

Topics in IC Design

5.1 Introduction to Delay-Locked Loop

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

School of Electrical and Computer Engineering
Seoul National University

2020 Fall

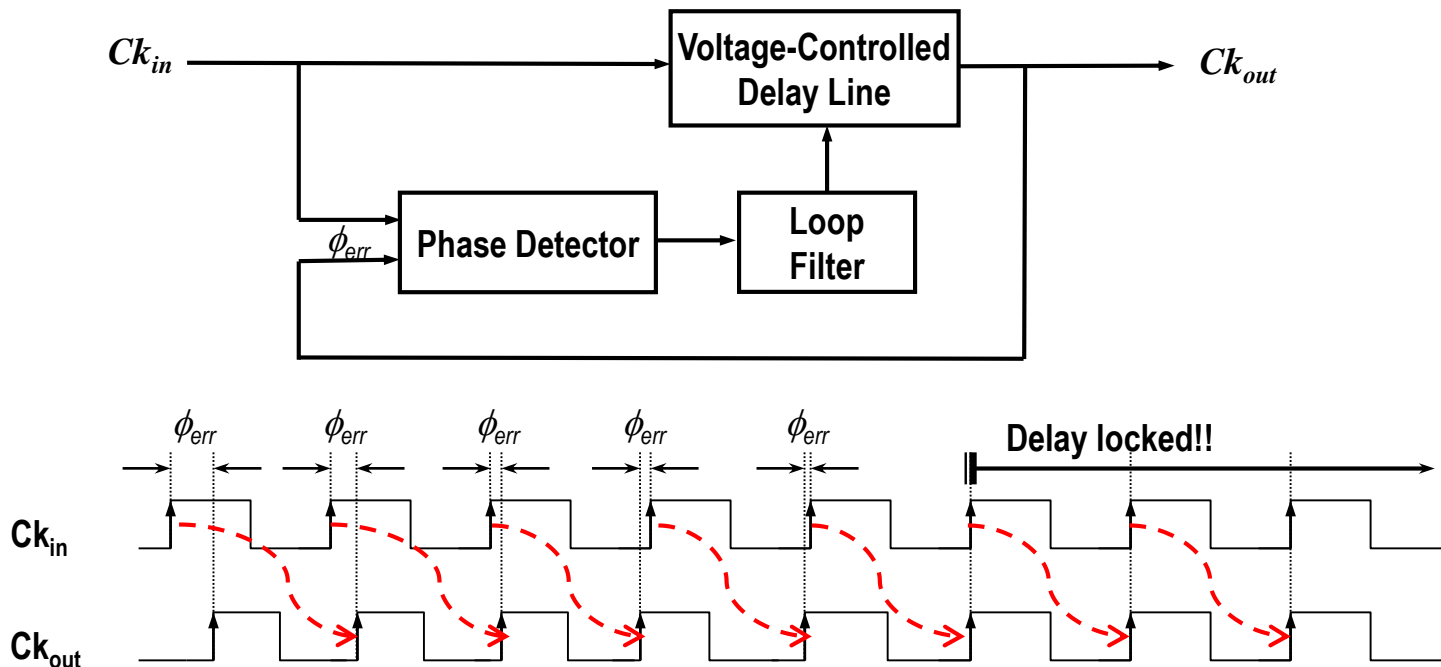
Outline

- Introduction
- Analysis of DLL
- DLL building blocks

What is DLL?

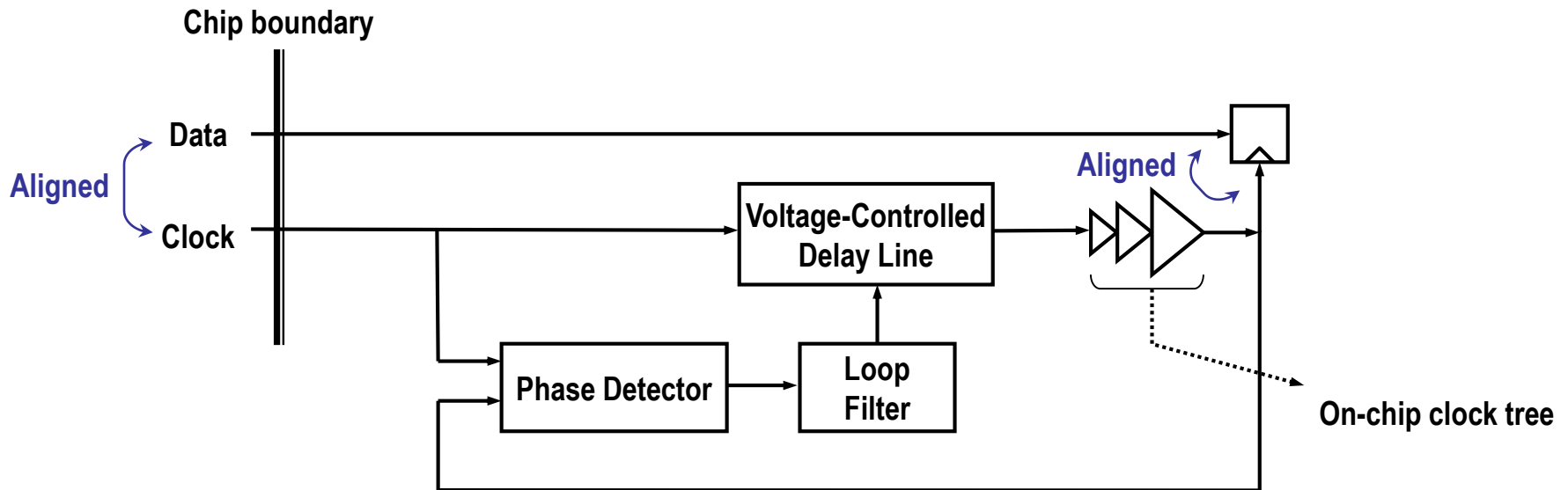
□ Delay-Locked Loop

- A negative feedback system where an delay-line-generated signal is locked to a reference signal
- Input and output frequencies are the same.
- Delayed output is automatically locked in phase to the input in a feedback loop.



DLL Applications (1)

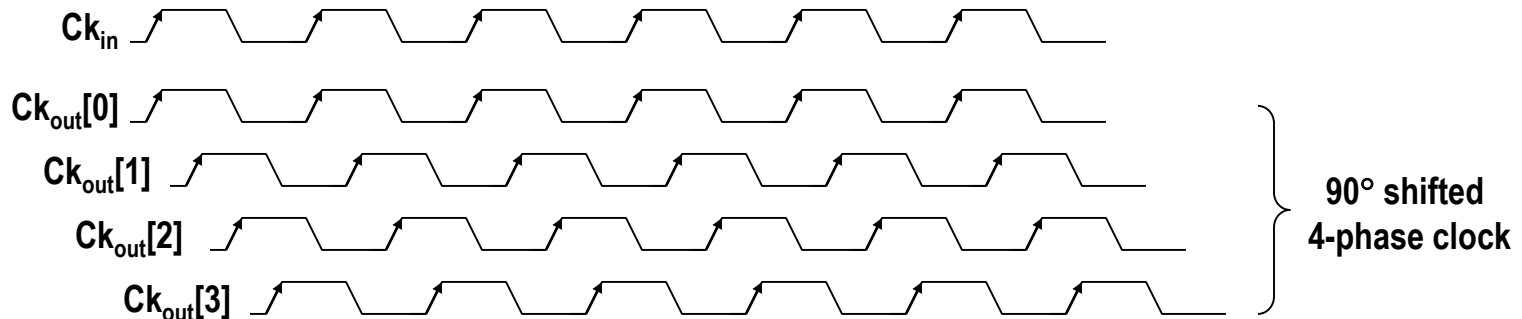
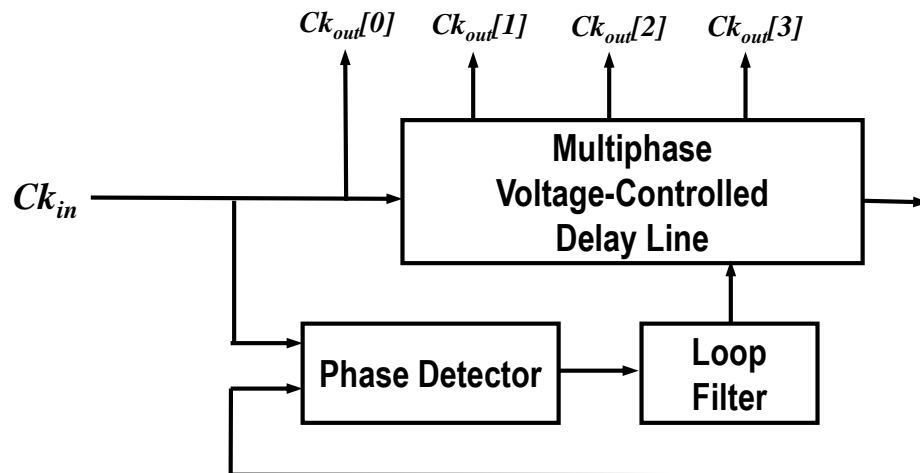
- ❑ Zero delay buffer
 - ❑ Resolves the skew problem due to on-chip clock tree



DLL Applications (2)

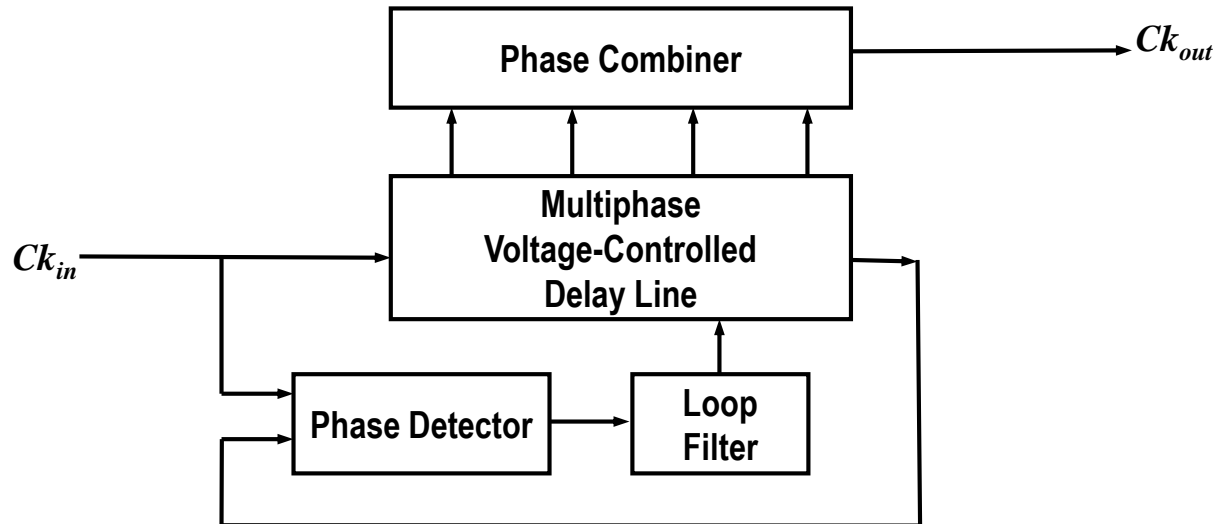
- Multi-phase clock generation

- Multi-phase clock is very useful in many applications



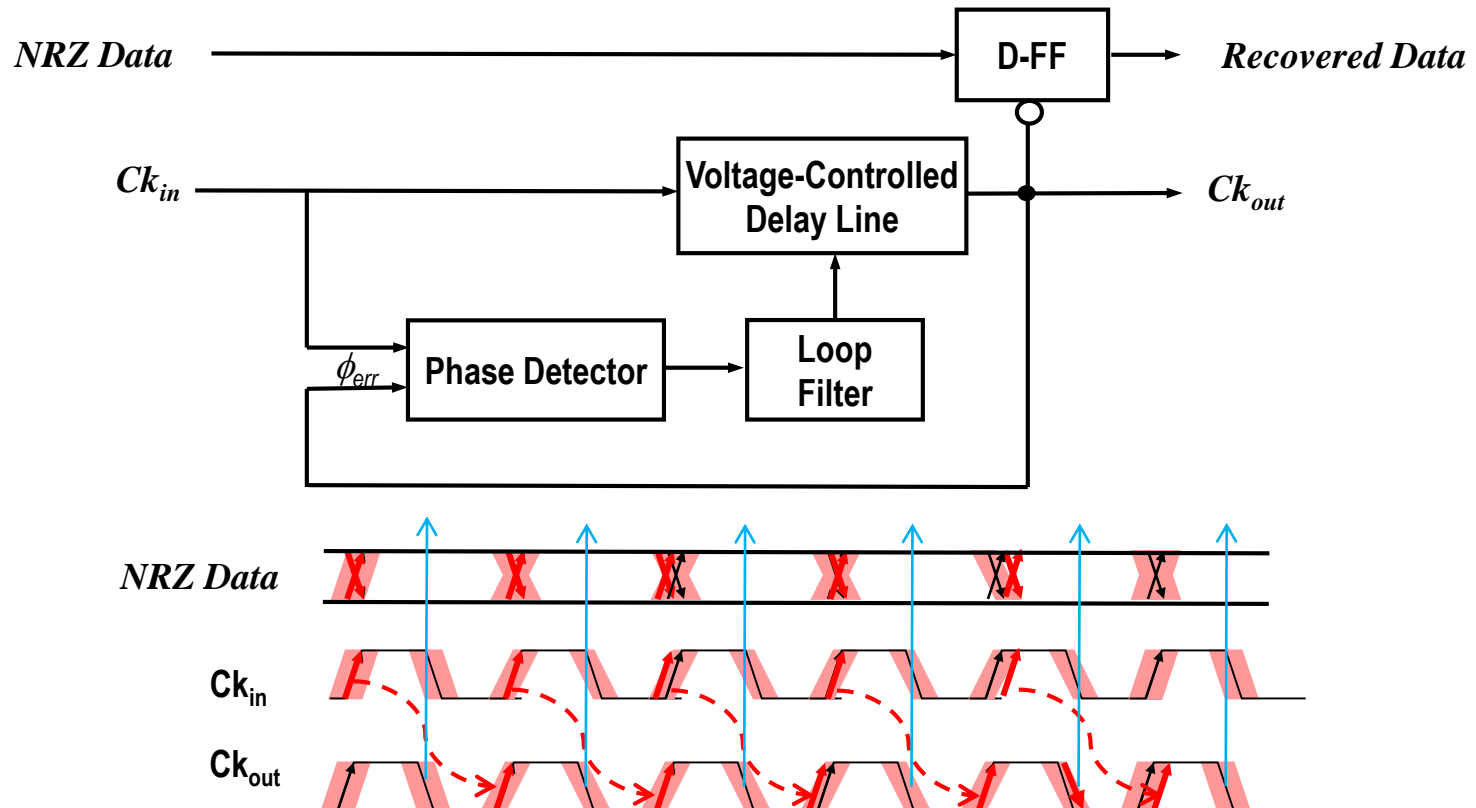
DLL Applications (3)

- ❑ Clock frequency multiplication with multiphase DLL
 - ❑ No jitter accumulation compared with PLL
 - ❑ Much spur generated - No RF applications



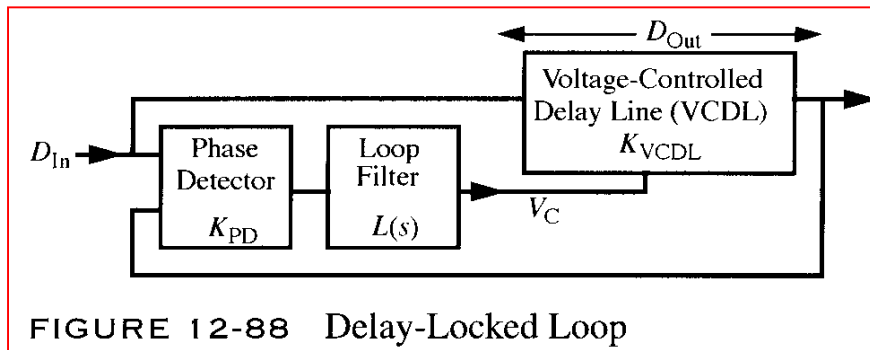
DLL Applications (4)

- ❑ Clock generation for source-synchronous systems
 - ❑ Jitter filtering is not possible
 - ❑ Infinite bandwidth – jitter passes through to the output with a 1 cycle delay

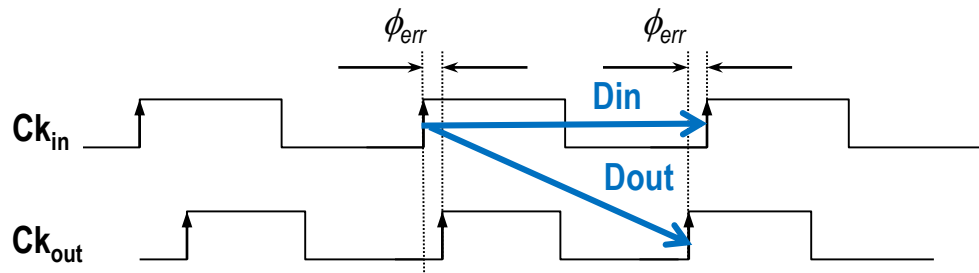


Delay-Locked Loop Transfer Function

- Loop filter for DLL is simpler
 - DLL is a first-order system
 - D_{in} is the period of input, D_{out} is the delay of the VCDL



- With $V_C = \omega_{In}(D_{In} - D_{Out})K_{PD}L(s)$ and $L(s) = 1/sRC$

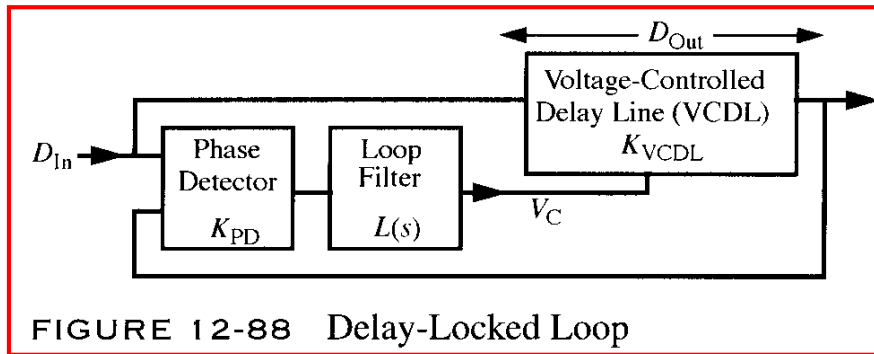


$$\frac{D_{Out}}{D_{In}} = \frac{1}{1 + s/\omega_1}$$

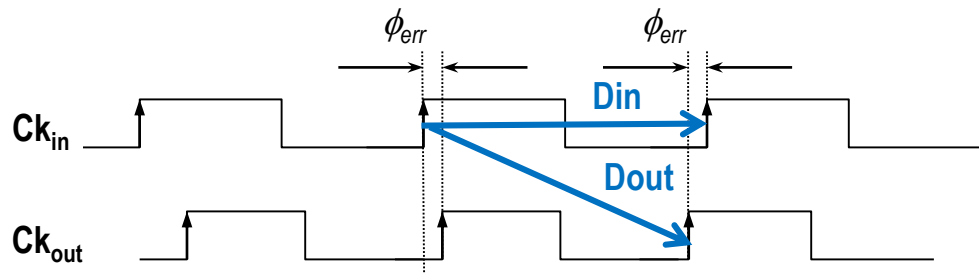
$$\omega_1 = \frac{K_{PD}K_{VCDL}\omega_{In}}{RC}$$

Delay-Locked Loop Transfer Function

- Loop filter for DLL is simpler
 - DLL is a first-order system
 - D_{in} is the period of input, D_{out} is the delay of the VCDL



- With $V_C = \omega_{In}(D_{In} - D_{Out})K_{PD}L(s)$ and $L(s) = 1/sRC$



$$D_{out}(s) = K_{VCDL}V_C(s)$$

$$\frac{D_{out}(s)}{D_{in}(s)} = \frac{K_{PD}K_{VCDL}\omega_{In}L(s)}{1 + K_{PD}K_{VCDL}\omega_{In}L(s)}$$

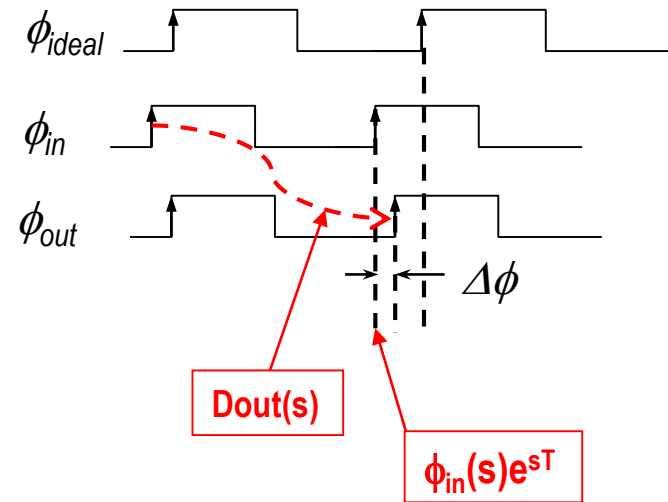
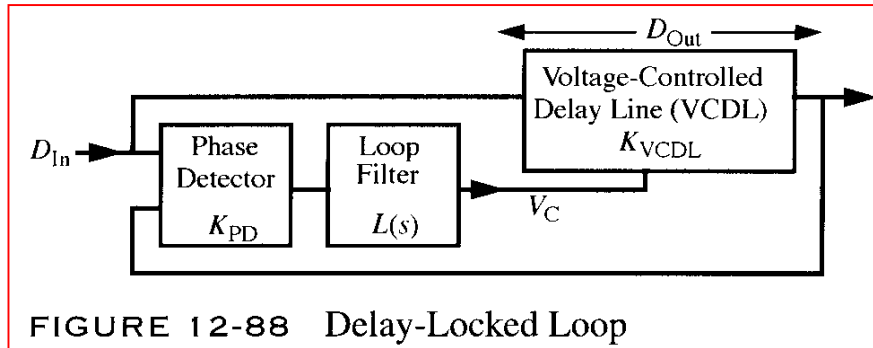
$$= \frac{1}{1 + s/\omega_1}$$

$$\omega_1 = \frac{K_{PD}K_{VCDL}\omega_{In}}{RC}$$

DLL Transfer Function

- What is the jitter transfer function?

- $H(s) = \Phi_{out}(s) / \Phi_{in}(s)$?



Using the following relations,

$$\begin{aligned} \Phi_{in}(s)e^{sT} - \Phi_{in}(s) &= \omega_{In} D_{in}(s) \\ \Phi_{out}(s) &= \Phi_{in}(s)e^{sT} - \omega_{In} D_{out}(s) \end{aligned}$$

Thus,

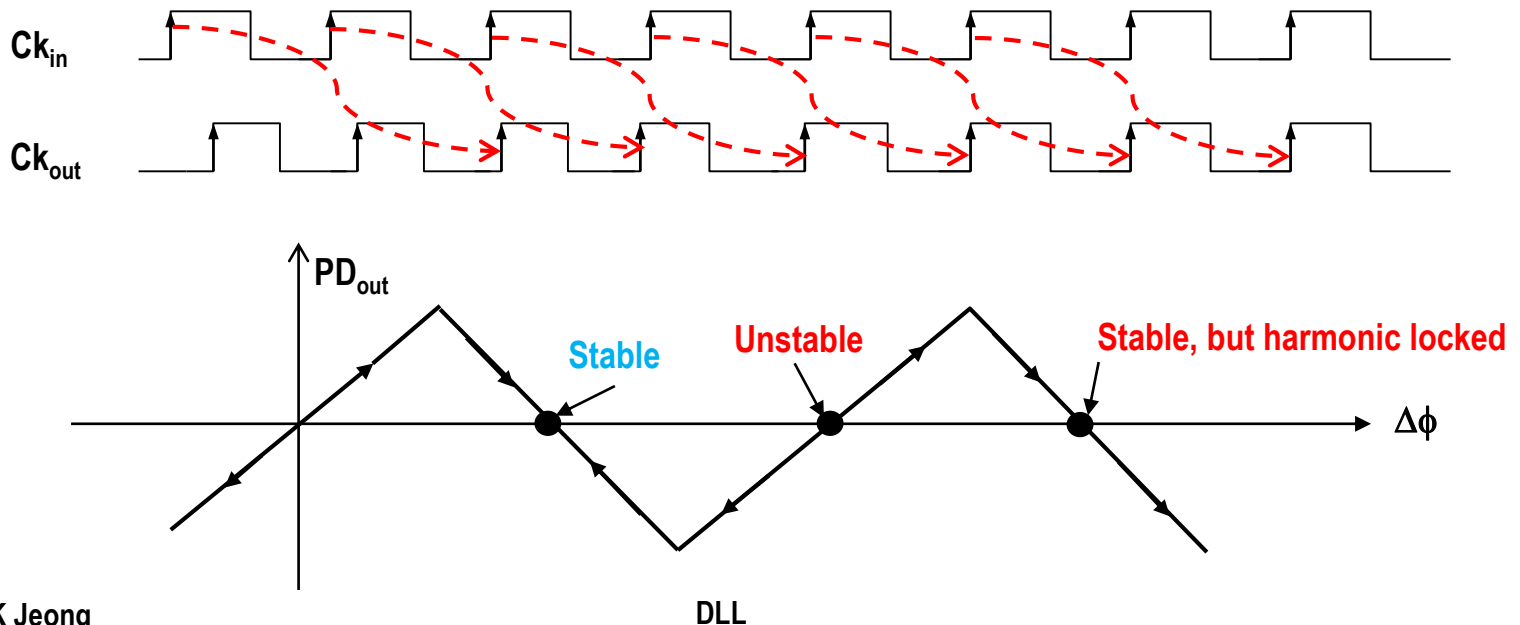
$$\begin{aligned} \frac{\Phi_{out}(s)}{\Phi_{in}(s)} &= \frac{e^{sT} + K_{PD}K_{VCDL}\omega_{In}L(s)}{1 + K_{PD}K_{VCDL}\omega_{In}L(s)} \\ &= \frac{1 + s/\omega_1 e^{sT}}{1 + s/\omega_1} \end{aligned}$$

\Leftarrow High-pass filter !!

Close to an all-pass filter !!

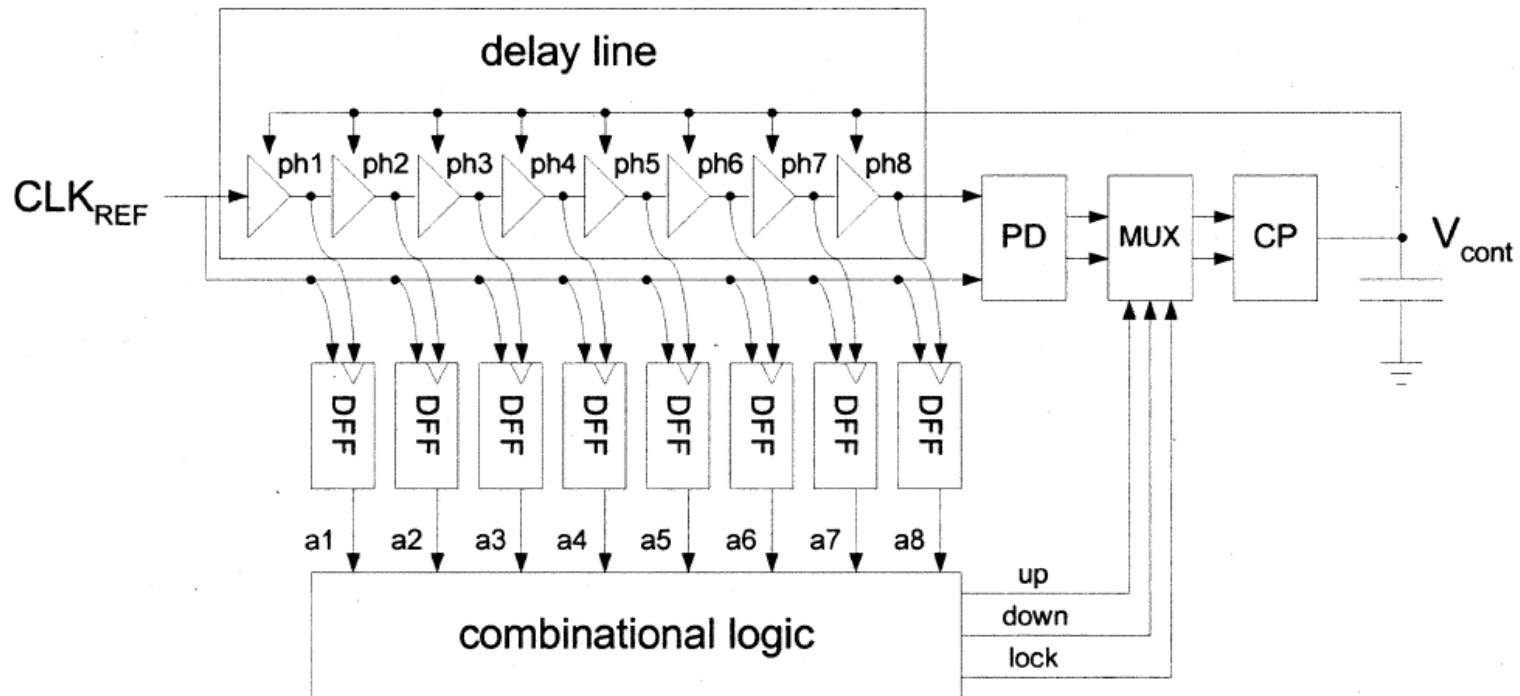
Harmonic Lock Problem

- **False lock or harmonic lock occurs when initial delay is greater than $2 * T_{in}$ with PD locking at 0° .**
 - In false lock, single phase output could be functionally ok but loop characteristics change
 - Multiphase clocks malfunction
 - Start DLL with reset. On reset, V_C forced to 0.



