Topics in IC Design

9.1 Samplers and Metastability

Deog-Kyoon Jeong

dkjeong@snu.ac.kr

School of Electrical and Computer Engineering

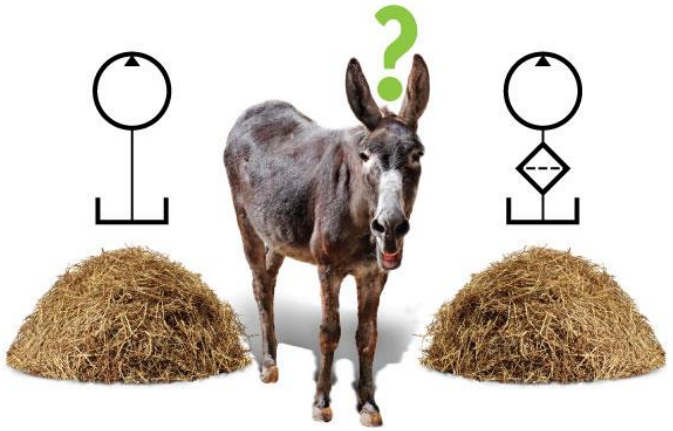
Seoul National University

2020 Fall

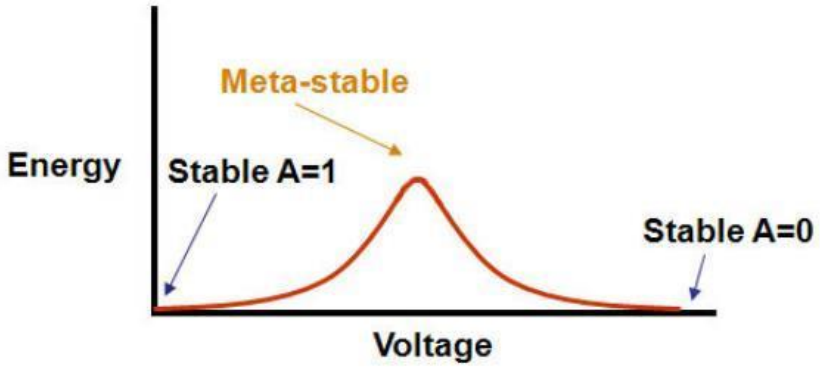
Compliments to Sanghee Lee

What is Metastability?

- **Buridan's donkey**



- **In a latch, hesitating decision between 0 and 1.**



Sampling Asynchrounous Signals

- **Signal from external world is asynchronous with Clk**
- **Asynchronous sampling occurs for the signal generated from different clock**
  - **Inputs to FFs must be stable when sampled by clkB**
  - **Sampling at aperture or keep-out region must be avoided**
- **D-FF has setup and hold times**

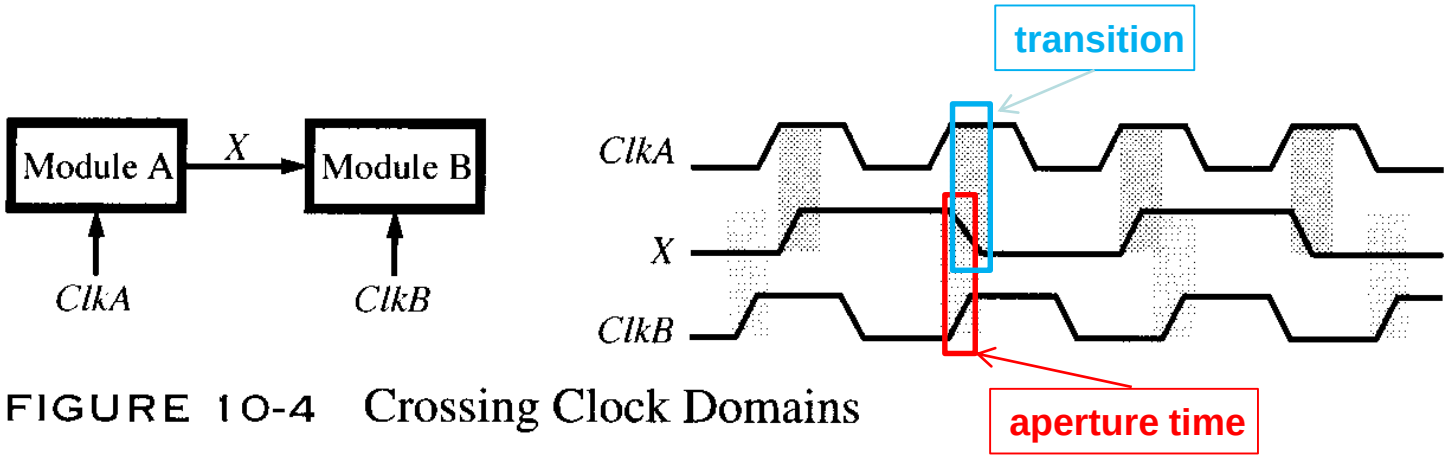


FIGURE 10-4 Crossing Clock Domains

Synchronizer

- **Inputs to synchronous systems must be synchronized - qualified in value and time.**
- **Inputs to synchronous systems must be synchronized - qualified in value and time.**
- **A synchronizer is similar to D-FF with sufficient delay**
- **X is sampled on the falling edge of clk and waits half cycle to decide and then asserts result at the rising edge of clk.**

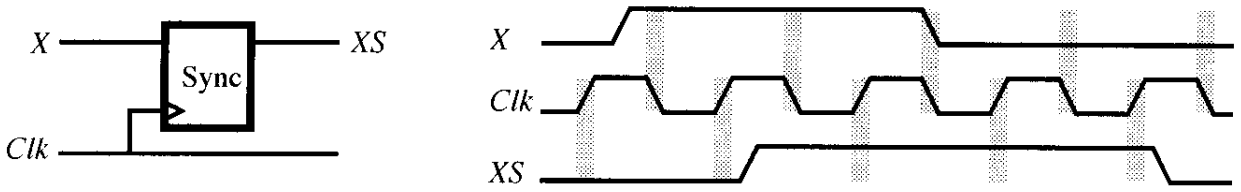
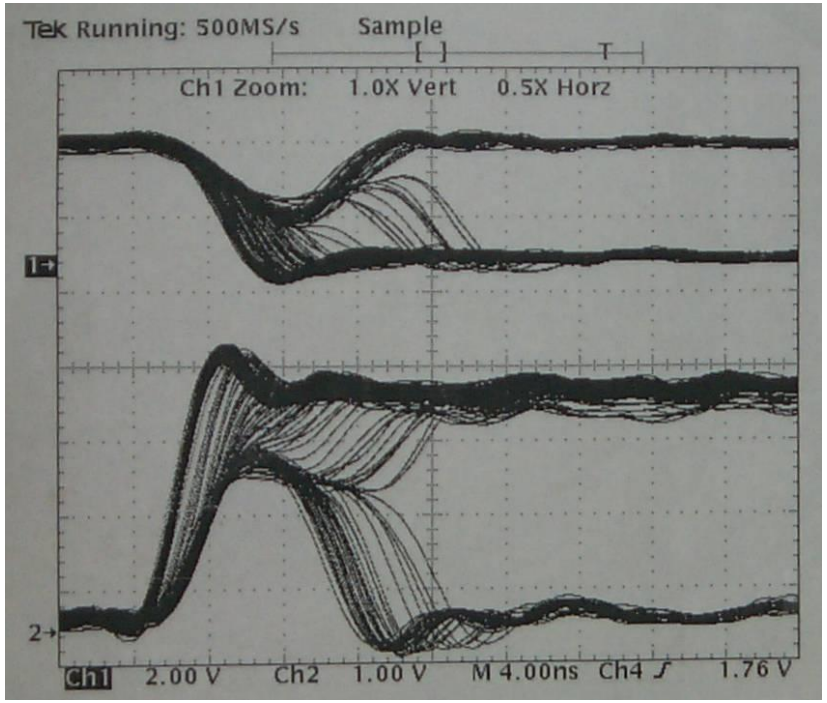


FIGURE 10-3 Sampling Asynchronous Signals

Synchronization Failure

- **When input level is close to the threshold, the decision can take longer than allotted time, which is called synchronization failure.**



Synchronizer Dynamics

- **Initially both inputs are low and both outputs are high.**
  - **When *A* rises first, *Afirst'* goes low and *Bfirst'* remains at high.**
  - **When *A* and *B* rise nearly at the same time, during Δt,**
  - $\Delta V_1 = K_s \Delta t$ **where** $K_s = I/C$
  - **When B arrives, NAND gates become cross-coupled inverters.**
  - $\Delta V(t) = \Delta V_1 \exp(t/\tau_s)$ **where τ_s is regeneration time constant**
  - **Decision time t_d required for ΔV to reach unit voltage is**

$$t_d = -\tau_s \log(\Delta V_1)$$

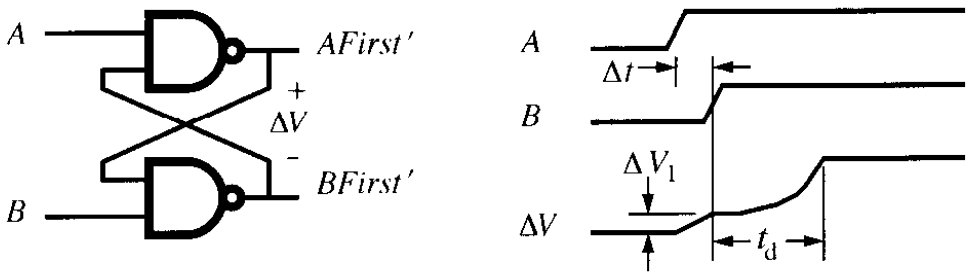
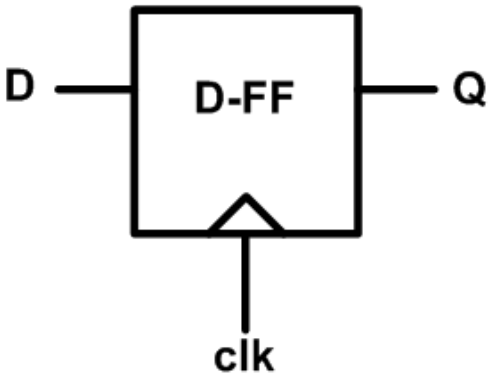
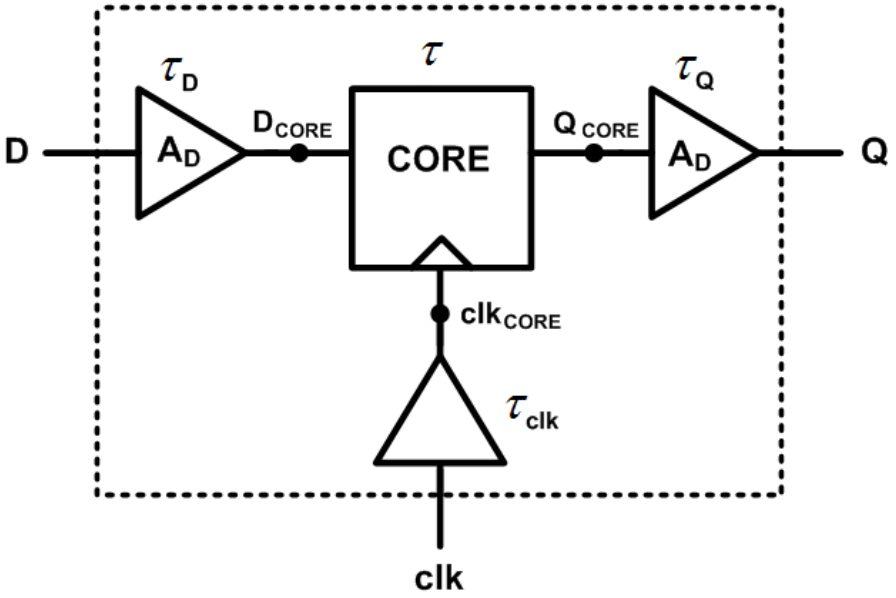


FIGURE 10-5 Synchronizer Dynamics

(Ref) Basic Model of Synchronizer

- Same function as positive-edge triggered D-FF
  - clk=0: outputs old data and tracks new data
  - clk=1: samples new data, regenerates, and outputs result
- The data is regenerated in synchronization core through a positive feedback loop





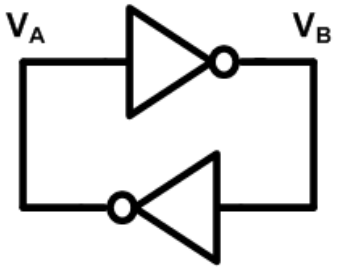
(Ref) Synchronization Core

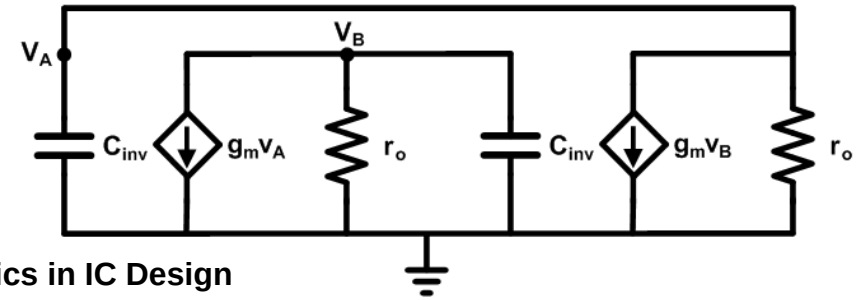
- **Pair of inverters connected back to back**
  - **Positive feedback loop of two inverters with unity-gain bandwidth f_1 and time constant τ.** $f_1 = \frac{1}{2\pi\tau}$
  - **Two time constants in the small-signal equivalent circuit**

$$\pm \frac{C_{inv}}{g_m + 1/r_o}$$

  - **Thus, metastability regeneration time constant is**

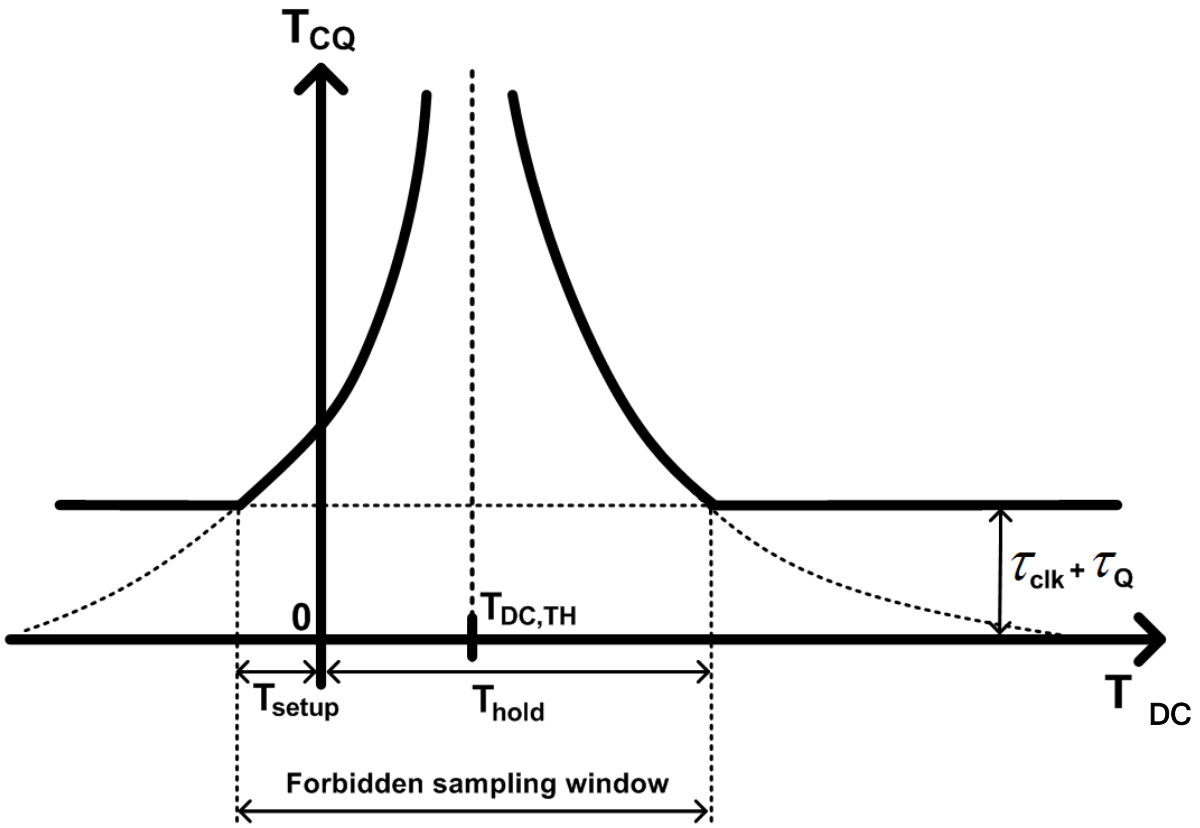
$$\tau = \frac{C_{inv}}{g_m + 1/r_o}, \quad f_1 = \frac{1}{2\pi\tau}$$



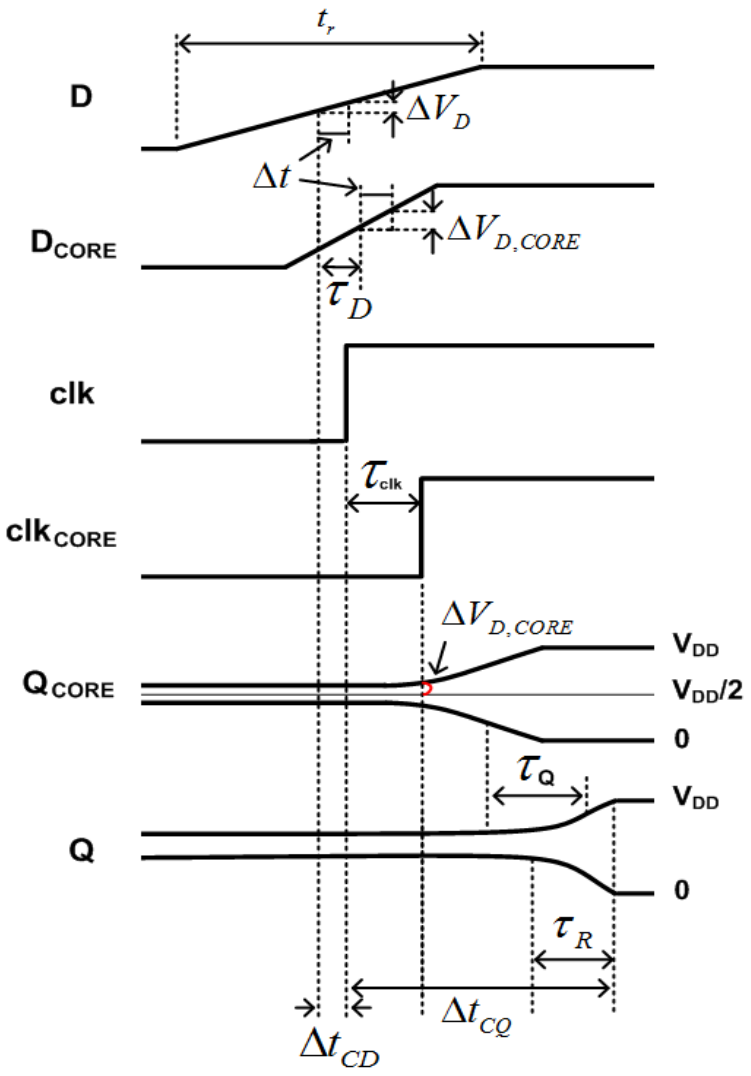


(Ref) Clock to Q Delay

- **Metastability appears as increased clk-to-Q delay, T_{CQ}, when setup or hold time is violated**



