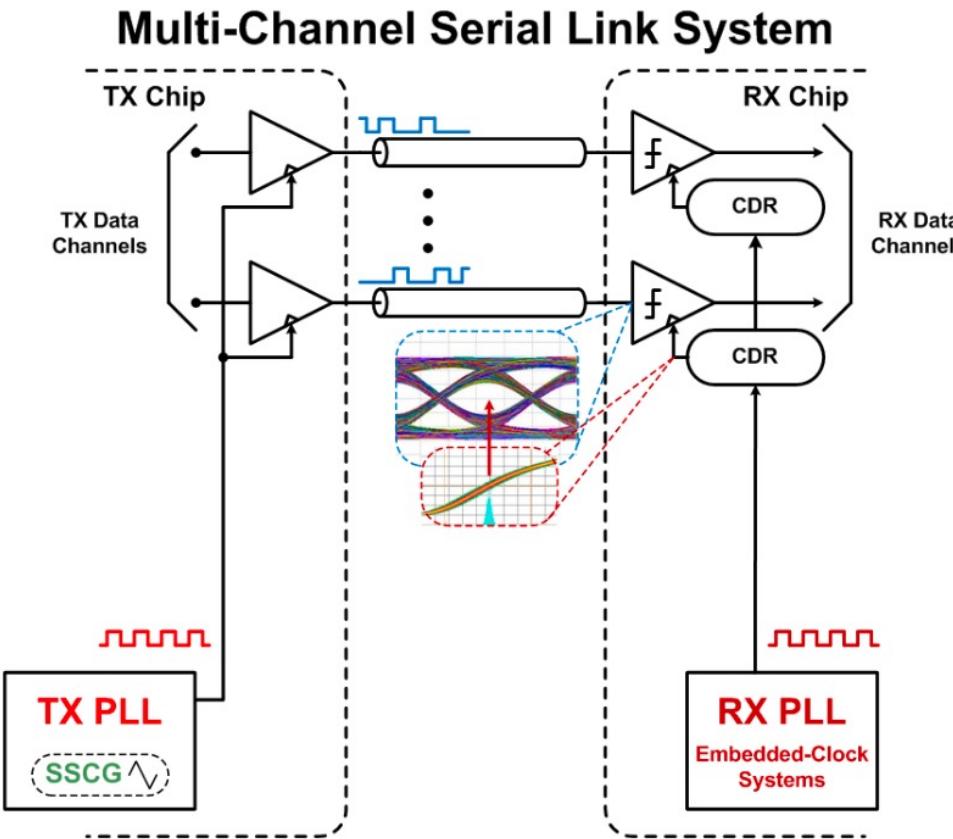
# Forward Clock I/O Circuits



- TX PLL
- TX Clock Distribution
- Replica TX Clock Driver
- Channel
- Forward Clock Amplifier
- RX Clock Distribution
- De-Skew Circuit
  - DLL/PI
  - Injection-Locked Oscillator

[Sam Palermo]