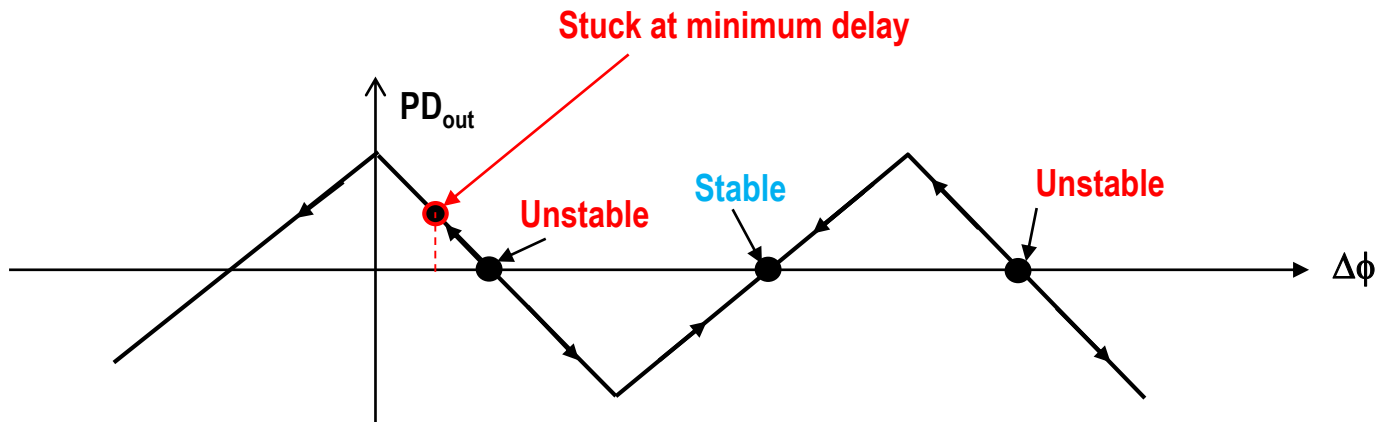
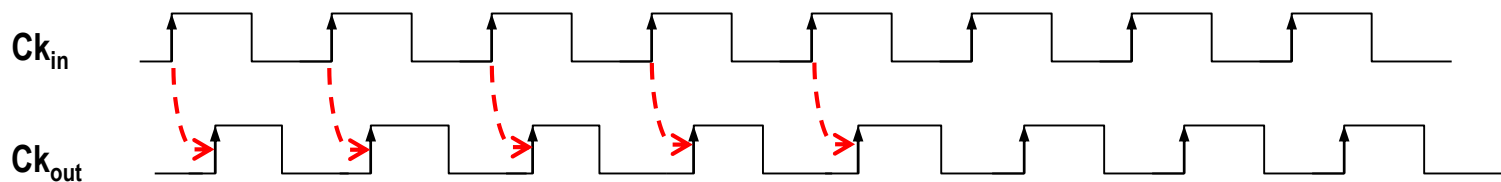
Harmonic Lock Problem

- Avoiding harmonic lock
 - Use intermediate outputs



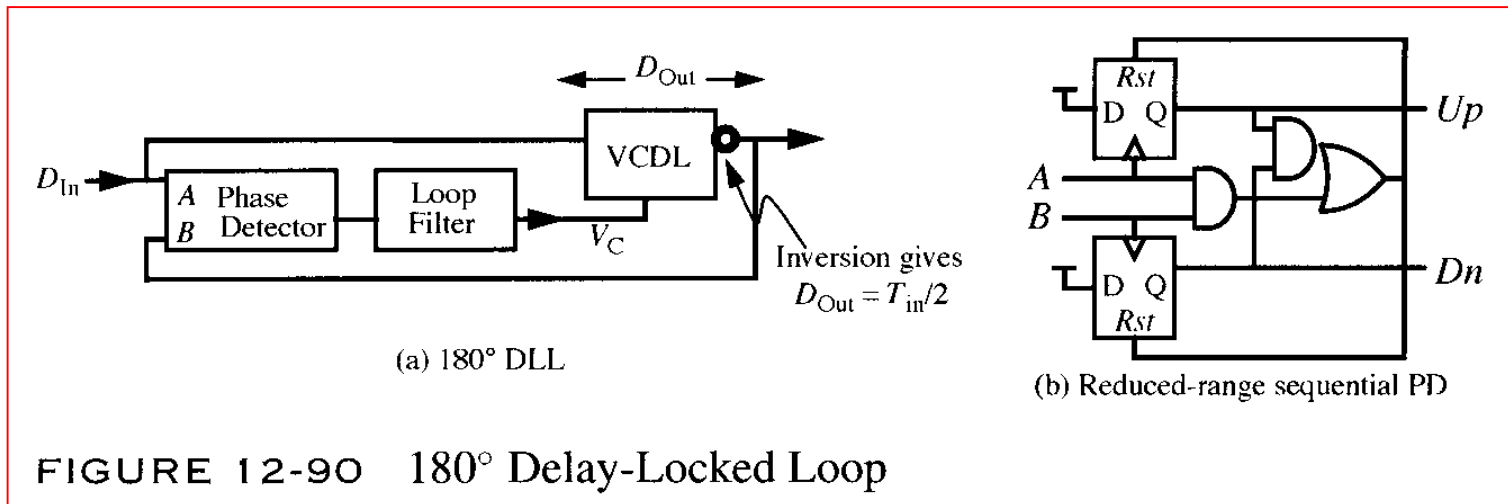
Stuck Problem

- Depending on initialization, stuck problem may arise.
 - Delay is at minimum but PD keeps comparing input with current-cycle output not with one-cycle delayed output.



Half Clock Delay DLL

- **DLL with only half a clock delay**
 - DLL locks at $D_{out} = T_{in}/2$
 - Limiting phase input difference with additional reset path for PFD can solve the stuck problem



References

- **[6.1.1] M. Lee, “Jitter Transfer Characteristics of Delay-Locked Loops—Theories and Design Techniques”, JSSC vol. 38, no 4, 2003**
- **[6.1.2] B. Kim, “A low-power CMOS Bluetooth RF transceiver with a digital offset canceling DLL-based GFSK demodulator,” JSSC vol. 38, no 10, 2003**

Delay-Locked Loops

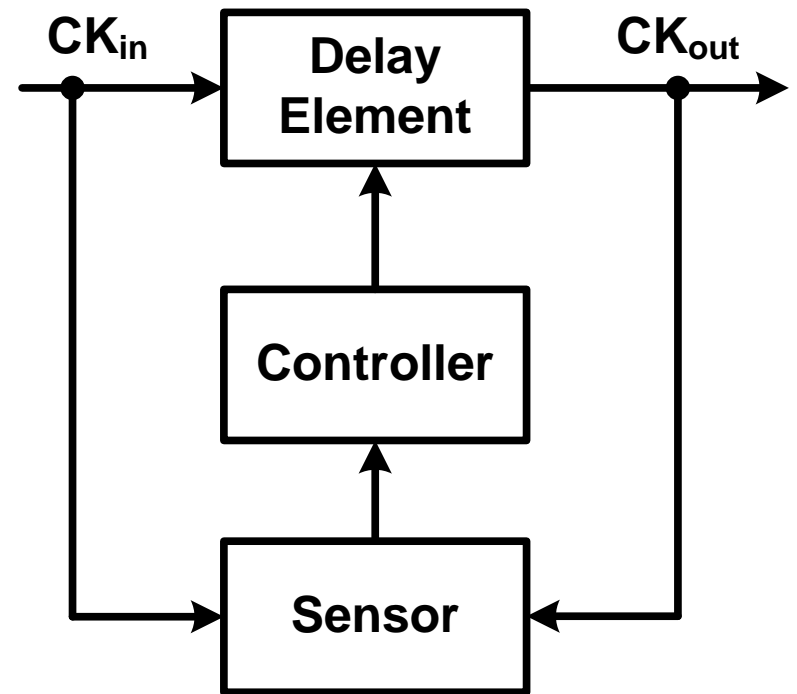
5.2 Structures

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

October 19, 2020

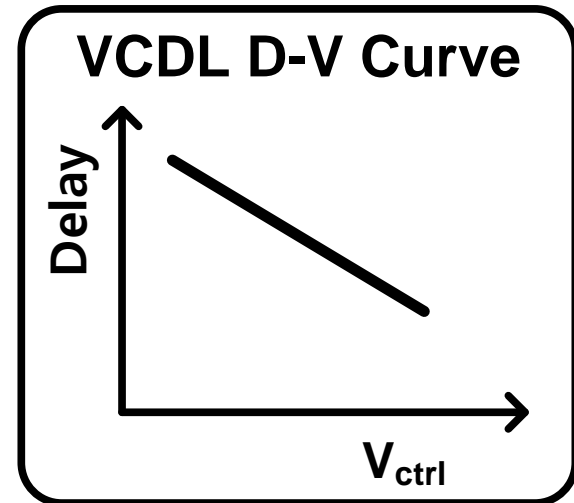
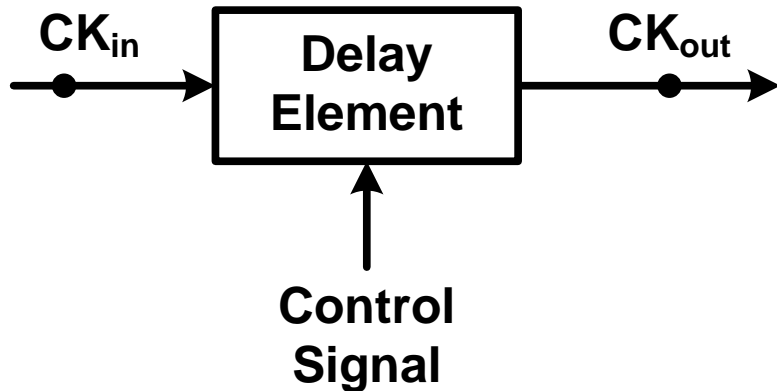
Delay-Locked Loop (DLL)

- Matching **TIMING** at separate two points by adjusting **DELAY**
- **Negative feedback loop**
- **DLL does not self-generate the output clock**



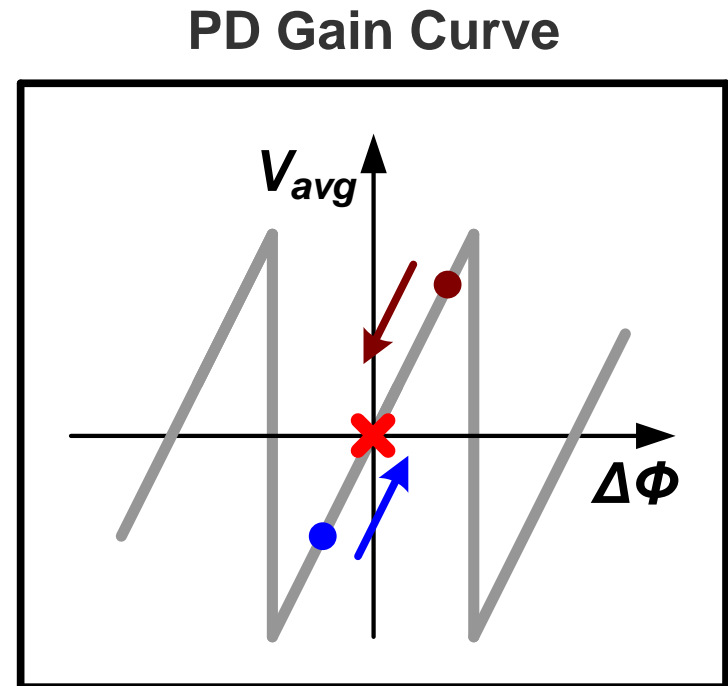
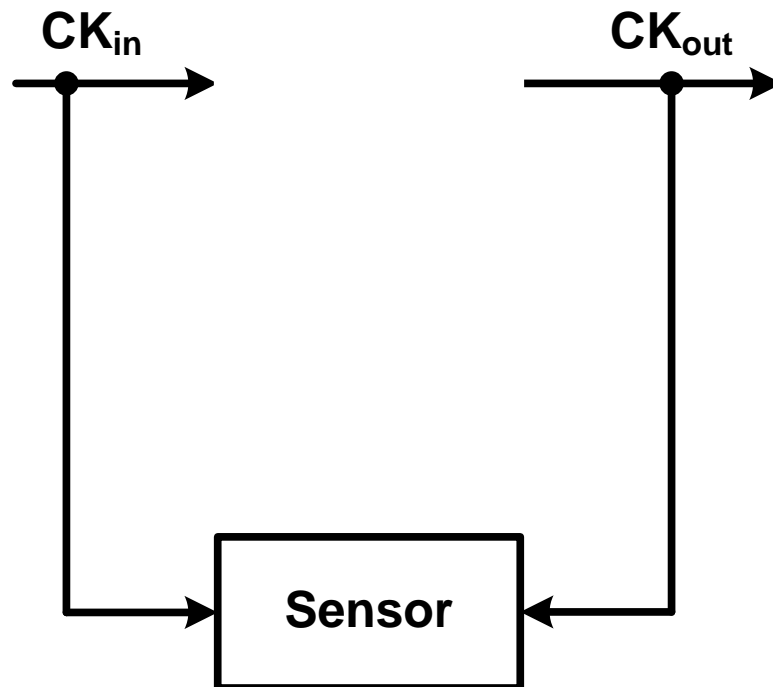
DLL Building Block: Delay Line

- Delay element with control port (Voltage-Controlled Delay Line)
- Control signal adjusting delay
- Typically, by tuning RC time constant



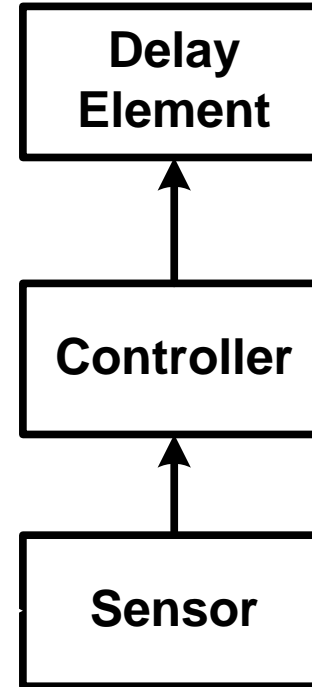
DLL Building Block: Phase Detector

- Phase detector (PD) for sensing the phase difference
- Linear phase detector, bang-bang phase detector



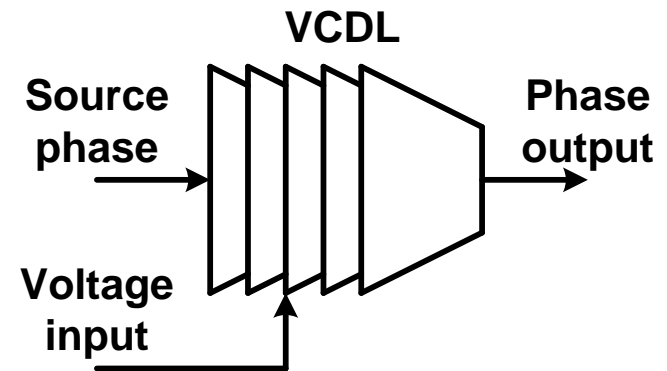
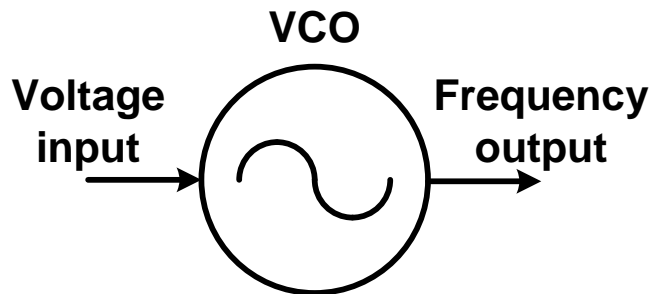
DLL Building Block: Loop Filter

- Loop filter for controlling delay element according to the sensed phase difference
- For DLL, only the first-order lowpass filter or an integrator with a charge-pump is sufficient.

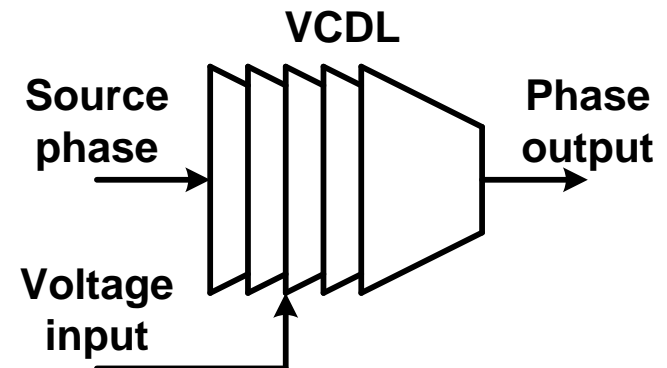
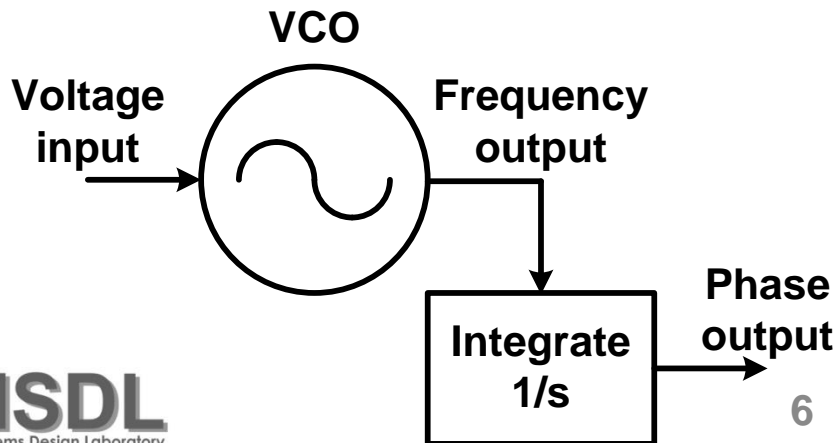


Comparison with Phase-Locked Loop

- Voltage-controlled oscillator vs. Voltage-controlled delay line



Phase Domain Model



Comparison with Phase-Locked Loop

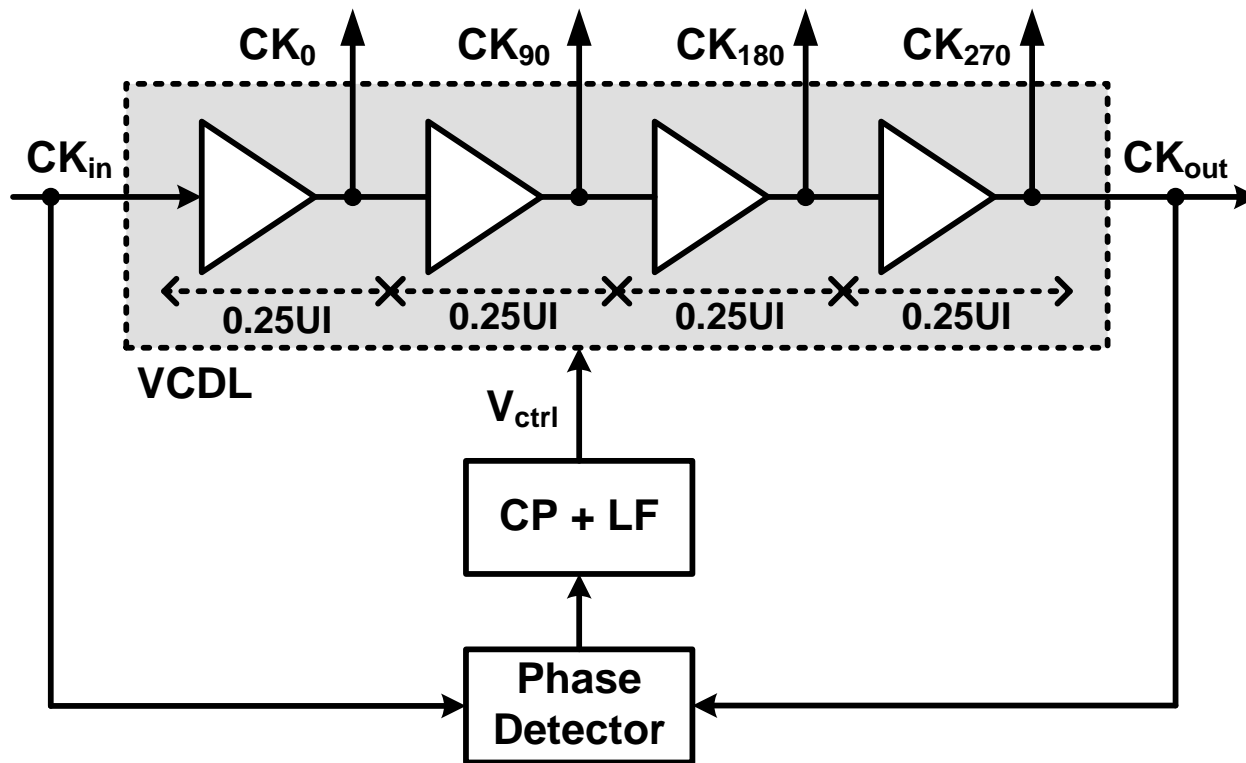
- **PLL has a pole at zero frequency inherently, but DLL does not**
- **Stability and settling issues are relaxed in DLL**
- **Jitter does not accumulate in VCDL**
- **But VCDL cannot adjust frequency:**
 - **Jitter is added, not removed.**
 - **difficult to generate different output frequency**
 - **Multiphase Delay Line or Multiplying DLL is used**

Types and Applications of DLL

- **Type I**
 - Multi-phase clock generation, zero-delay buffer
 - Only one input
 - PD compares input and output of the VCDL
- **Type II**
 - Data recovery
 - Two inputs (i.e. data and clock)
 - PD compares VCDL output and the other input

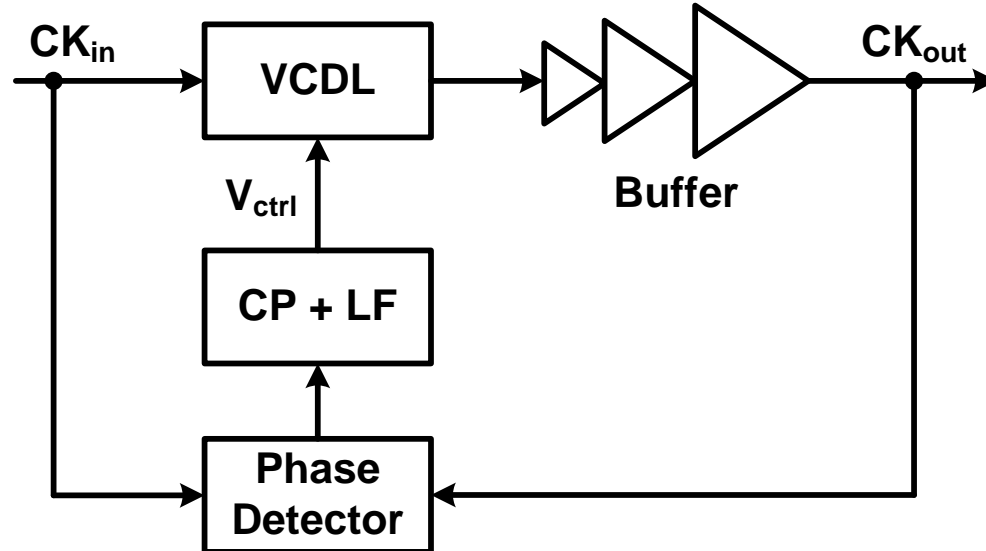
DLL Applications - 1

- Multi-phase clock generation (Type I)



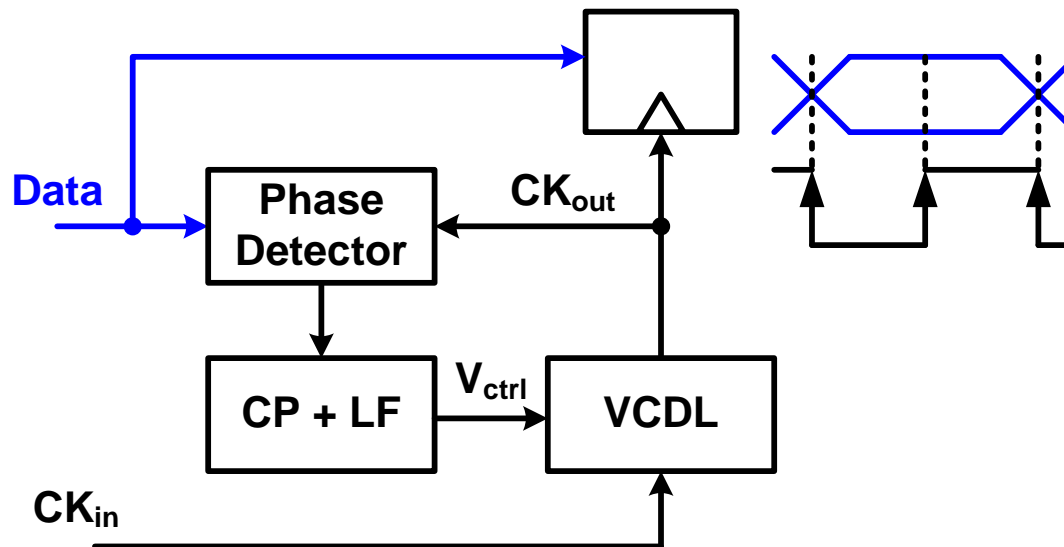
DLL Applications - 2

- Zero-delay buffer (Type I)
- Driving the clock to a large load without adding skew
- Resolving the skew problem due to on-chip clock tree



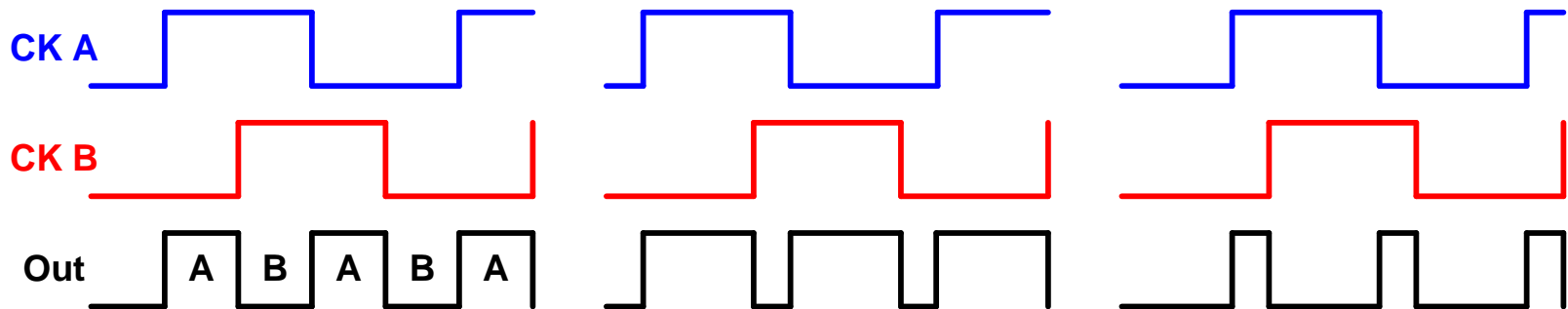
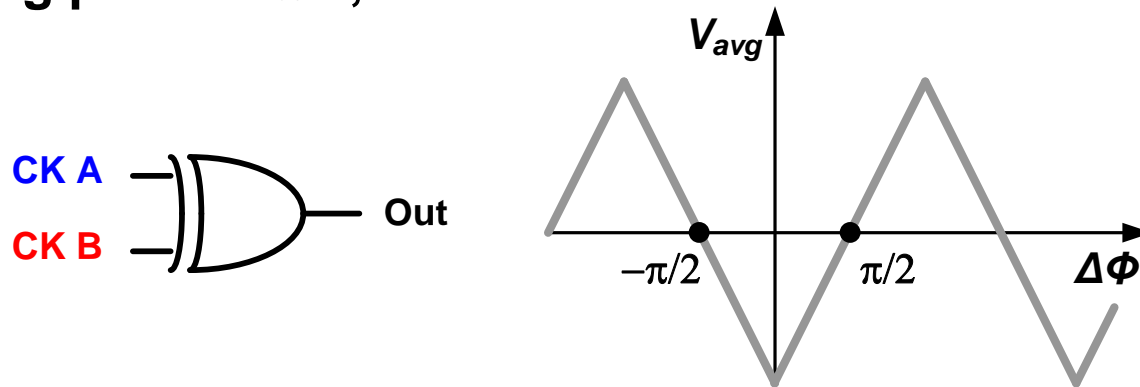
DLL Applications - 3

- Data recovery (Type II)
- Recovering data by sampling at the eye center
- Jitter is not filtered



Building Block Examples: PD

- XOR phase detector - Duty ratio sensitive
- Locking point at $\pi/2$, not zero



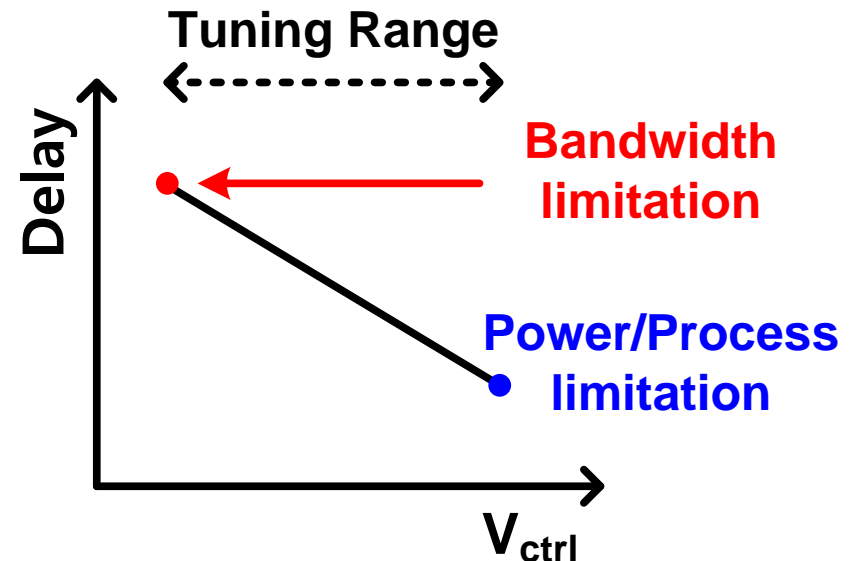
Phase-locked

A lead

B lead

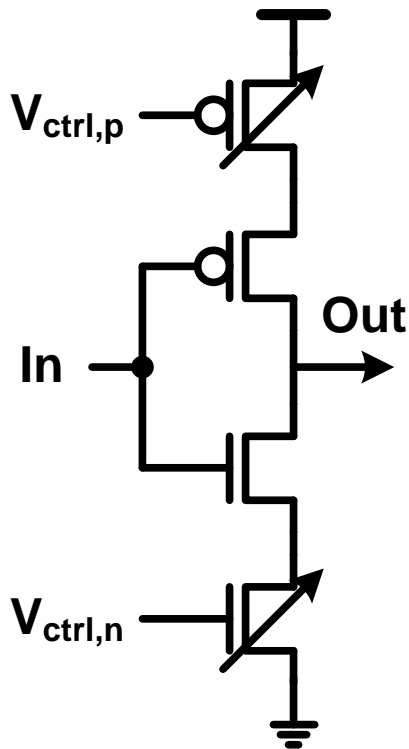
Building Block Examples: Delay

- For most of CMOS circuits, RC time constant determines circuit BANDWIDTH and DELAY (Time constant $\cong 1/\text{bandwidth}$)
- **Upper limit:** process or power
- **Lower limit:**
- VCDL bandwidth should be higher than the clock frequency even when the VCDL is **SLOW**
- **Duty ratio must be maintained**