(Ref) Metastable State



- **At sampling time Δt= t_{DC}-$t_{DC,TH}$ from threshold crossing time $t_{DC,TH}$**

$$\Delta V_D = V_{DD} \cdot \Delta t / t_r, \quad \Delta V_{D,CORE} = \frac{1}{2} A_D V_{DD} \cdot \Delta t / t_r$$

- **In the regeneration phase**

$$V_{Q,CORE}(t) = \frac{V_{DD}}{2} + \Delta V_{D,CORE} \cdot e^{t/\tau}$$

$$V_Q(t+\tau_Q) = A_Q V_{Q,CORE}(t) = \frac{V_{DD}}{2} + \frac{1}{2} A_Q A_D V_{DD} \cdot \Delta t / t_r \cdot e^{t/\tau}$$

- **Regeneration time**

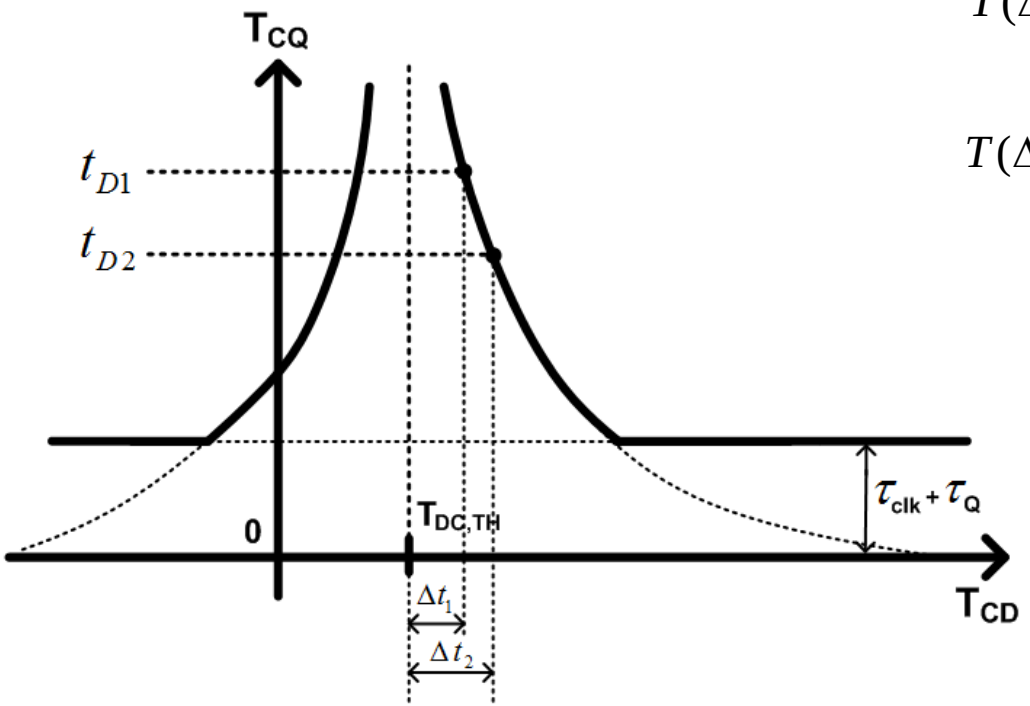
$$V_Q(\tau_Q + \tau_R) = V_{DD}, \quad \tau_R = \tau \ln\left(\frac{t_r}{A_Q A_D \Delta t}\right)$$

(Ref) Measurement of τ

- **Metastability resolving time constant τ**

$$T(\Delta t) = \tau_{ck} + \tau_Q + \max\left(\tau \ln(\frac{t_r}{A_Q A_D \Delta t}), 0\right)$$

In case of none-metastable state



$$T(\Delta t_1) = \tau_{ck} + \tau_Q + \tau \ln(\frac{t_r}{A_Q A_D \Delta t_1}) = t_{D1}$$

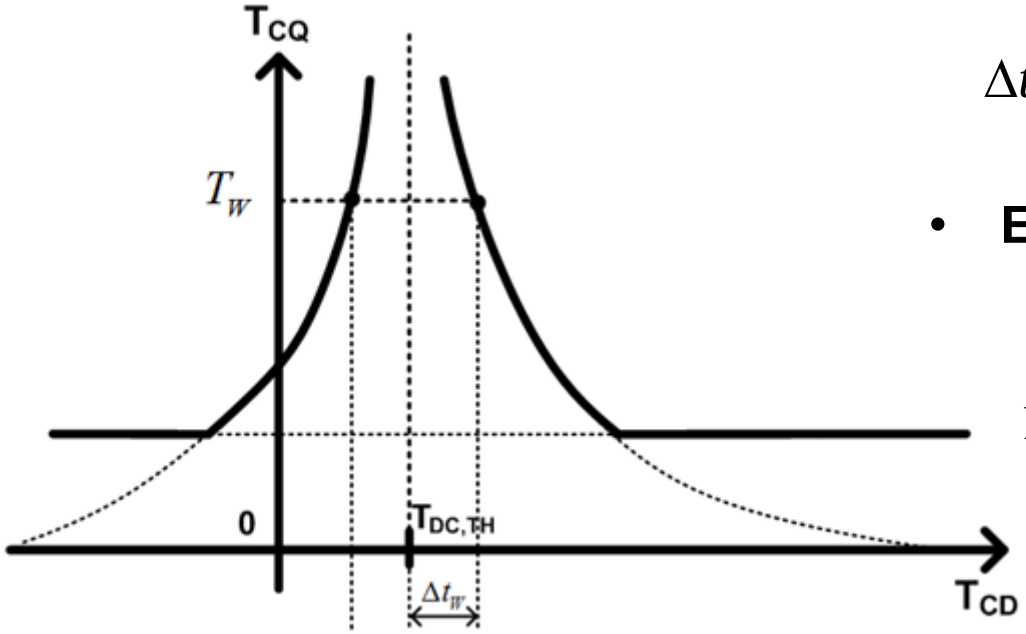
$$T(\Delta t_2) = \tau_{ck} + \tau_Q + \tau \ln(\frac{t_r}{A_Q A_D \Delta t_2}) = t_{D2}$$

From $t_{D1}, t_{D2}, \Delta t_1, \Delta t_2$

$$\tau = \frac{t_{D1} - t_{D2}}{\ln(\Delta t_2 / \Delta t_1)}$$

(Ref) Calculation of Error Probability

- **Probability of clk-to-Q delay exceeding wait time** T_W



$$T(\Delta t_W) = \tau_Q + \tau_{ck} + \tau \ln\left(\frac{t_r}{A_Q A_D \Delta t_W}\right) = T_W$$

$$\Delta t_W = \frac{t_r}{A_D A_Q} e^{(\tau_{ck}+\tau_Q-T_w)/\tau}$$

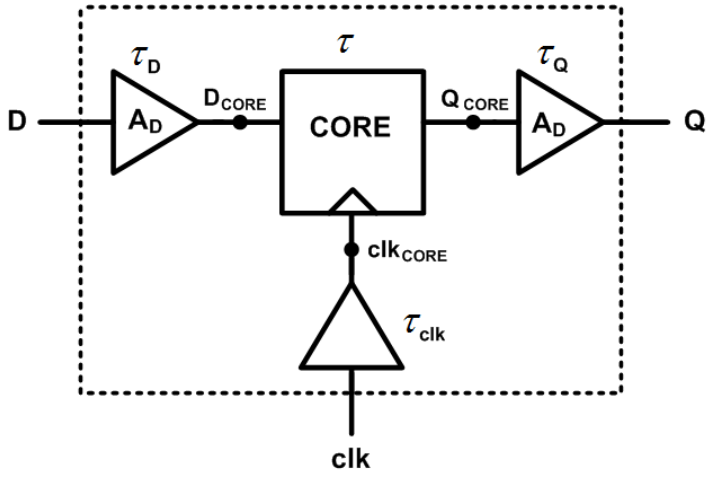
- **Error probability**

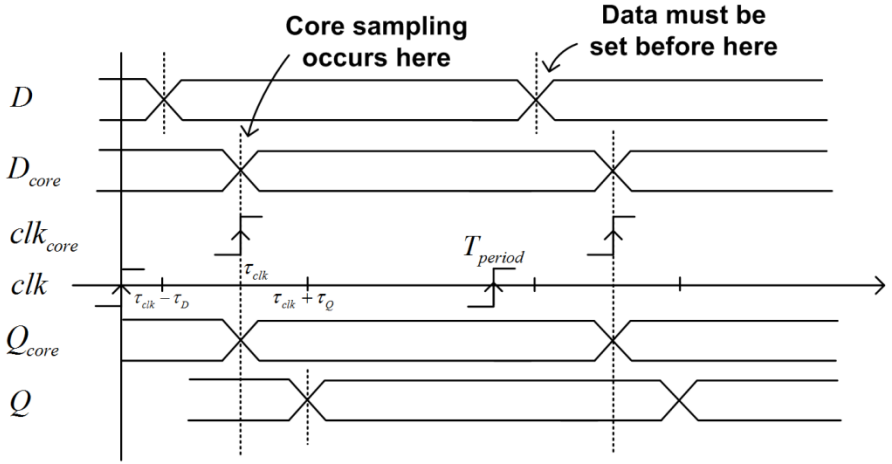
$$\mathrm{Prob}(T_W) = \frac{\frac{2t_r}{A_D A_Q} e^{(\tau_{ck}+\tau_Q-T_w)/\tau}}{T_{period}}$$

(Ref) Error Probability

- **Timing window expands due to sampling.**
  - **Error occurs when clock to Q delay exceeds** $T_{period} + (\tau_{ck} - \tau_D)$

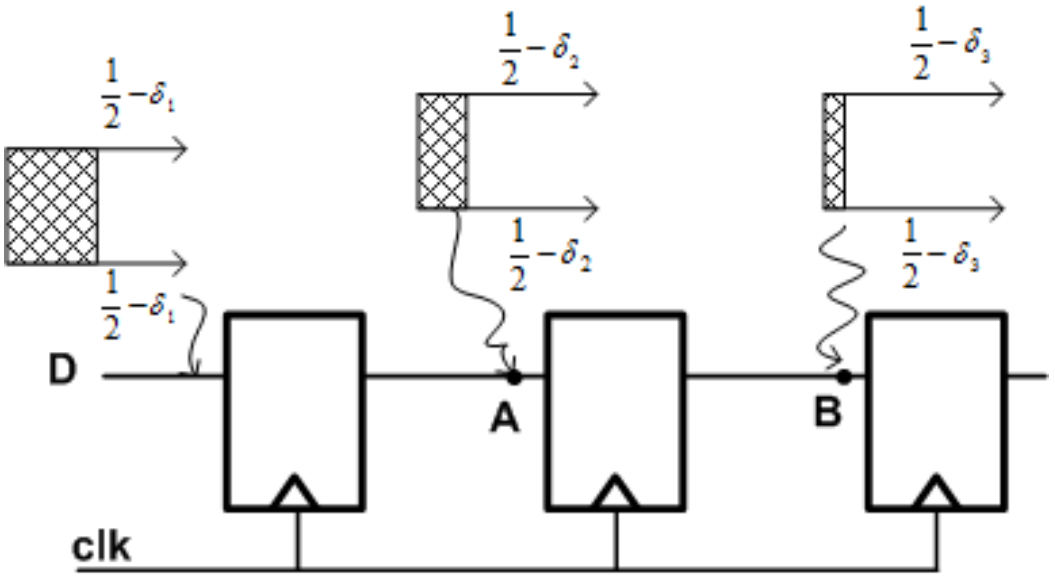
$$\Pr ob(error\ at\ Q) = \frac{\frac{2t_r}{A_D A_Q} e^{(\tau_{ck} + \tau_Q - (T_{period} + \tau_{ck} - \tau_D))/\tau}}{T_{period}} = \frac{\frac{2t_r}{A_D A_Q} e^{(\tau_D + \tau_Q - T_{period})/\tau}}{T_{period}}$$





(Ref) Error Probability

- **Cascaded Synchronizer**
  - **Error probability at A →** $\mathrm{Prob}(C2Q\,delay\,exceeds\,(T_{period}+\tau_{ck}-\tau_{D}))$
  - **Error probability at B →** $\mathrm{Prob}(C2Q\,delay\,exceeds\,(T_{period}+\tau_{ck}+\tau_{Q}))$



(Ref) Error Probability

- **Stacked synchronizer**
  - **Error probability at A →** $\dfrac{\dfrac{2t_r}{A_D A_Q} e^{(\tau_D+\tau_Q-T_{period})/\tau}}{T_{period}}$
  - **Error probability at B →** $\dfrac{\dfrac{2t_r}{A_D A_Q} e^{(\tau_D+\tau_Q-T_{period})/\tau}}{T_{period}} \cdot \left[\dfrac{1}{A_D A_Q} e^{(\tau_{clk}+\tau_Q-T_{period})/\tau} \right]$

$$= \frac{2t_r}{T_{period}} \cdot \left(\frac{1}{A_D A_Q} \right)^2 \cdot e^{\left(\tau_D-\tau_{clk}-2(T_{period}-\tau_{clk}-\tau_Q)\right)/\tau}$$

  - **Error probability at N'th stage**

$$\frac{2t_r}{T_{period}} \cdot \left(\frac{1}{A_D A_Q} \right)^N \cdot e^{\left(\tau_D-\tau_{clk}-N(T_{period}-\tau_{clk}-\tau_Q)\right)/\tau}$$

Metastability

- **If the two inputs rise exactly at the same time, ΔV_1 is zero and no signal to amplify.**

$$\Delta V(t) = \Delta V_1 \exp(t/\tau_s)$$

- **The circuit with $\Delta V(t)=0$ is in a metastable state.**
- **Tiny amount of perturbation (noise) will drive the circuit into either one of the two stable state.**

Probability of Sync Failure

- **Wait time is the cycle time of A**

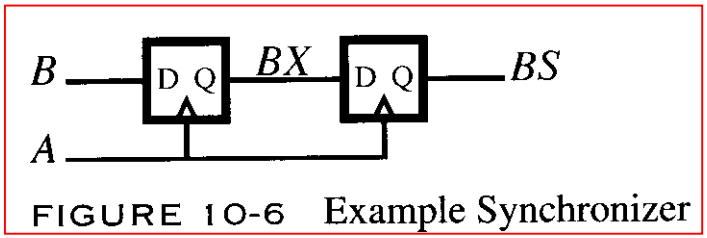
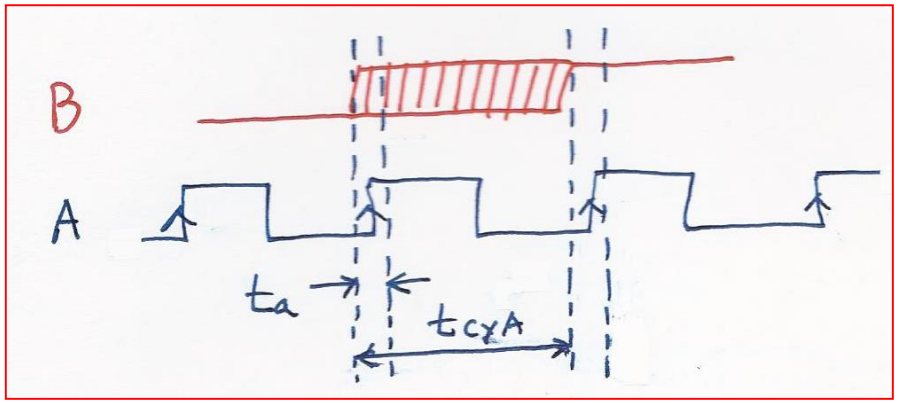


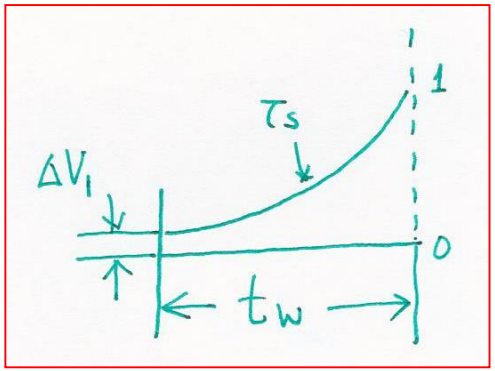
FIGURE 10-6 Example Synchronizer

- **Prob (B rises/falls during aperture time) = $t_a/t_{cyA}=t_a \cdot f_A$**



Probability of Sync Failure

- **Initial condition of ΔV that grows exponentially and reaches 1 after wait time of $t_w = \Delta V_1 = 1 \cdot \exp(-t_w/t_s)$**



- **Prob (Edge of B causes metastability after t_w)**
  $= t_a \cdot f_A \cdot \Delta V_1 = t_a \cdot f_A \cdot exp(-t_w/t_s)$
- **Frequency of synchronization failure**
  $= t_a \cdot f_A \cdot f_B \cdot exp(-t_w/t_s)$

Example Sync Calculation

- **Synchronized signal *BS* has a very low probability of failure**

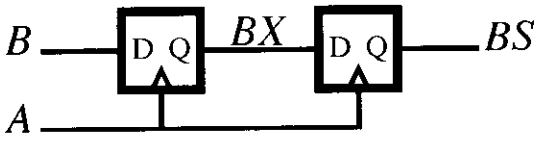


FIGURE 10-6 Example Synchronizer

Regeneration time constant t_s = 200ps

TABLE 10-1 **Example Synchronizer Calculation**

Symbol	Description	Value	Units
f_A	Event frequency on line A	100	MHz
f_B	Event frequency on B	1	MHz
t_a	Aperture time of flip-flop	200	ps
t_w	Waiting time	10	ns
P_{sf}	Probability of synchronization failure for each event on B	3.86×10^{-24}	
f_{sf}	Frequency of synchronization failures	3.86×10^{-18}	Hz
MTBF	Time between synchronization failures	2.59×10^{17}	s

$t_a \cdot f_A \cdot \exp(-t_w/t_s)$

Completion Detection

- **During metastability (decision time), outputs are not validated**
  - **Only after the outputs diverge beyond decision voltage V_d, completion signal *Done* is asserted.**