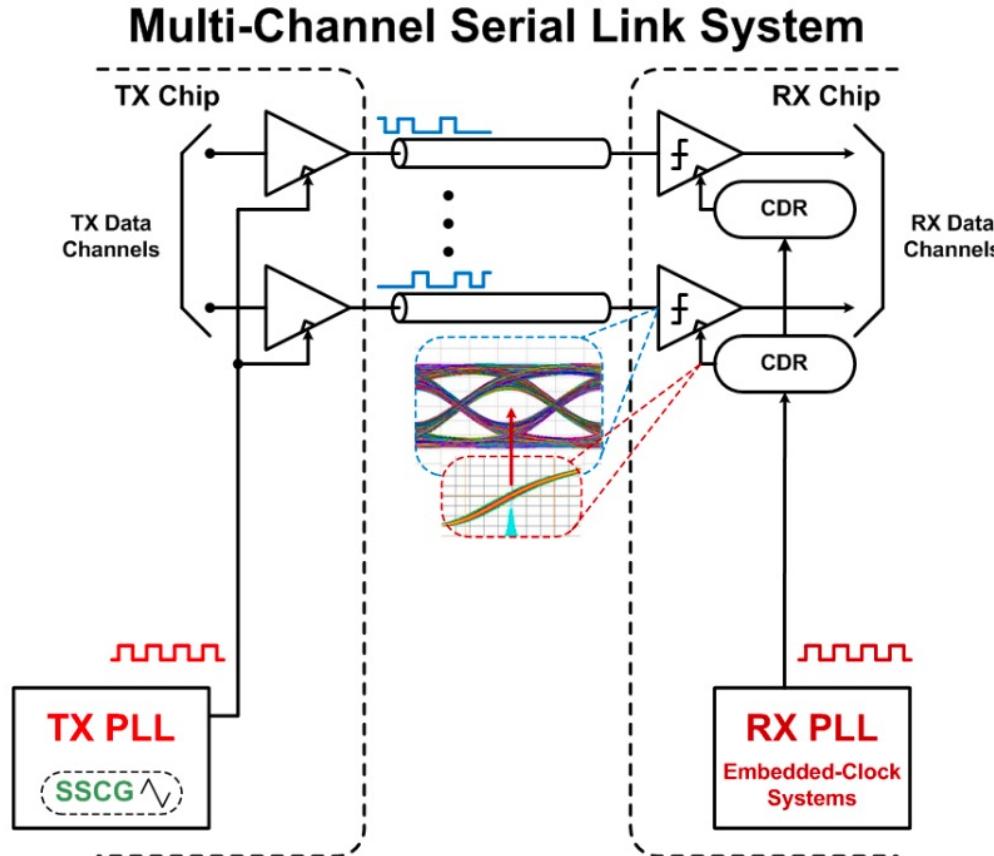
# Embedded Clock I/O Architecture



- Can be used in mesochronous or plesiochronous systems
- Clock frequency and optimum phase position are extracted from incoming data stream
- Phase detection continuously running
- CDR Implementations
  - Per-channel PLL-based
  - Dual-loop w/ Global PLL &
    - Local DLL/PI
    - Local Phase-Rotator PLLs

[Sam Palermo]

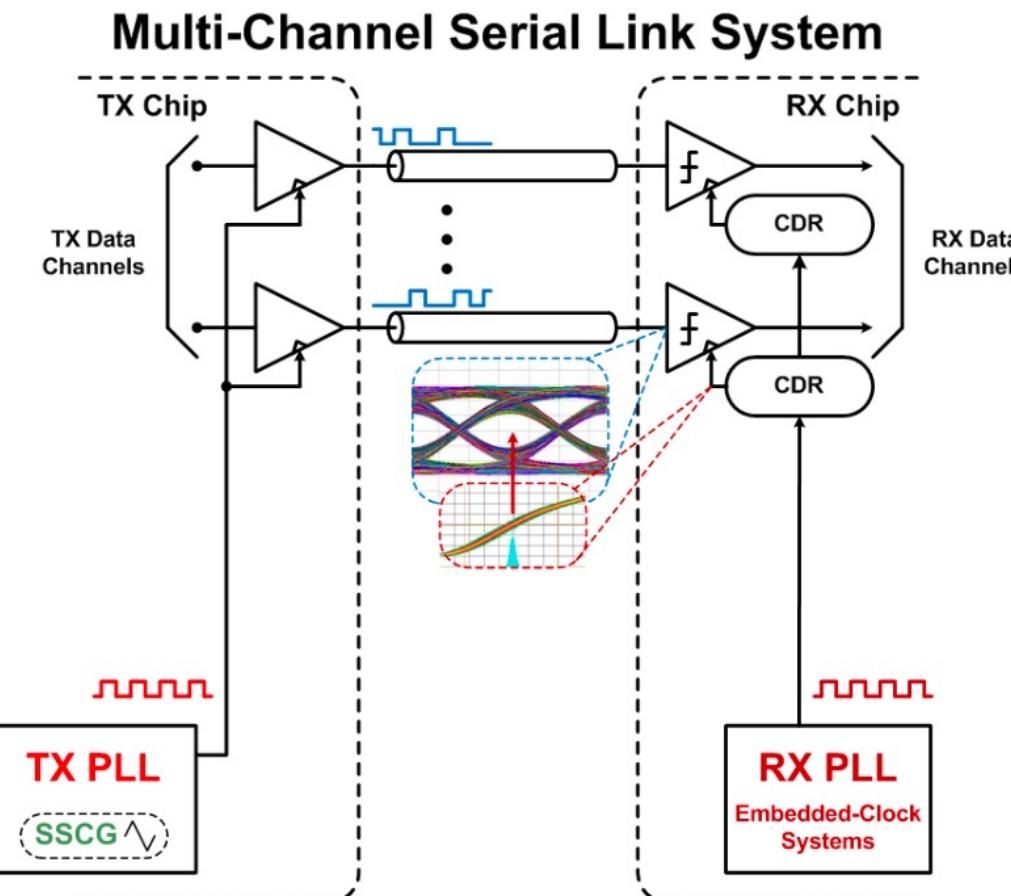
# Embedded Clock I/O Limitations



- Jitter tracking limited by CDR bandwidth
  - Technology scaling allows CDRs with higher bandwidths which can achieve higher frequency jitter tracking
- Generally more hardware than forward clock implementations
  - Extra input phase samplers

[Sam Palermo]

# Embedded Clock I/O Circuits



- TX PLL
- TX Clock Distribution
- CDR
  - Per-channel PLL-based
  - Dual-loop w/ Global PLL &
    - Local DLL/PI
    - Local Phase-Rotator PLLs
    - Global PLL requires RX clock distribution to individual channels

[Sam Palermo]