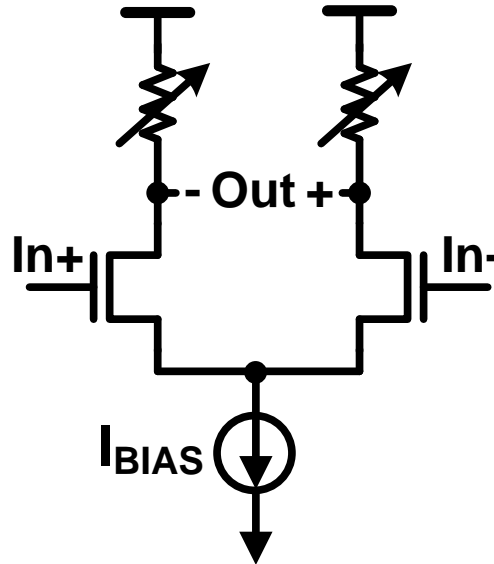


Building Block Examples: Delay

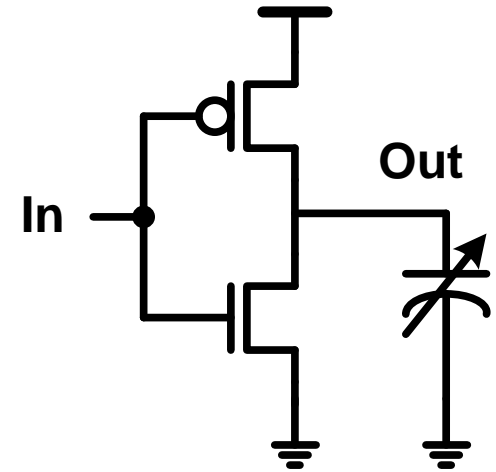
- Delay element examples



R tuning



R Tuning



C tuning

Delay-Locked Loops

5.3 False Locking Issue

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

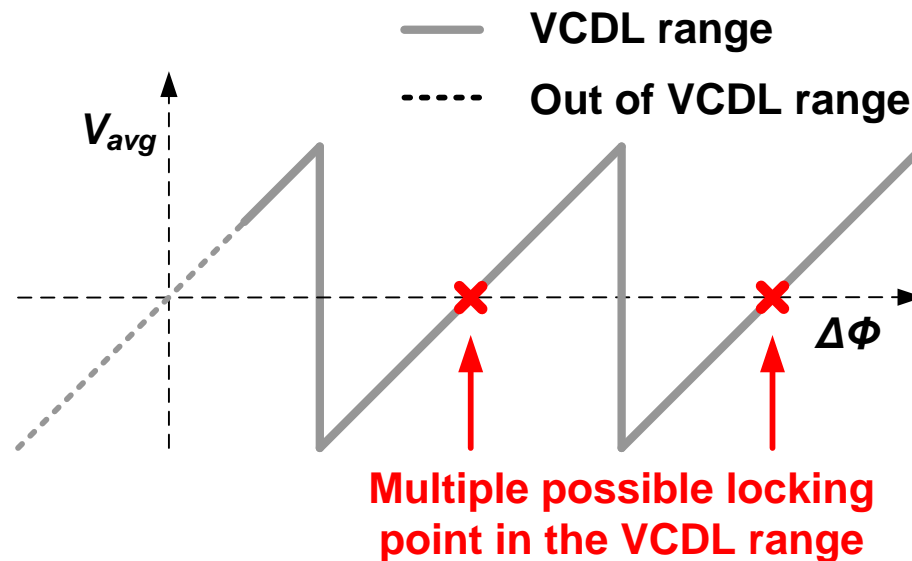
October 19, 2020

False Locking Issue in DLL

- **Harmonic locking**
 - DLL adjust DELAY, according to the information from PHASE detector
 - Same phase, but different delay: Harmonic
- **Stuck locking**
 - Finite delay range of a delay line
 - When the desired phase of the output signal is out of the range, the loop will be stuck at the edge of the range

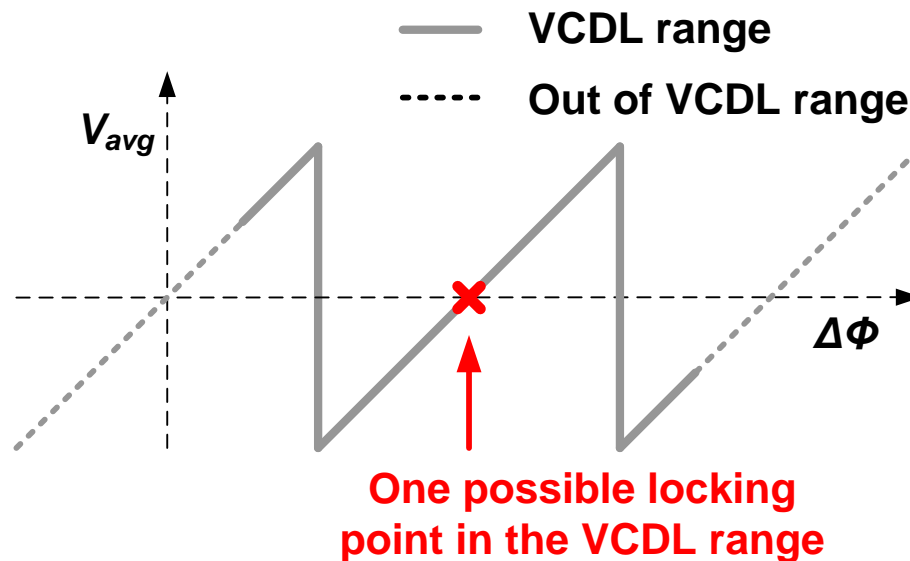
Harmonic Locking

- Harmonic locking occurs when the VCDL range is wide



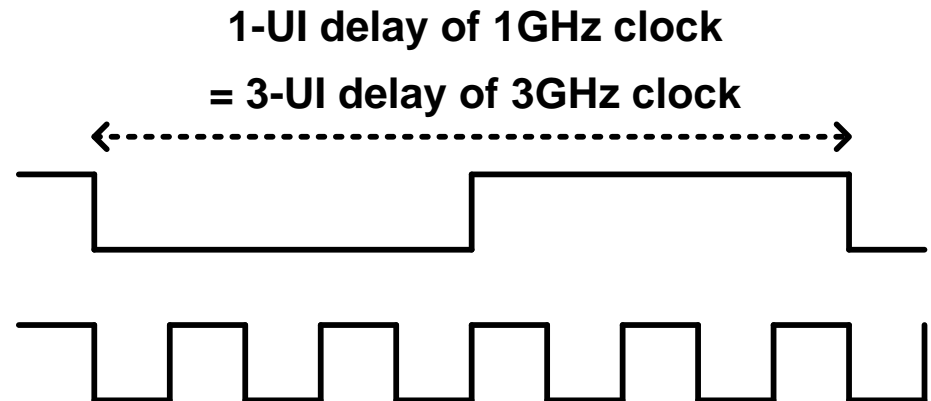
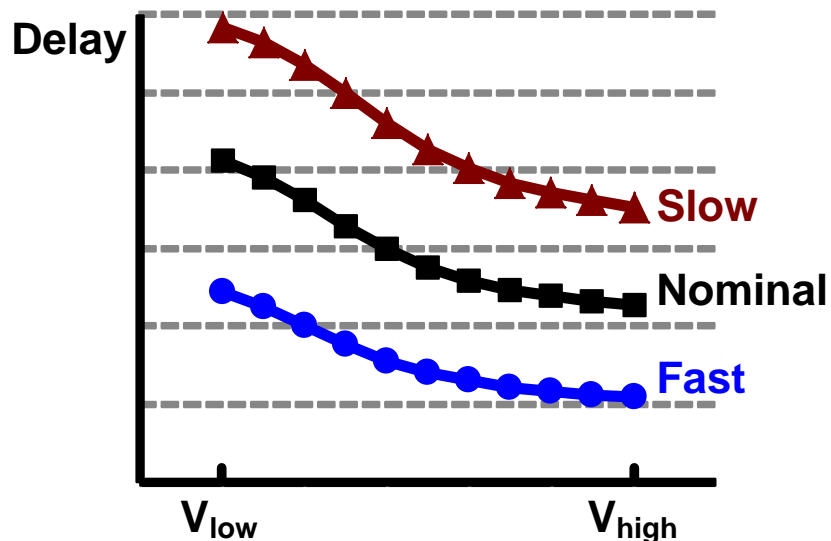
Harmonic Locking

- Harmonic locking occurs when the VCDL range is wide
- Limiting the VCDL range eliminates harmonic locking, but



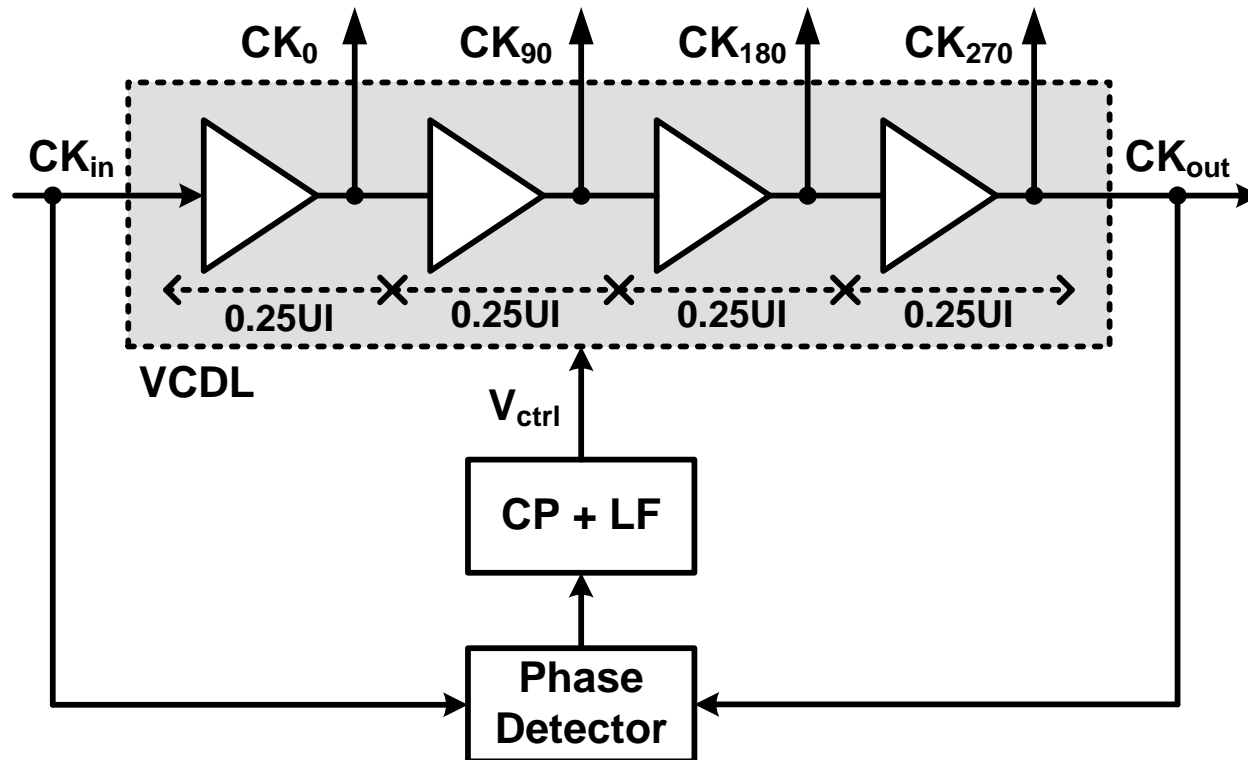
Harmonic Locking

- Harmonic locking occurs when the VCDL range is wide
- Limiting the VCDL range eliminates harmonic locking, but
- PVT variations, wide operating frequency range



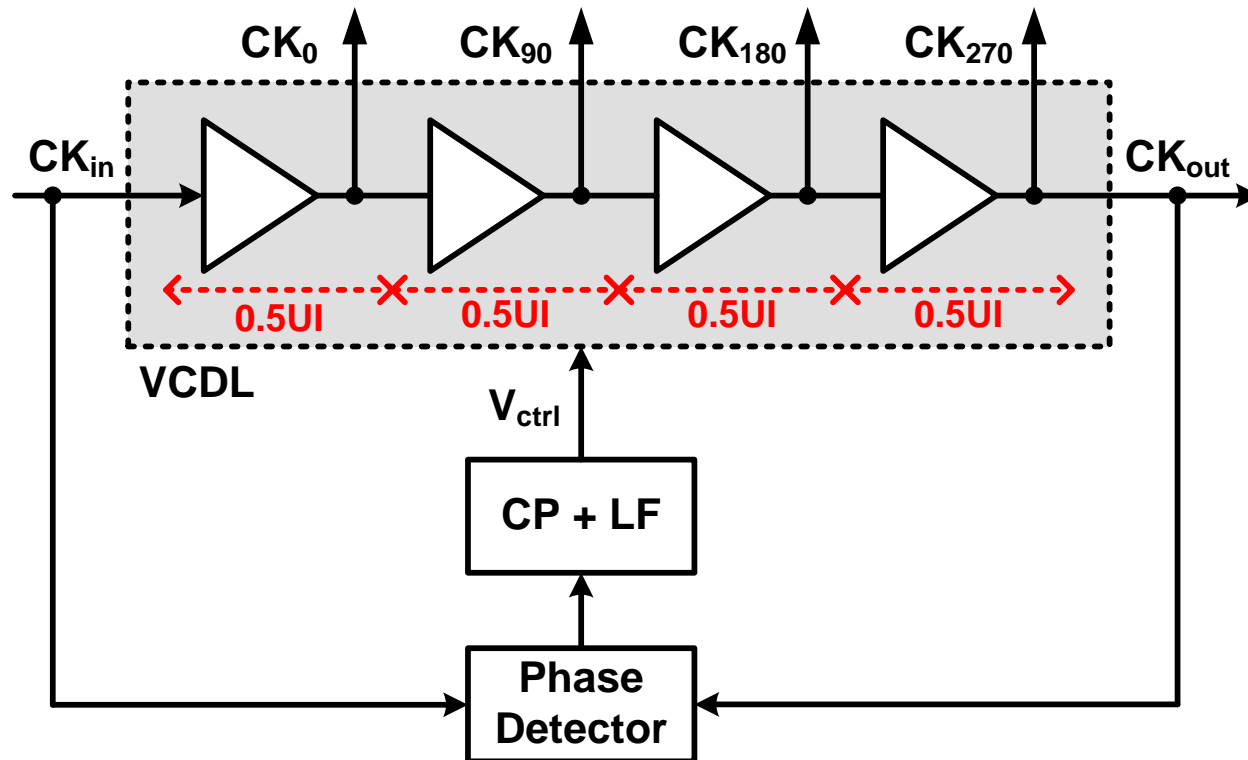
Harmonic Locking

- Example: Type-I DLL for multi-phase generation
- Desired phase shift per stage: $0.25 UI$



Harmonic Locking

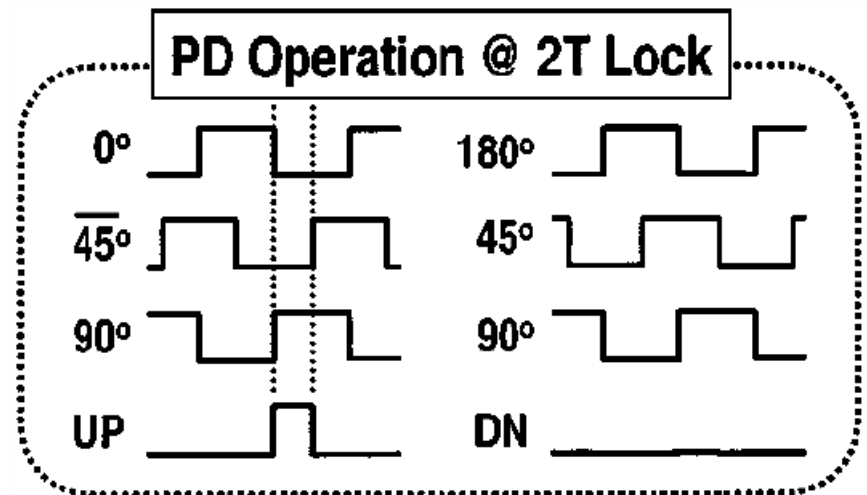
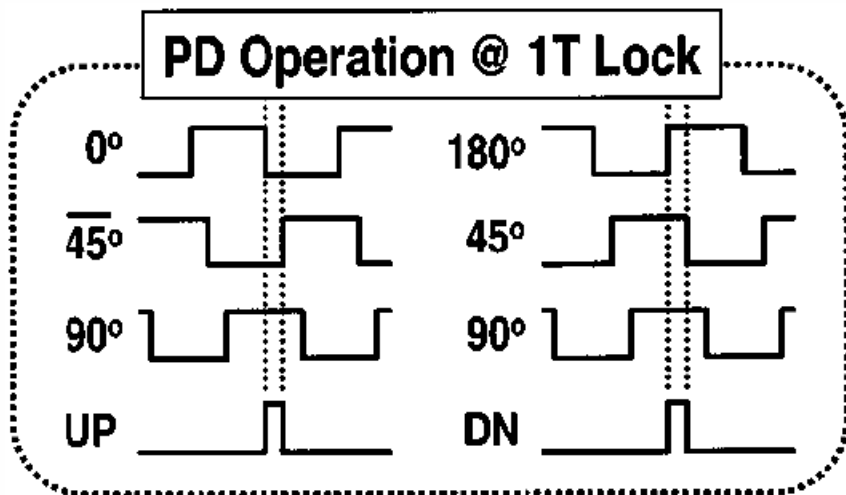
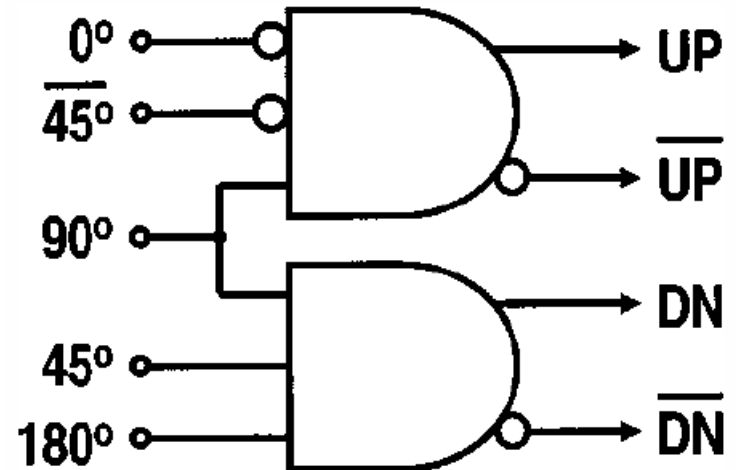
- PD does not know what's wrong: because phase is locked
- Maybe, a delay detector solves the issue



Delay Detector Example

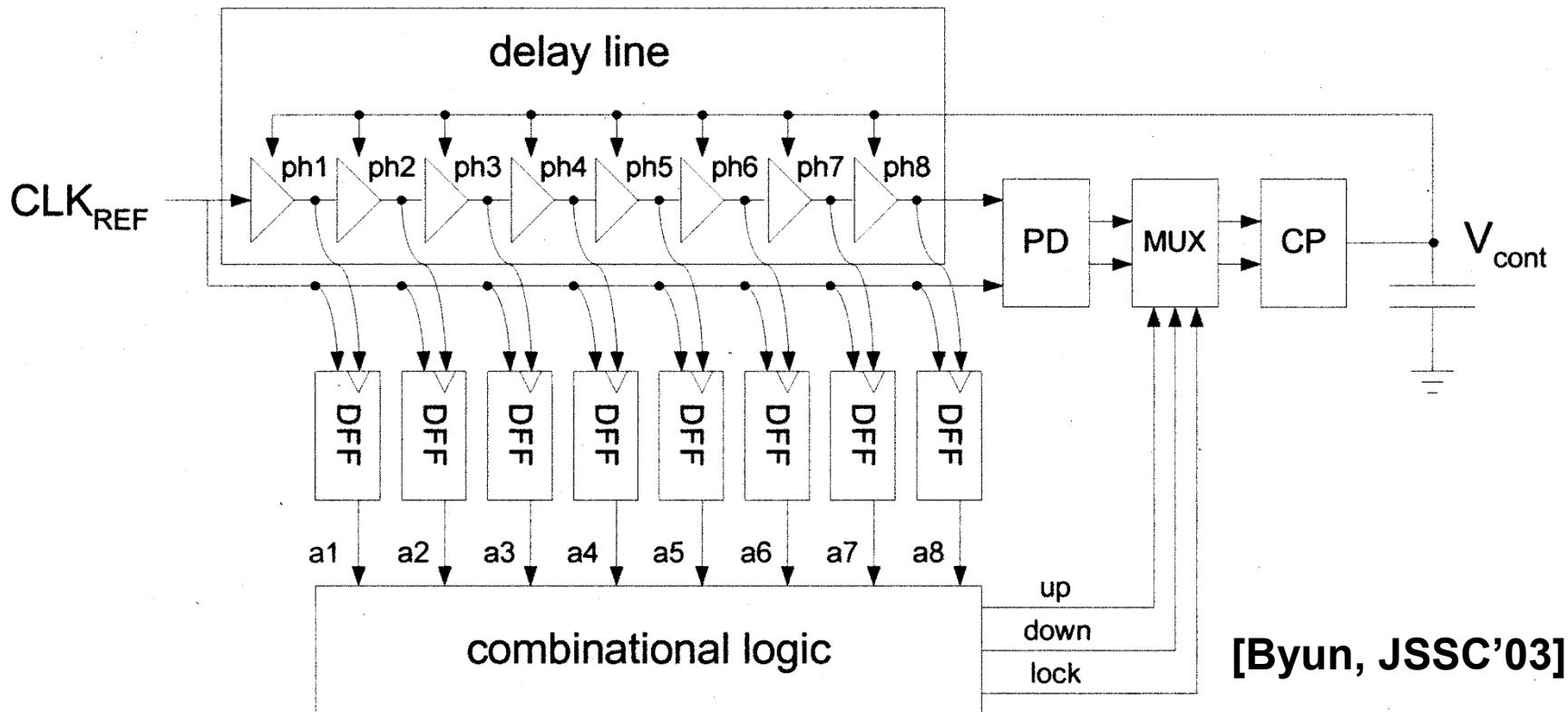
- Phase detection using multiple phases

[Jung, JSSC'01]



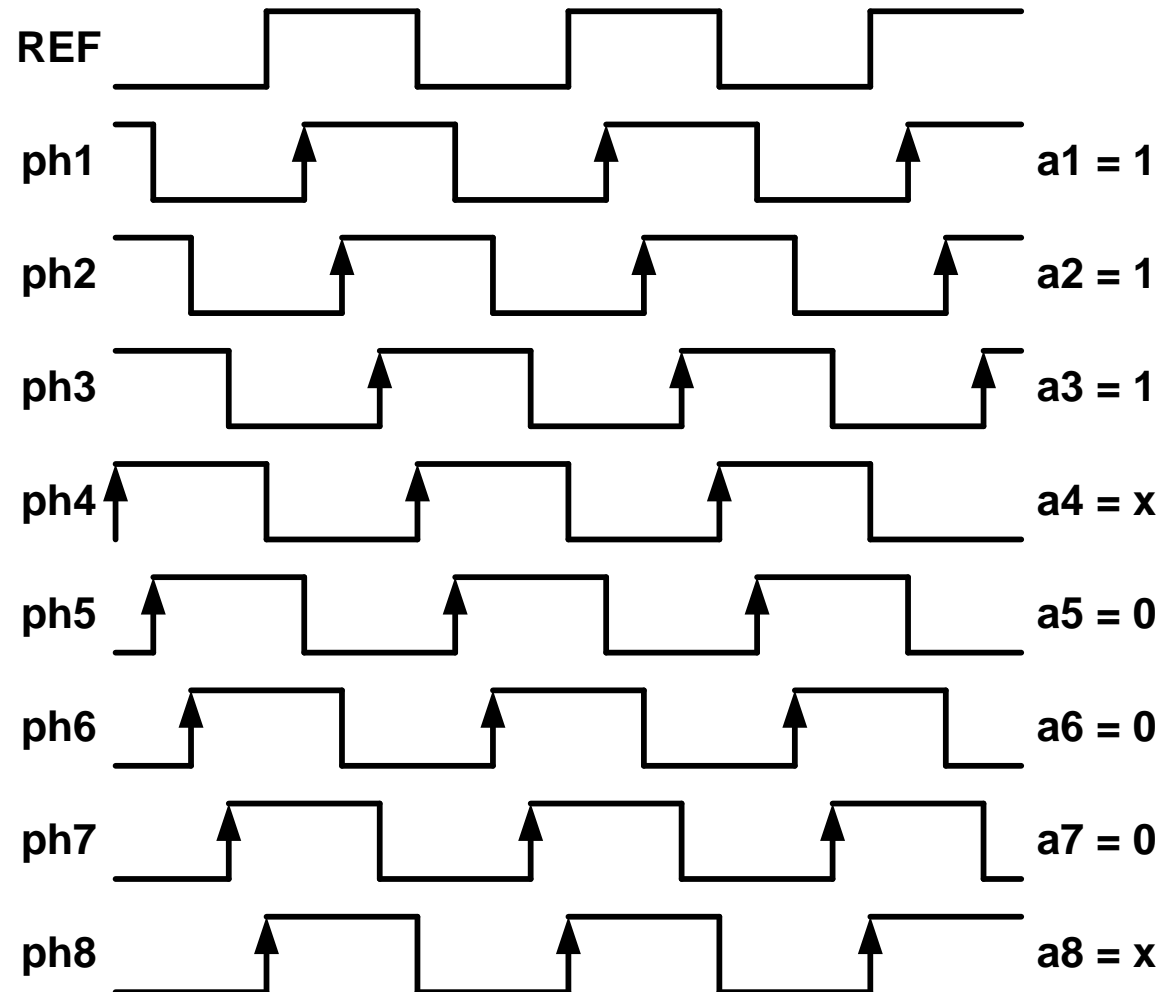
Harmonic Lock Detection

- Comparing all the multi-phases with the reference phase
 - Assuming that duty-cycle is about 50%