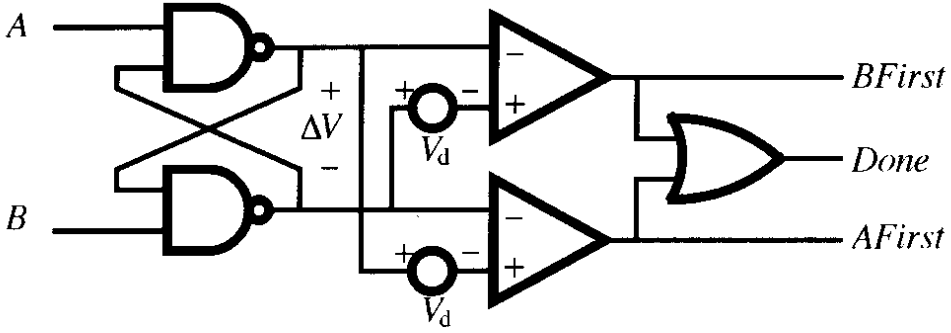


FIGURE 10-7 Completion Detection

  - **Asynchronous circuit following arbiter operates only after Done is asserted. No sync failure!**
  - **Done is a request signal for the next stage**
  - **Less latency, no worst-case waiting time.**

Common Sync Mistakes

- **Properly designed synchronizer exhibits very low probability of failure**
- **Common errors in synchronization**
  - **Using dynamic latch w/o regeneration**
  - **Load cap inside regeneration loop**

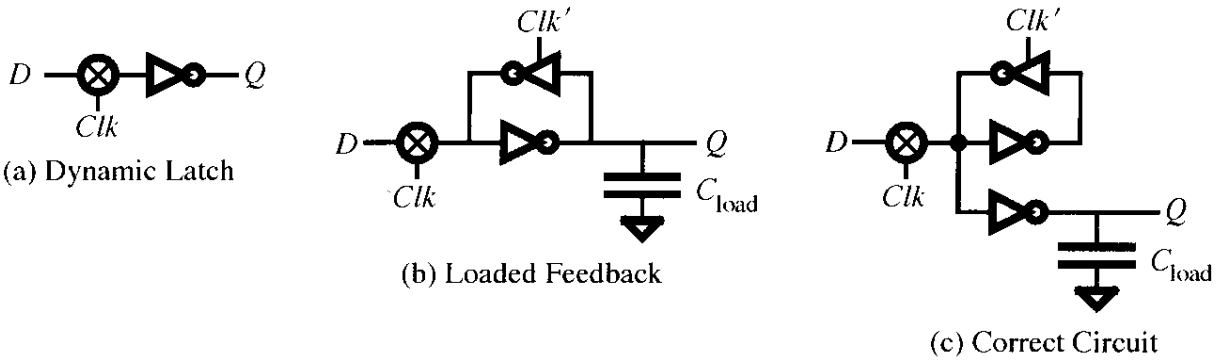
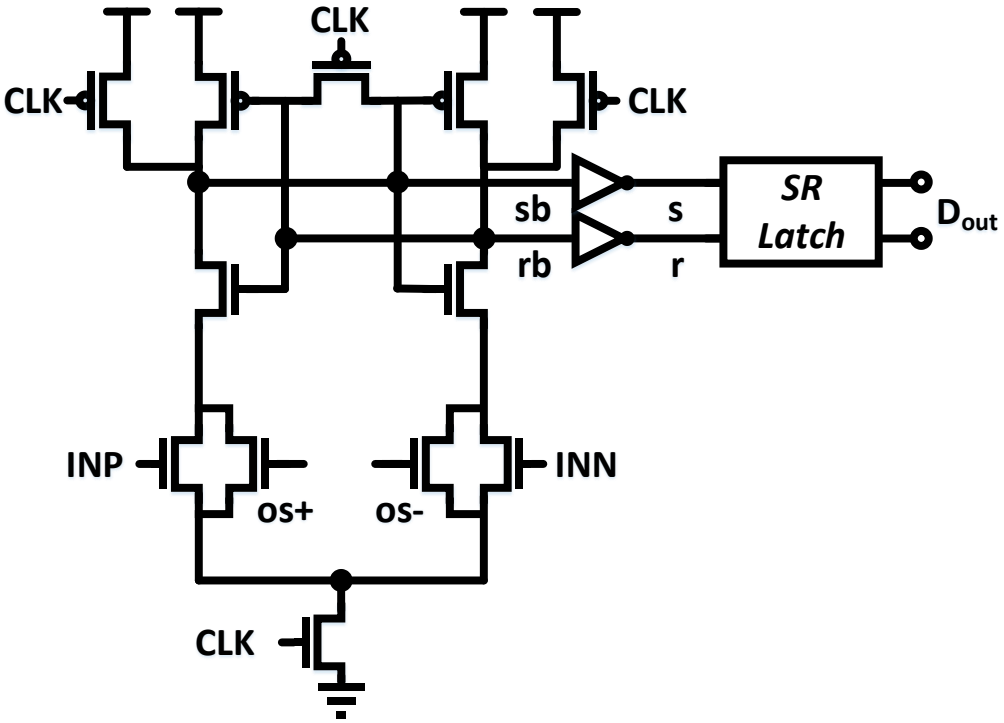


FIGURE 10-8 Common Synchronizer Errors

StrongArm Latch(schematic)

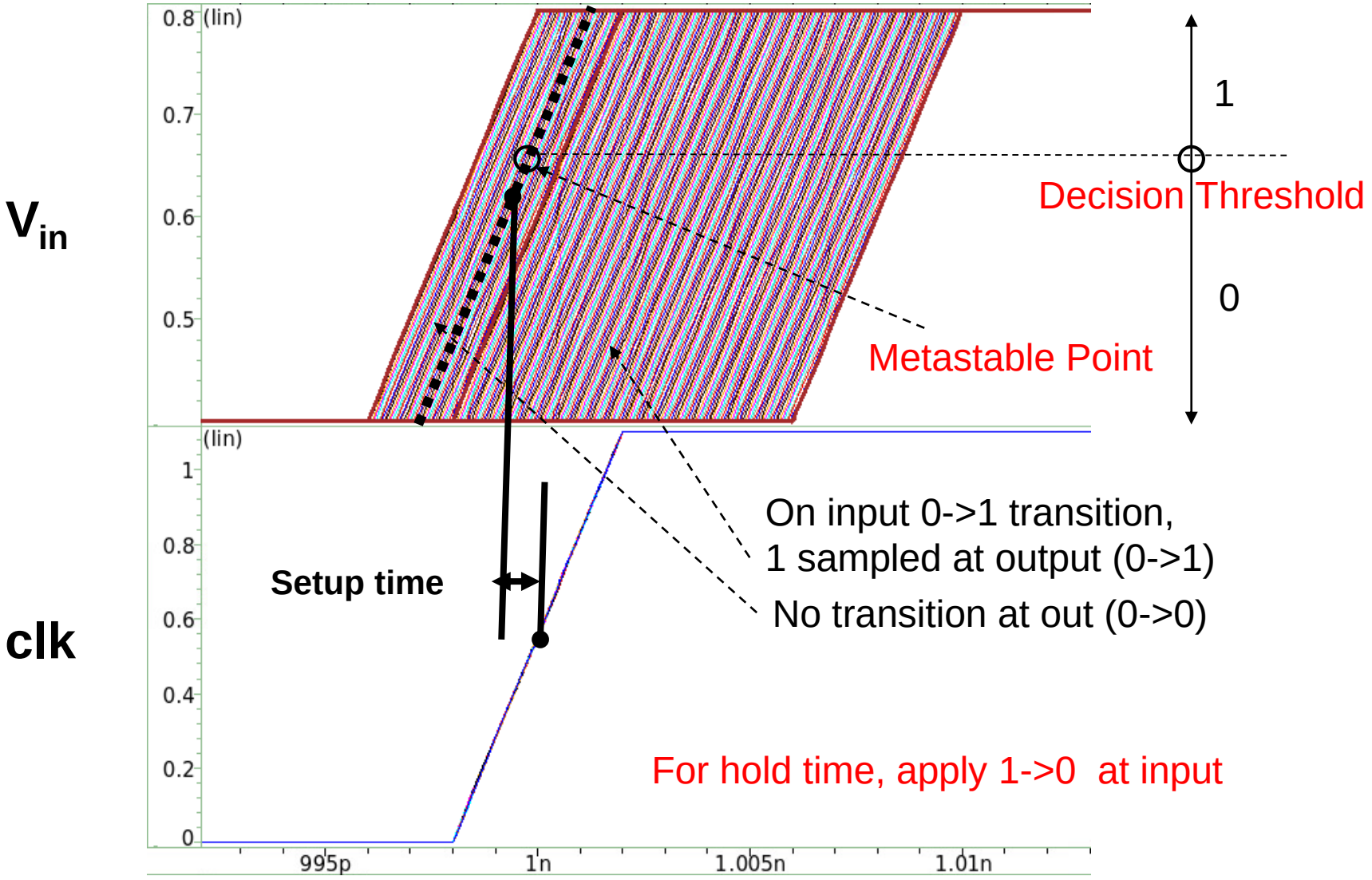


When CLK=L (precharge), both sb and rb are H.

When CLK=H (evaluation), one goes down to L and the other remains H.

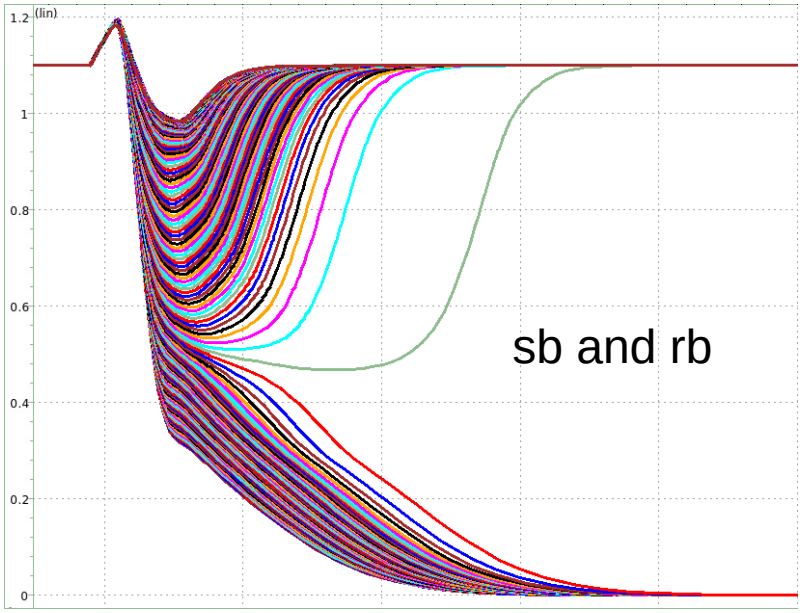
When CLK=L, data must be held. Both s and b are L. So SR-latch with NOR gates must be used.

StrongARM Latch t_{setup}/t_{hold} Sim.

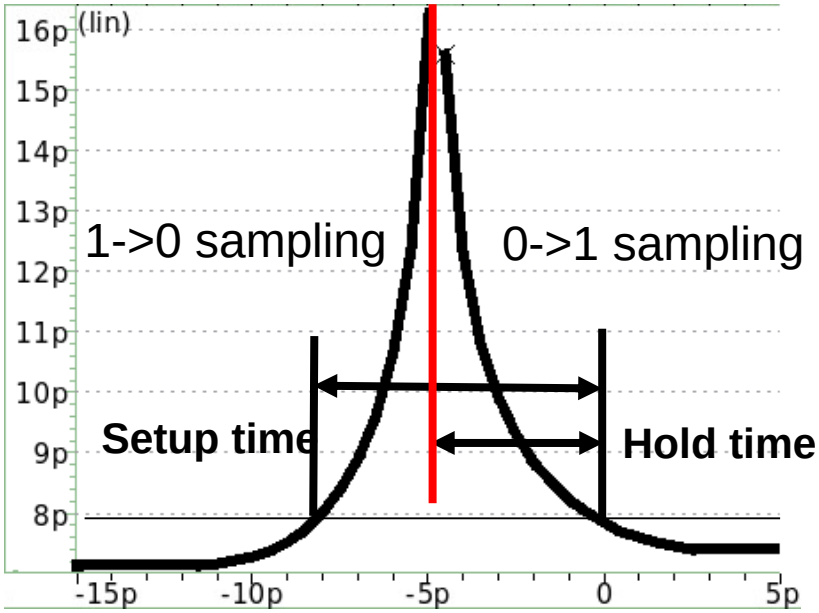


StrongARM Latch t_{setup}/t_{hold} Sim.

- **Decision takes longer due to setup/hold time violation.**
- **Metastable point is varied with PVT variation.**



Metastable Point



Offset of StrongARM Latch

- **“Offset” is due to mismatch in the supposedly identical matched pair.**

- **From process variations:**
  - Random dopant fluctuations
  - Interface-state density fluctuations
  - Mask misalignment, W and L variations

- **Offset voltage is determined by mismatch of**
  - Threshold voltage, transconductance, capacitance

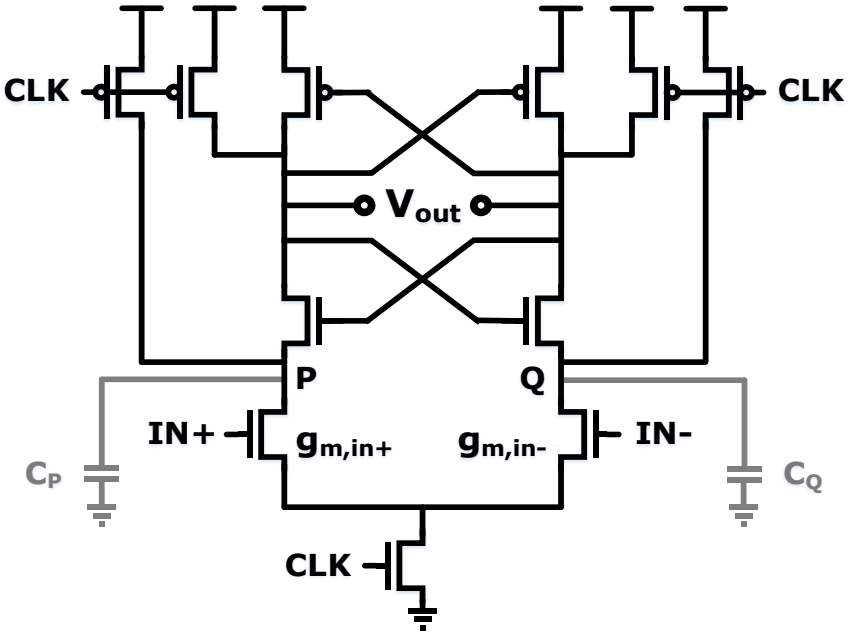
Offset of StrongARM Latch

- **Different discharge rates at P and Q**

$$V_P = V_{DD} - \frac{g_{m1}(V_{in1} - V_{in2})}{2C_P} t - \frac{I_{CM}}{C_P} t$$

$$V_Q = V_{DD} + \frac{g_{m2}(V_{in1} - V_{in2})}{2C_Q} t - \frac{I_{CM}}{C_Q} t$$

$$V_P - V_Q = -\frac{g_{m1,2}}{2} \cdot \frac{C_P + C_Q}{C_P C_Q} (V_{in1} - V_{in2})\, t + \frac{C_P - C_Q}{C_P C_Q} I_{CM}\, t$$



References

1. H. J. Veendrick, “The behavior of flip-flops used as synchronizers and prediction of their failure rate", IEEE Journal of Solid-State Circuits, Volume: 15, Issue: 2, pp. 169-176, 1980.
2. J. Montanaro et al., “A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor", IEEE Journal of Solid State Circuits, Volume: 31, Issue: 11, pp. 1703-1714, 1996.
3. R. Singh and N. Bhat, “An offset compensation technique for latch type sense amplifiers in high-speed low-power SRAMs”, IEEE Transaction on Very Large Scale Integration Systems, Volume: 12, Issue: 6, pp. 652-657, 2004.
4. J. Stephenson et al., “Understanding metastability in FPGAs”, White Paper WP-01082-1.2, Altera Ltd, 2009.
5. J. Kim et al., “Simulation and analysis of random decision errors in clocked comparators", IEEE Transactions on Circuits and Systems 1: Regular Papers, Volume: 56, Issue: 8, pp. 1844-1857, 2009.
6. B. Razavi, “The StrongARM latch [a circuit for all seasons]", IEEE Solid-State Circuits Magazine, Volume: 7, Issue: 2, pp. 12-17, 2015.

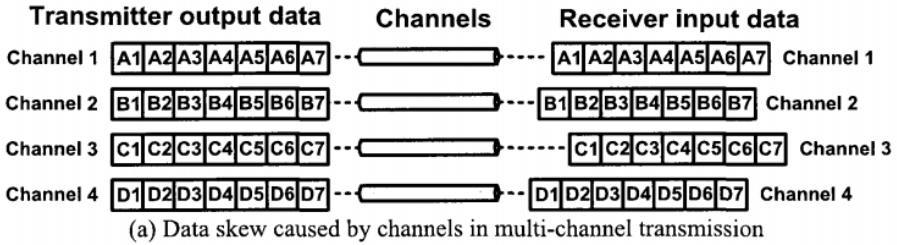
Topics in IC Design

8.2 Serializer

Deog-Kyoon Jeong
dkjeong@snu.ac.kr
School of Electrical and Computer Engineering
Seoul National University
2020 Fall
Compliments to Jung Min Yoon

What is a serializer?