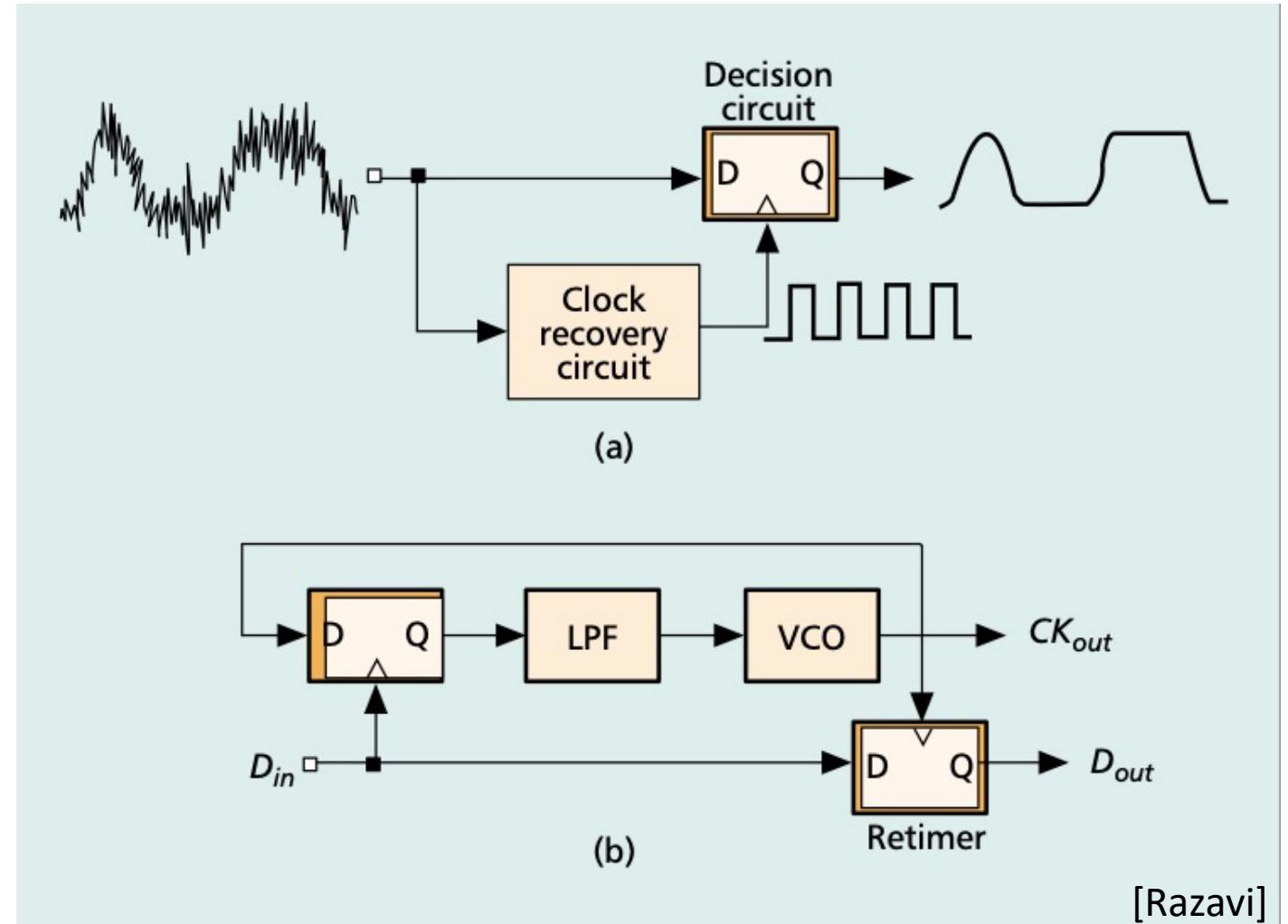
# In General: CDR

---

- **CDR = Clock and Data Recovery**
  - Recover clock phase and/or frequency based on data itself
  - If phase only, need a frequency reference
- **Several advantages vs. fixed timing**
  - Don't have to match delays/paths (**mesochronous**)
  - Allows separate crystals (**plesiochronous**)
- **But, CDR isn't free**
  - And places some requirements on data

[Elad Alon]

# Conceptual CDR



Difference between PLL & CDR ?