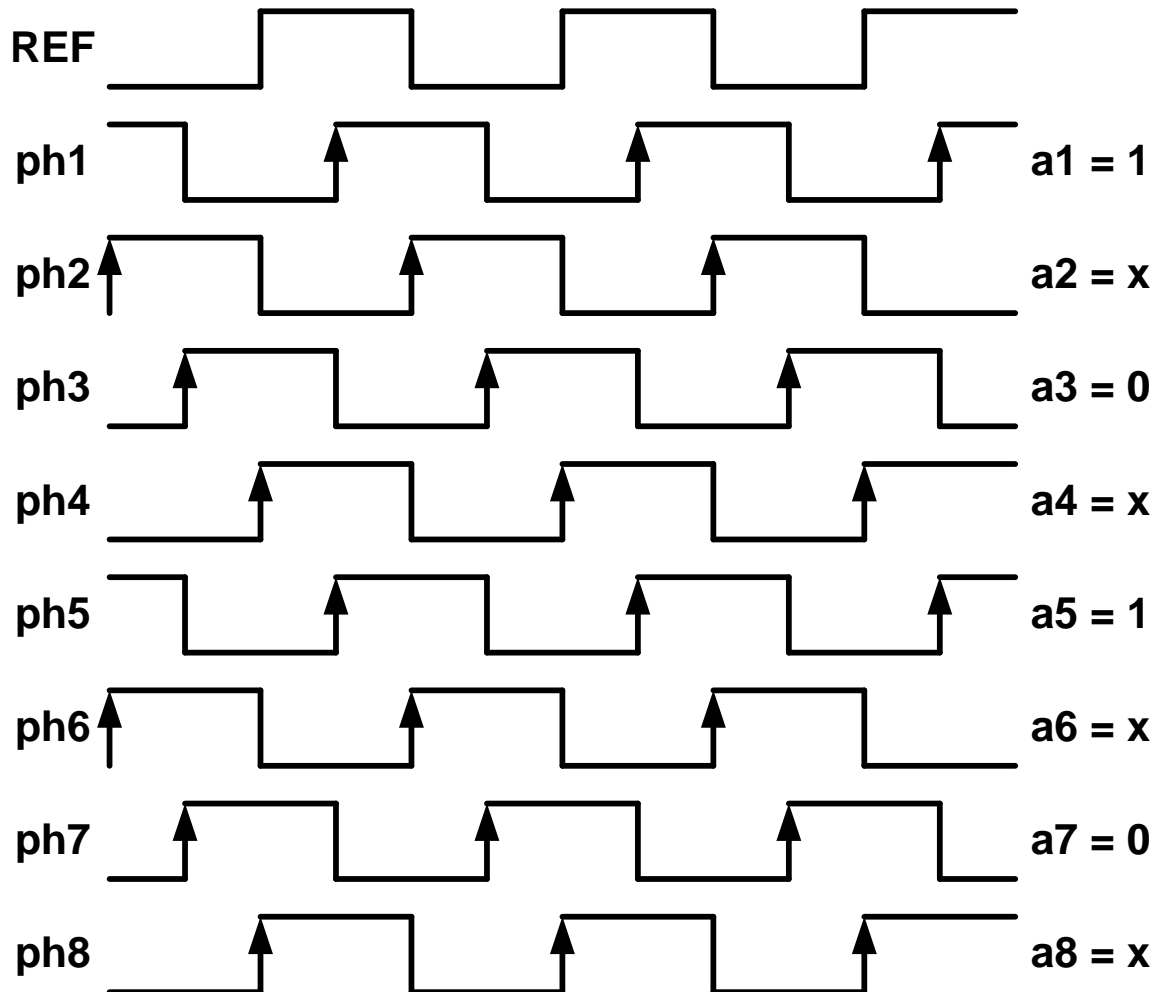
Harmonic Lock Detection

- Proper locking



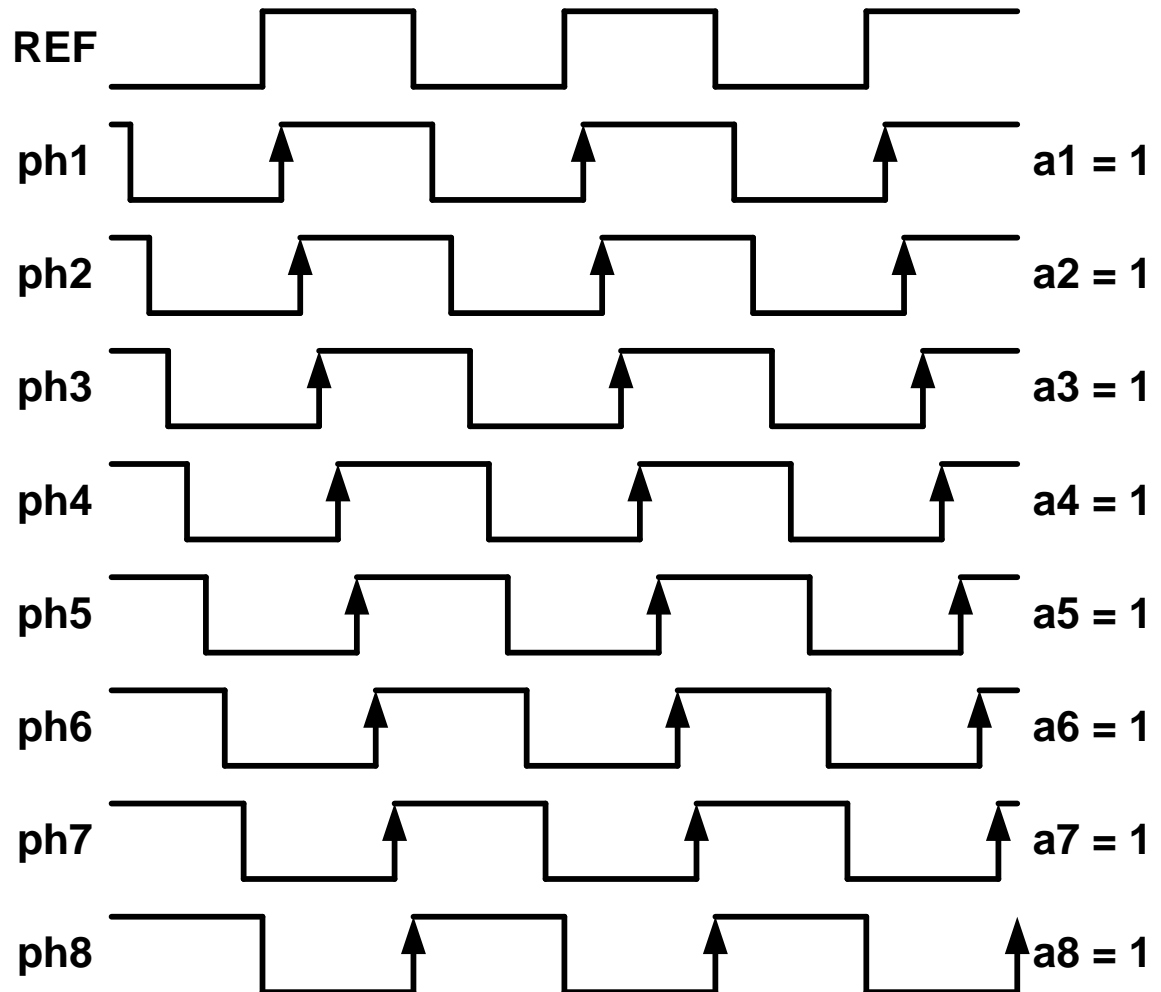
Harmonic Lock Detection

- Harmonic locking
- Phase = $2T$



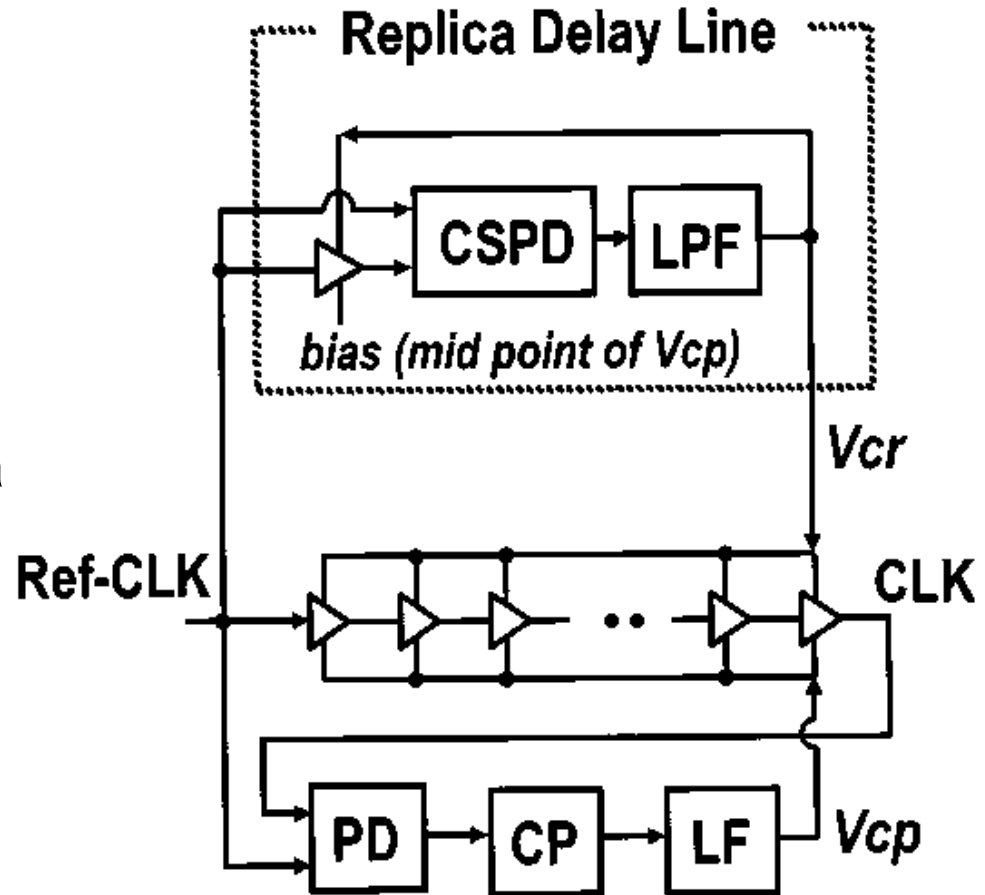
Harmonic Lock Detection

- Stuck locking
- Phase $< 0.5T$



Relaxing Harmonic Lock

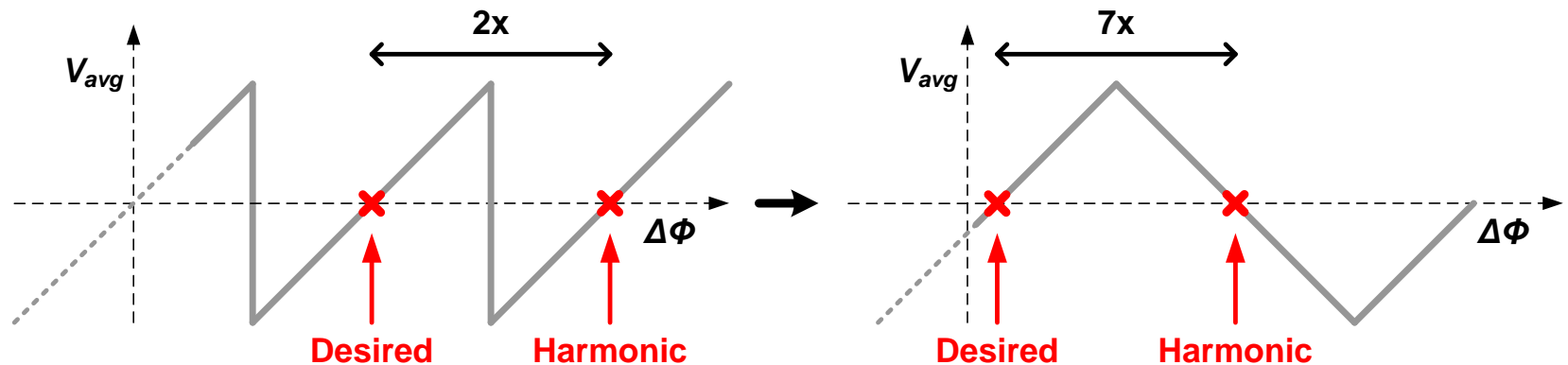
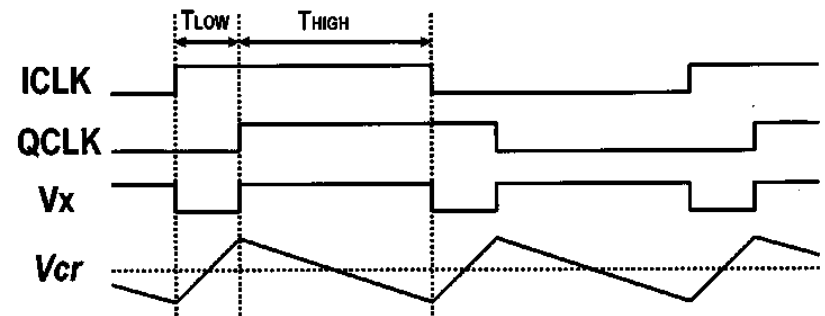
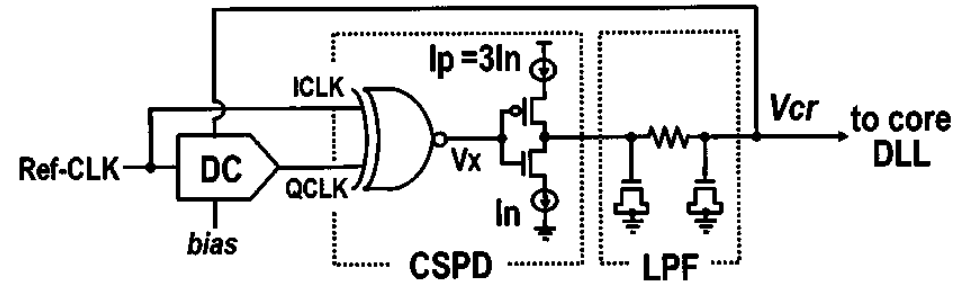
- PD locking point repeats every 2π
- VCDL range wider than 2π results in harmonic lock
- Using a $1/N$ of VCDL (replica VCDL) to relax the range requirement



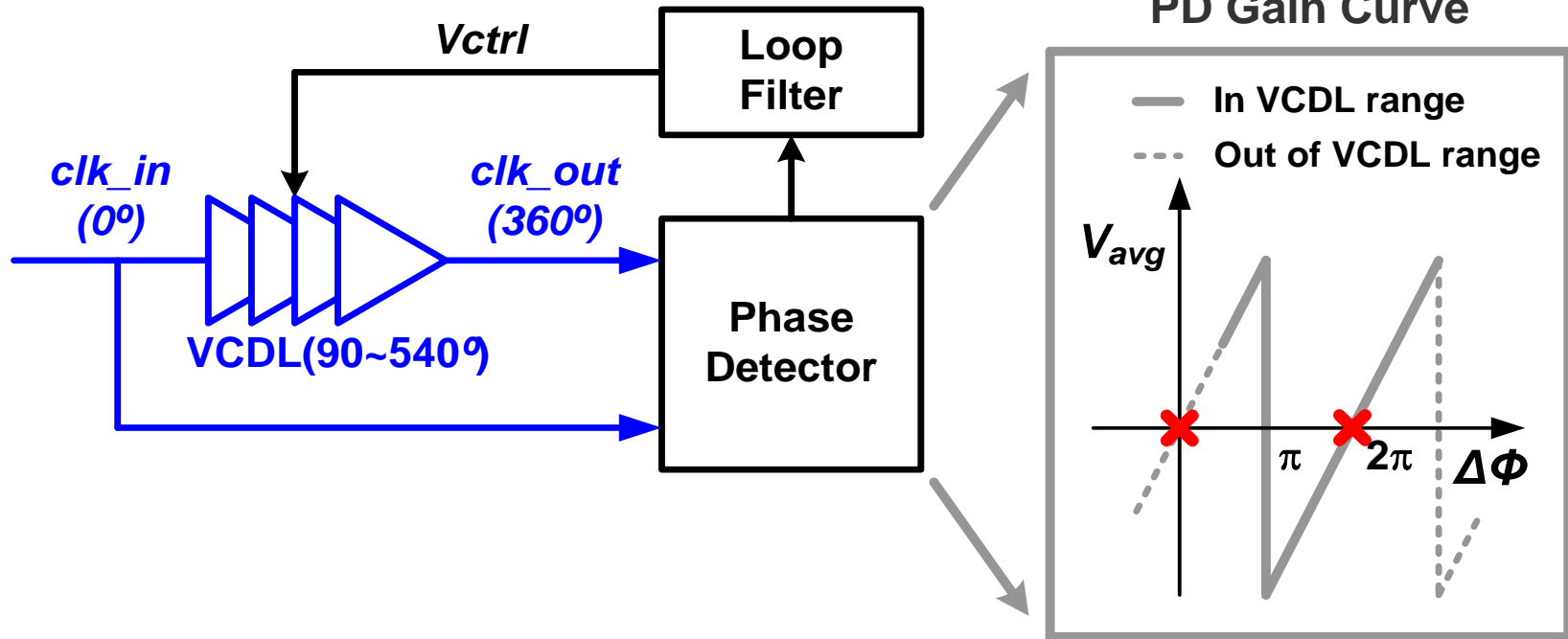
[Moon, JSSC'00]

Relaxing Harmonic Lock

- 1/8 of VCDL locks at $\pi/4$
- XOR PD offers $\pi/2$ locking
- Intentional current offset of charge-pump shifts the locking point

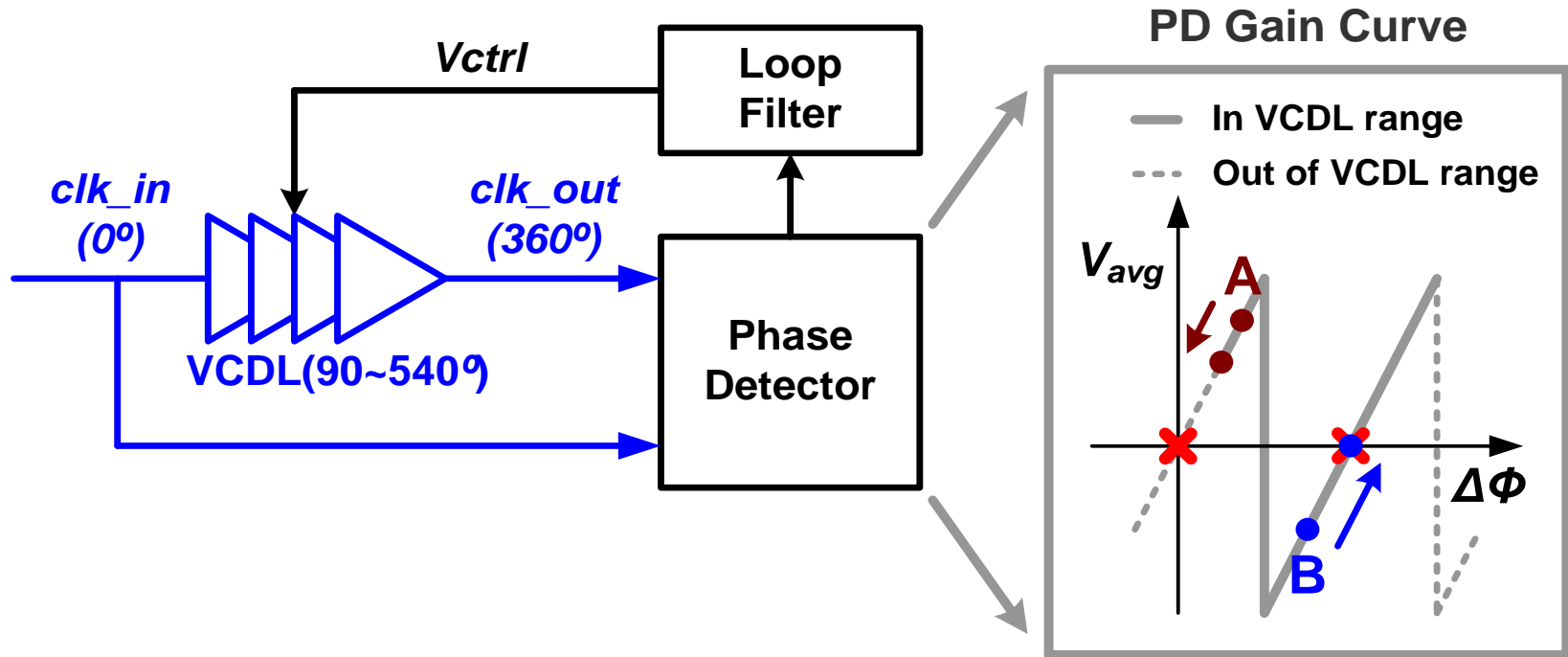


Stuck Locking in DLL : Type-I



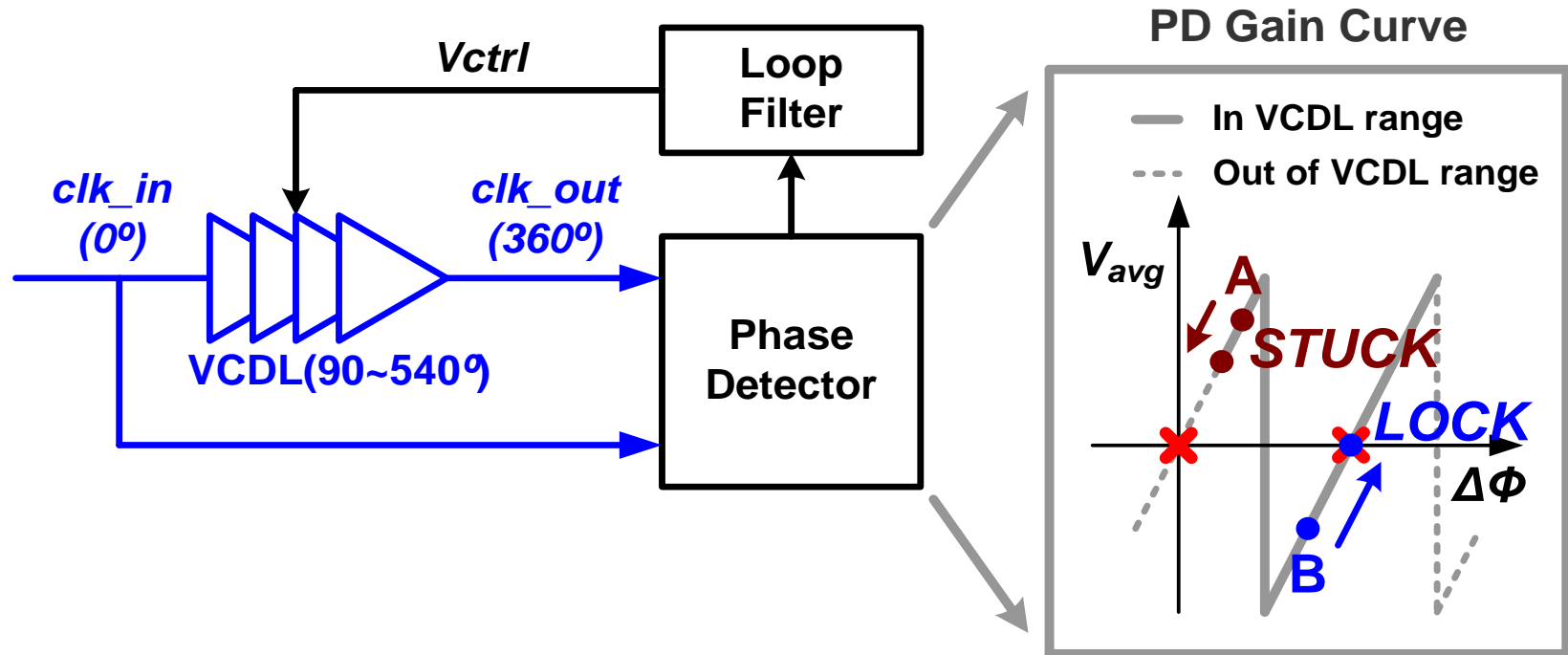
- Desired phase shift by the VCDL: 360°

Stuck Locking in DLL : Type-I



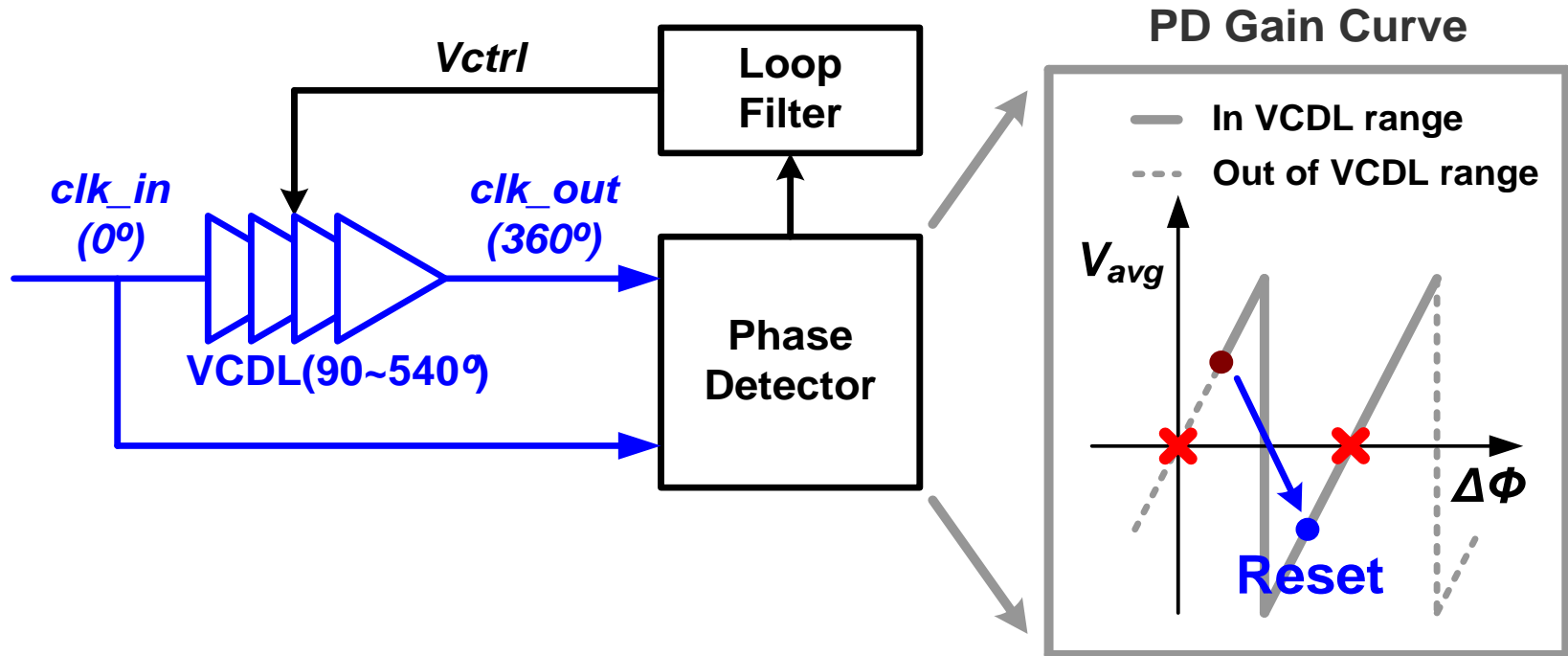
- Desired phase shift by the VCDL: 360°
 - Initial condition (A, B)
 - I.C @ A : PD drives the locking point beyond the VCDL range

Stuck Locking in DLL : Type-I



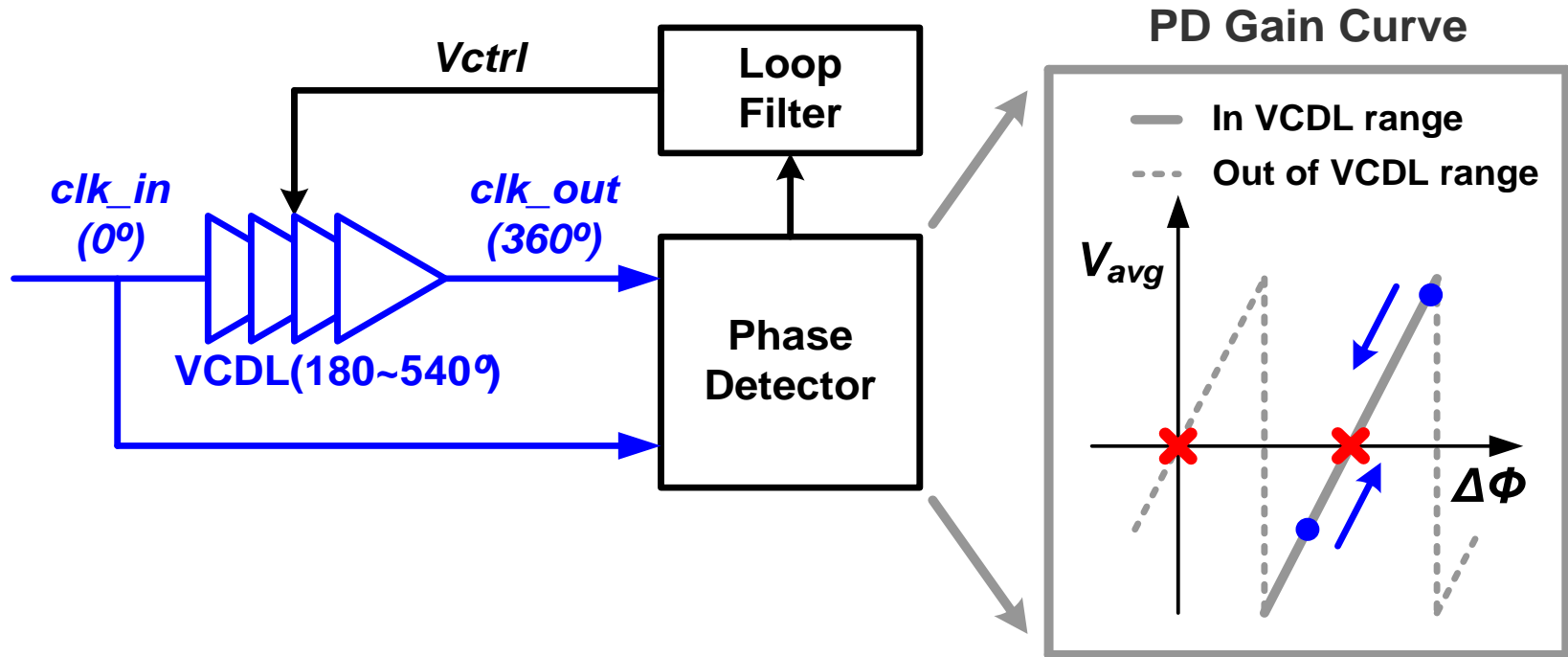
- Initial condition causes **STUCK**
 - To avoid the stuck,

Avoiding Stuck in Type-I DLL



- Initial condition causes **STUCK**
 - To avoid the stuck,
 - Set initial condition by **RESET**

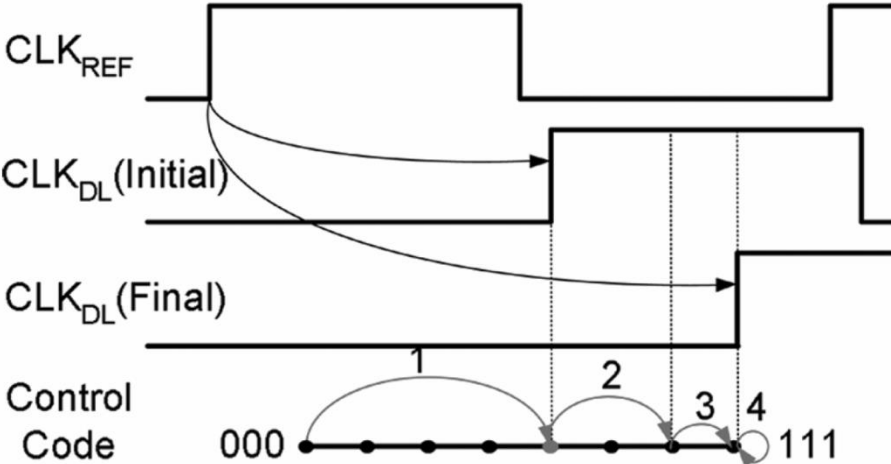
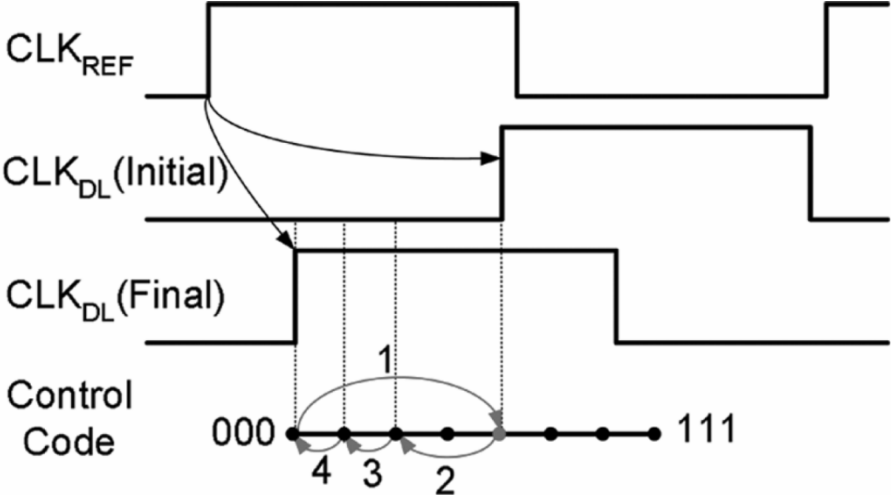
Avoiding Stuck in Type-I DLL



- Initial condition causes **STUCK**
 - To avoid the stuck,
 - Set initial condition by RESET
 - Limit the VCDL range

Stuck Detection

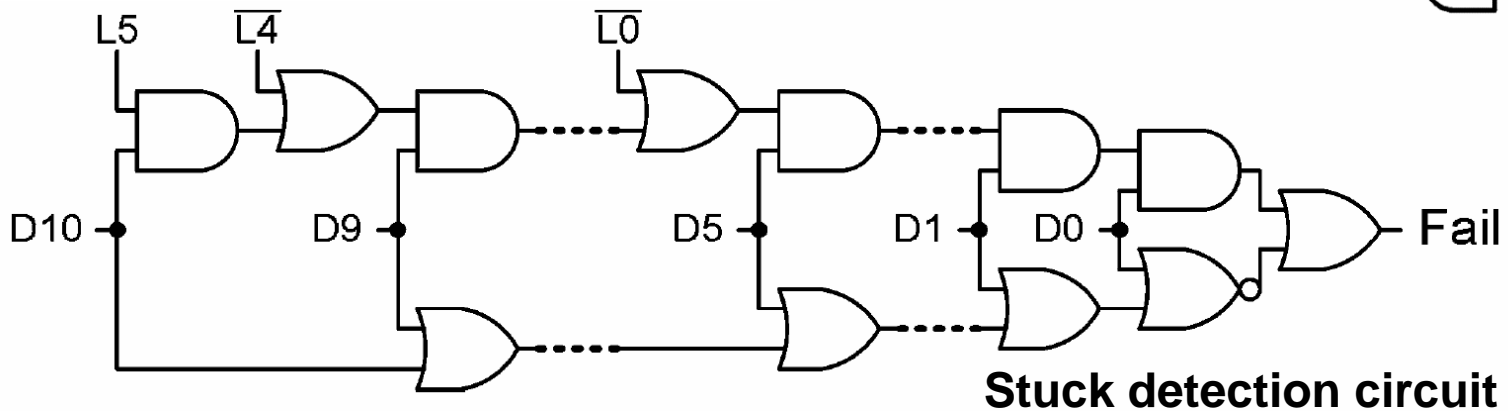
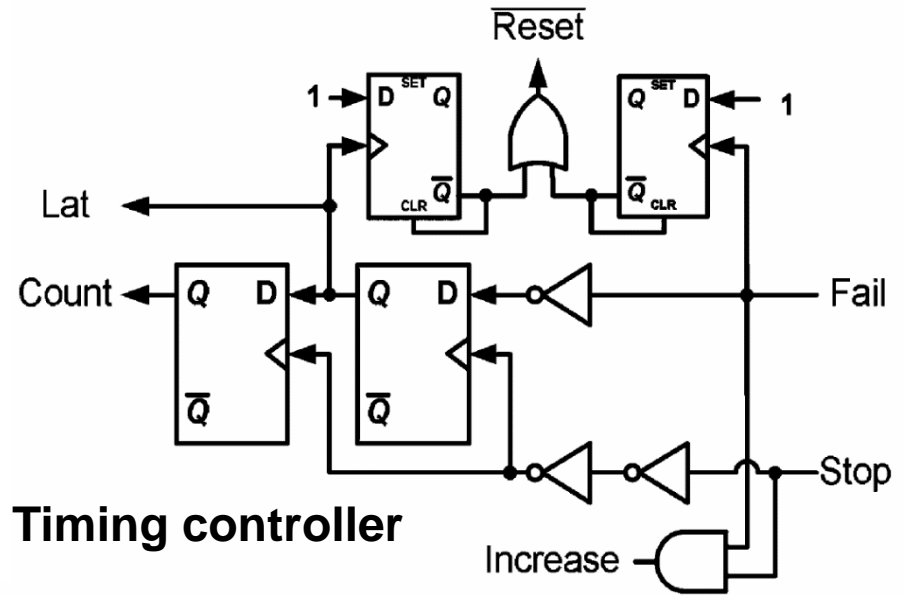
- When a DLL is stuck, delay codes are all ones or zeros