- **Takes *n*-bits of parallel data changing at rate *y* and transforms into a stream at a rate of *n*y***
- **Converts input parallel data-bits into serial for inter-IC transmission across lossy channel**
- **Serial link reduces the number of required pins**



(a) Data skew caused by channels in multi-channel transmission

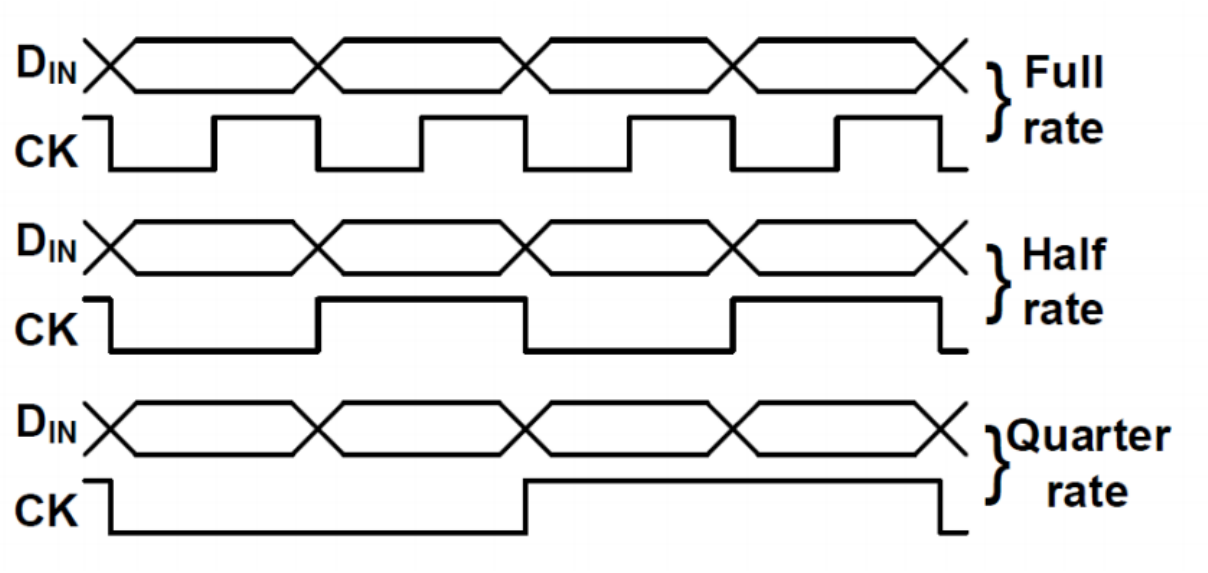
(b) Data skew is eliminated through serializer/deserializer

Figure 1-1 Data skew elimination with SerDes

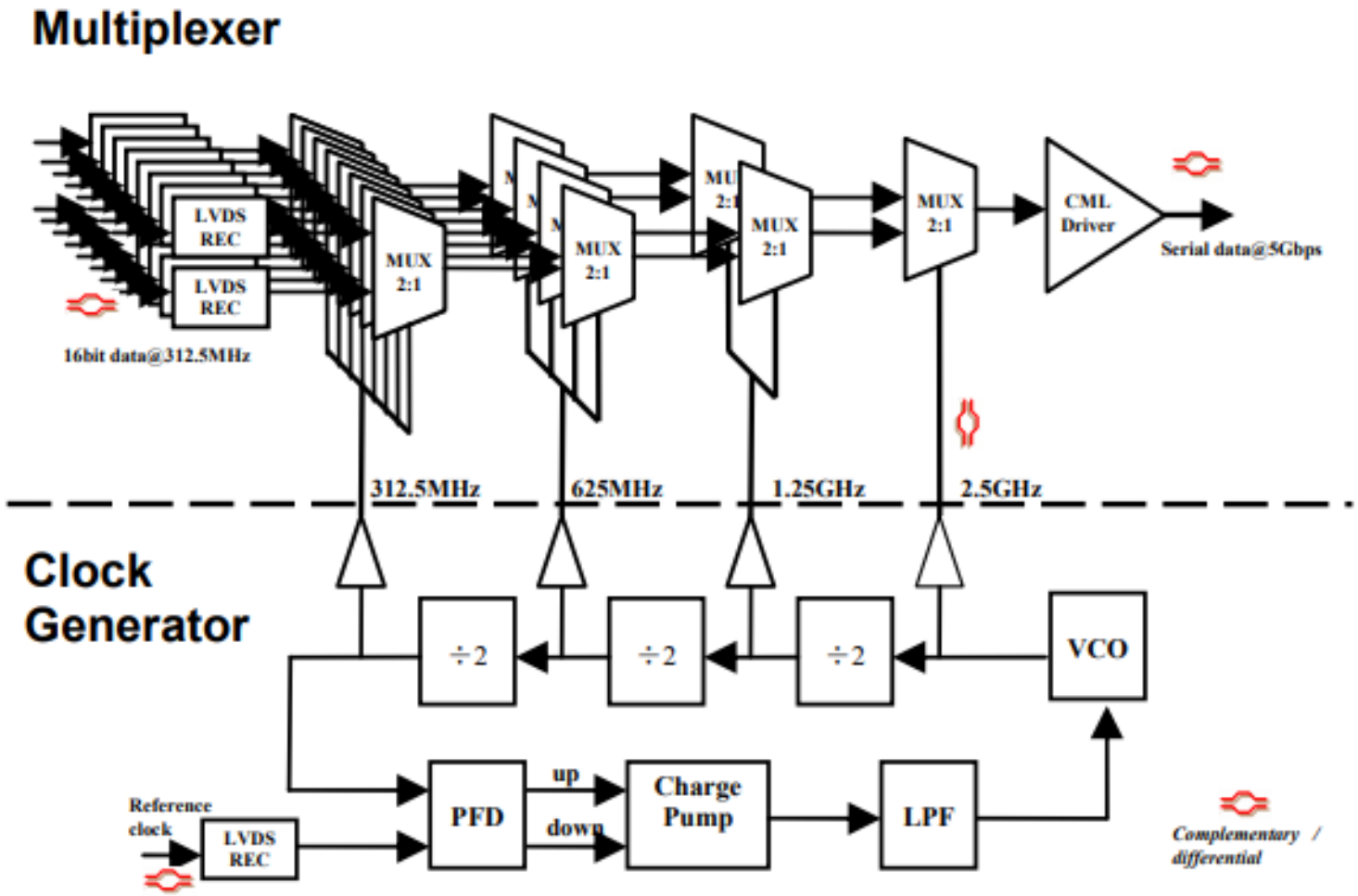
[3] 'A high speed serializer deserializer design', Y. Luo, 2010

Serializer for high-speed links

- **In serializing data, clock rate is important**
- **Usually uses half-rate clock (2:1) / quarter-rate clock (4:1) depending on the process, power, and speed.**

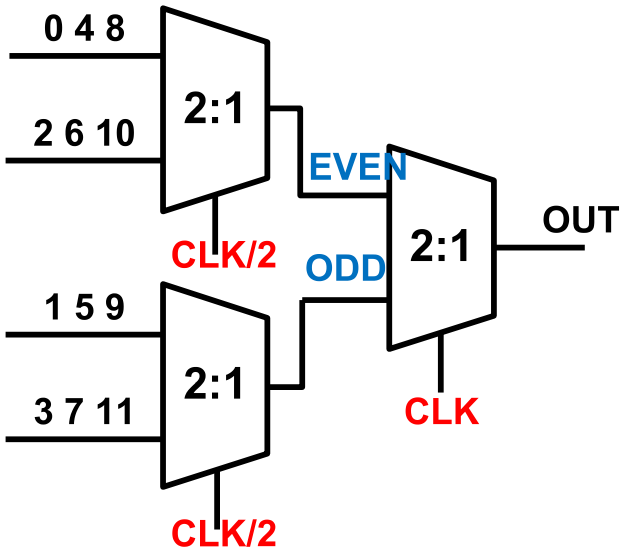


Typical Serializer Architecture

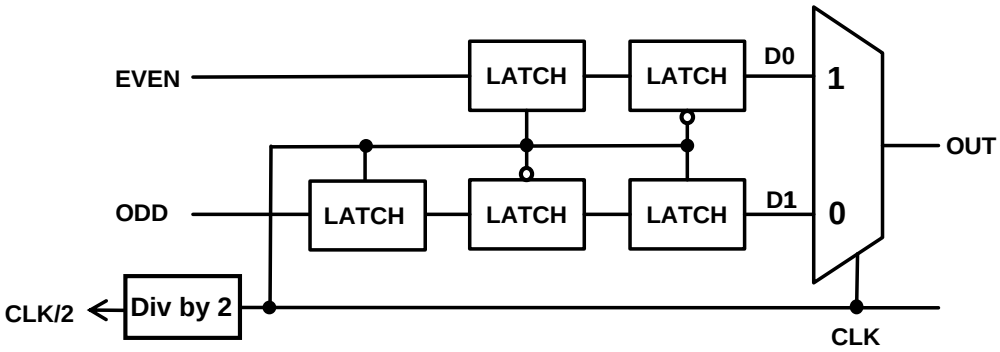


'Development of A 16:1 serializer for data transmission at 5 Gbps', Gong et al

2:1 Serializer Using MUX

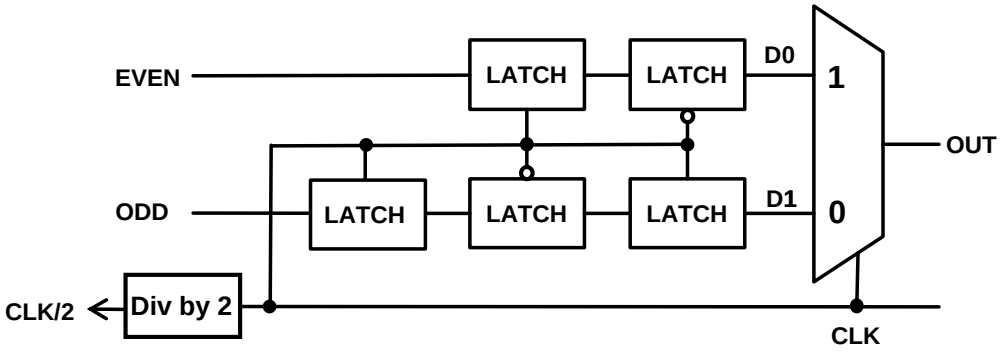


(a) Schematic

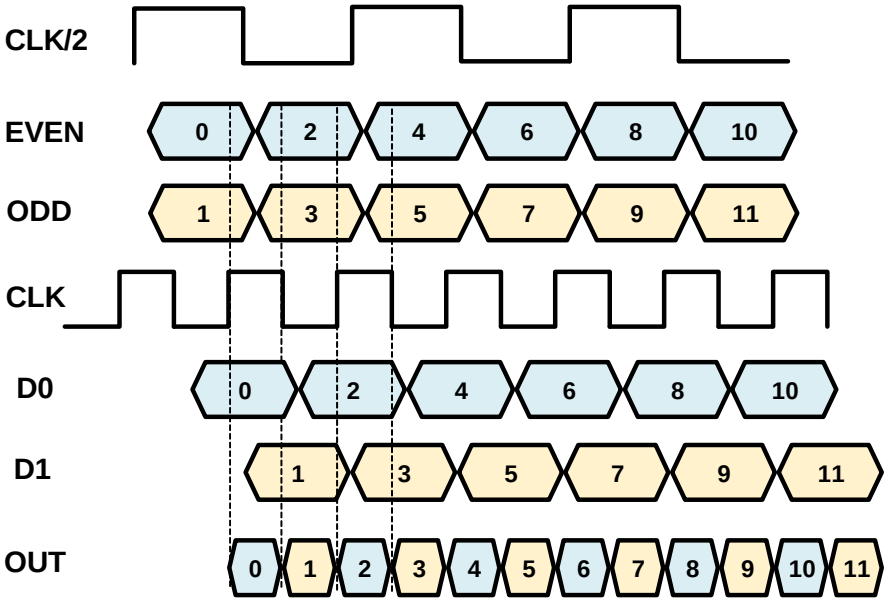


(b) Block diagram of 2:1 Serializer

2:1 Serializer Using MUX



(a) Block diagram of 2:1 Serializer



(b) Timing diagram of 2:1 Serializer

Timing Constraints for N:1 MUX

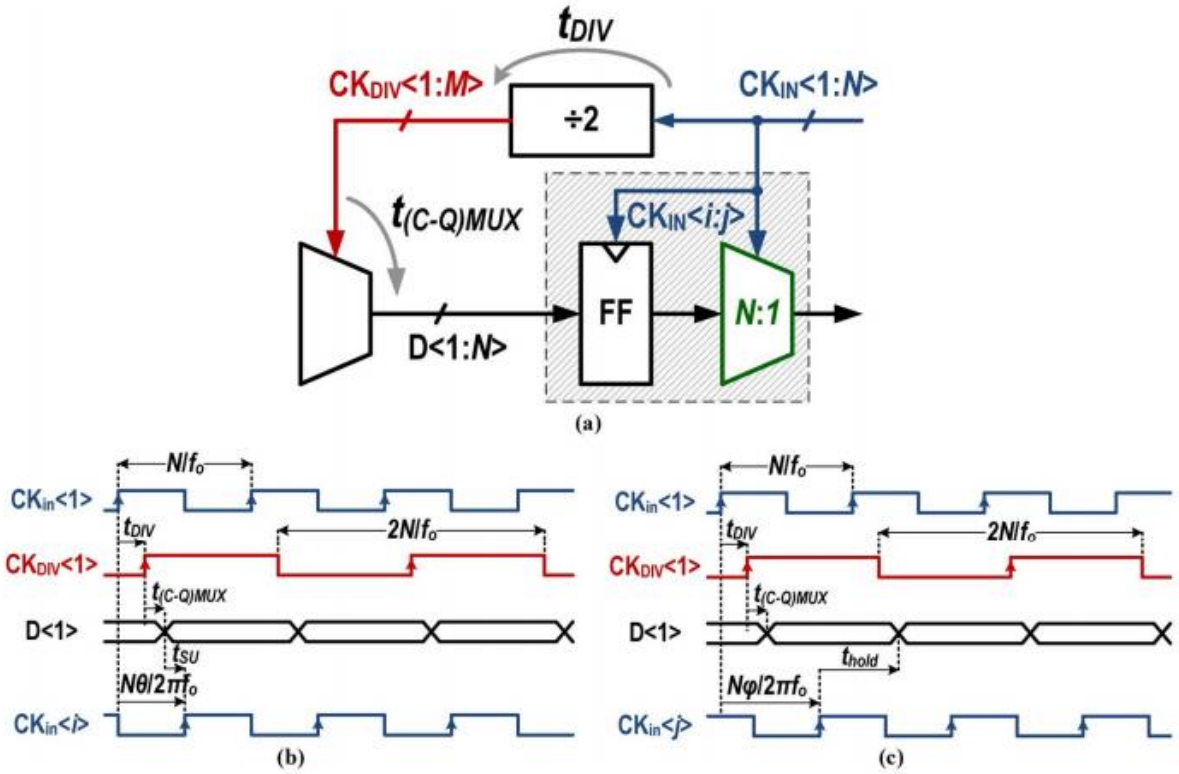


Fig. 1. Timing constraints for an N:1 MUX. (a) Block diagram (b) Setup time. (c) Hold time.

[7] 'A 32-48 Gb/s Serializing Tx using Multiphase Serialization in 65nm CMOS Tech.', JSSC,2015, UCLA

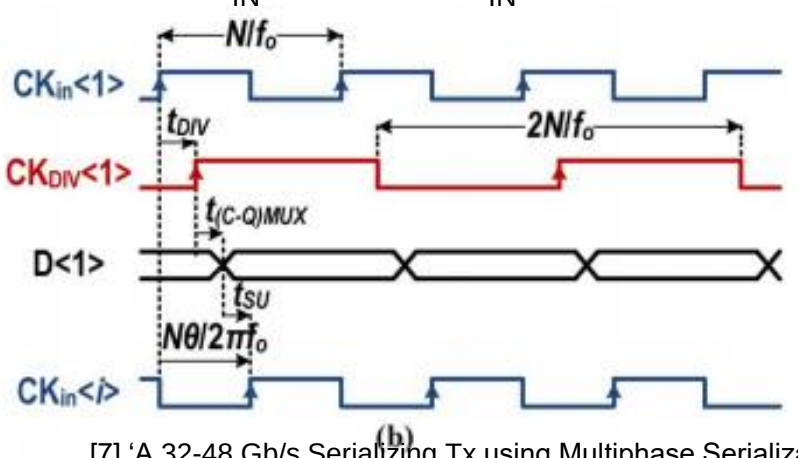
Timing Constraints for N:1 MUX

- **Setup time**
  - **N:1 stage driven by N-phase input clock of f_0/N (f_0 = output bitrate)**
  - **CK_{IN}<1> triggers transition at CK_{DIV}<1> after delay t_{DIV}**
  - **Triggers a transition after delay $t_{(C\text{-}Q)MUX}$**

$$N\frac{\theta}{2\pi f_o} \geq t_{\mathrm{DIV}} + t_{(C-Q)\mathrm{MUX}} + t_{\mathrm{SU}}$$

t_{su}= set up time

θ : phase shift between CK_{IN}<1> and CK_{IN}<i>



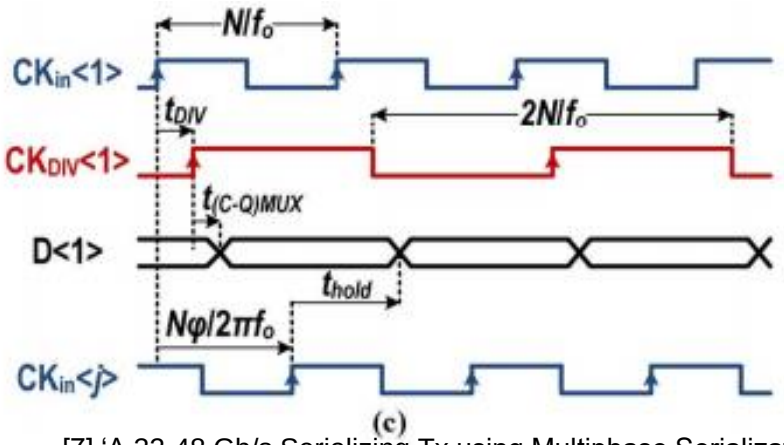
[7] 'A 32-48 Gb/s Serializing Tx using Multiphase Serialization in 65nm CMOS Tech.', JSSC,2015, UCLA

Timing Constraints for N:1 MUX

- **Hold time**
  - **Delay from CKIN<1> to the last sampling edge, CK_{IN}<j> has to be short enough to avoid data switch**

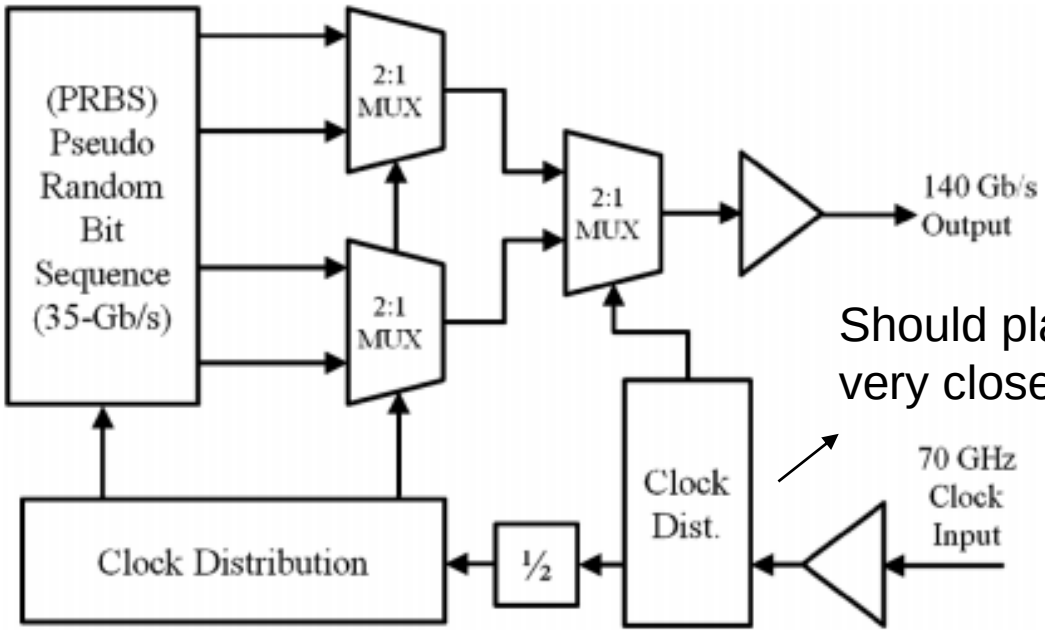
$$\left(\frac{\varphi}{2\pi}-1\right)\times\frac{N}{f_o}t_{\mathrm{DIV}}+t_{(C-Q)\mathrm{MUX}}-t_{\mathrm{hold}}$$

t_{hold}= hold time
φ : phase shift between CK_{IN}<1> and CK_{IN}<j>



[7] 'A 32-48 Gb/s Serializing Tx using Multiphase Serialization in 65nm CMOS Tech.', JSSC,2015, UCLA