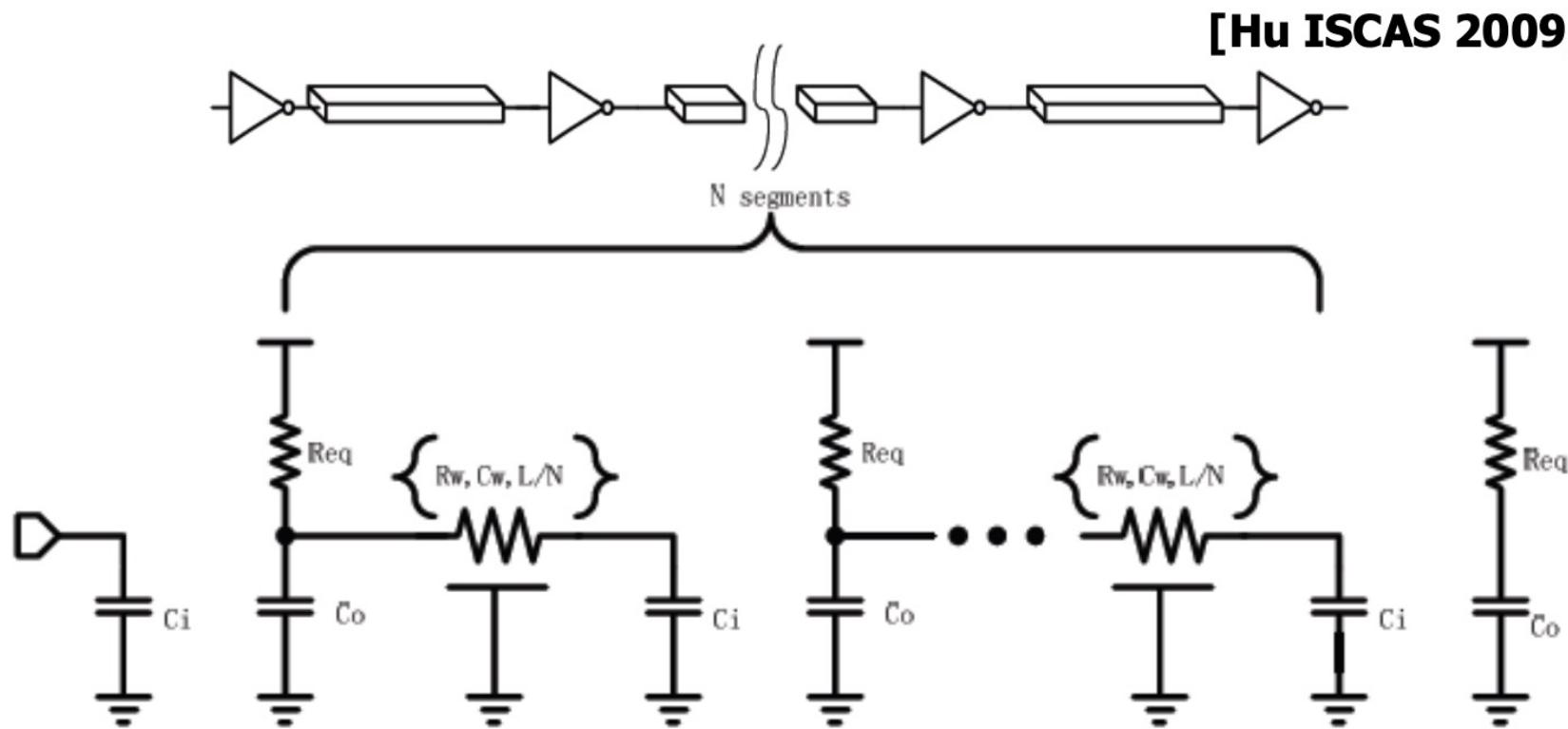
[Razavi]

# Final Notes: Clock Distribution

---

[Elad Alon]