[Yang, JSSC'07]

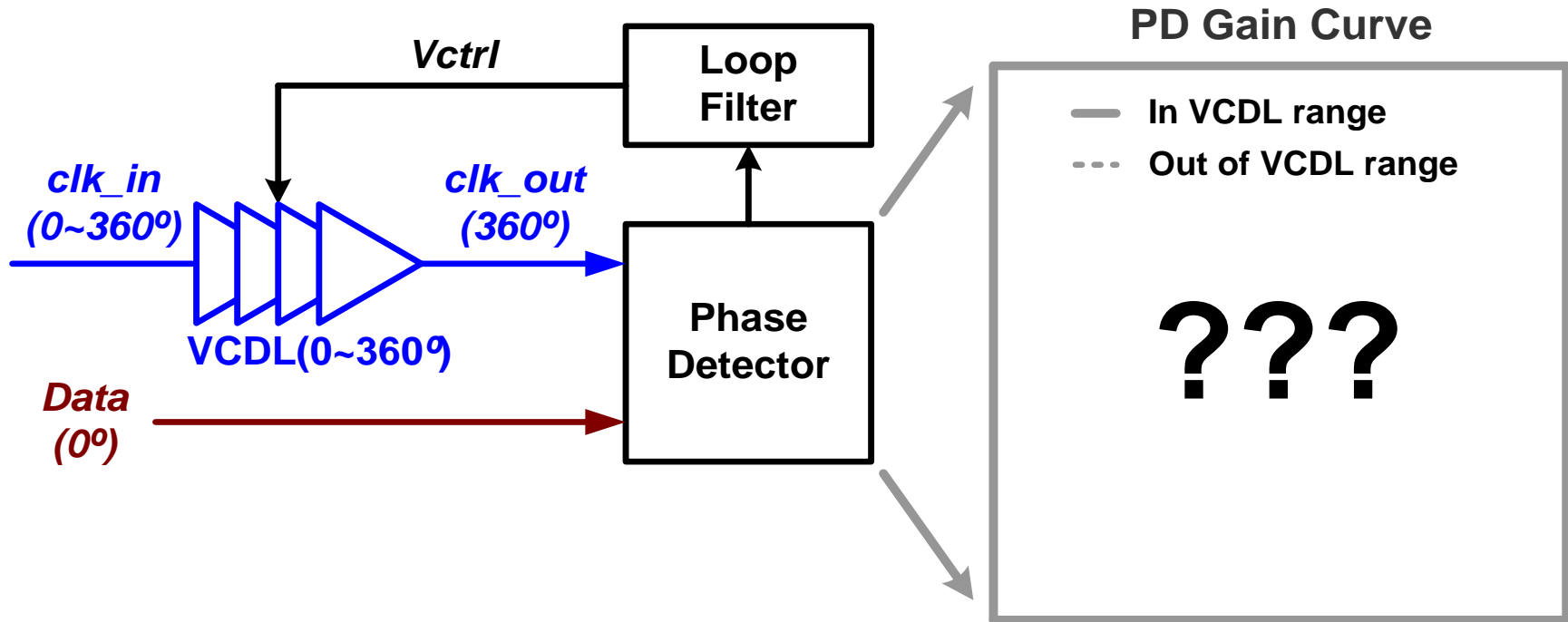
Stuck Detection

- All-ones and all-zeros detector
- Timing controller generates required control and reset signals



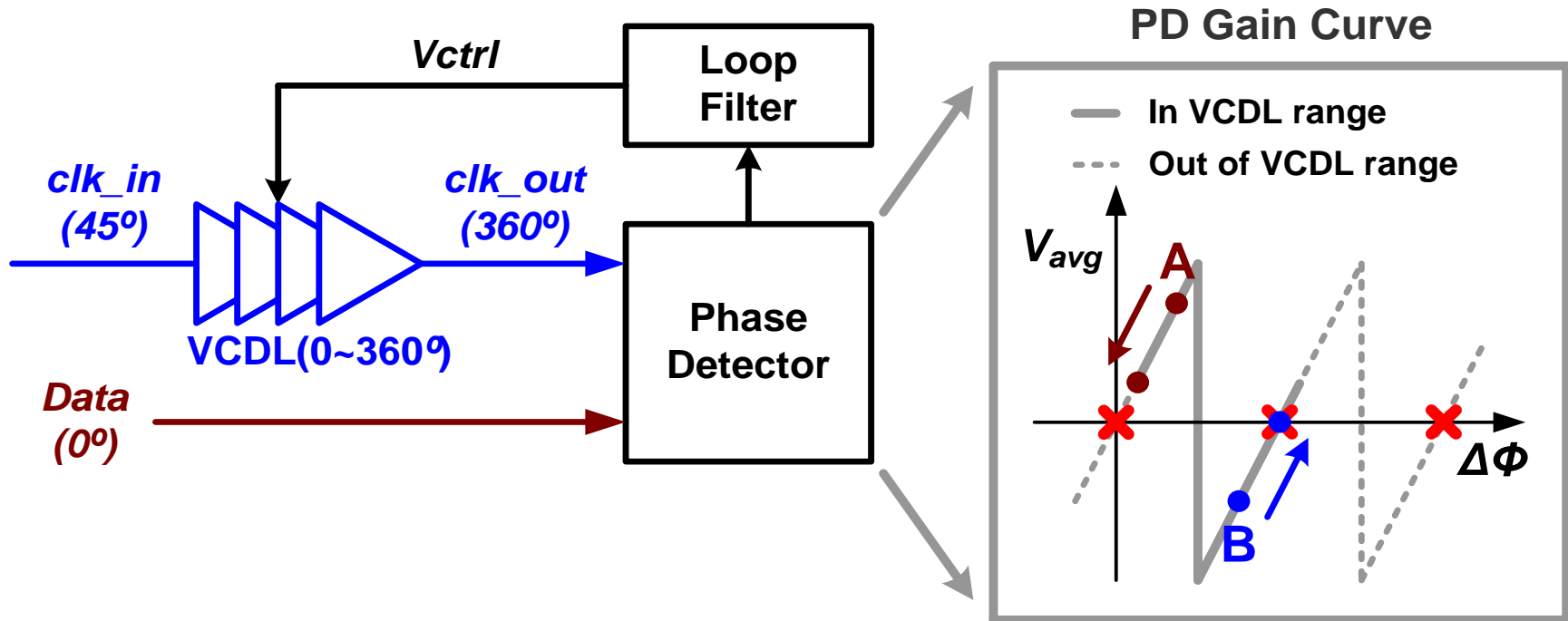
[Yang, JSSC'07]

Stuck Locking in DLL : Type-II



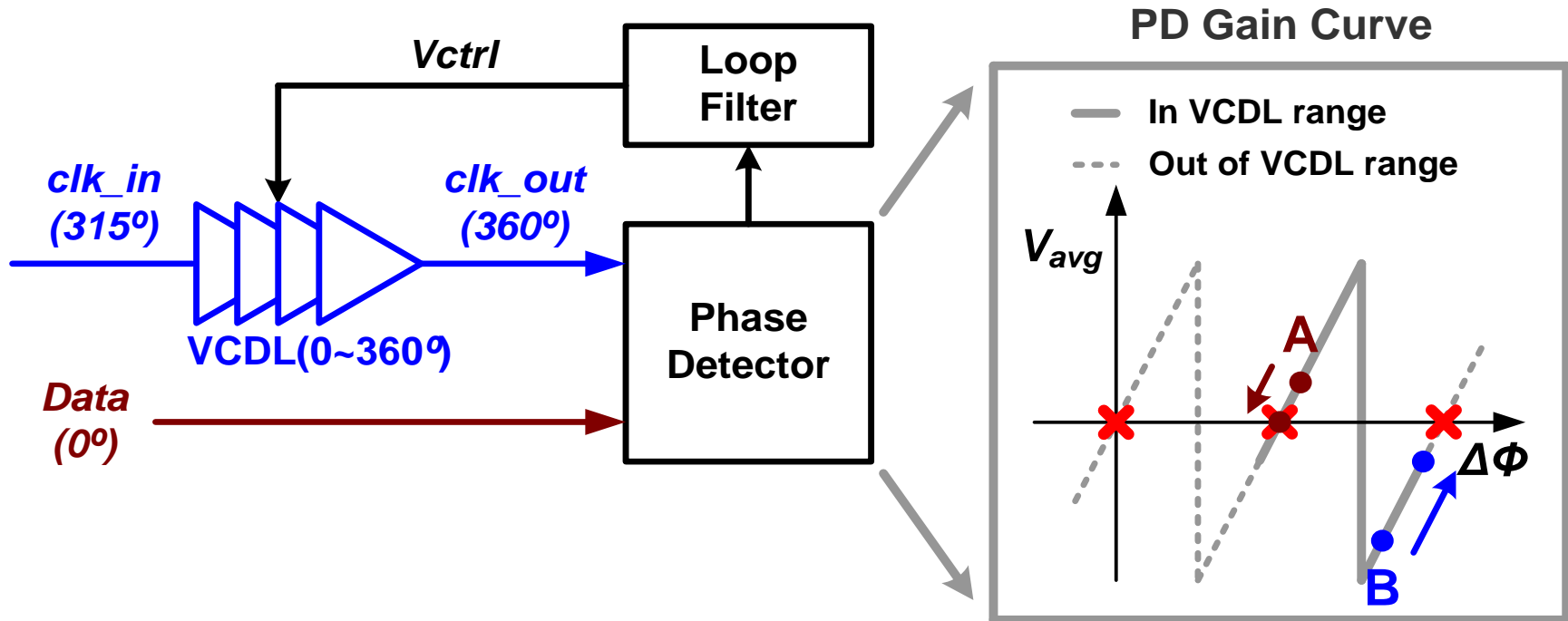
- Desired phase shift by the VCDL: ???
 - Arbitrary phase difference between clock and data

Stuck Locking in DLL : Type-II



- Case example 1: 45° skew between clk & $data$
 - **I.C A** results in **STUCK**

Stuck Locking in DLL : Type-II

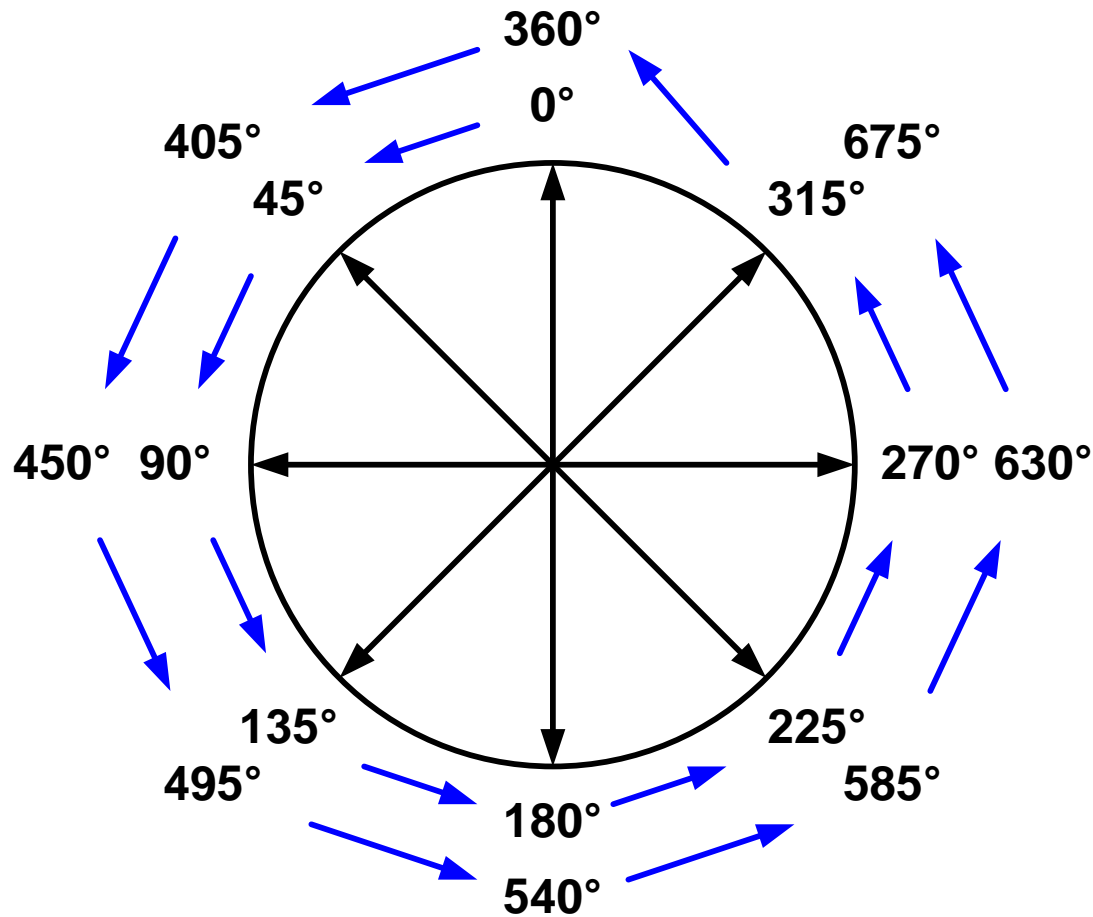


- Case example 2: 315° skew between clk&data
 - I.C B results in STUCK
 - Setting initial condition cannot solve the problem

Obviously, VCDL range cannot be limited also

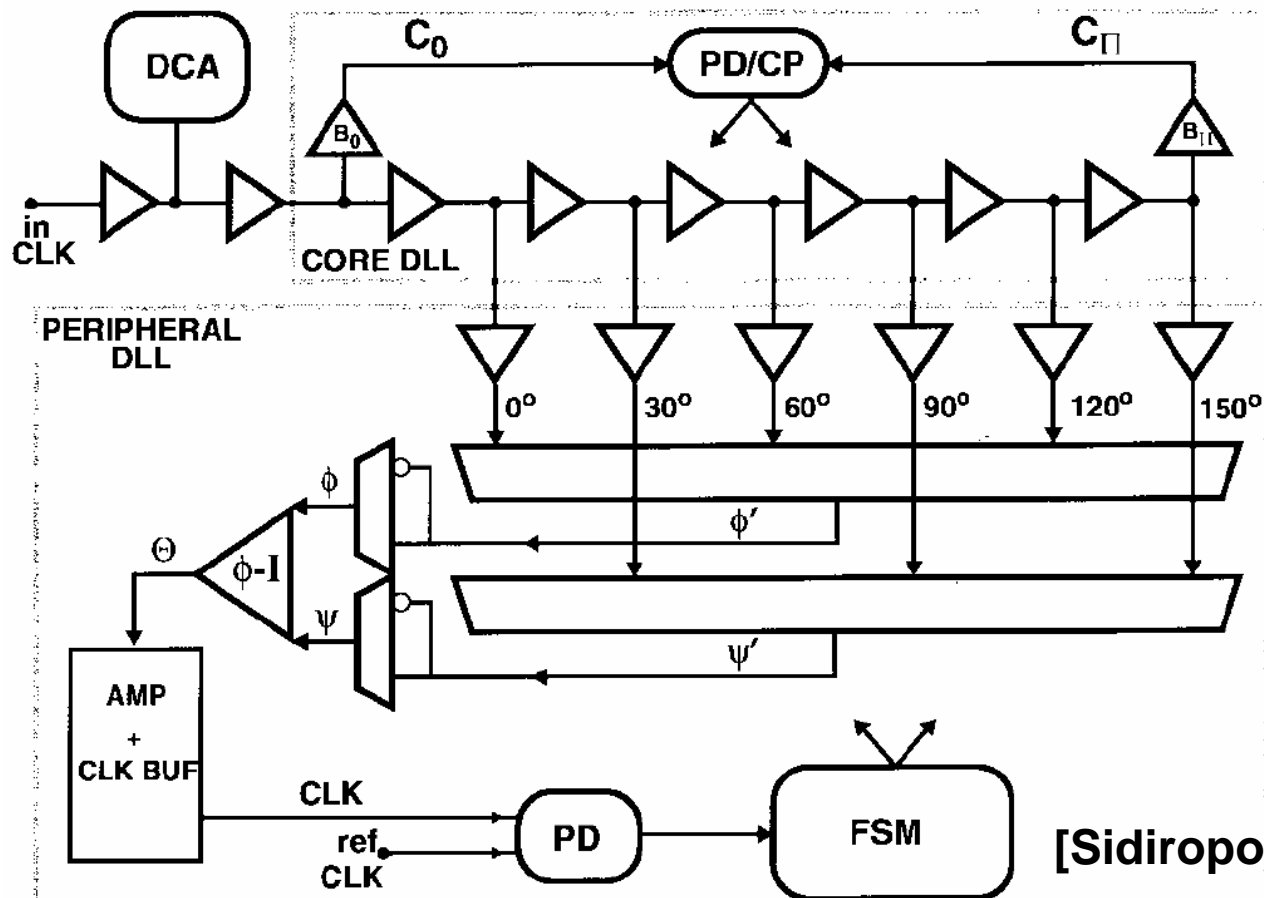
Stuck-Free DLL

- Infinite range DLL using phase-interpolator (PI)



Stuck-Free DLL

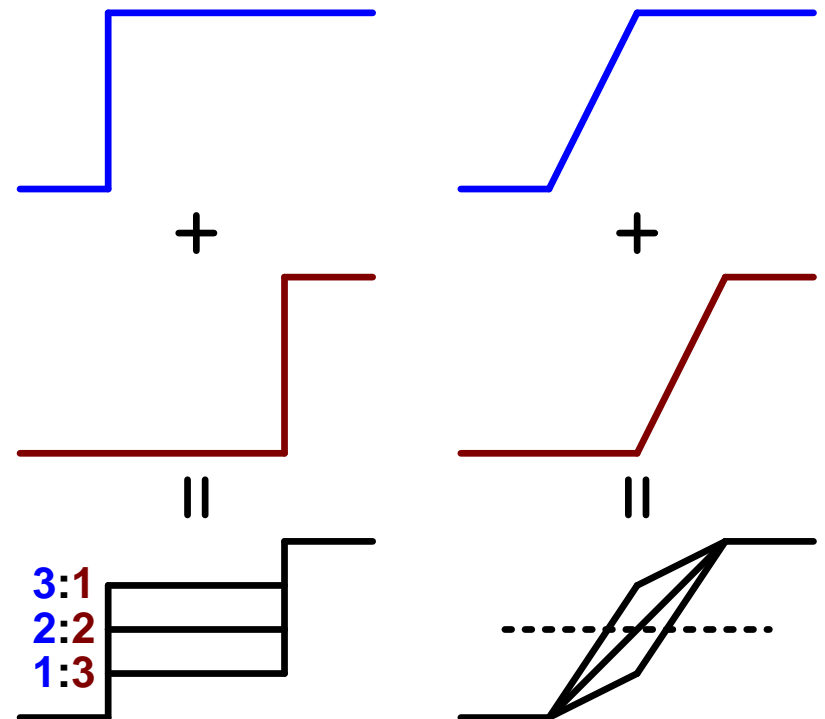
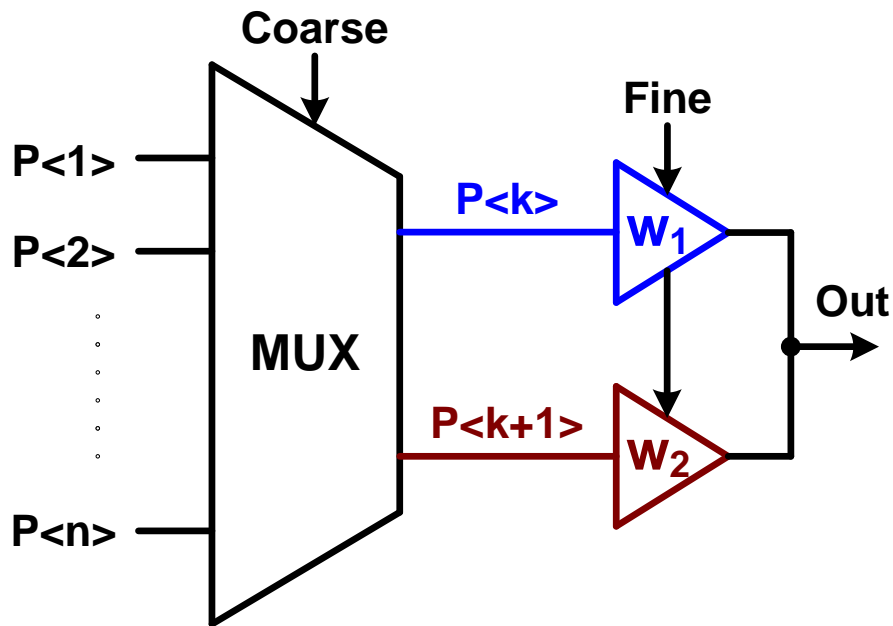
- Dual-loop DLL: Type-I DLL (multi-phase) + type-II DLL (PI)



[Sidiropoulos, JSSC'97]

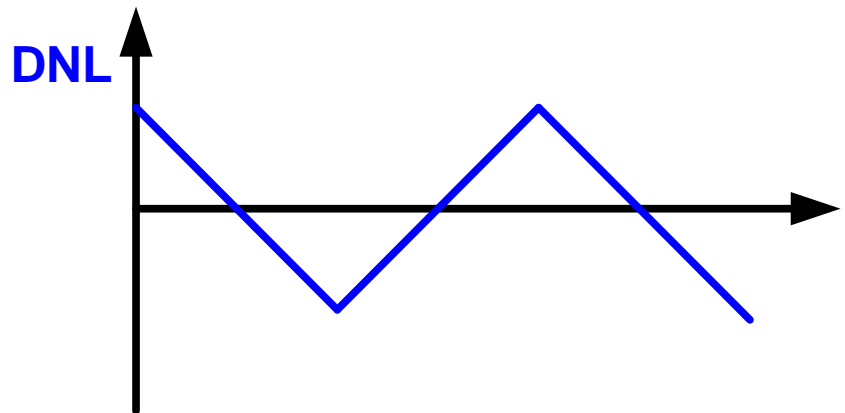
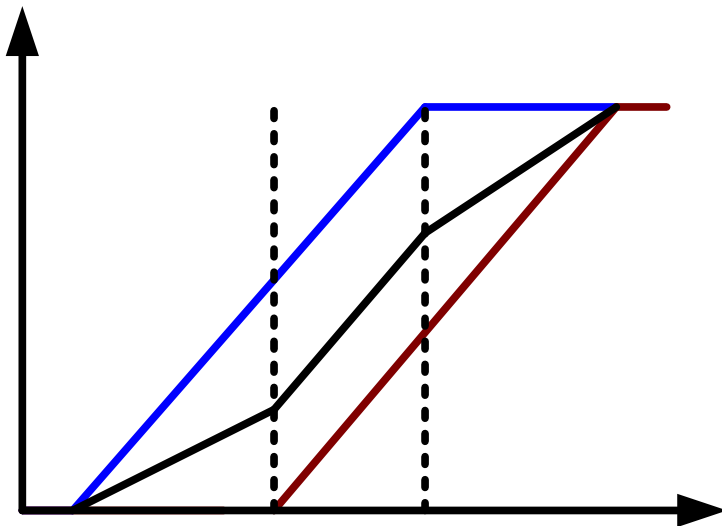
Phase Interpolator

- Voltage interpolation for phase interpolation
- Gradual clock edge is required: noise, speed, and power issues



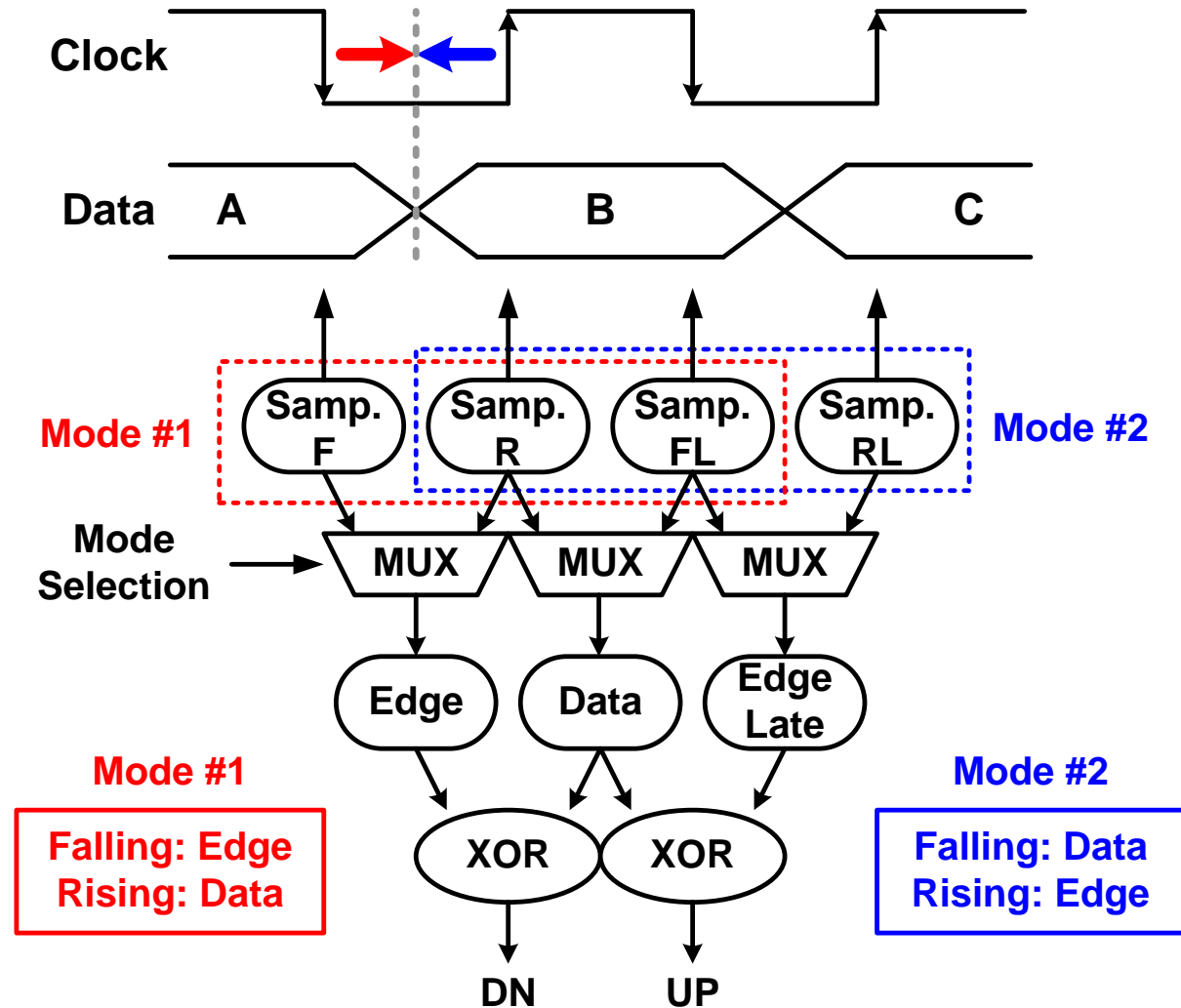
Phase Interpolator

- Linear combination of two clocks does not mean linear combination of the two phases
- Because interpolated phase is a function of slew rate
- Non-ideal INL, DNL issue



Stuck Locking Escape

- Sample-swapping technique
- Equivalent to 0.5UI phase shift of PD gain curve



Delay-Locked Loops

5.4 Phase Detector

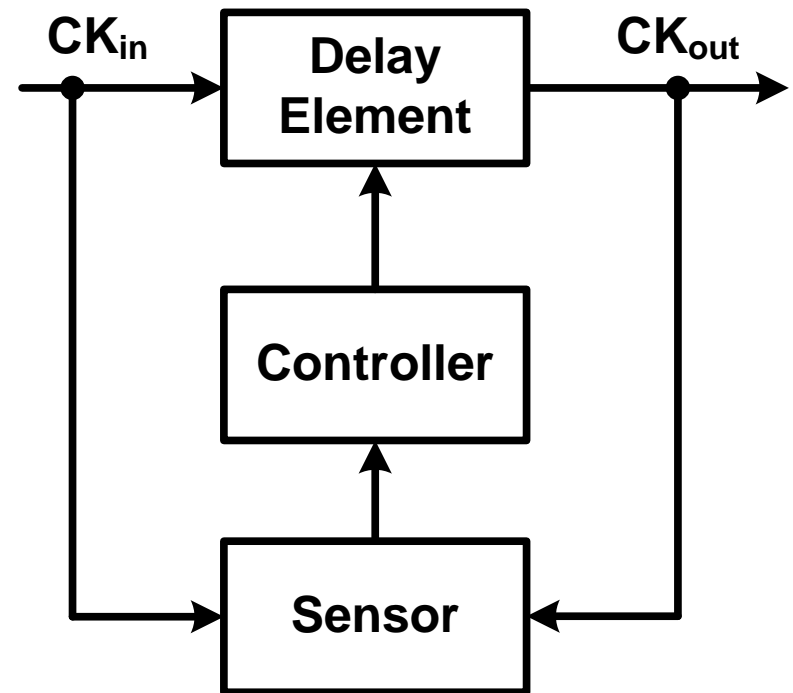
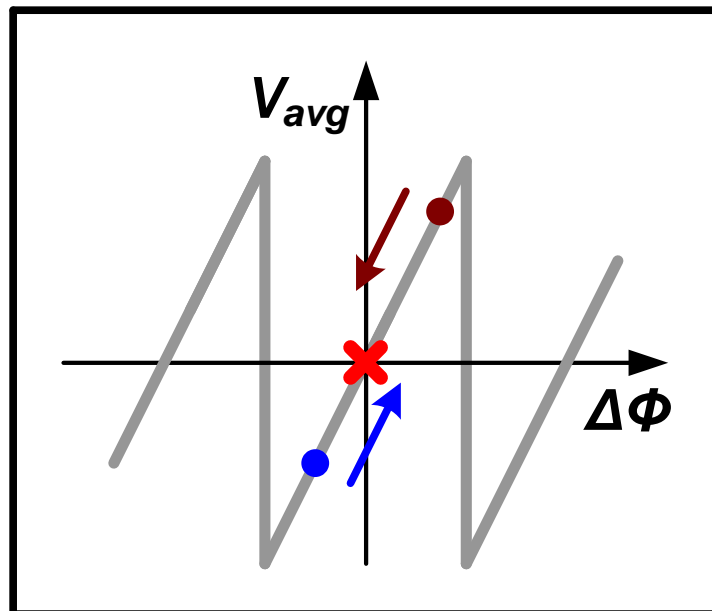
Deog-Kyoon Jeong
dkjeong@snu.ac.kr