Conventional half-rate 4:1 Serializer



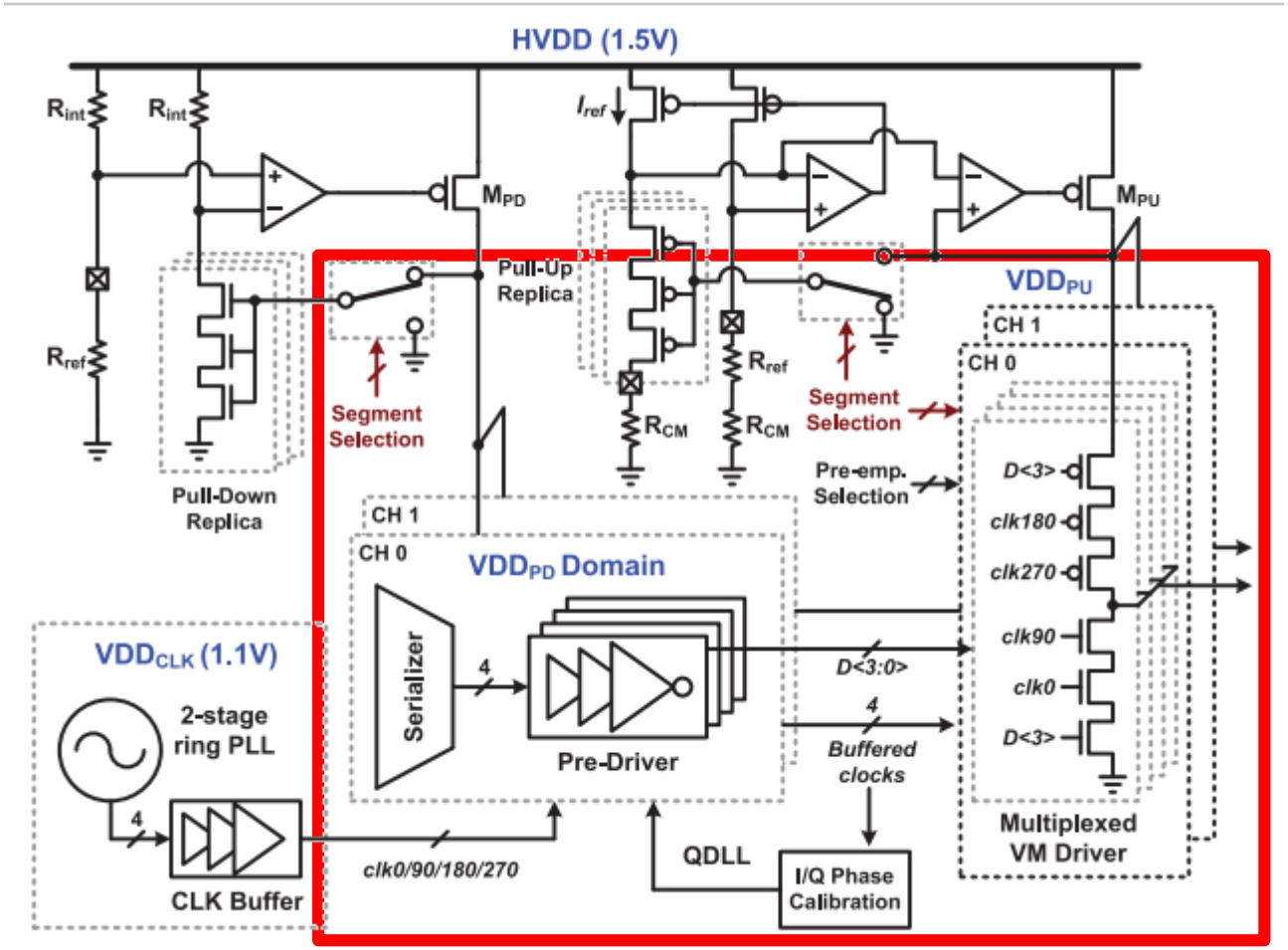
Should place clock source very close to the last stage

Conventional 4:1 half-rate serializer architecture

Ex) For 140Gbps serial output, 70 GHz clock is required
→ Hard to implement such high-speed clock
→ Problems with PLL, clock distribution

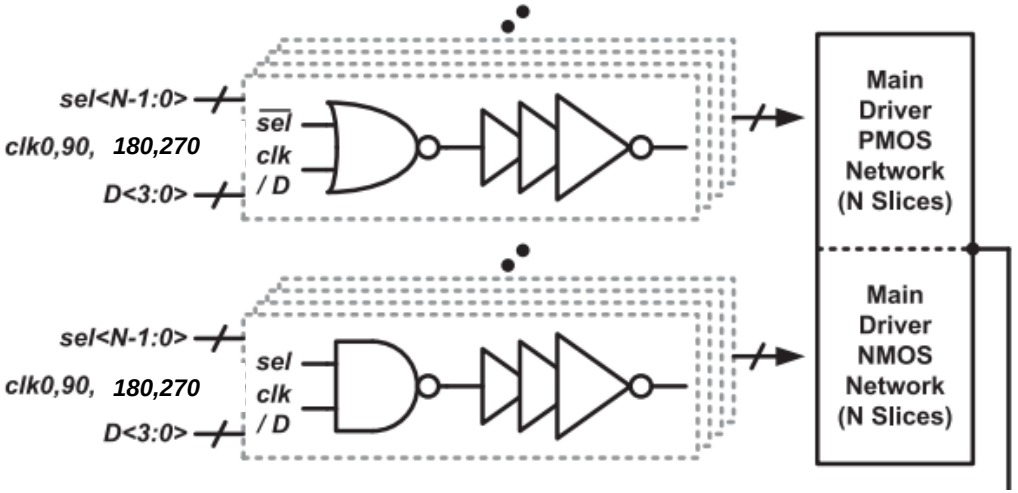
[8] '140 Gb/s Serializer Using Clock Doublers in 90 nm SiGe Technology', JSSC,2015, Rensselaer Polytechnic Institute

Example (1) MUX-based triple stack



[5] 'A Supply-Scalable-Serializing Transmitter With Controllable Output Swing and Equalization for Next-Generation Standards', Transactions on Industrial Electronics ,2018, WRBae

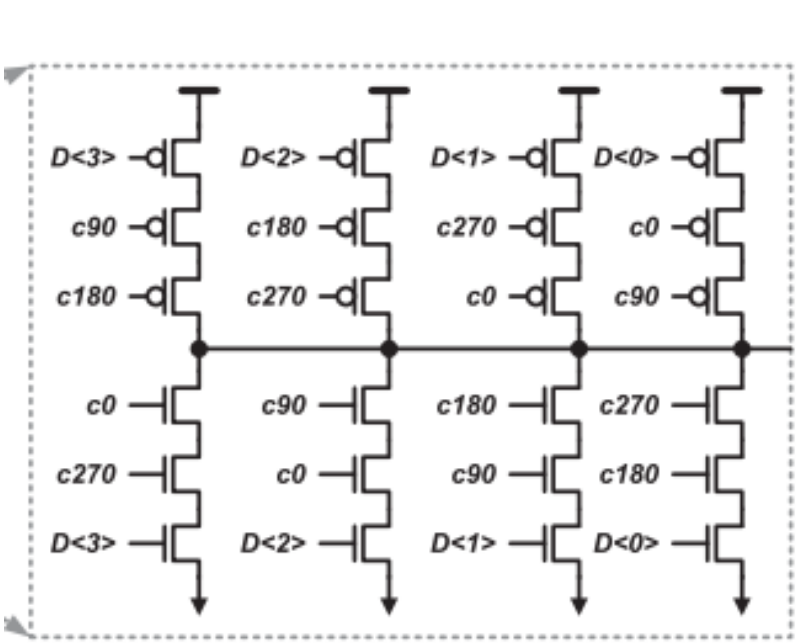
Example (1) MUX-based triple stack

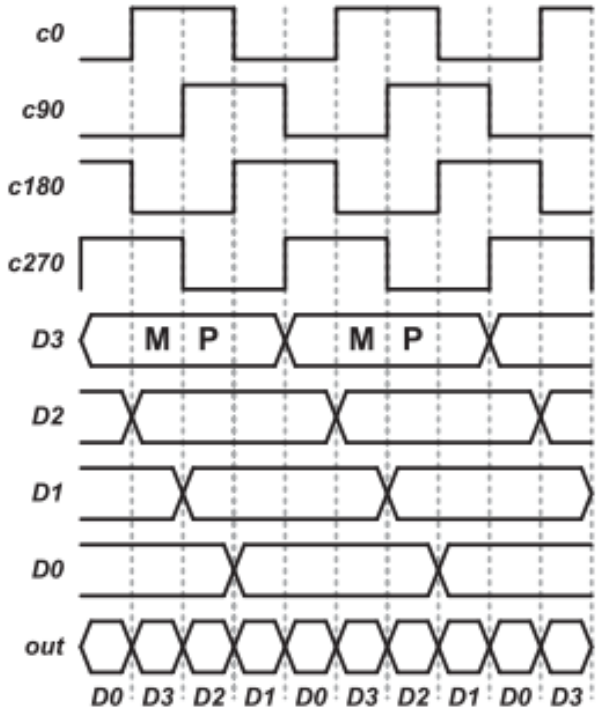


Pre-driver aligns data with the clock using NAND gate and inverter chains

[5] 'A Supply-Scalable-Serializing Transmitter With Controllable Output Swing and Equalization for Next-Generation Standards', Transactions on Industrial Electronics ,2018, WRBae

Example (1) MUX-based triple stack

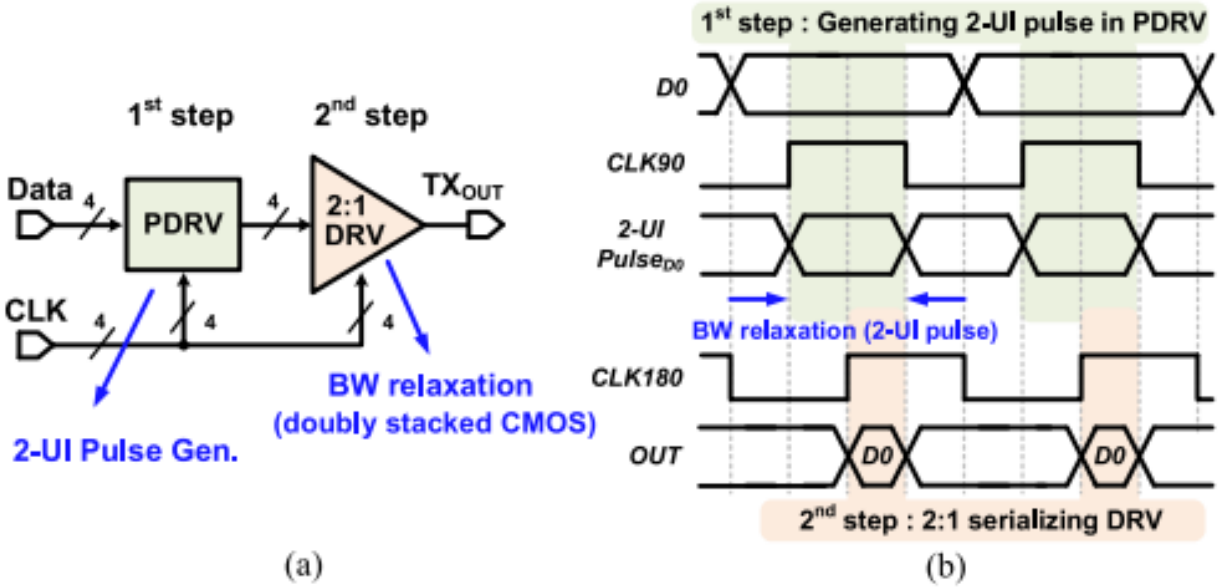




- **Aligned-data is then MUXed with clocks (4:1 MUX)**
- **Has issues with ISI**

[5] 'A Supply-Scalable-Serializing Transmitter With Controllable Output Swing and Equalization for Next-Generation Standards', Transactions on Industrial Electronics ,2018, WRBae

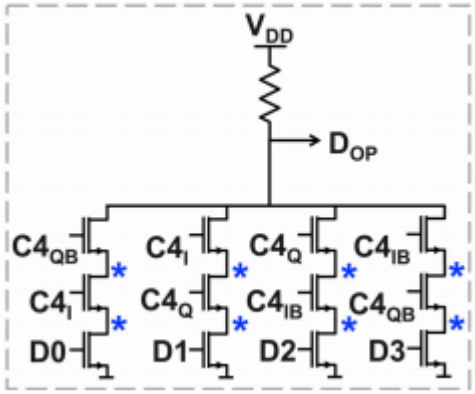
Examples (2) 2-step time multiplexing

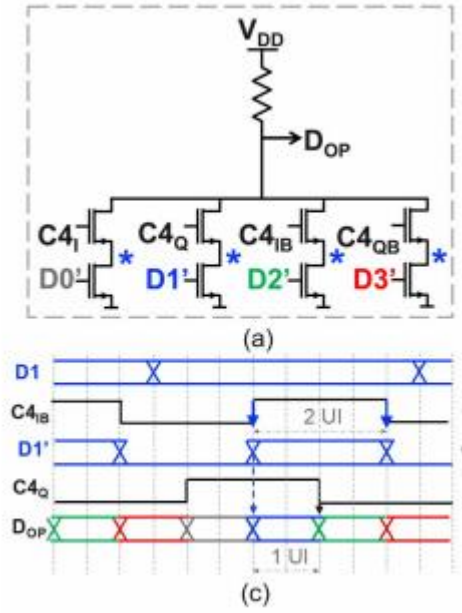


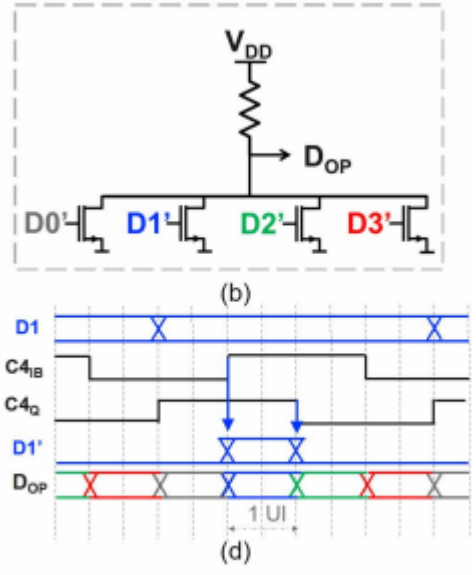
- **Create 2-UI RZ data and MUX it with clock**
- **Timing relaxation**
- **Reduced ISIs compared to example (1)**

[4] 'A 2.5–28 Gb/s Multi-Standard Transmitter With Two-Step Time-Multiplexing Driver', TCAS-II, 2018, M. Choi

Example (3) Unstacked 4:1 MUX





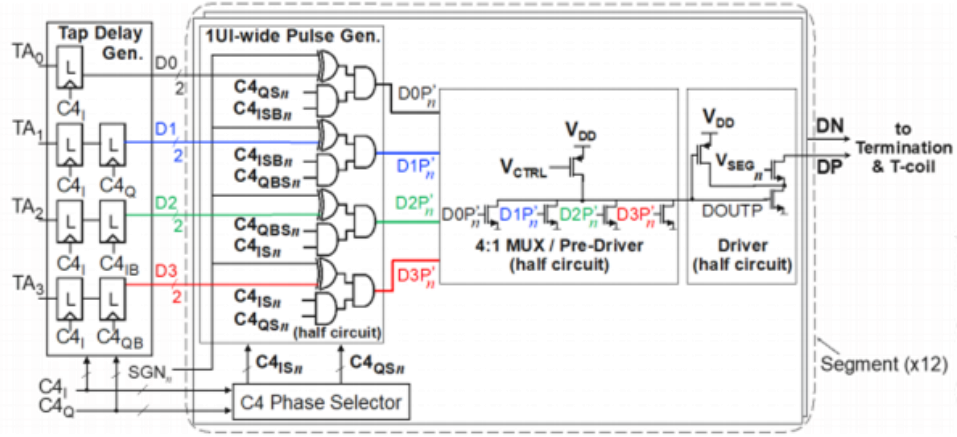


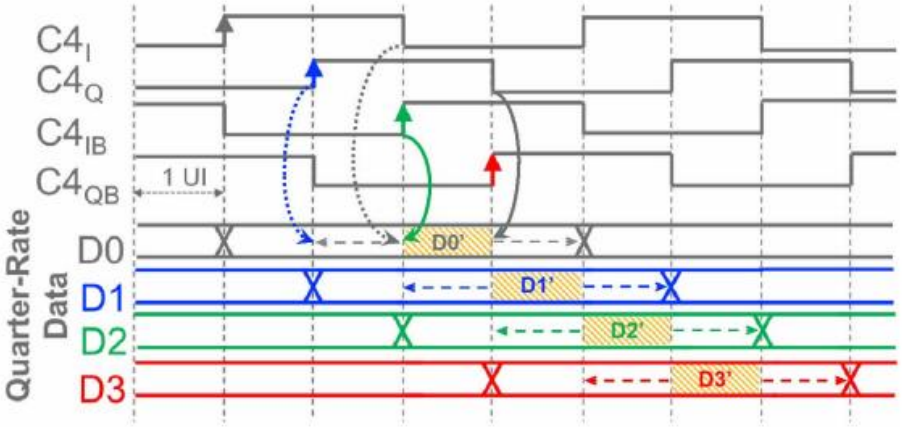
Triple-stacked 4:1 MUX **double-stacked 4:1 MUX** **unstacked 4:1 MUX**

- **Nodes highlighted with * indicates undriven nodes, which results in ISIs**

[6] 'A 128-Gb/s 1.3-pJ/b PAM-4 Transmitter With Reconfigurable 3-Tap FFE in 14-nm CMOS', JSSC, 2020, IBM

Example (3) Unstacked 4:1 MUX





- **AND 2 clocks to create 1-UI wide pulse and then AND it with data**
- **Reduced un-driven nodes for ISI reduction**

[6] 'A 128-Gb/s 1.3-pJ/b PAM-4 Transmitter With Reconfigurable 3-Tap FFE in 14-nm CMOS', JSSC, 2020, IBM