# Inverter Chain Distribution

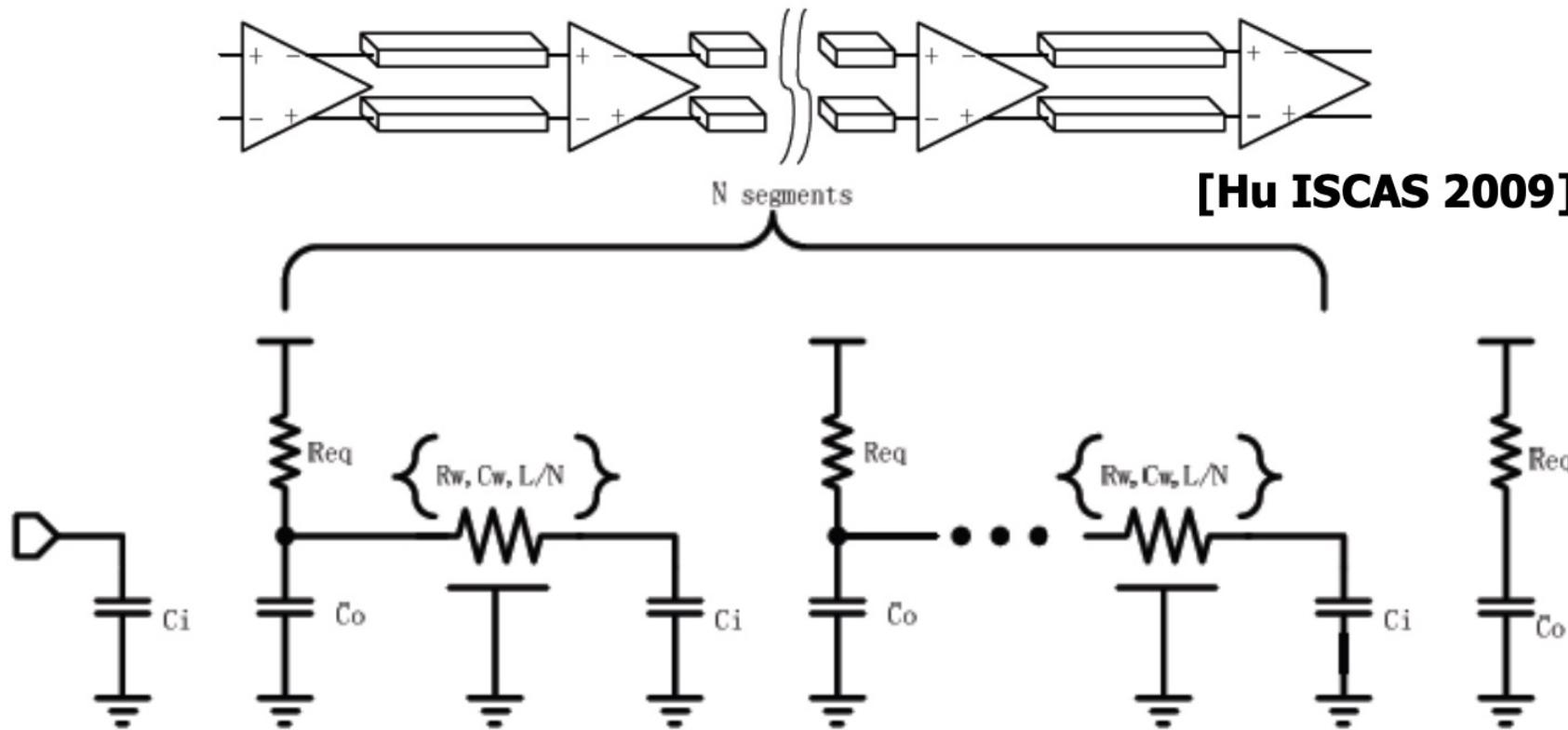


- Instead of driving the long low-bandwidth wire with one huge inverter, break wire up into  $N$  segments driven by  $N$  inverters

9

[Sam Palermo]

# CML Chain Distribution



- Relative to inverter-based buffers, low-swing CML buffers offer increased bandwidth and PSRR
- Same model used to analyze CML distribution

[Sam Palermo]

# Clock Distribution Performance Comparison

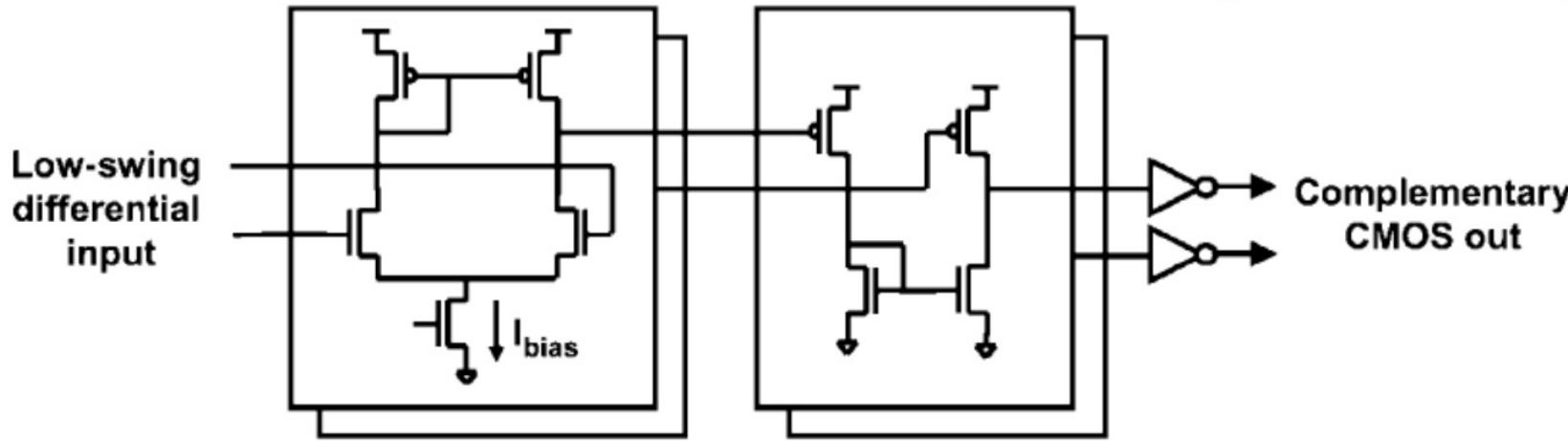
[Hu ISCAS 2009]