October 19, 2020

Review: Phase Detector in DLL

- Phase detector (PD) for sensing the phase difference

PD Gain Curve

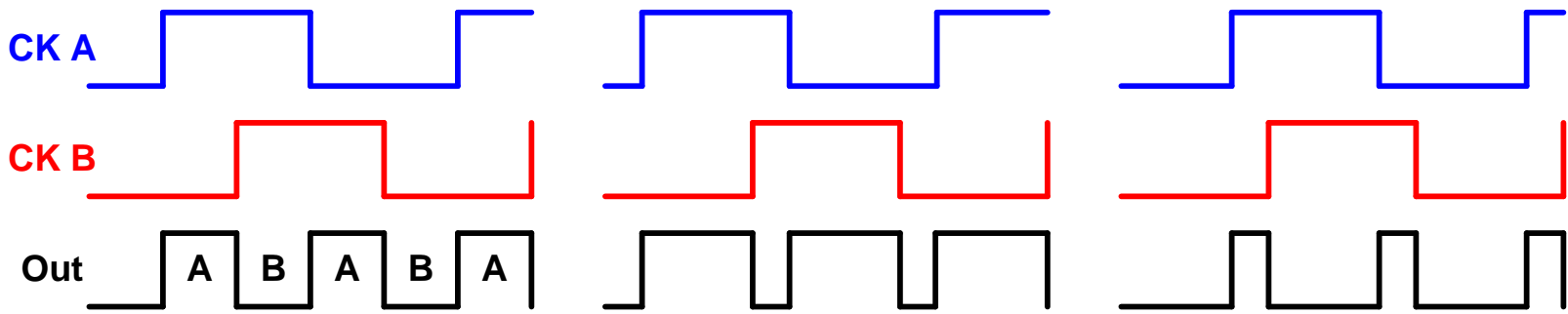
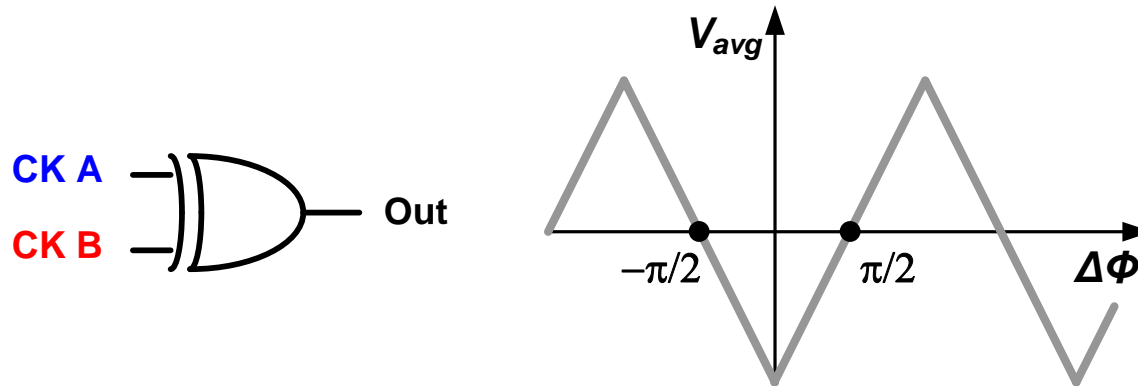


Classification of Phase Detector

- **Phase Frequency Detector (PFD)**
- **Phase Detector (PD)**
- **Static PD**
- **Dynamic PD**
 - **Fast operation speed and potential saving in power**

XOR PD

- Phase only detector
- Locking point at $\pi/2$, not zero



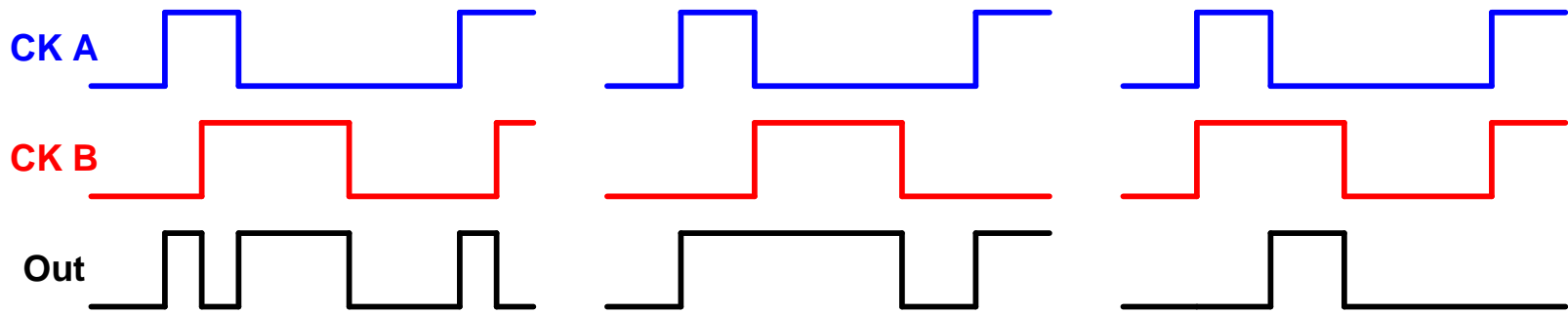
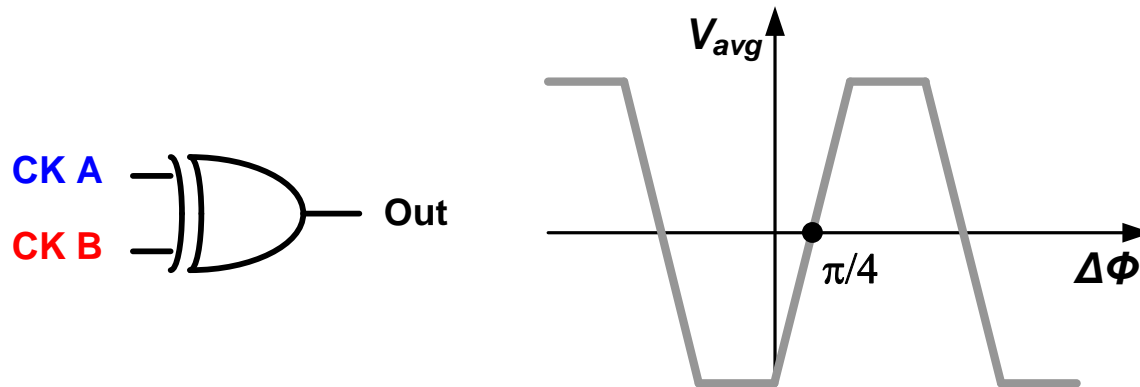
Phase-locked

A lead
4

B lead

XOR PD with Non-Ideal Duty-Cycle

- PD gain curve shifts
- Level sensitive: Two inputs need to be symmetrical



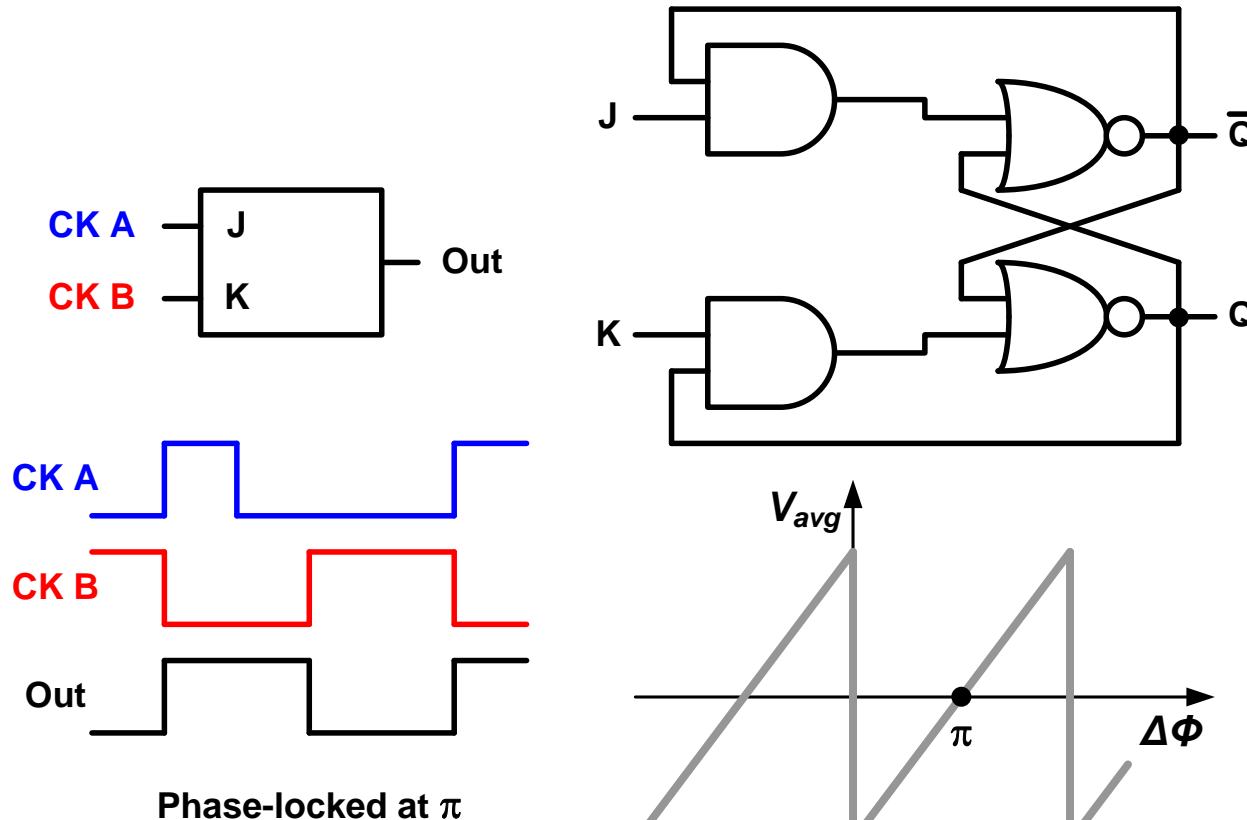
Phase-locked

A lead
5

B lead

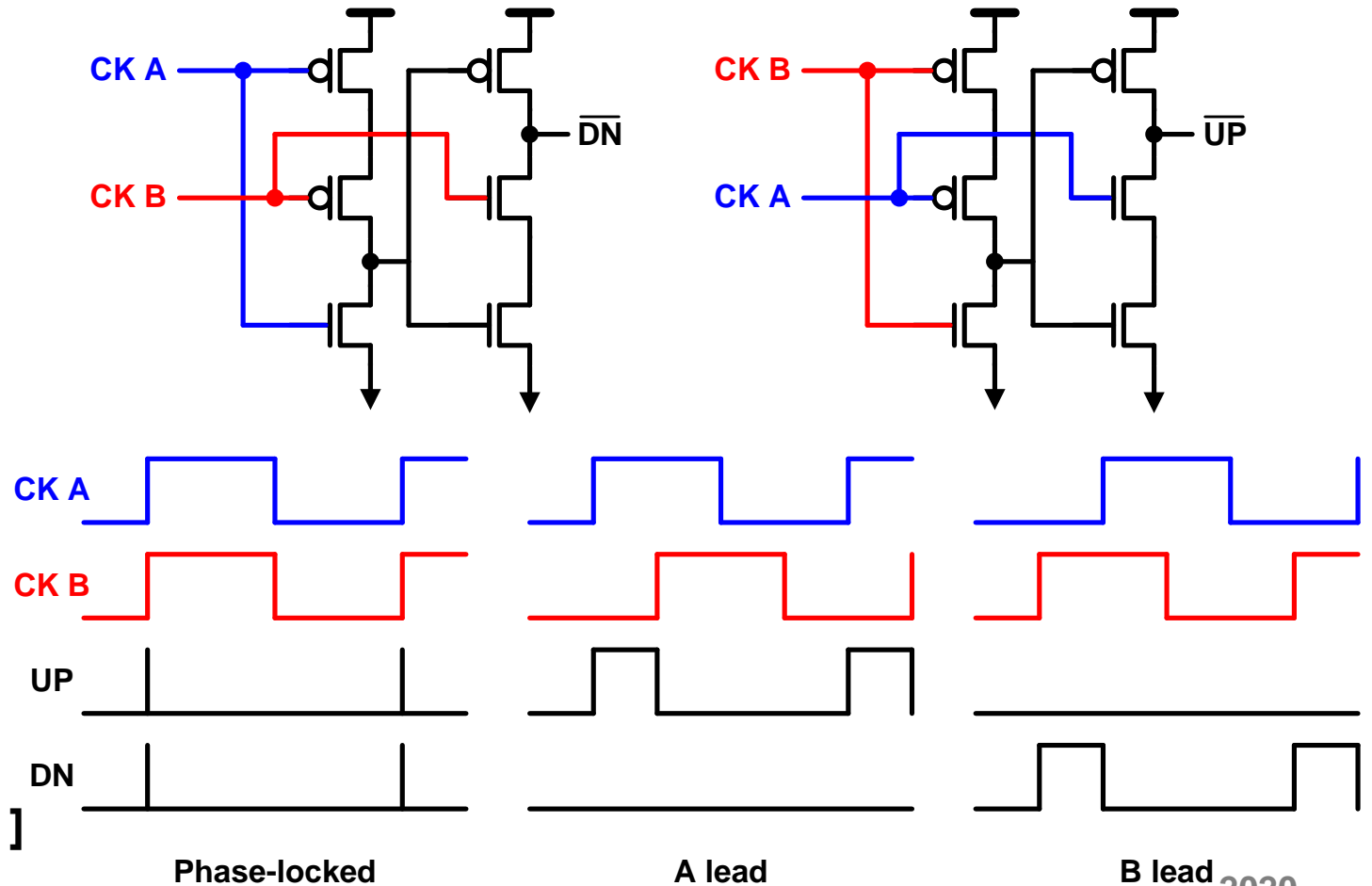
Edge Sensitive PD: JK Latch

- Positive edge at J input triggers the output to high
- Positive edge at K input triggers the output to low



Dynamic PD

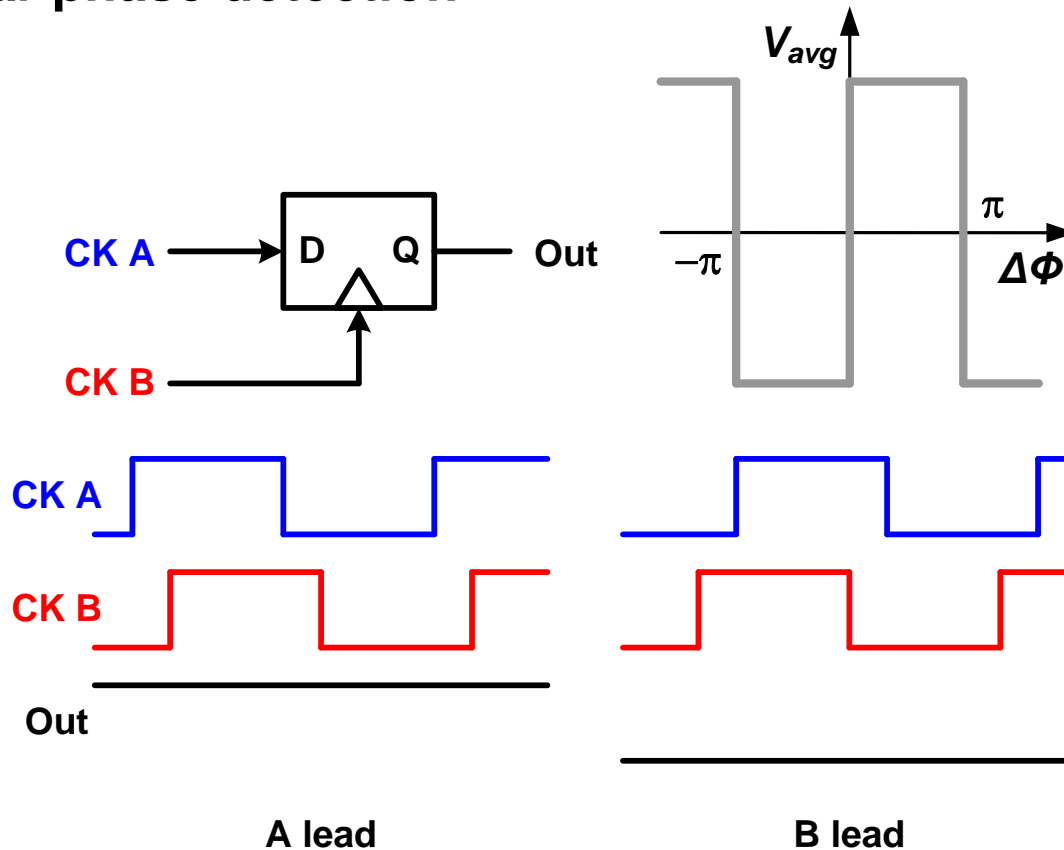
- Less phase offset and high speed



[Moon, JSSC'00]

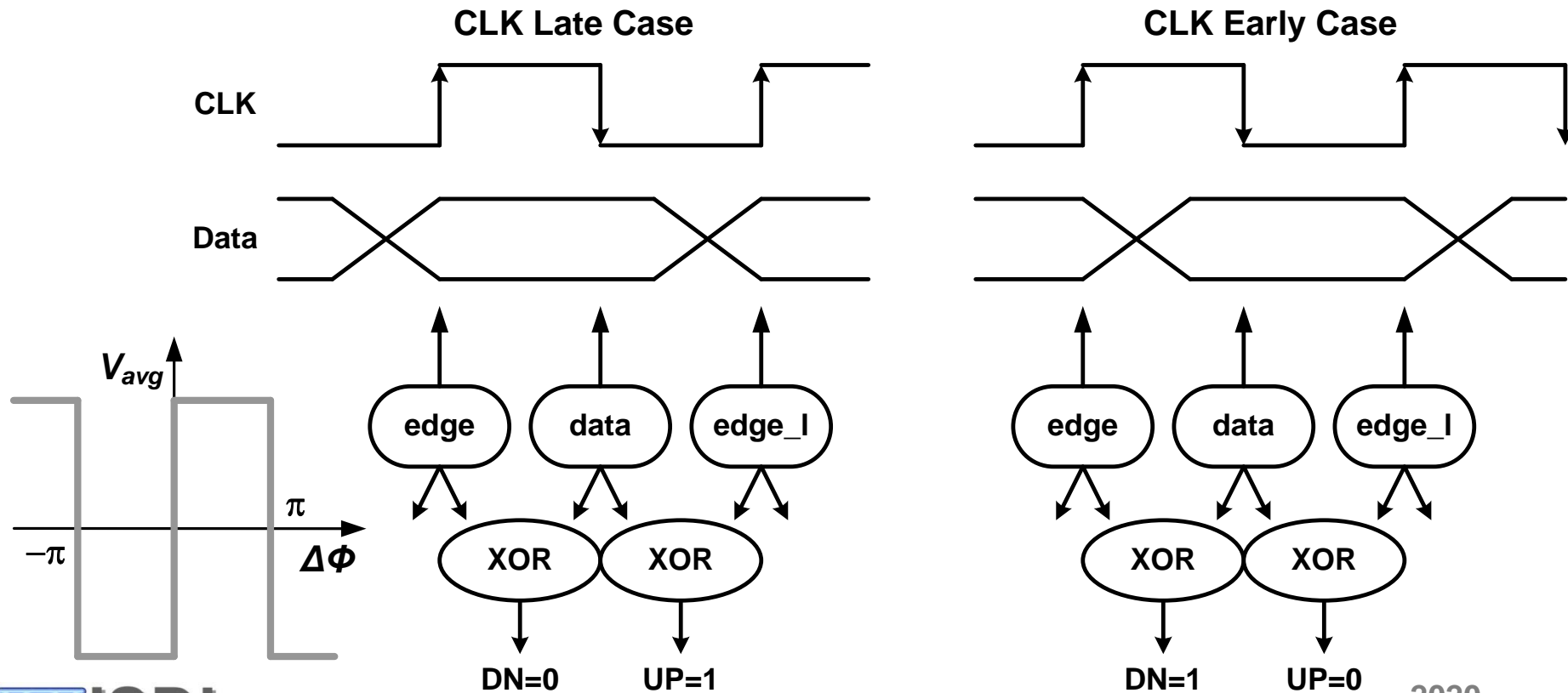
Bang-Bang PD - 1

- Sampling one clock with the other clock
- Non-linear phase detection



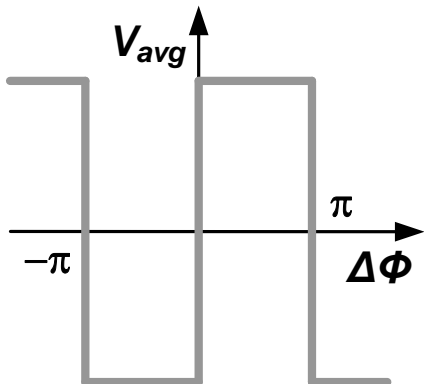
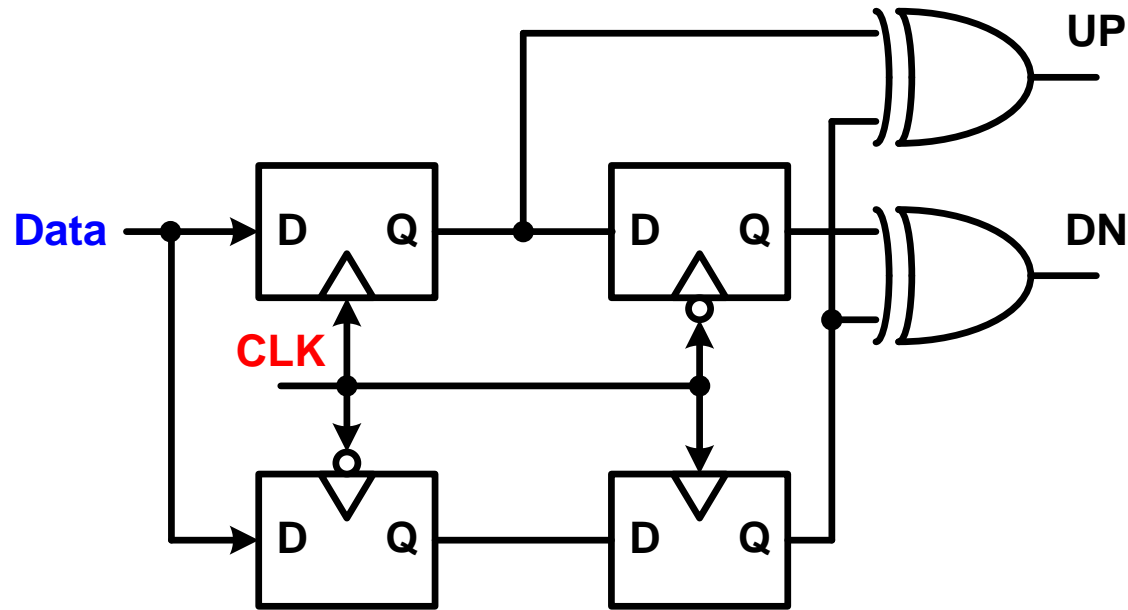
Bang-Bang PD - 2

- For a special case of type-II DLL: one of the input is not a CLK
- Alexander phase detector



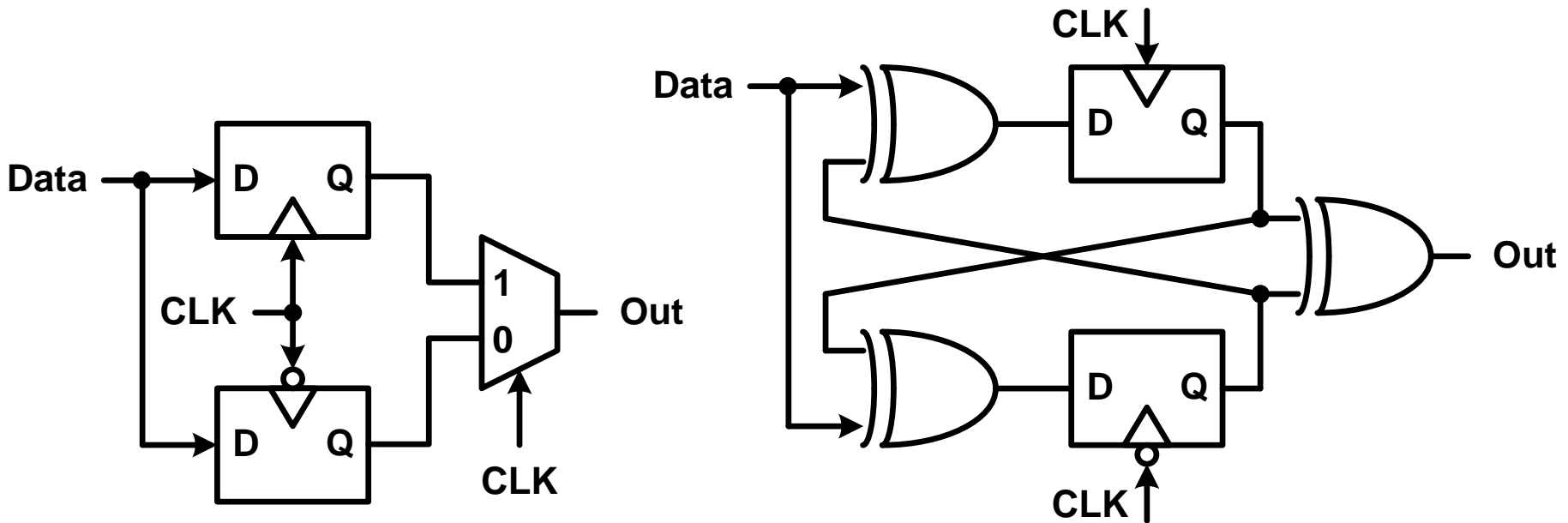
Bang-Bang PD - 2

- For a special case of type-II DLL: one of the input is not a CLK
- Alexander phase detector



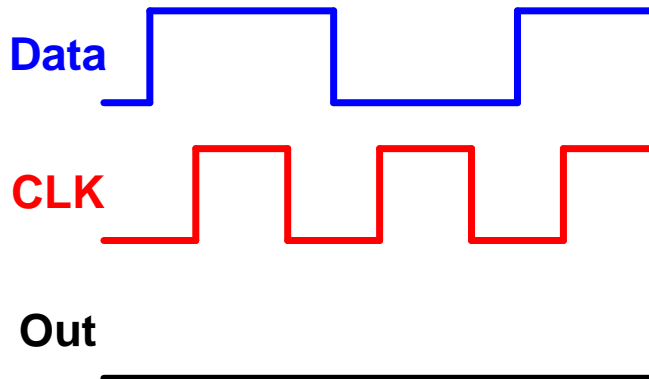
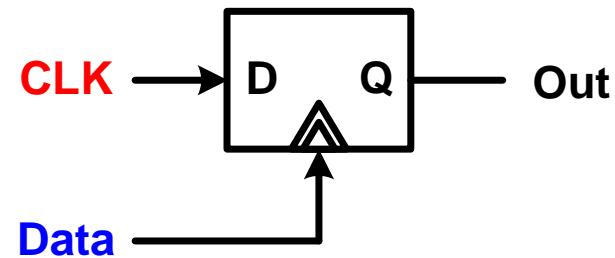
BBPD with Dual-Edge Flipflop

- Dual edge triggered flipflop

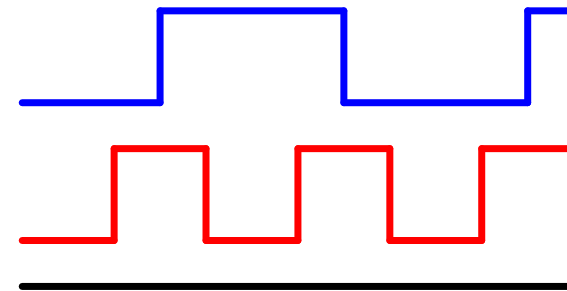


BBPD with Dual-Edge Flipflop

- Dual edge triggered flipflop
- Clock sampled by Data



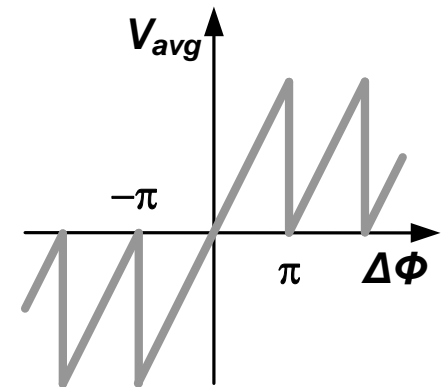
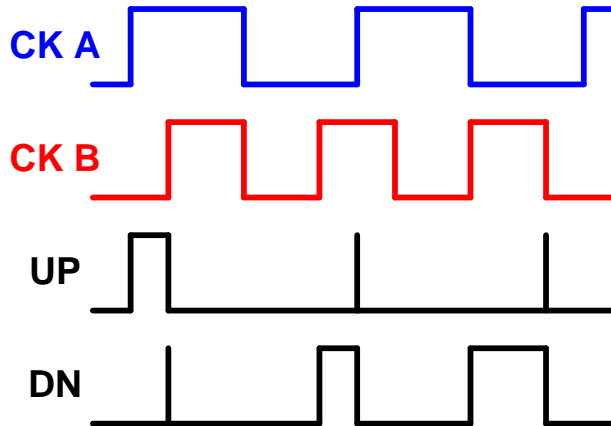
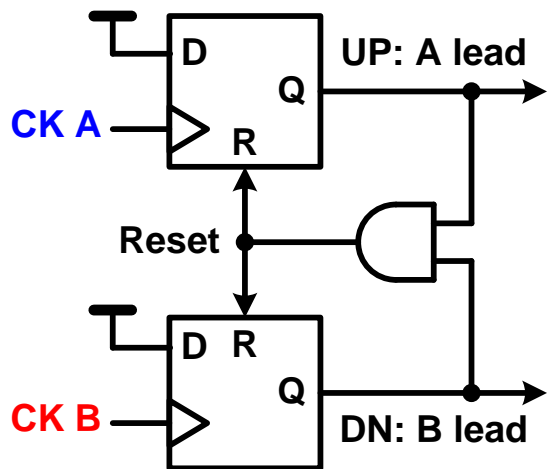
Data lead



CLK lead

Phase-Frequency Detector (PFD)