Applications

- **Examples of widely used serial links**

- Memory Bus (Single-ended, Parallel)
  - DDR (4.266 Gbps)
  - LPDDR4 (4.266 Gbps)
  - GDDR (7 Gps)
  - XDR (differential, 4.8 Gbps)
  - Wide IO2, HBM
- Front Side Bus (Differential, Parallel)
  - QuickPath Interconnect (6.4 Gbps)
  - HyperTransport (6.4 Gbps)
- Computer IO (Differential, Parallel)
  - PCIe (8 Gbps)
  - InfiniBand (10 Gbps)
- Cable (Differential, Serial)
  - USB (4.266 Gbps)
  - HDMI (4.266 Gbps)
  - Firewire: Cat 5, Cat 5e, Cat 6
- Storage (Differential, Serial)
  - eMMC, UFS (6 Gbps)
  - SAS, STATA (6 Gbps)
  - FiberChannel (10 – 20 Gbps)
- Ethernet (Differential, Serial)
  - XAUI (10 Gbps)
  - XFI (10 Gbps)
  - CEI-6GLR
  - SONNET (10 Gbps)
  - 10GBase-x, 100GBase (25 Gbps)

References

[1] Lecture note, Seralizer, Yonsei University

[2] Lecture note, High-Speed Links, UIUC

[3] Thesis ‘‘A high speed serializer deserializer design’, Y. Luo, 2010

[4] ‘A 2.5–28 Gb/s Multi-Standard Transmitter With Two-Step Time-Multiplexing Driver’, TCAS-II, 2018, M. Choi

[5] ‘A Supply-Scalable-Serializing Transmitter With Controllable Output Swing and Equalization for Next-Generation Standards’, Transactions on Industrial Electronics ,2018, W. Bae

[6] ‘A 128-Gb/s 1.3-pJ/b PAM-4 Transmitter With Reconfigurable 3-Tap FFE in 14-nm CMOS’, JSSC, 2020, IBM

[7] ‘A 32-48 Gb/s Serializing Tx using Multiphase Serialization in 65nm CMOS Tech.’, JSSC,2015, UCLA

[8] ‘140 Gb/s Serializer Using Clock Doublers in 90 nm SiGe Technology’, JSSC,2015, Rensselaer Polytechnic Institute

Topics in IC Design

9.3 Phase Interpolator

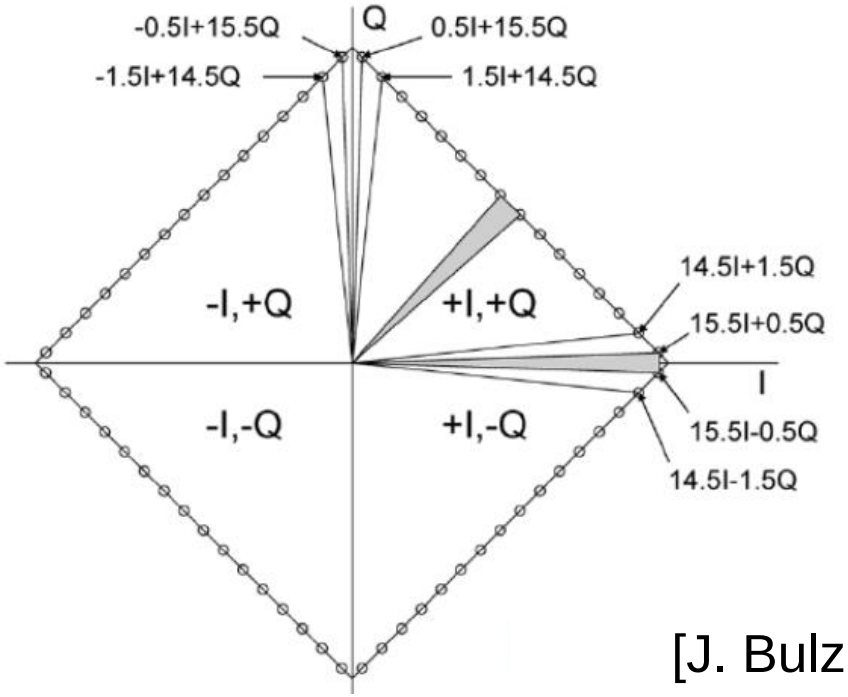
Deog-Kyoon Jeong
dkjeong@snu.ac.kr
School of Electrical and Computer Engineering
Seoul National University
2020 Fall
Compliments to Moon-Chul Choi, Minkyo Shim

Contents

- Phase Interpolator
- Operation principle
- Linearity
- Structure
- Case study
- Reference

Phase Interpolator

- **Phase interpolator mixes between input phases to produce a fine sampling phase**
  - **Ex) Quadrature 90˚ PI inputs with 5 bit resolution provides sampling phases spaced by $90^{\circ}/(2^5-1) = 2.9^{\circ}$**



[J. Bulzacchelli, JSSC 06']

Phase Interpolator(PI) Based CDR

- **The D/PLL and PI combination produces adjustable clock phase clock generator**

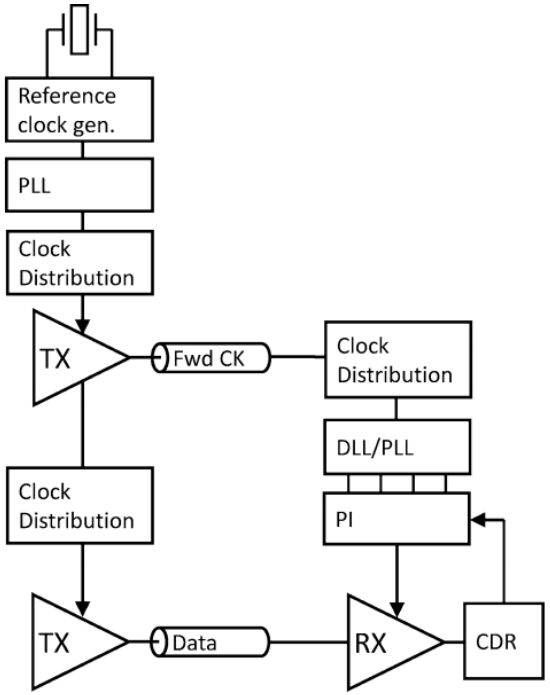


Fig. 3. DLL/PLL-based forwarded clock architecture.

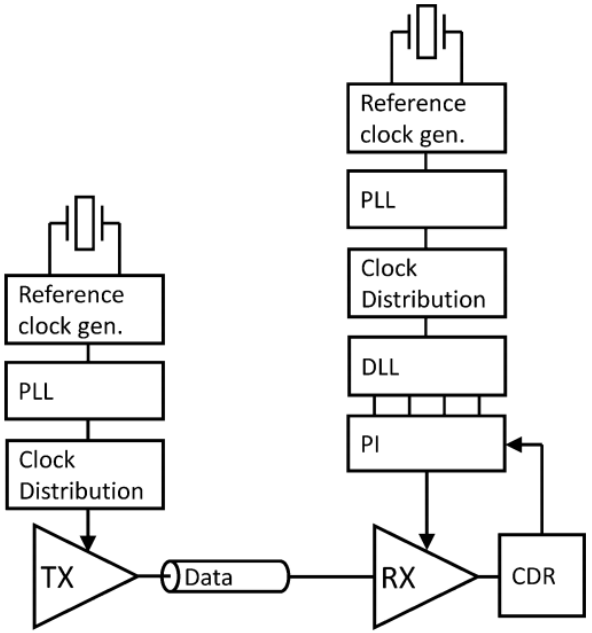
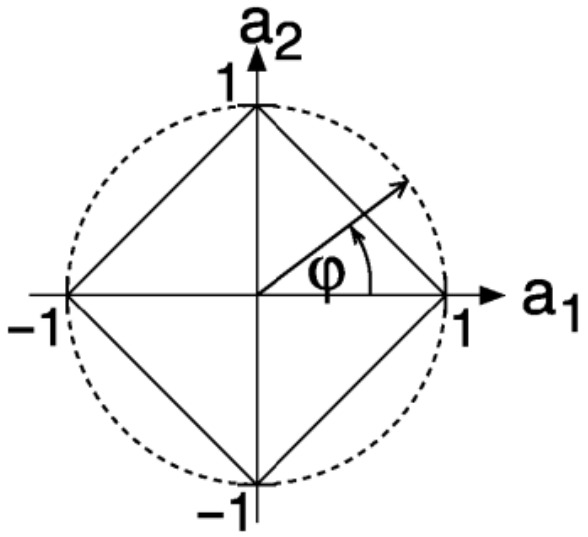
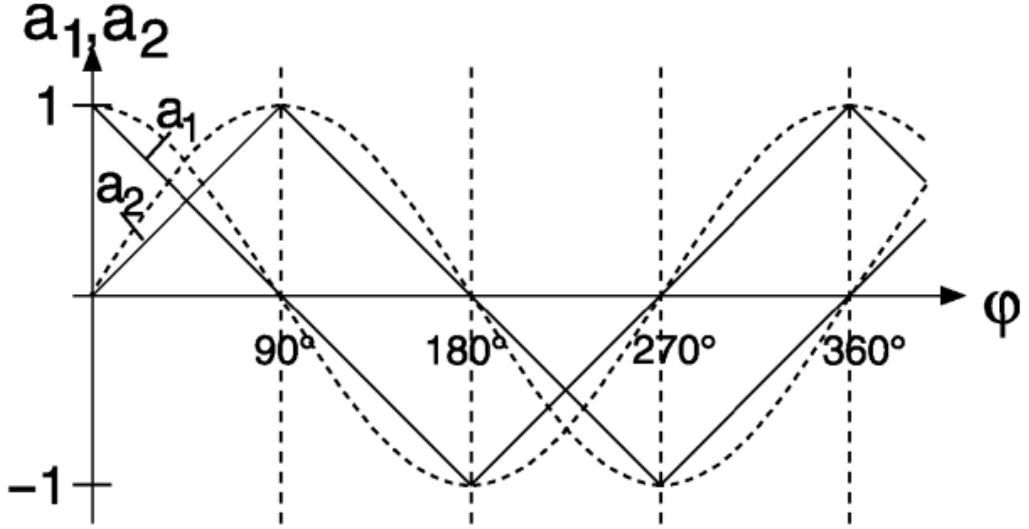


Fig. 5. PI-based (or mixer-based) embedded clock architecture.

[B. Casper, TICAS1 09']

Operation Principle





[R. Kreienkamp, JSSC 05']

$$V_{in,I} = A\sin(\omega t)$$

$$V_{in,Q} = A\sin(\omega t - \pi/2)$$

$$V_{out} = A\sin(\omega t - \phi)$$

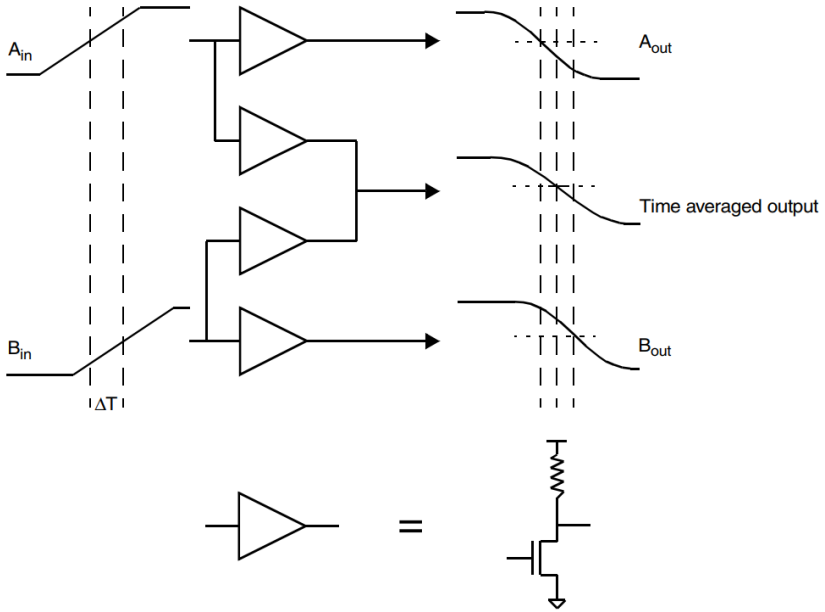
$$= \cos(\phi)\, V_{in,I} + \sin(\phi)\, V_{in,Q} = a_1 V_{in,I} + a_2 V_{in,Q}$$

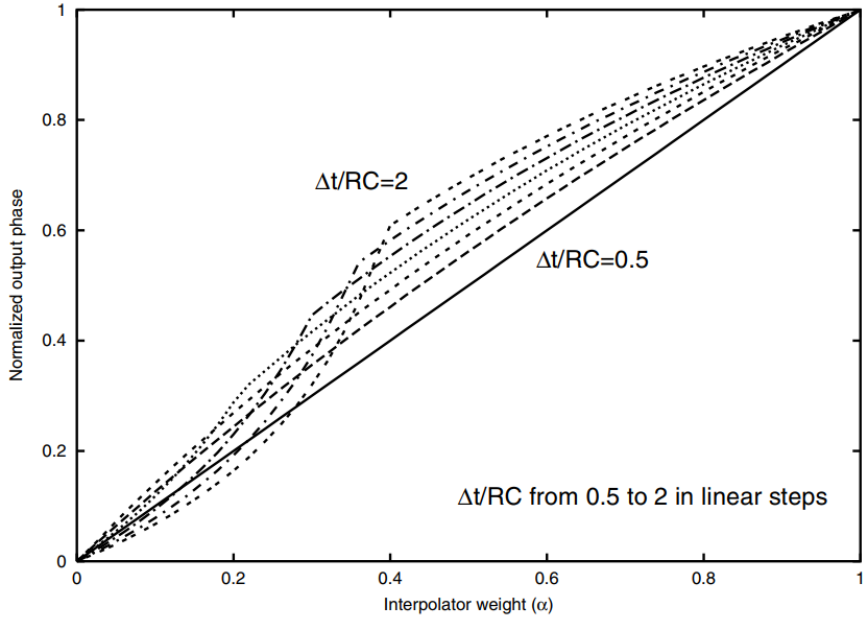
$$a_1^2 + a_2^2 = 1$$

Linearity

- **PI linearity is a function of the phase spacing, Δt , to output time constant, RC, ratio**

$$V_o(t) = V_{cc} + R \cdot I \cdot \left[(1-\alpha) \cdot u(t) \cdot \left(e^{-\frac{t}{RC}} - 1 \right) + \alpha \cdot u(t-\Delta t) \cdot \left(e^{-\frac{t-\Delta t}{RC}} - 1 \right) \right], \quad (3.1)$$

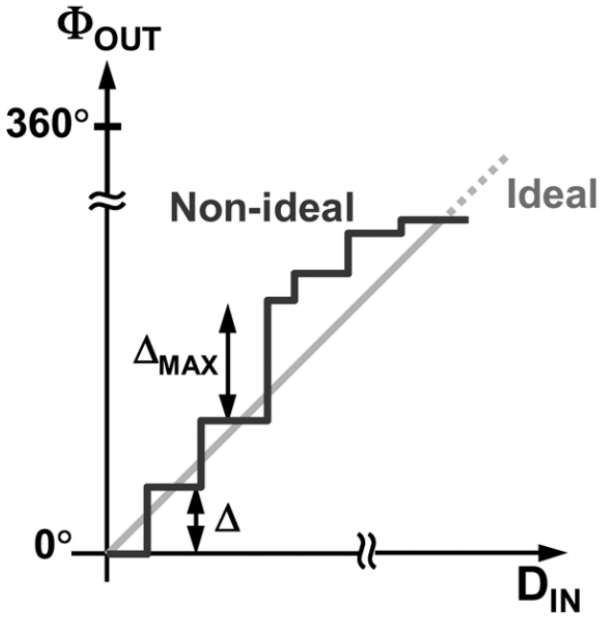




[Weinlader, PhD Thesis]

Non-Linearity Parameter

- **INL : suppressed by the large feedback gain**
- **DNL : severely degrades the recovered clock jitter**



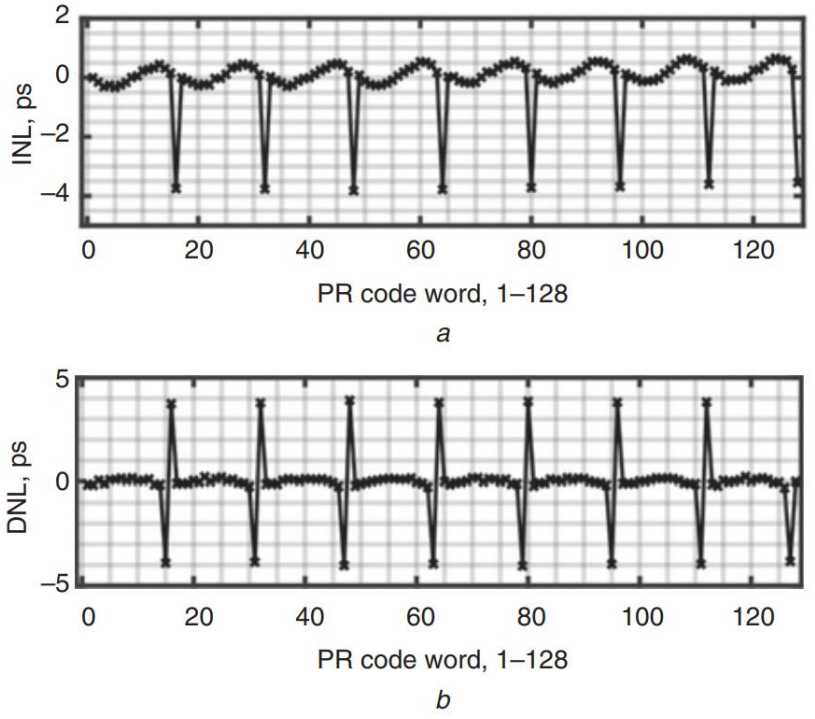


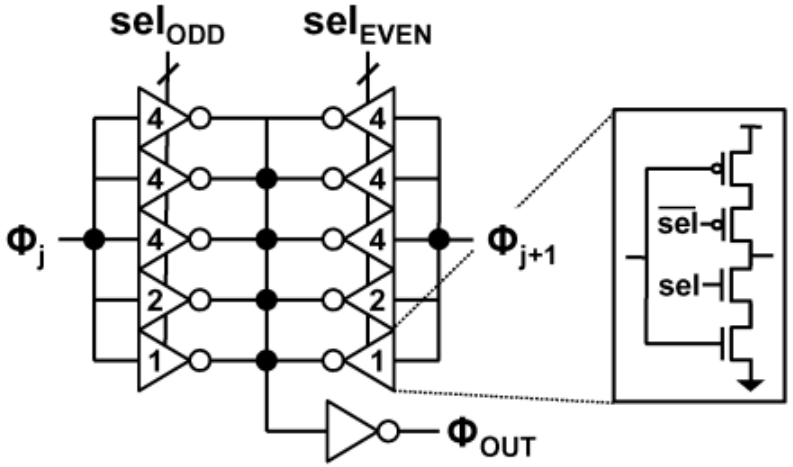
Fig. 3. Representative phase interpolator transfer function.