Technology	1.2V 90nm CMOS		
Methods (with optimal tradeoff)	Performance		
	jitter(ps)	delay(ps)	power(mW)
Inverter chain (N=3, m=128)	36	321	11.5
CML chain (N=2, m=1)	1	221	2
Transmission line	0.18	43	4
Inductive load (L=6nH, Q=2)	0.42	55	4
CDW (Cc=50f)	1.98	116	0.62

- Transmission-line distribution offers best jitter and delay performance
- CDW offers minimum jitter-power and delay-power product
- Note, everything but inverter-chain distribution is low-swing
- If CML2CMOS converter is not designed well, that can kill your nice distribution network performance

# CML2CMOS Converter (1)

[Balamurugan JSSC 2008]

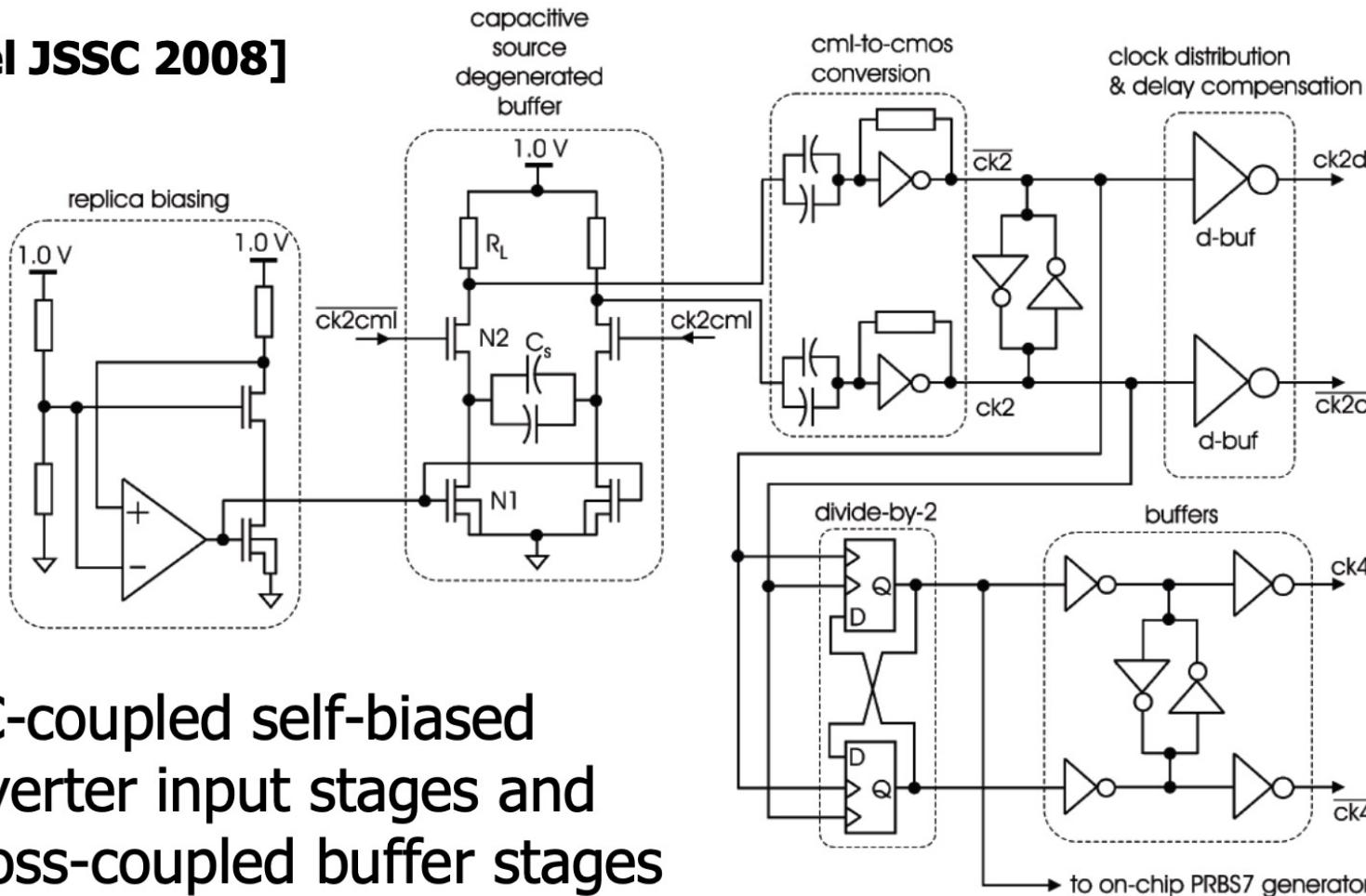


- Differential input stage followed by high-swing output stage
- Can be sensitive to power-supply noise and reduce jitter benefits of low-swing distribution techniques
- Often require some type of duty-cycle control

[Sam Palermo]

# CML2CMOS Converter (2)

[Koscel JSSC 2008]