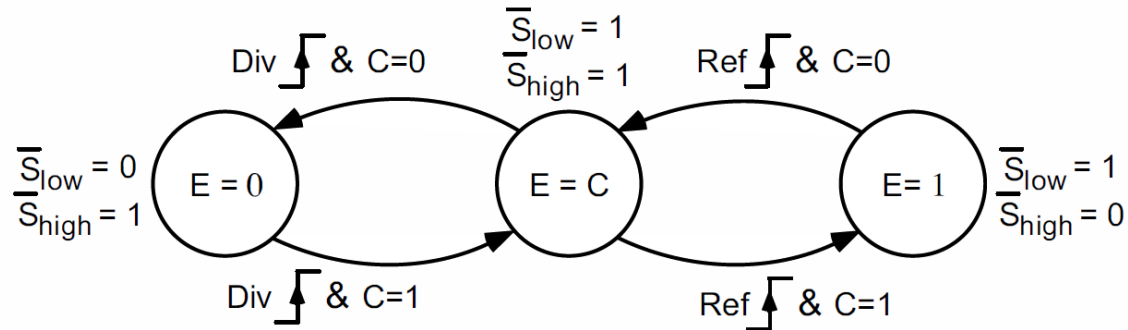
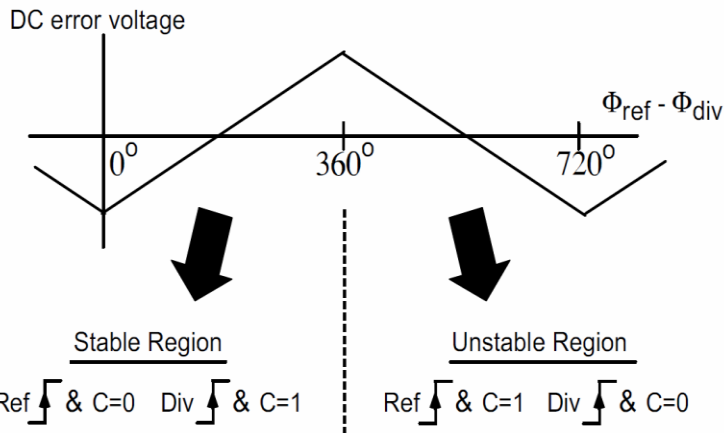
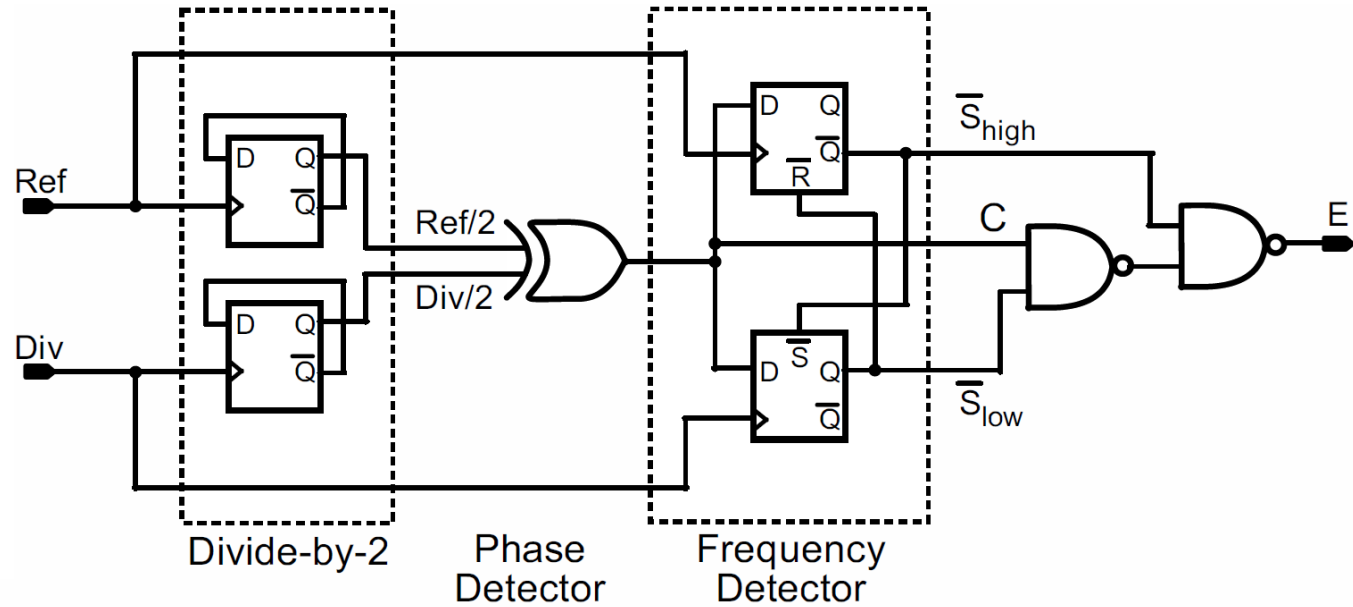
- Flip-flops to remember the edges PFD has been comparing
- Detects cycle slipping
- Dead zone and blind zone issues



XOR-PFD

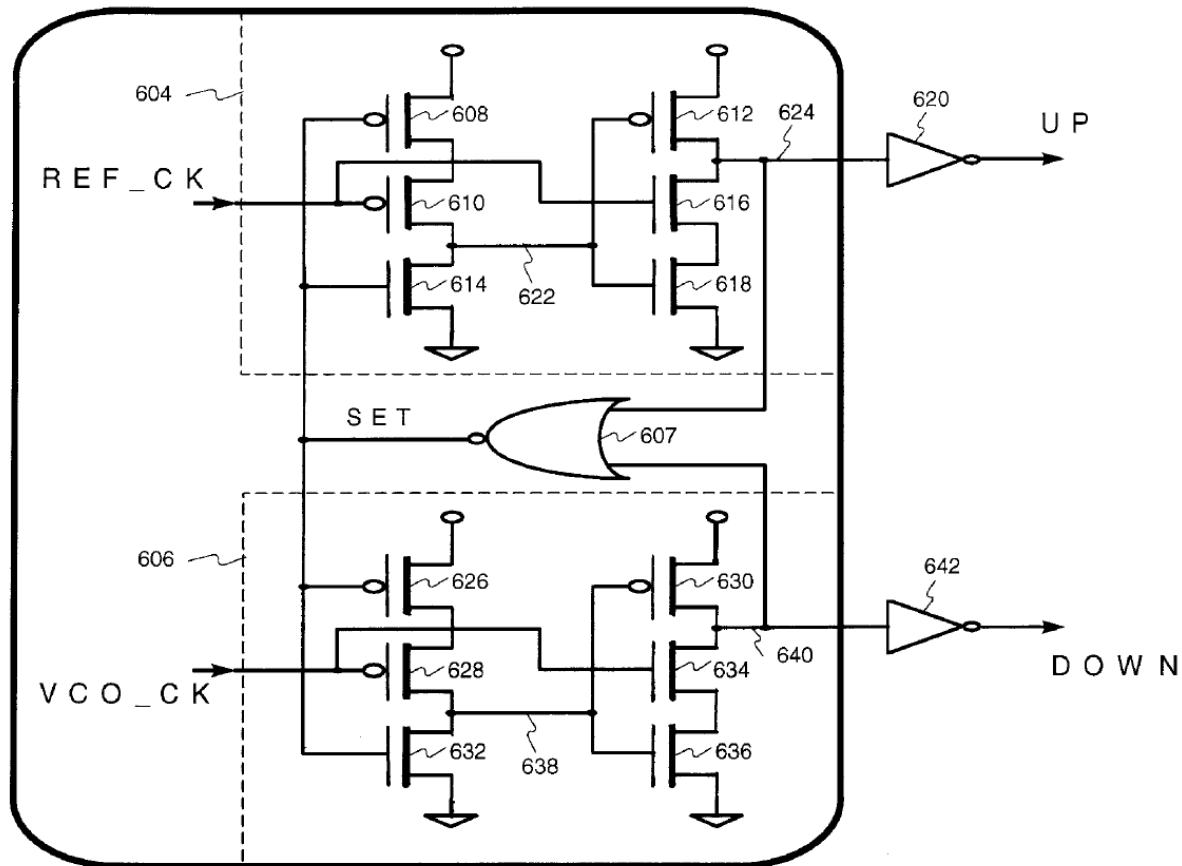
- Static PFD

[M. Perrott, Ph.D
Dissertation]



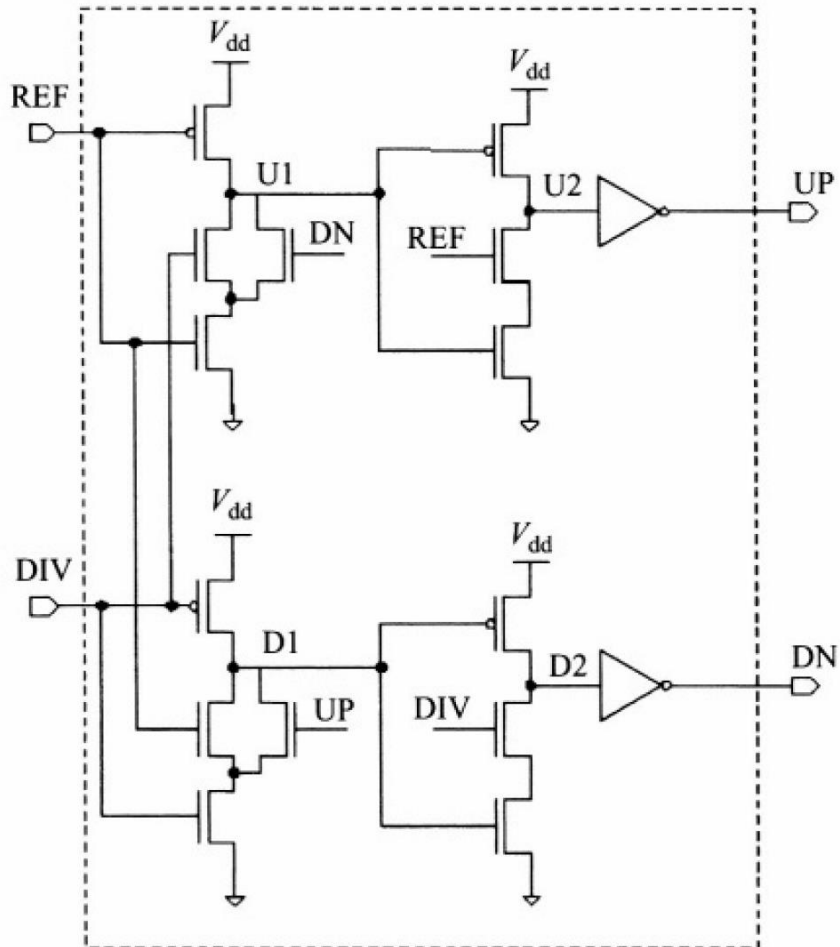
Dynamic PFD

- K. Lee, US patent 5,815,041

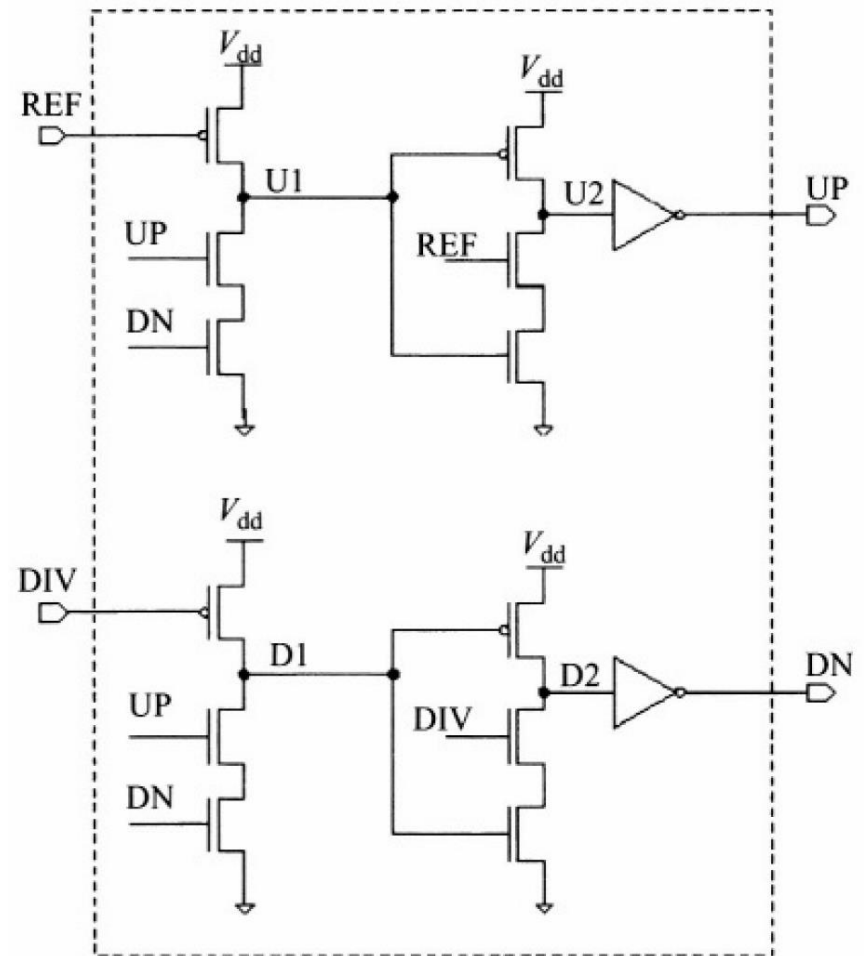


Dynamic PFD

[Kondoh, IEICE TE'95]

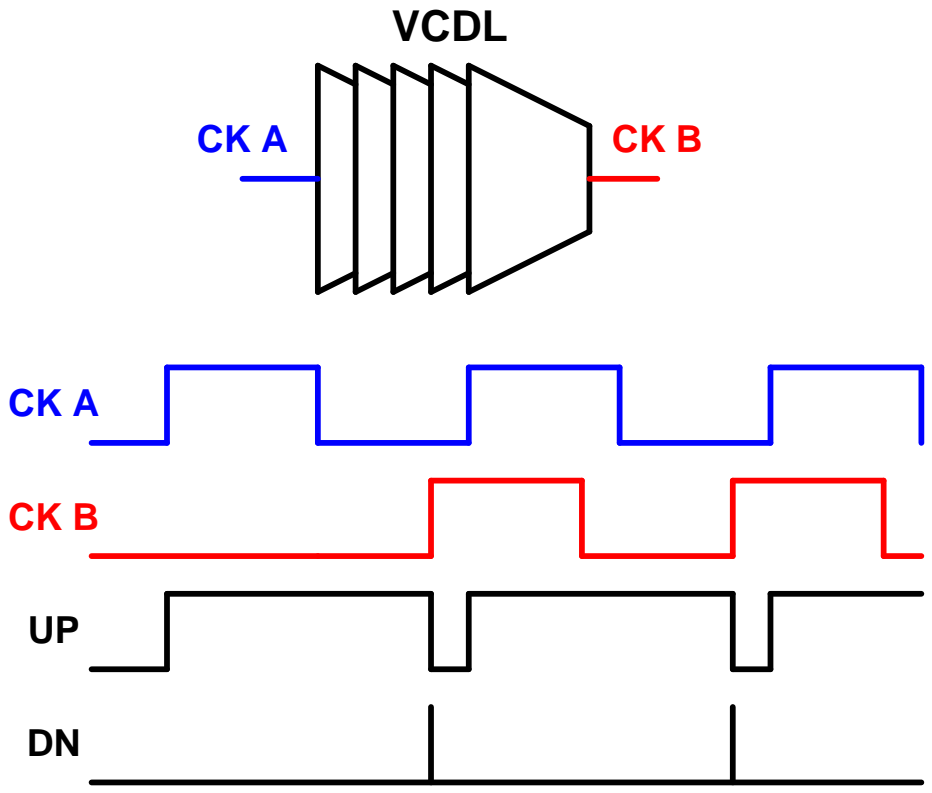
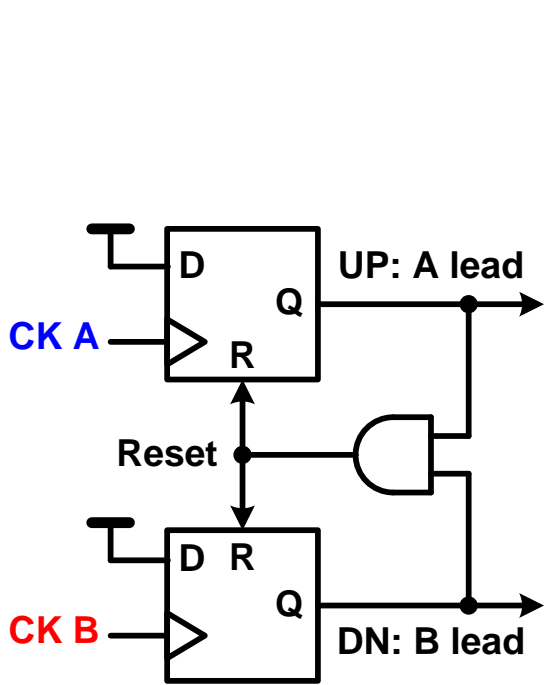


[Lee, MWCAS'99]



PFD in DLL

- Start-up issue



B lead, but PFD tells 'A lead'

Delay-Locked Loops

5.5 Voltage Controlled Delay Line

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

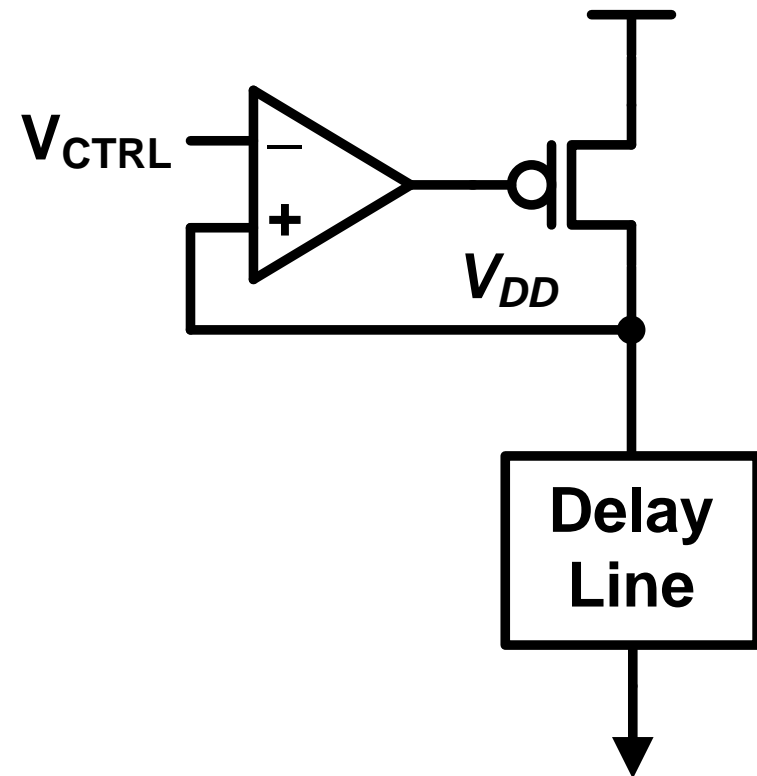
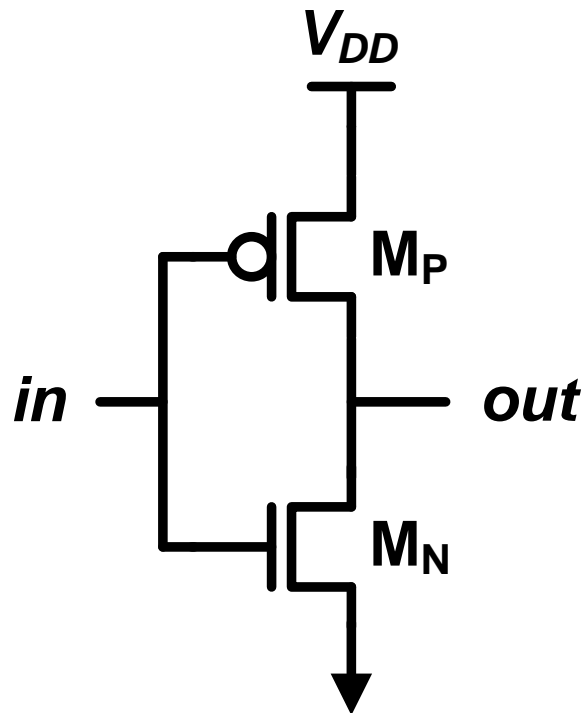
October 20, 2020

Delay Lines

- **Analog Controlled**
 - Starved Inverter
 - Load Capacitor
 - Supply Controlled inverter
 - Current-controlled fixed-swing CML with replica biasing
 - Swing-controlled fixed-current CML with replica biasing
- **Digitally Controlled**
 - CMOS NAND Lattice
 - Switched capacitor
 - Resistor-string DAC based current controlled inverter

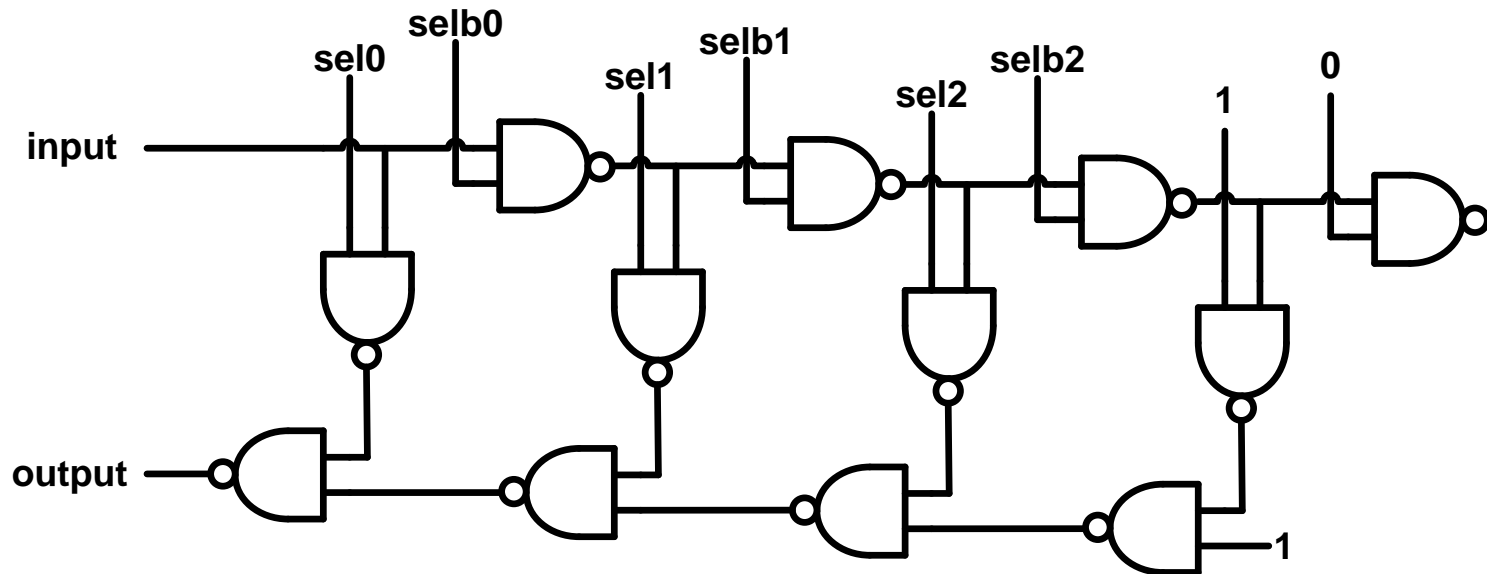
Supply Controlled Inverter

- Circuit diagram



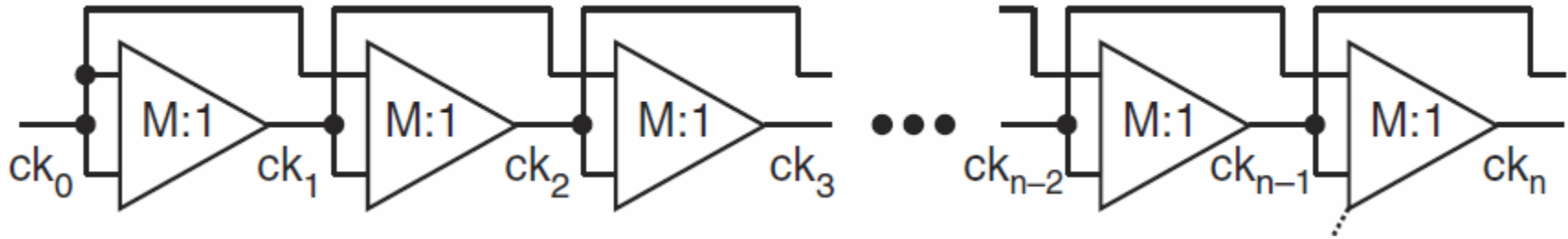
NAND Lattice Delay Line

- Circuit diagram



Fast Delay Line

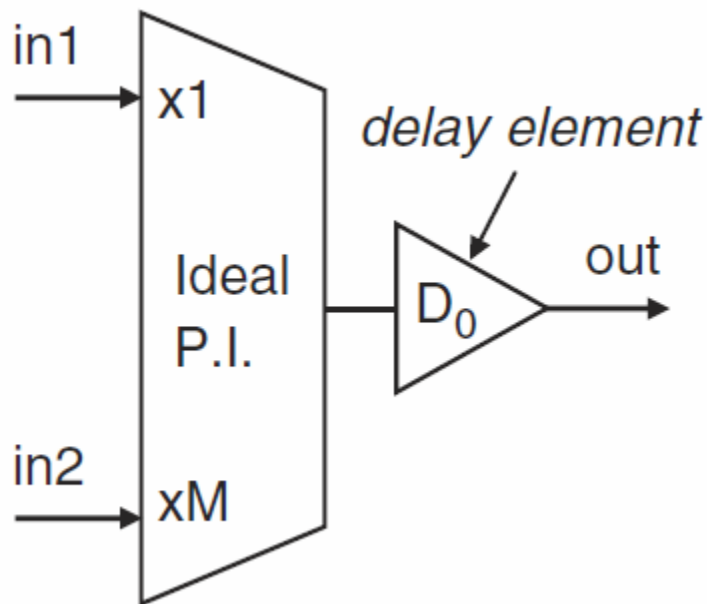
- For high frequency MDLL and finely-spaced multi-phase generation DLL, a delay cell with a short delay is required
- Dual-input interpolating delay cell



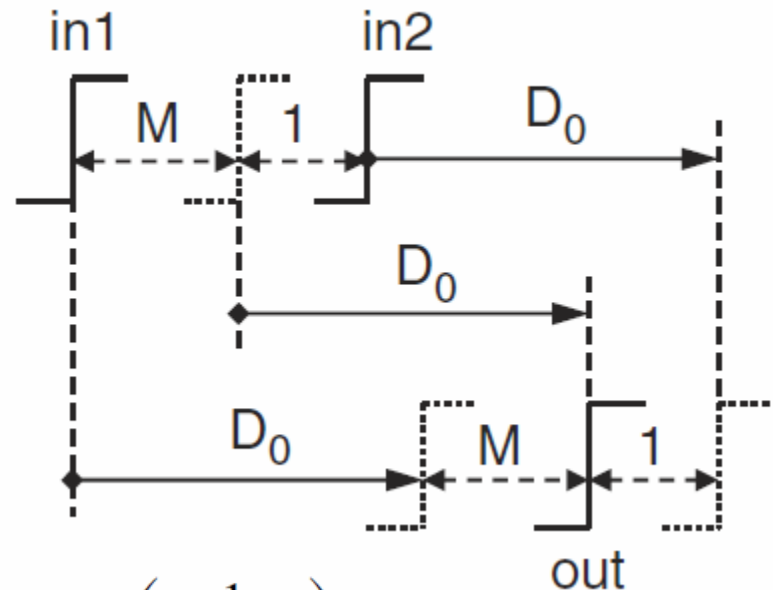
[Song, EL'10]

Fast Delay Line

- DIDC can be modeled as a combination of an ideal phase interpolator and a delay element



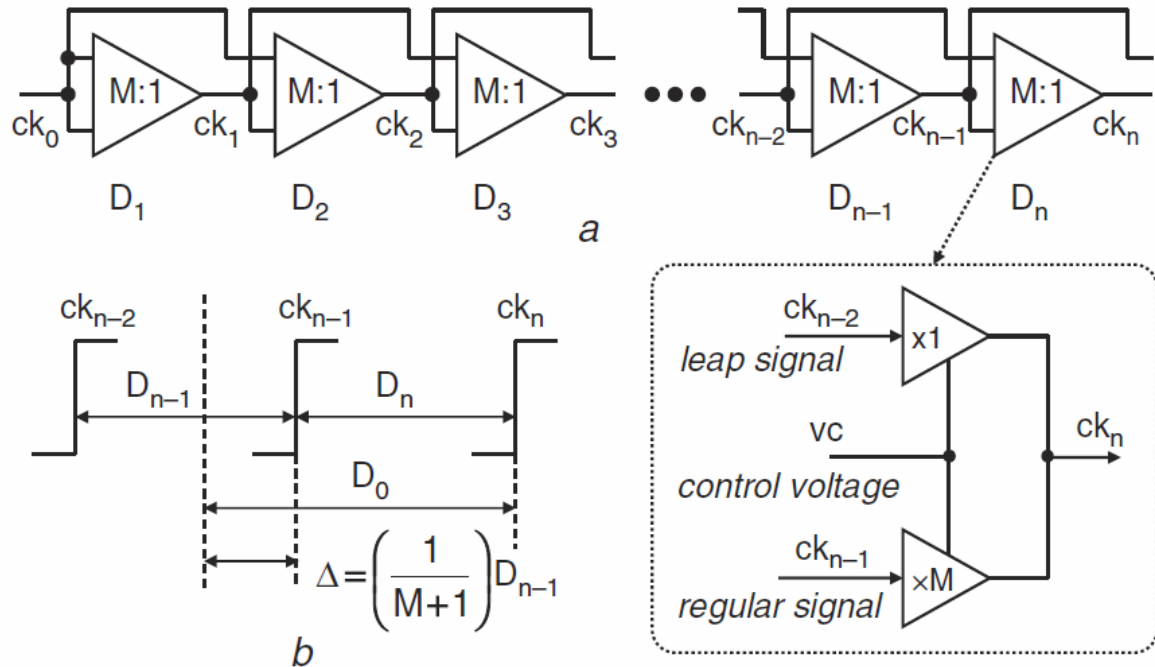
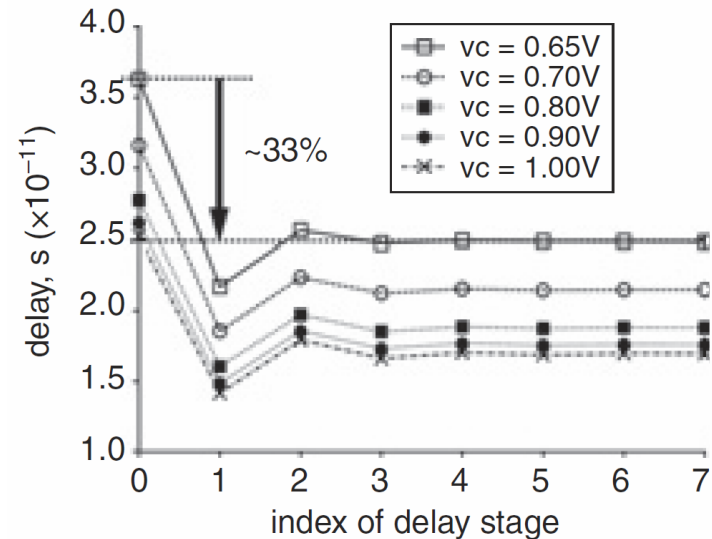
[Song, EL'10]



$$D_n = D_0 - \left(\frac{1}{M+1} \right) D_{n-1}$$

Fast Delay Line

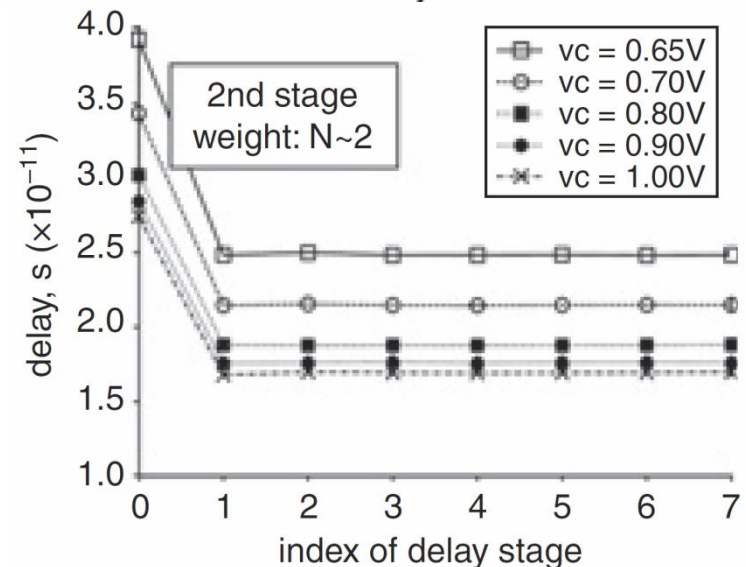
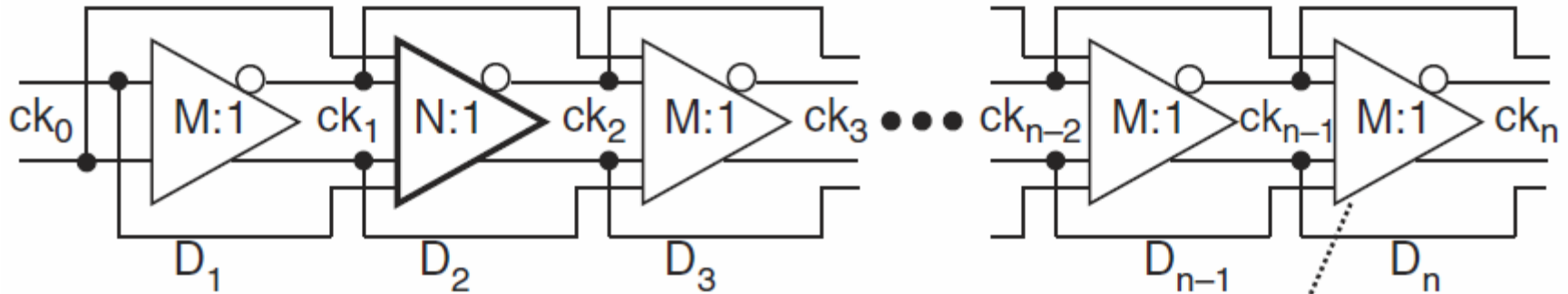
- Delay of the n-th delay cell $D_n = D_0 \left[\left(\frac{M+1}{M+2} \right) + \frac{1}{M+2} \left(\frac{-1}{M+1} \right)^{n-1} \right]$
- Not a constant delay: dummy delay cells are required