[P. Hanumolu, JSSC 08']

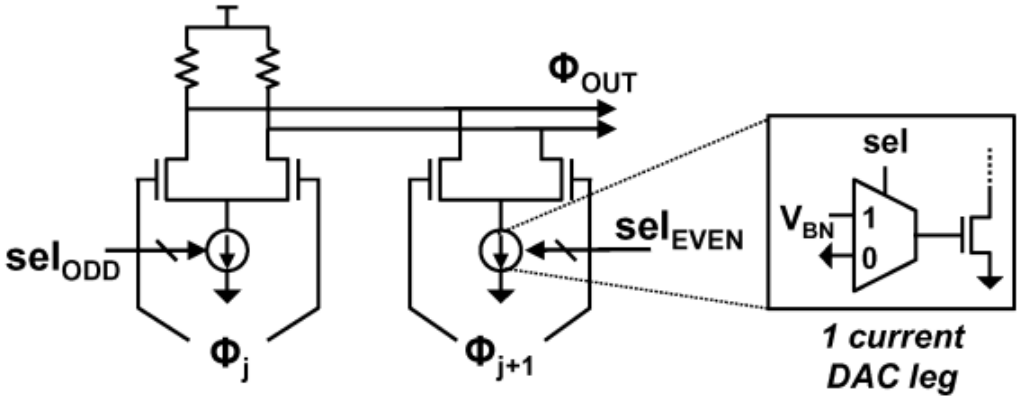
[H. Kim, EL 20']

PI structures

- **Voltage-mode[CMOS] PI**
- **Current-mode[CML] PI**



Voltage-mode

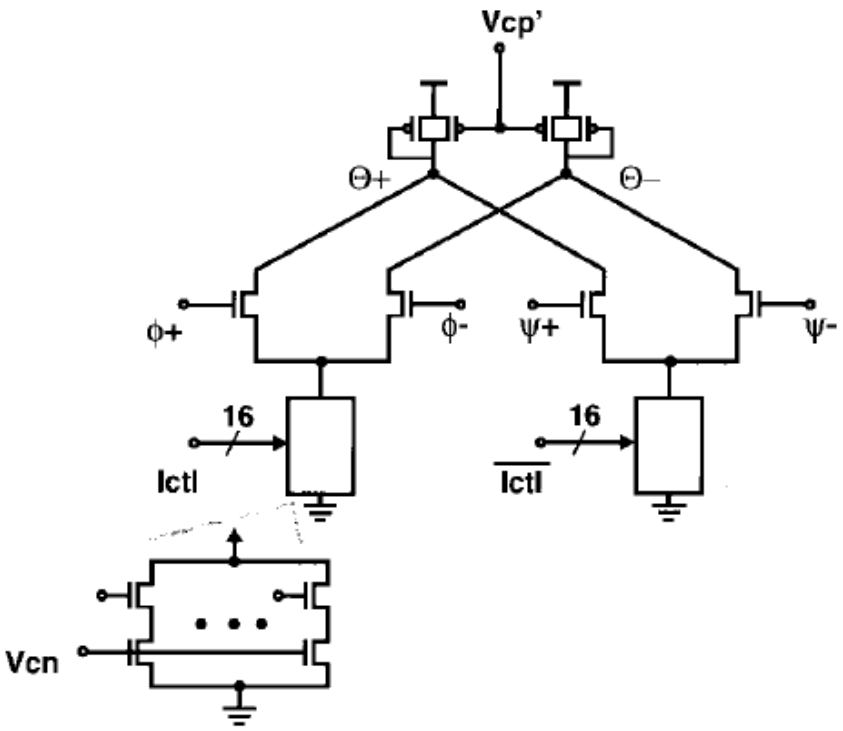


Current-mode

[B. Casper, TICAS1 09']

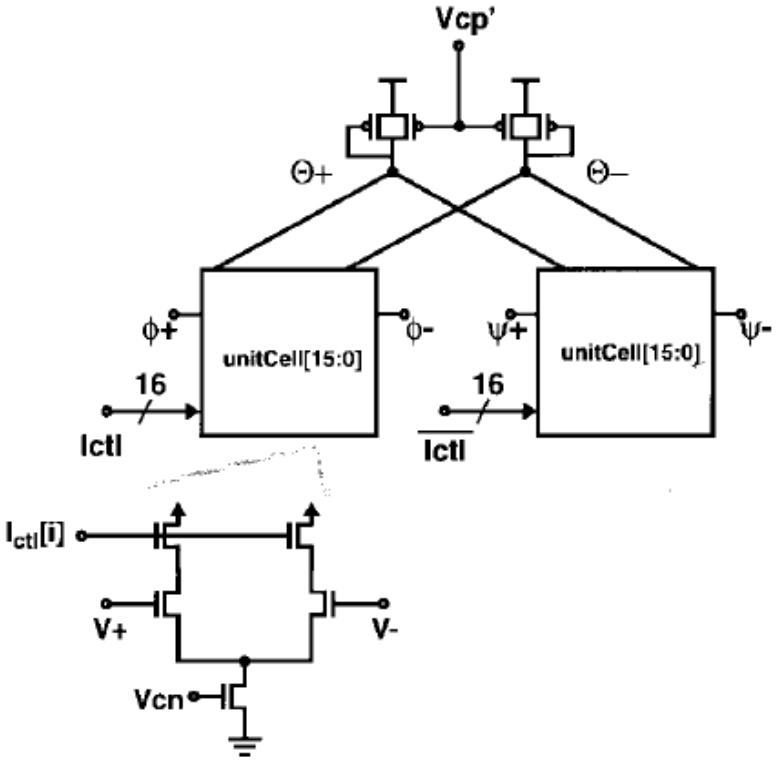
Comparison of PI

Type-I PI



Does not satisfy seamless boundary switching (C_{GD})

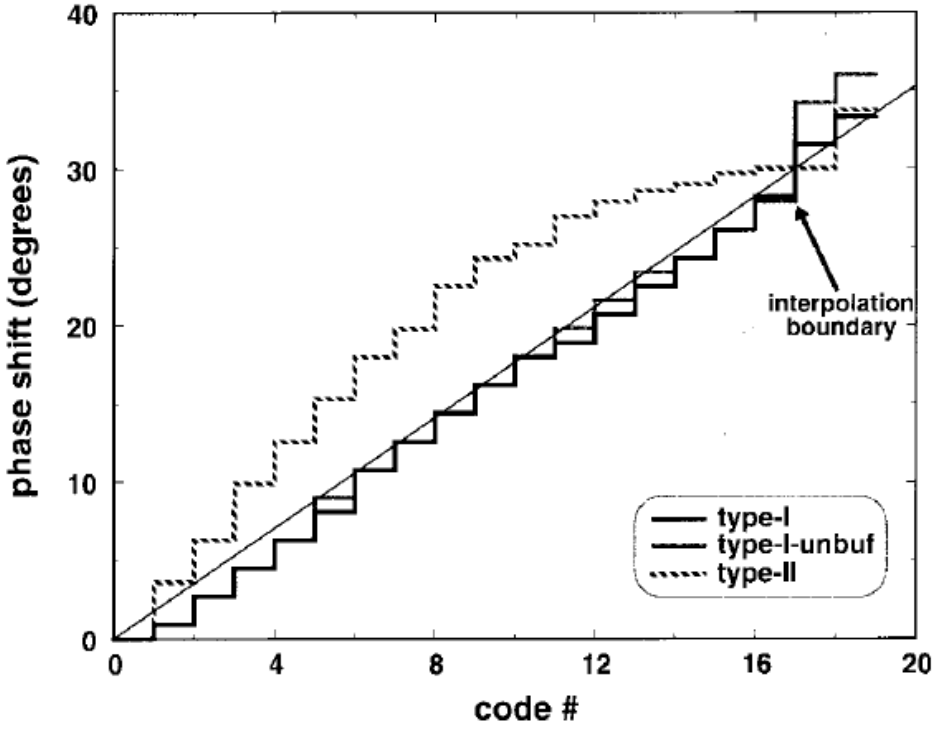
Type-II PI



Seamless boundary switching (input is isolated from output)

[S. Sidiropoulos, JSSC 97']

Simulated PI transfer function

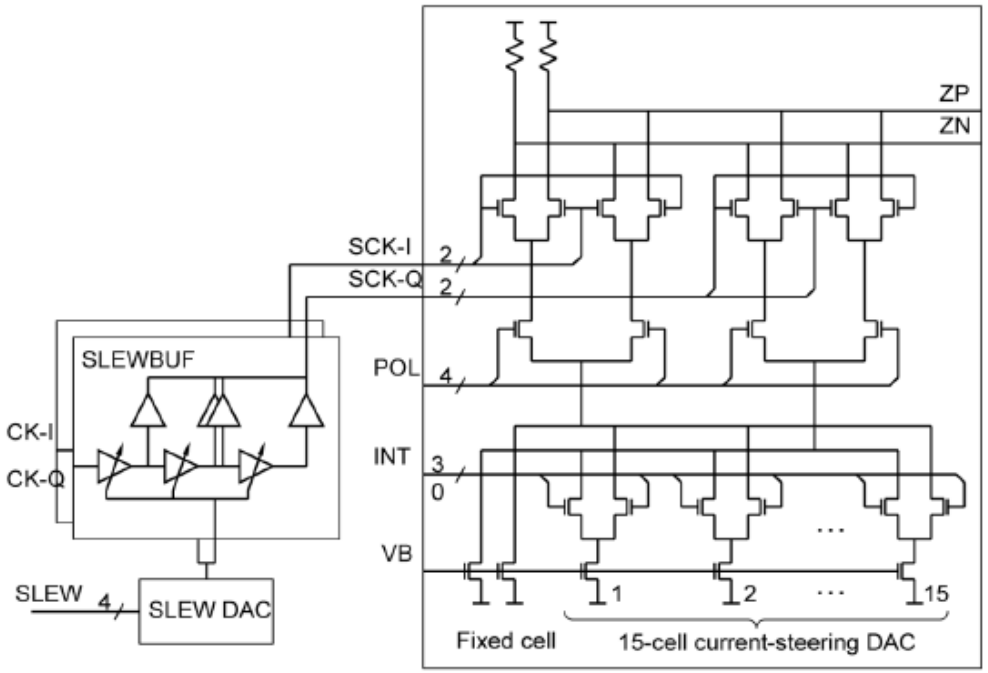


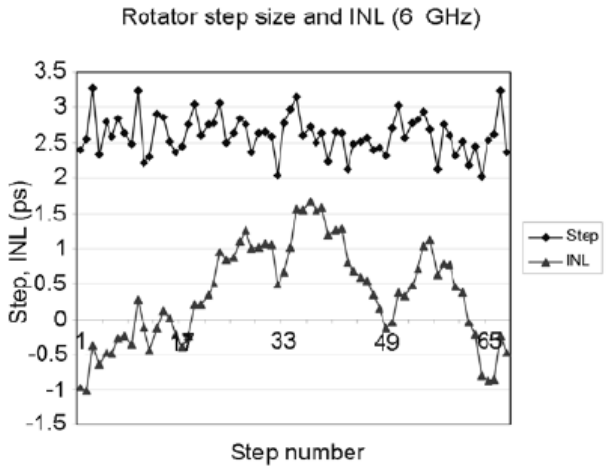
- **Type-I PI: Maximum step of 3.8° due to C_{GD}**
- **Type-II PI: large area, more nonlinear characteristics due to data-dependent loading**

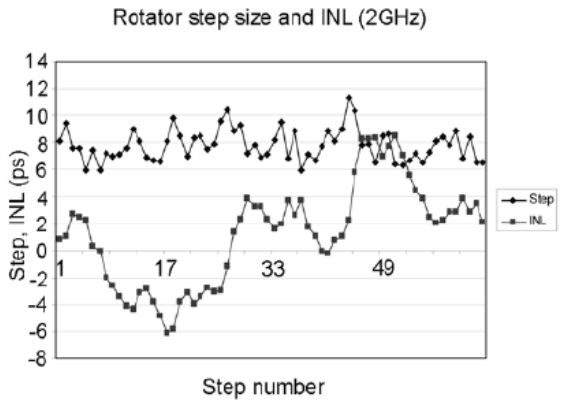
[S. Sidiropoulos, JSSC 97']

Case Study: PI [1/5]

- **I/Q polarity selection (POL)**
- **Slew rate control @ input nodes**



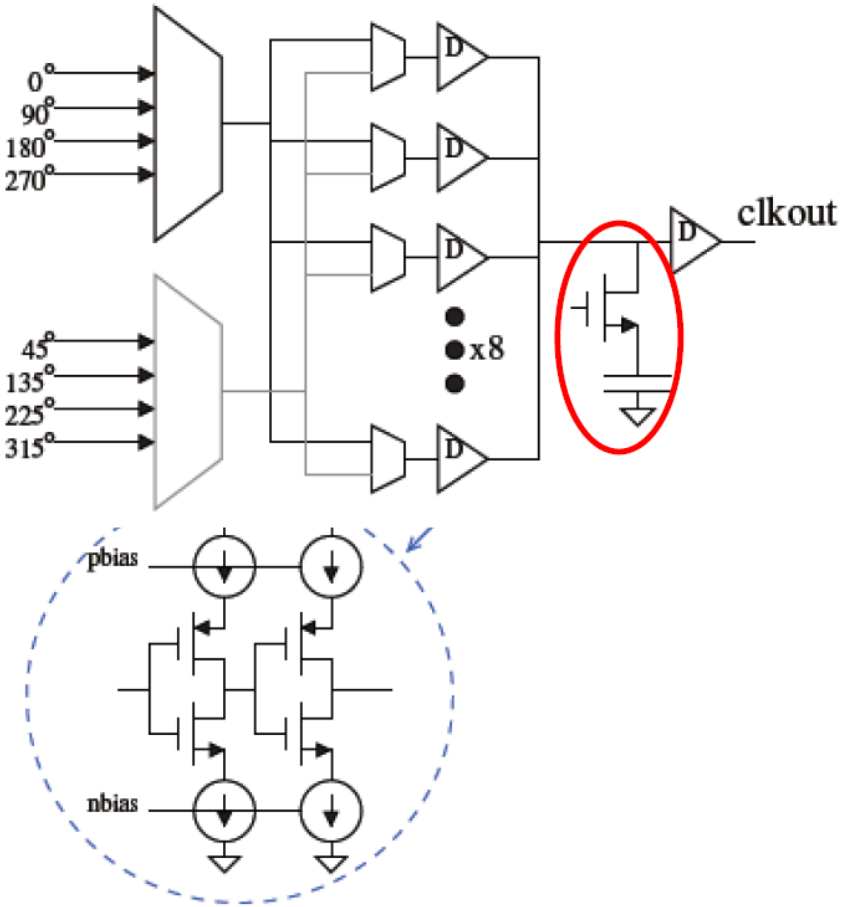


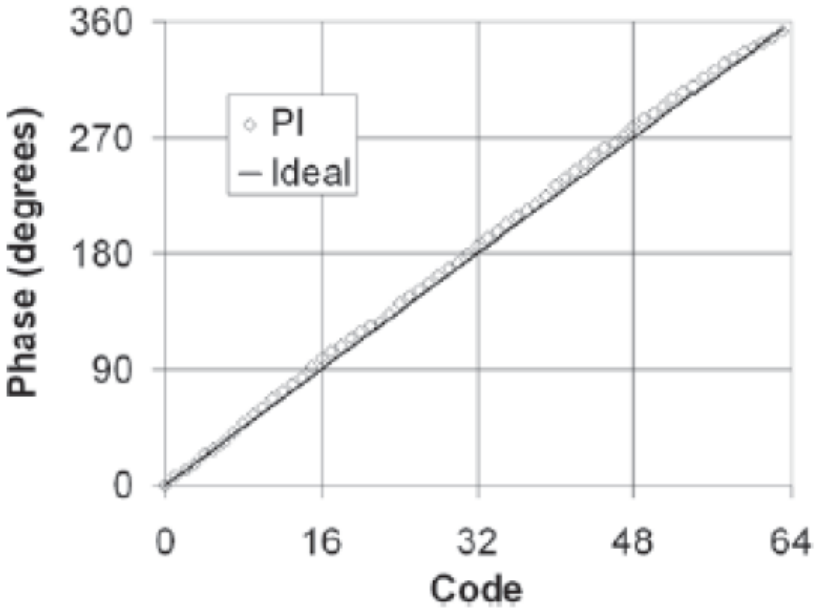


[J. Bulzacchelli, JSSC 06']

Case Study: PI [2/5]

- **Cap @ output nodes for slew rate control**

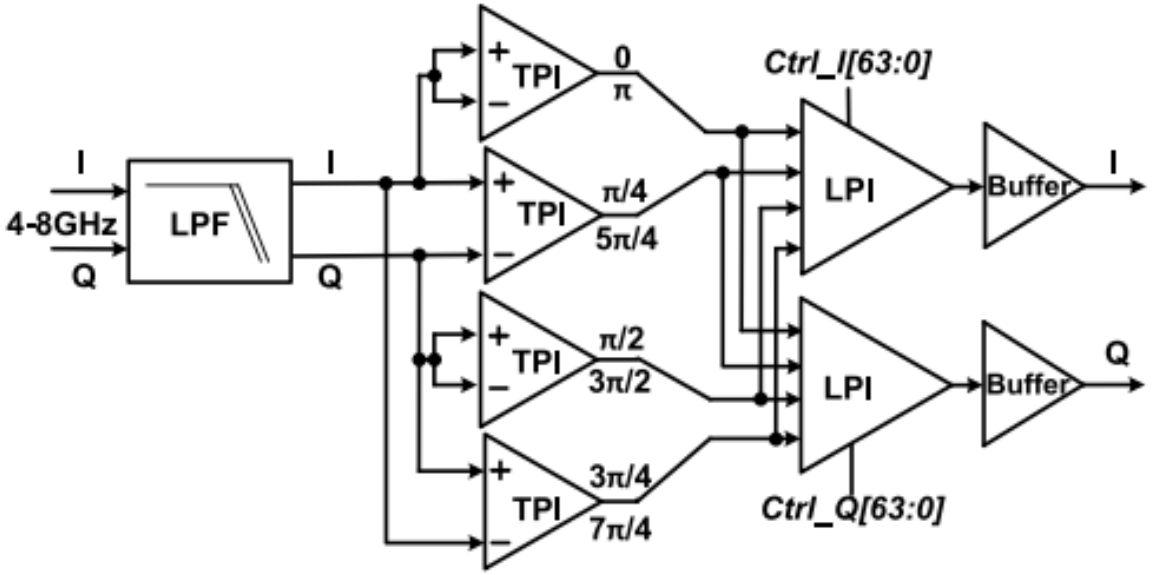




[S. Joshi, VLSI 09']

Case Study: PI [3/5]

- **Two step PI (TSPI): trigonometric PI (TPI) + linear PI (LPI)**
- **Tunable LPF for wide-range operation (4-8GHz)**