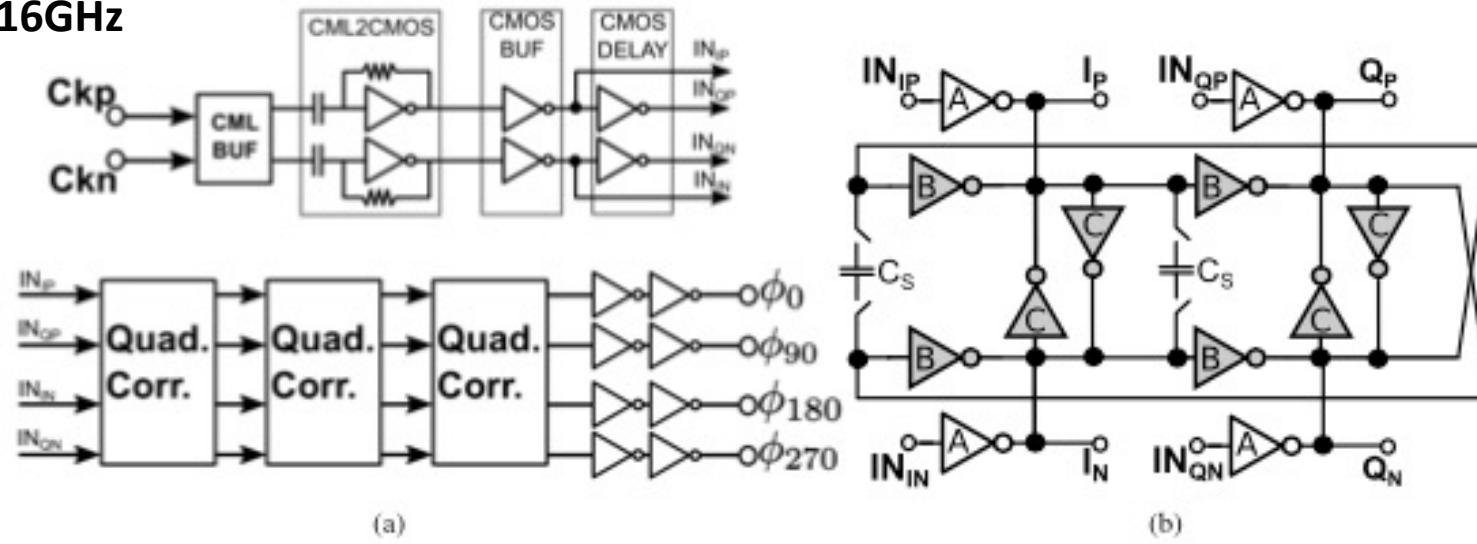


- AC-coupled self-biased inverter input stages and cross-coupled buffer stages can help improve duty cycle performance

[Sam Palermo]

# Examples: IBM 64Gb/s NRZ Rx

16GHz



19. Clock path. (a) Block diagram. (b) Quadrature corrector stage.

TABLE I

RX POWER BREAKDOWN AT 64 Gb/s

	VDAH (1V)	VDAL (0.9V)	VDD (0.9V)	Energy (fJ/b)
TIA	3.5 mA			55
VGA	21.5 mA			335
12 SLICERS		18 mA		253
CLK BUF		25 mA		351
ALIGNER		9.7 mA		136
VDACs		1 mA		14
DFE logic			12.3 mA	172
DMUX 4:32			7.1 mA	100
Total	25 mA	53.7 mA	19.4 mA	1416

[IBM, JSSC 2017]