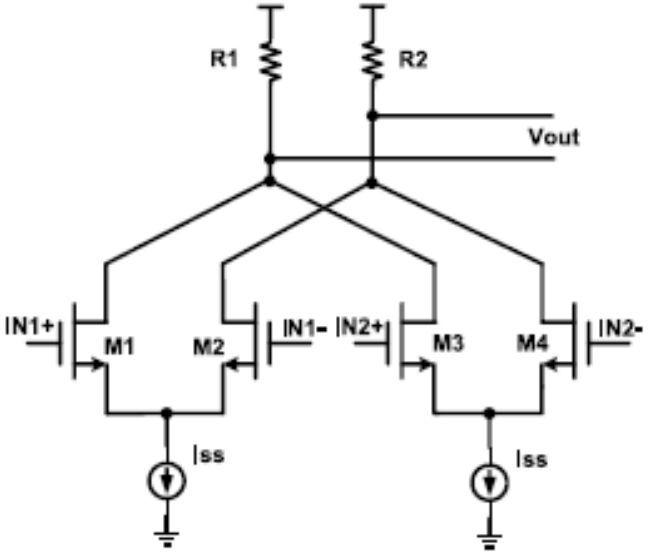


- **TPSI achieves high bandwidth & linearity**

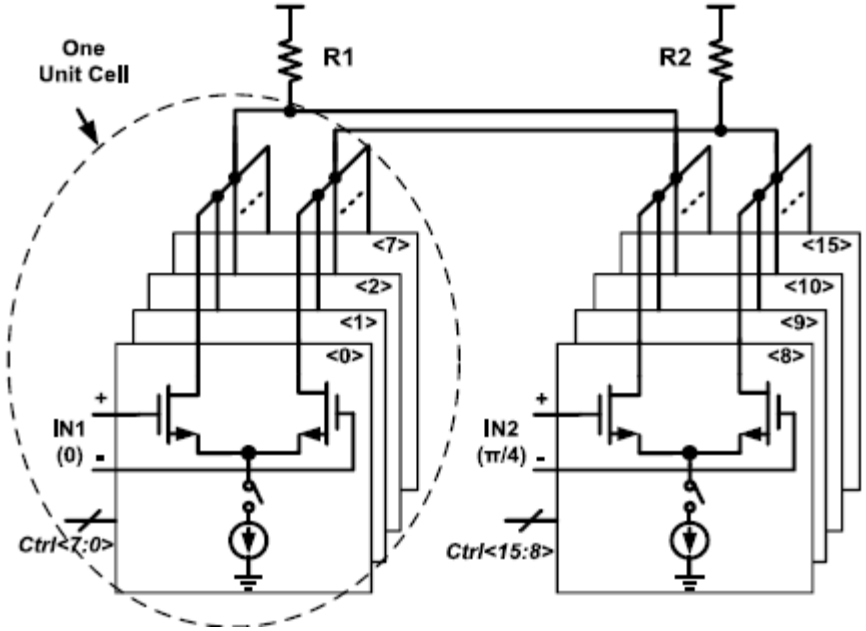
[B. Wu, T-VLSI 16']

Case Study: PI [3/5]

Trigonometric PI (TPI)



Linear PI (LPI)

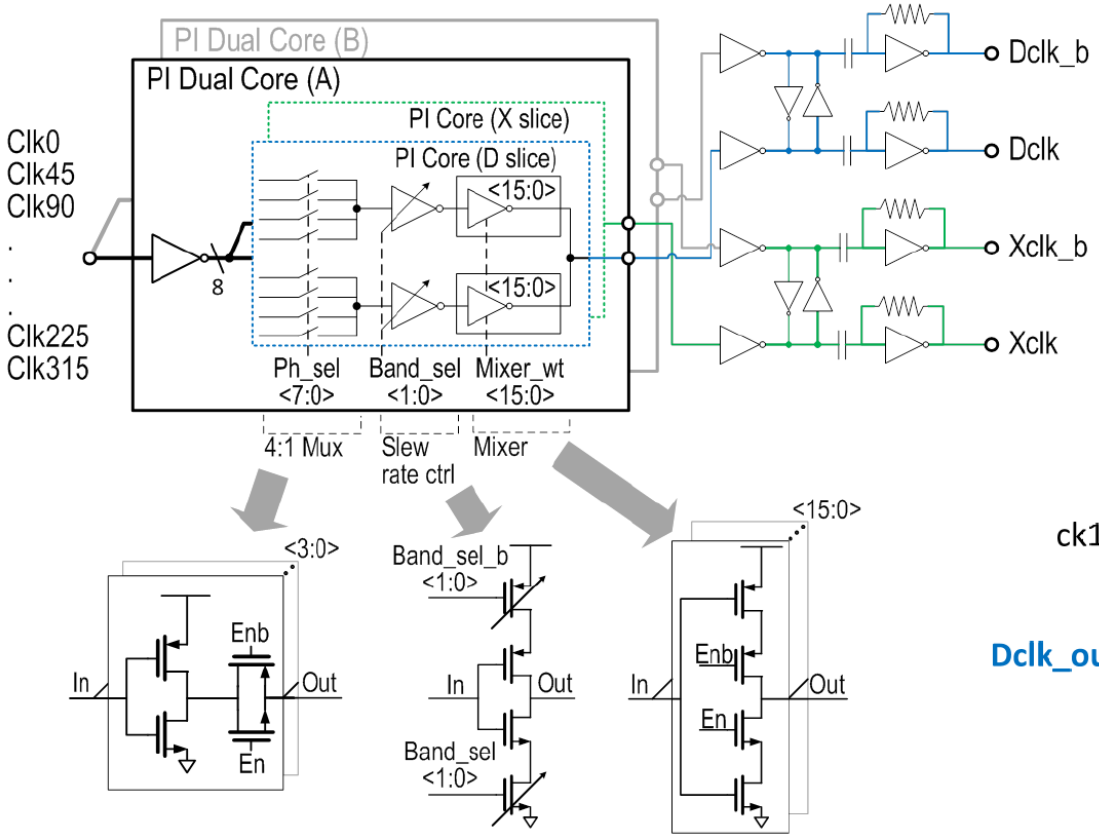


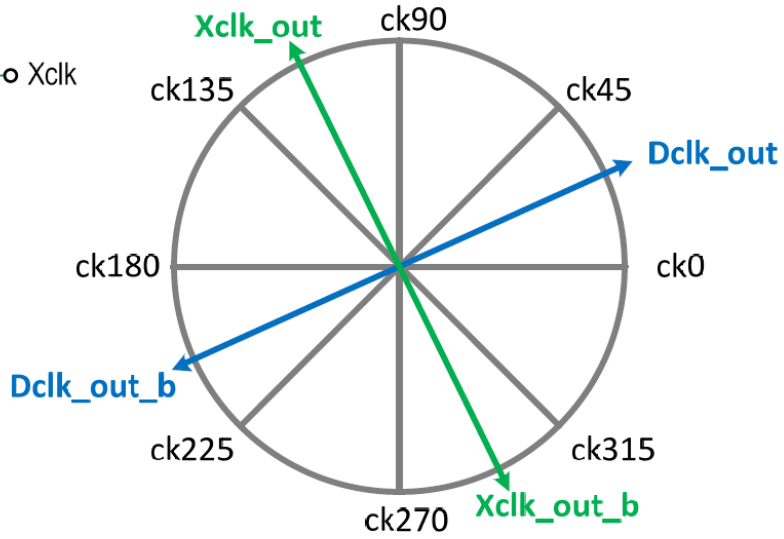
- **TPI generates middle phase**
- **LPI mixes the phases between 0 & $\pi/4$**

[B. Wu, T-VLSI 16']

Case Study: PI [4/5]

- **4-16 GHz CMOS-based PI w/ slew rate controller**

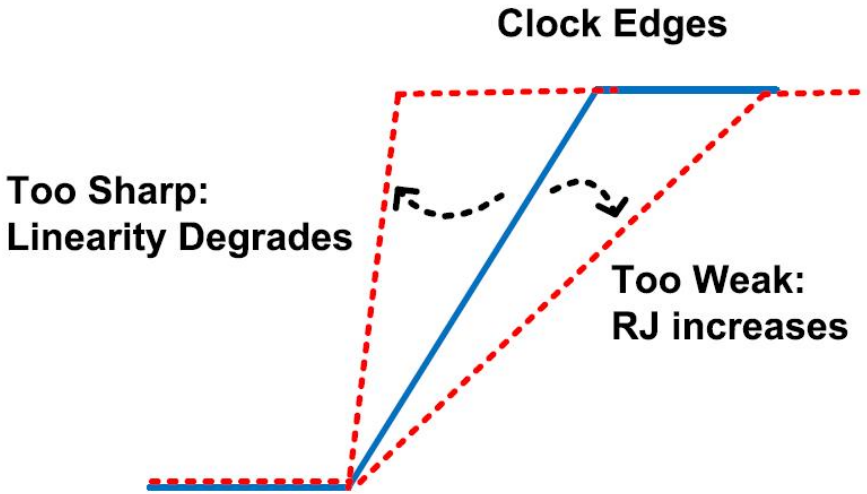


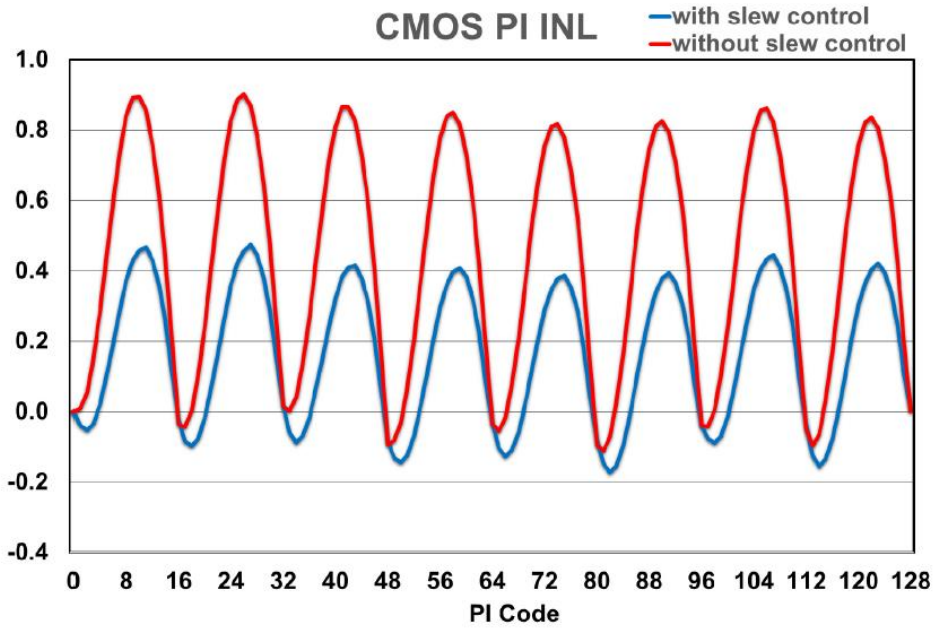


[S. Chen, ISSCC 18']

Case Study: PI [4/5]

- **PI supports a wide frequency range (4-16 GHz)**
  - **Slew rate needs to be adjusted and optimized for target bands**
  - **Achieve better linearity without significant RJ increase**

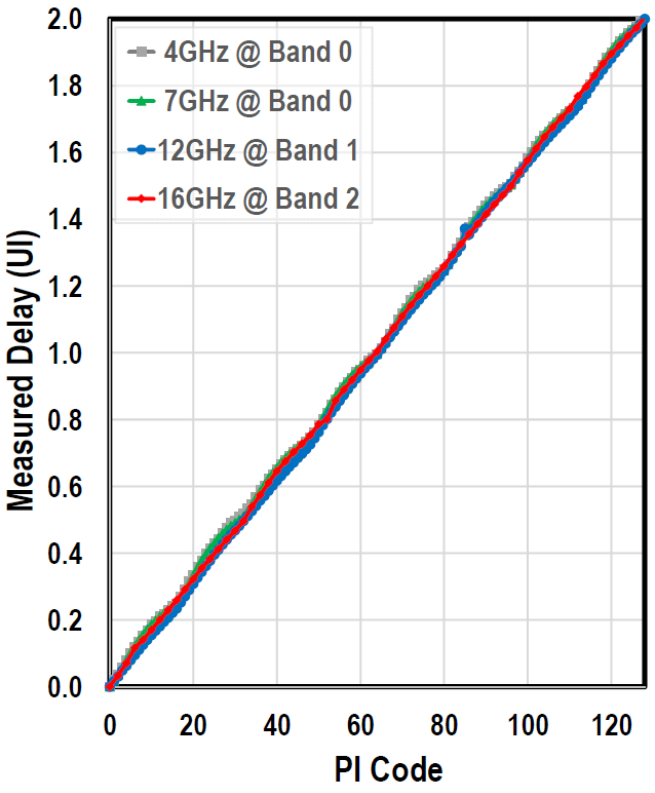


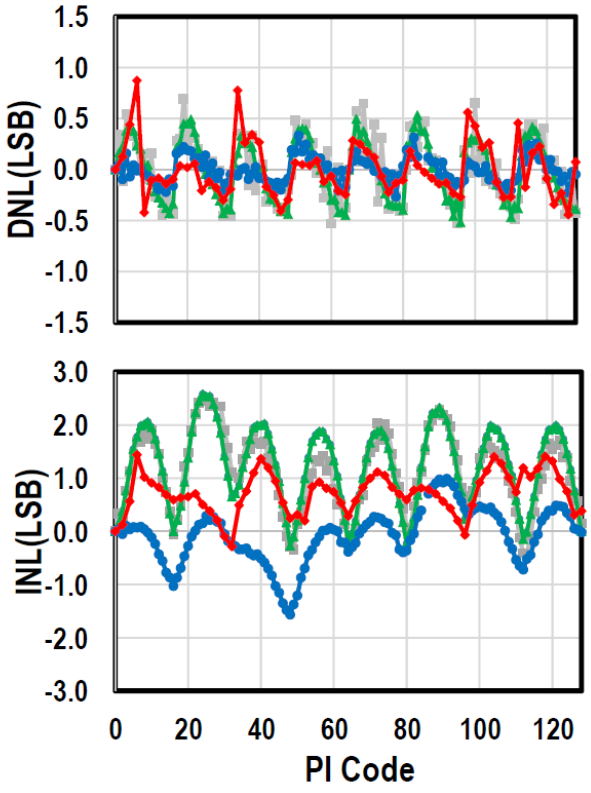


[S. Chen, ISSCC 18']

Case Study: PI [4/5]

- **Measured PI linearity**





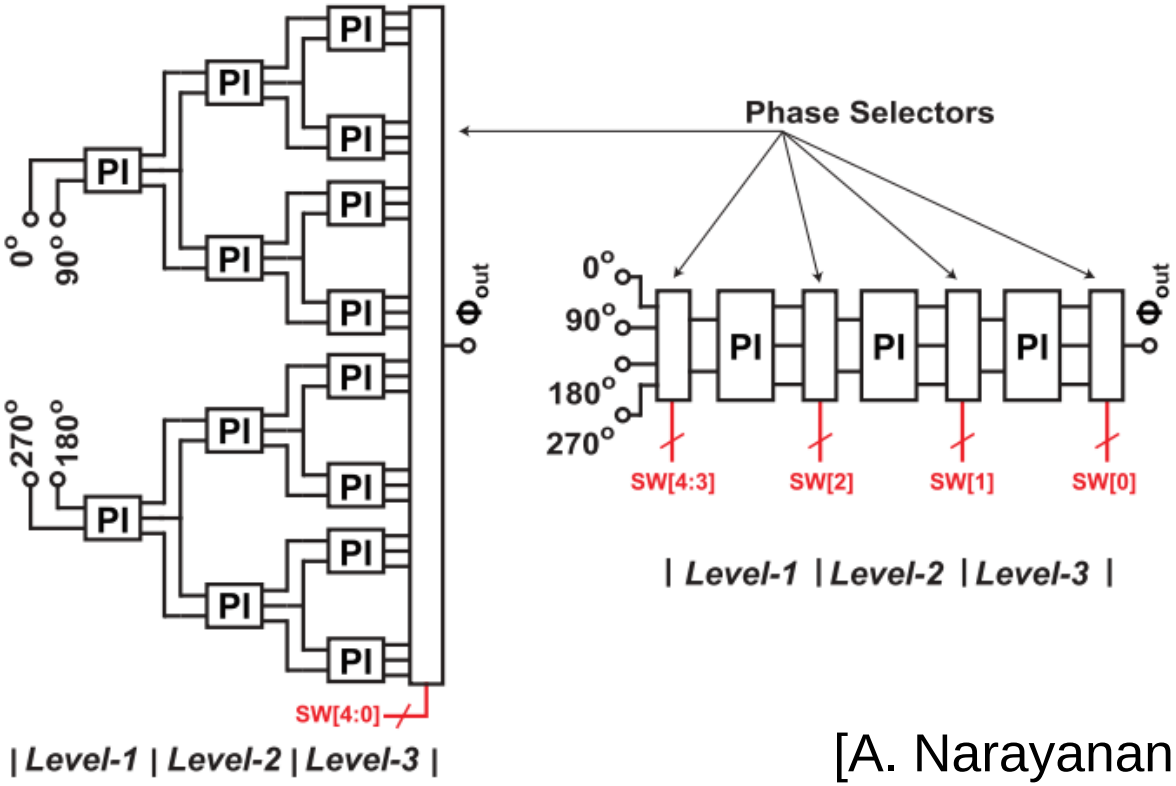
At 16GHz
DNL: 0.87 LSB
INL: 1.44 LSB

Worst Case INL:
2.58LSB at 7GHz

[S. Chen, ISSCC 18']

Case Study: PI [5/5]

- **Tournament-based PI**



[A. Narayanan, JSSC 2016]

References

1. S. Sidiropoulos and M. A. Horowitz, "A semidigital dual delay-locked loop," JSSC, vol. 32, no. 11, pp. 1683-1692, Nov. 1997.
2. J. F. Bulzacchelli et al., "A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology," JSSC, vol. 41, no. 12, pp. 2885-2900, Dec. 2006.
3. S. Joshi et al., "A 12-Gb/s transceiver in 32-nm bulk CMOS," VLSI, 2009.
4. G. Wu et al., "A 1-16-Gb/s All-Digital Clock and Data Recovery With a Wideband, High-Linearity Phase Interpolator," in IEEE Transactions on Very Large Scale Integration (VLSI), 2016.
5. S. Chen et al., "A 4-to-16GHz inverter-based injection-locked quadrature clock generator with phase interpolators for multi-standard I/Os in 7nm FinFET," ISSCC, 2018.

References

6. B. Casper and F. O'Mahony, "Clocking analysis implementation and measurement techniques for high-speed data linksA tutorial", *IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 56, no. 1, pp. 17-39, Jan. 2009.
7. R. Kreienkamp, "A 10-Gb/s CMOS clock and data recovery circuit with an analog phase interpolator", *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 735-743, Mar. 2005.
8. Weinlader, D.K.: `Precision CMOS receivers for VLSI testing applications', November 2001, PhD, Stanford University, Palo Alto, CA, USA.
9. P. Hanumolu, G.-Y. Wei and U.-K. Moon, "A wide-tracking range clock and data recovery circuit", *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 425-439, Feb. 2008.
10. H. Kim, K. Lee, and D.-K. Jeong, “Synthesisable glitch-less phase rotator with conditionally cascading scheme,” IEEE Electronic Letters, vol.56, no. 11, pp.536-538, May 2020.
11. A. T. Narayanan et al., "A fractional-N sub-sampling PLL using a pipelined phase-interpolator with an FoM of −250 dB", *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1630-1640, Jul. 2016.