# Example: Intel 112Gb/s PAM4 TRx

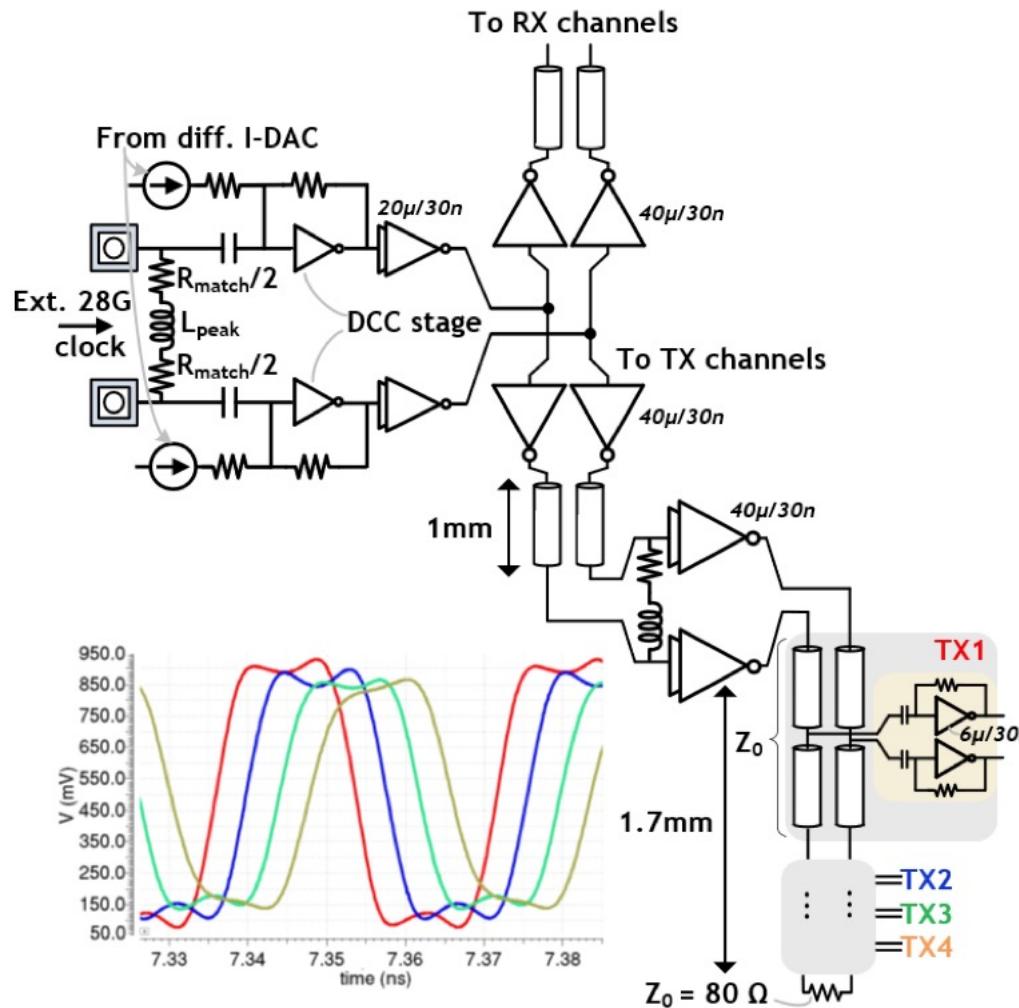


Fig. 14. Half-rate external clock buffering and global distribution.

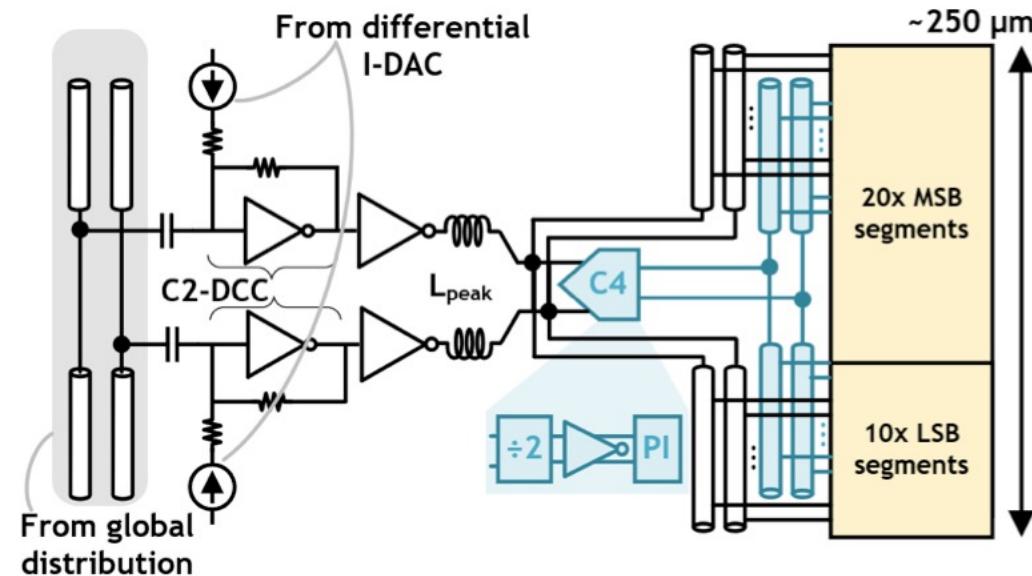
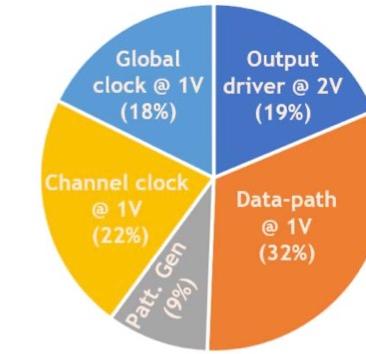
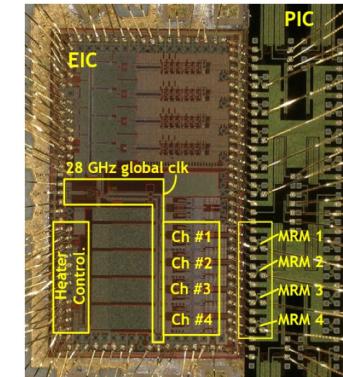


Fig. 13. Per-channel half- and quarter-rate clock distribution.

[Intel, JSSC 2022]