[Song, EL'10]

Fast Delay Line

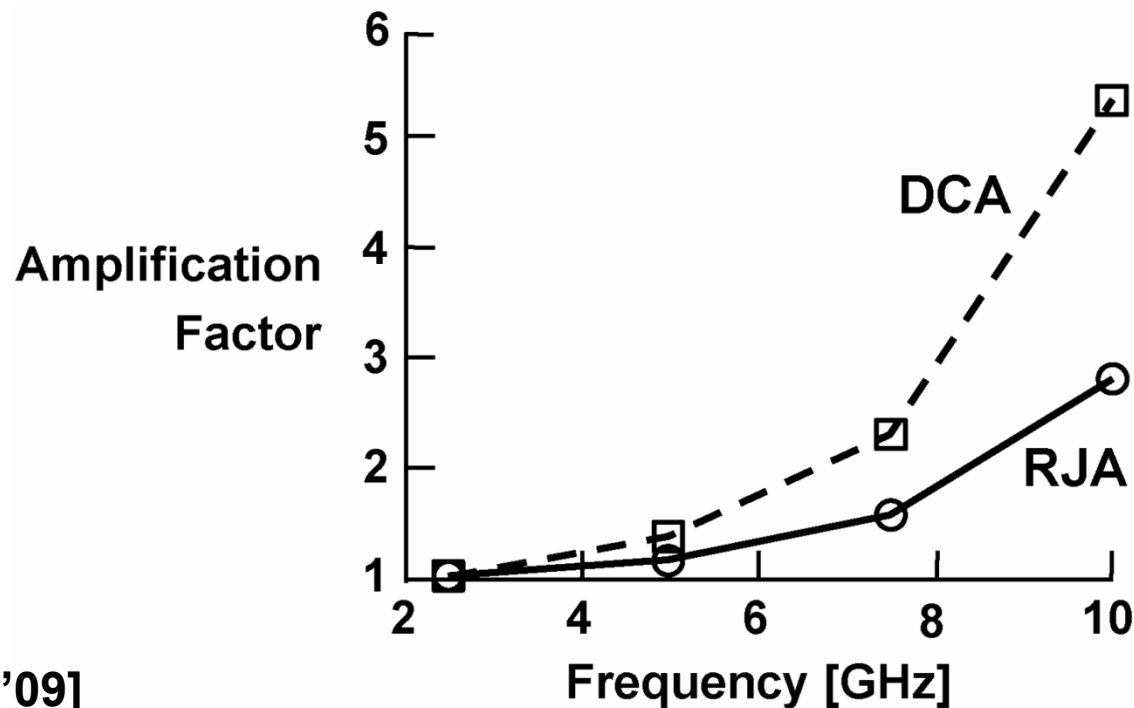
- Weight adjusted DIDC chain: the initial error disappears if the weight of the second delay cell is adjusted to $(M+1):1$



[Song, EL'10]

Duty-Cycle Error Amplification

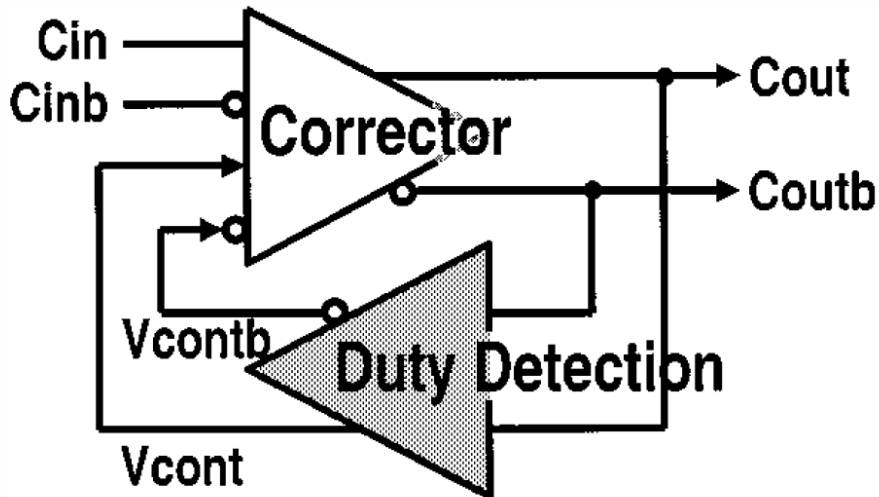
- Insufficient circuit bandwidth of delay element causes severe duty-cycle error amplification



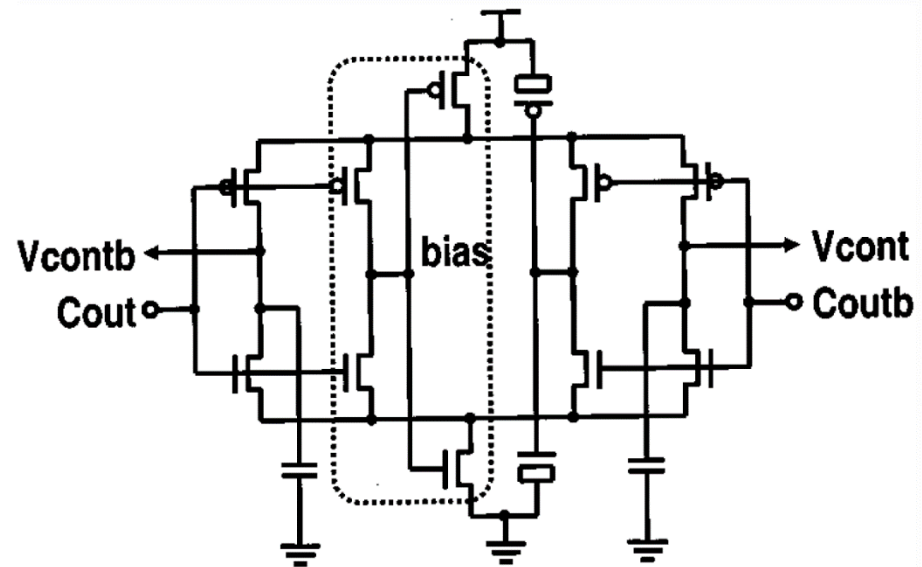
[Casper, TCAS-I'09]

Duty-Cycle Error and Correction

- DC component of clock \cong Duty-cycle of clock
- Measure the duty-cycle error by measuring DC portion of a differential clock



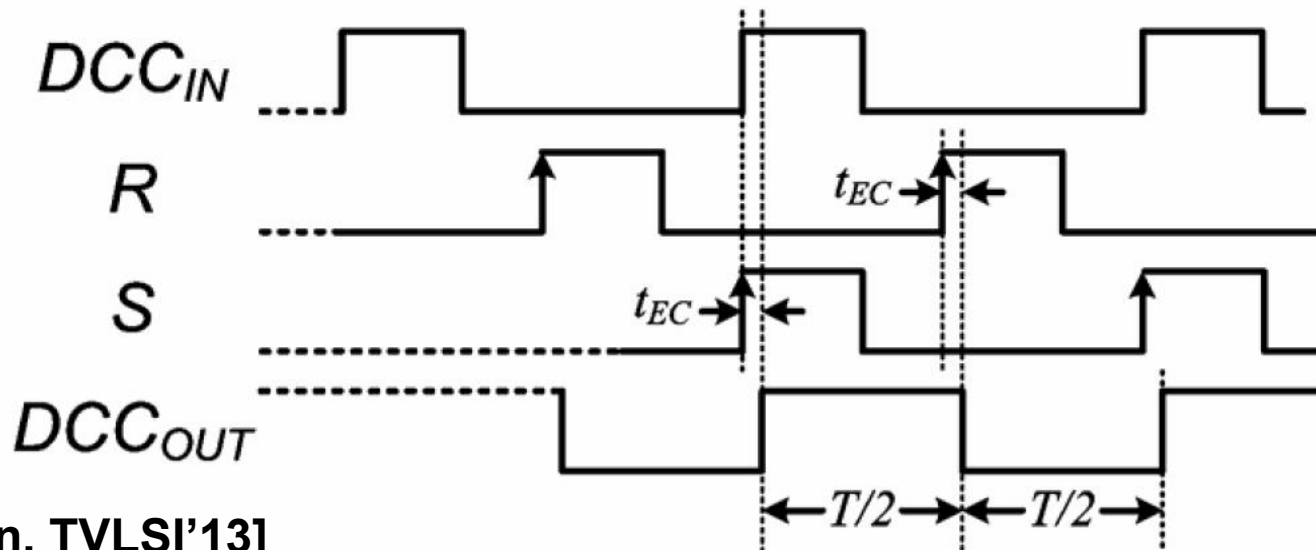
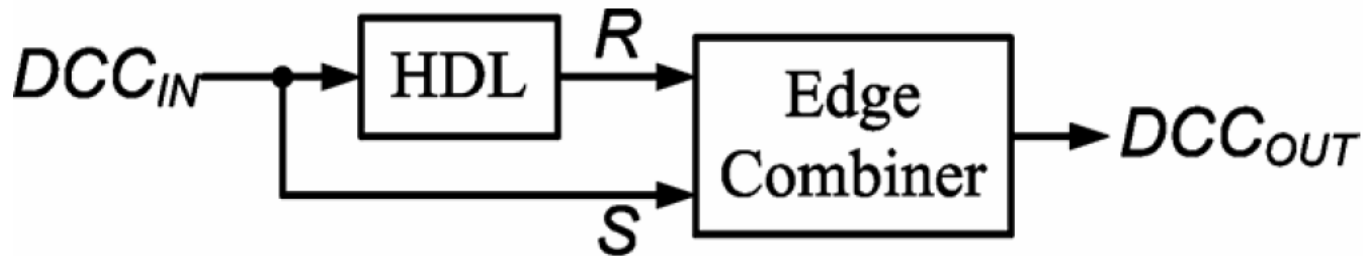
[Jung, JSSC'01]



Duty Detection Circuit

DCC with a Half Delay Line

- Duty-cycle correction (DCC) example: use of Half Delay Line

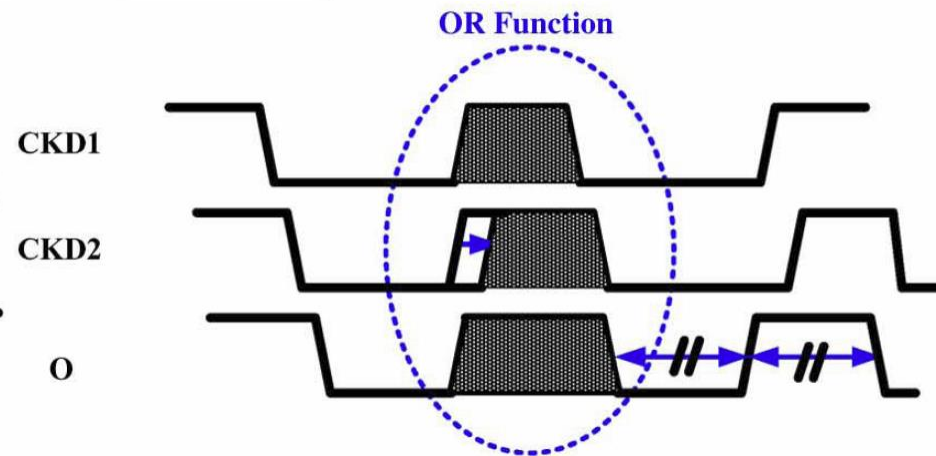
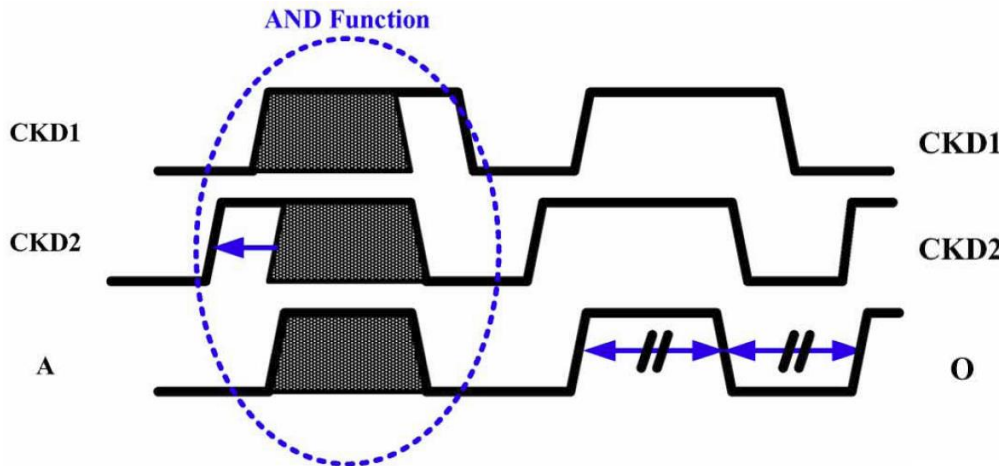
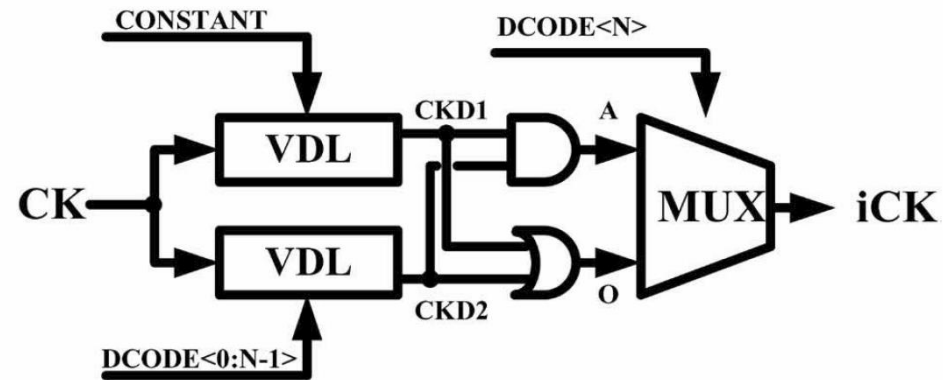


[Chen, TVLSI'13]

OR-AND DCC

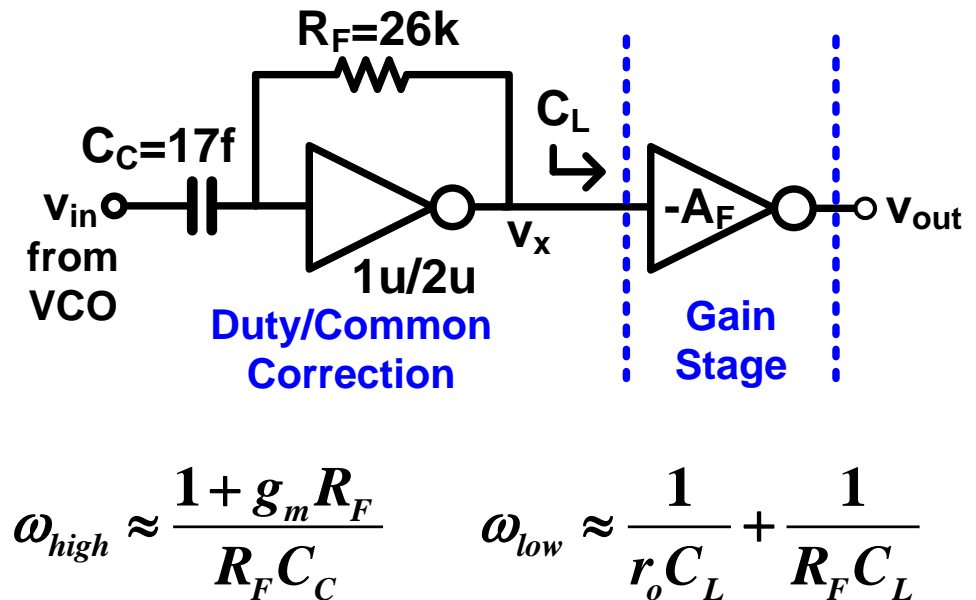
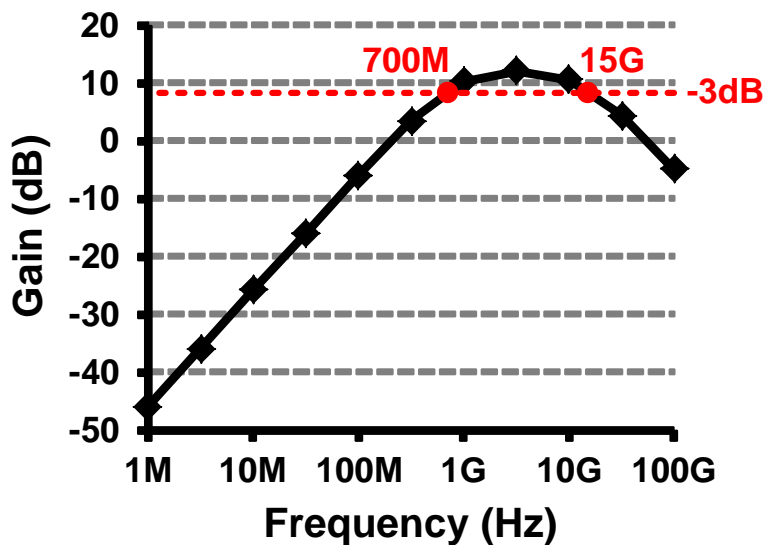
- Duty-cycle correction (DCC) example
- At first, duty detector selects either AND or OR path
- Changing the delay of the VDL based on the duty detector

[Lee, JSSC'12]



Open Loop Duty Correction

- Duty correction using band-pass filter
- Small capacitor at high frequency



[Bae, A-SSCC'15]

Delay-Locked Loops

5.6 All-Digital DLL

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

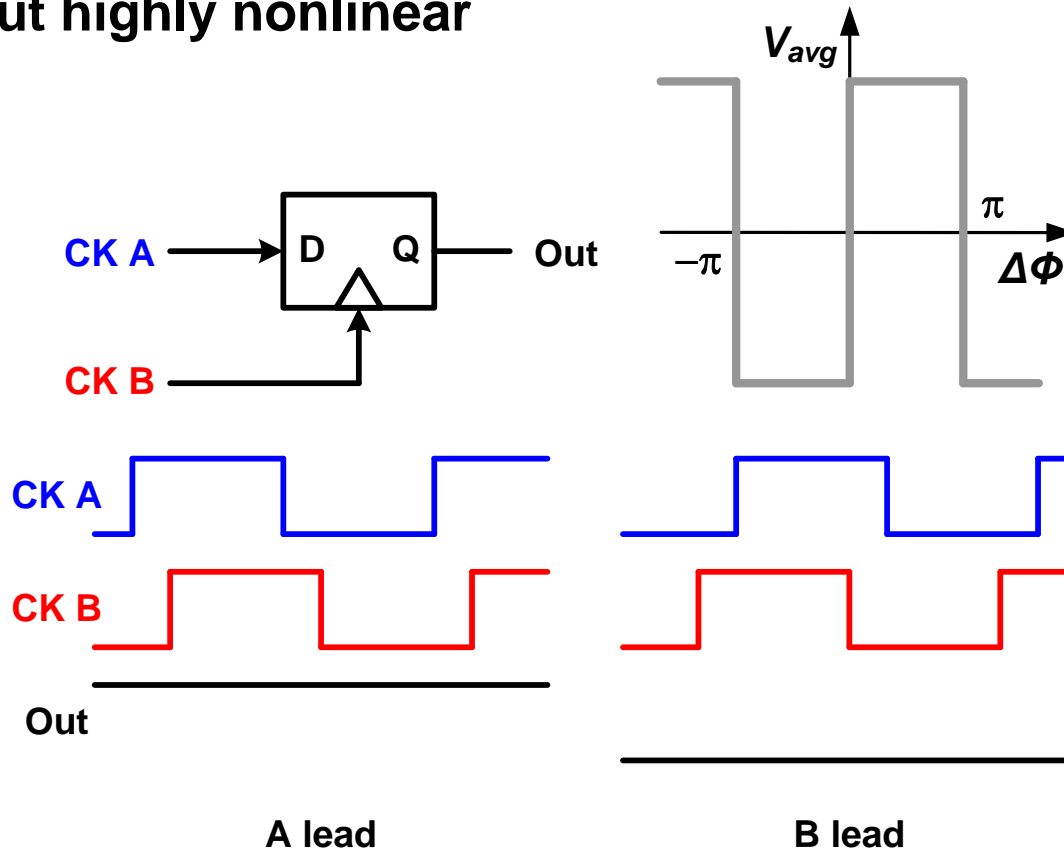
October 27, 2020

All-Digital DLL

- All components provide digital interface only
- Suitable for deep-submicron technology using low supply
- PVT variation can be mitigated
- Less sensitive to gate leakage
- Fast locking
- Storing locking information during power down mode

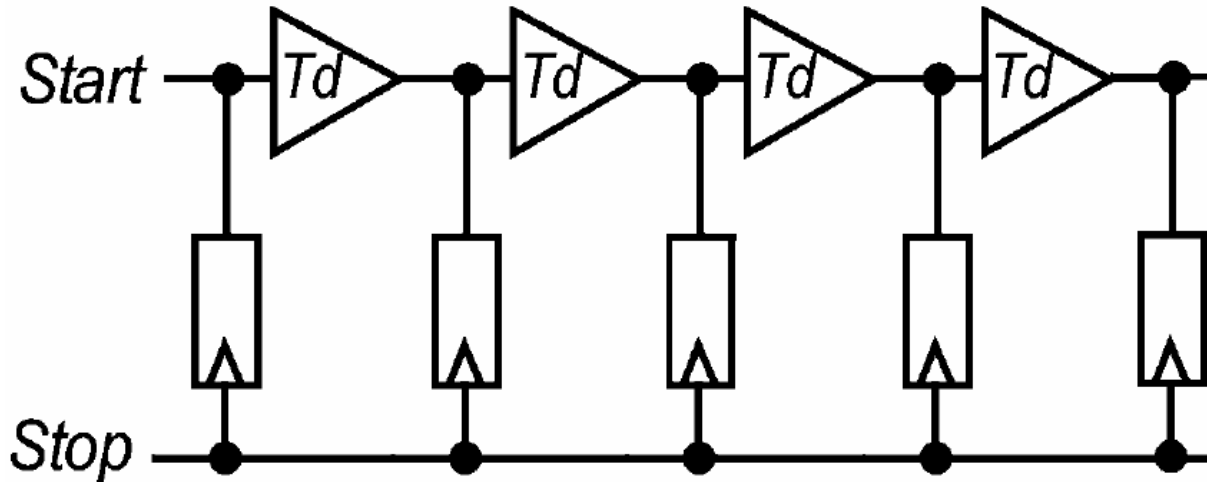
Digital Phase Detector - 1

- Binary phase detector
- Simple but highly nonlinear



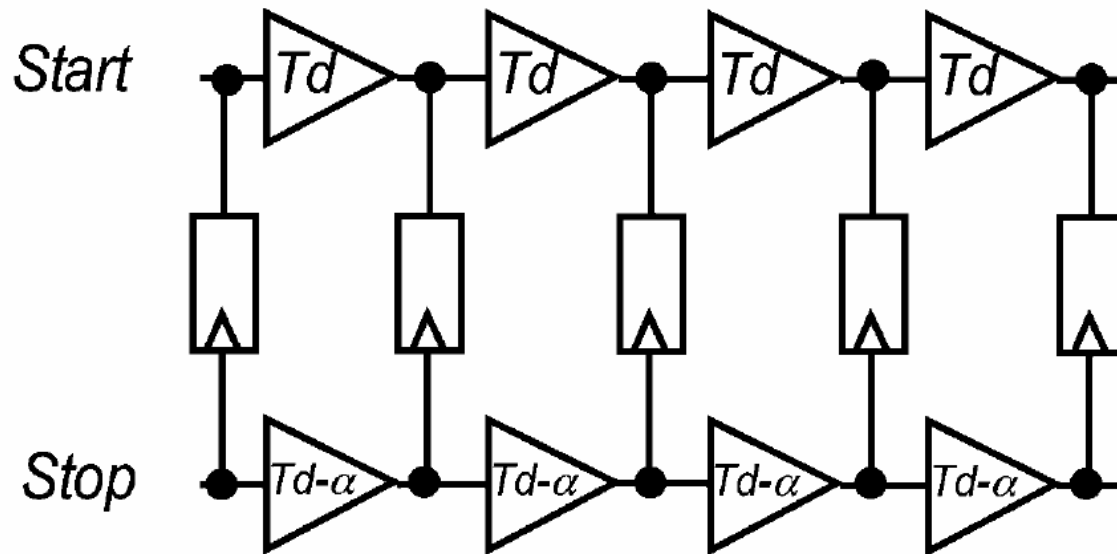
Digital Phase Detector - 2

- Linear: Time-to-digital converter (TDC)
- Conventional TDC: delay chain and samplers
- Resolution: Intrinsic gate delay



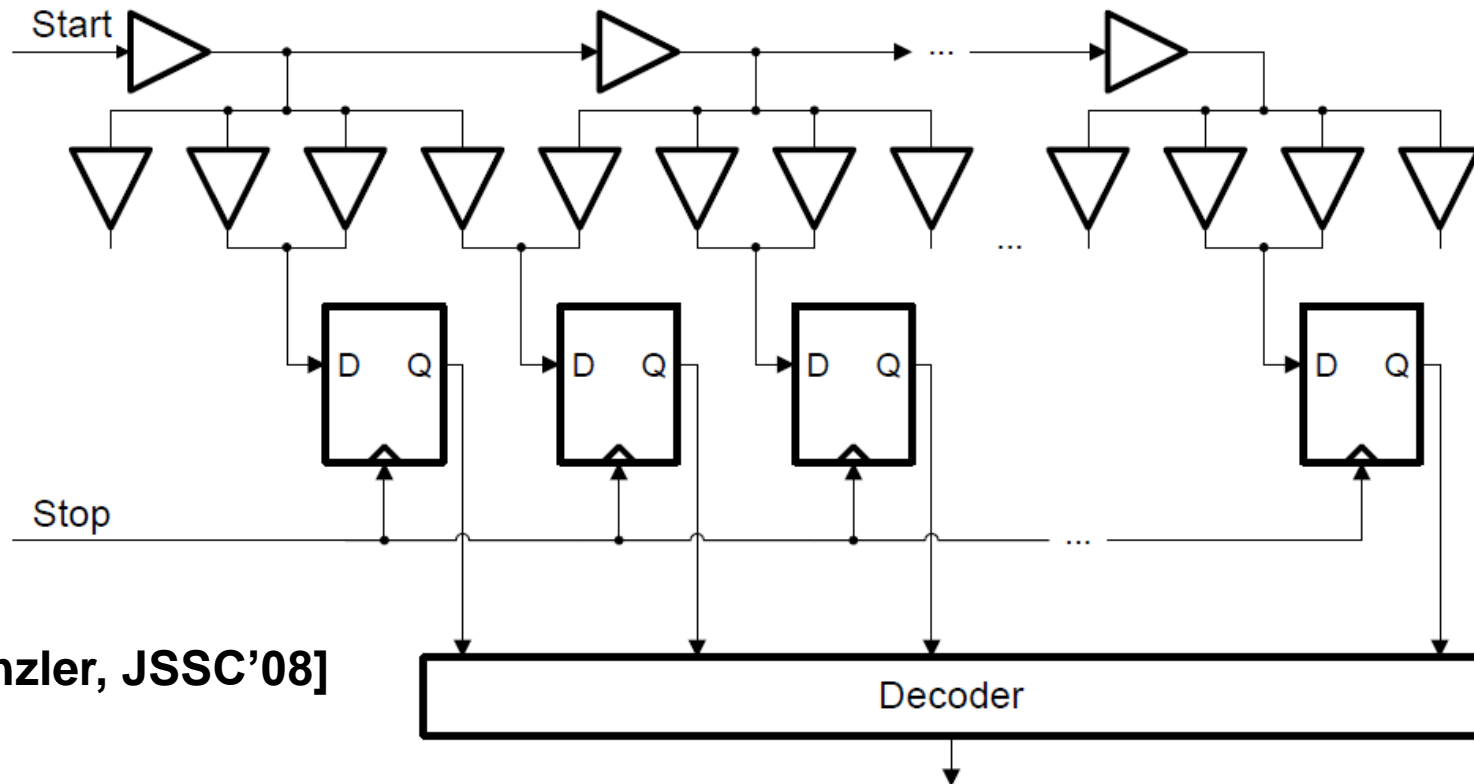
Digital Phase Detector - 3

- Vernier TDC
- Fine resolution but large area, high power consumption



Digital Phase Detector - 4

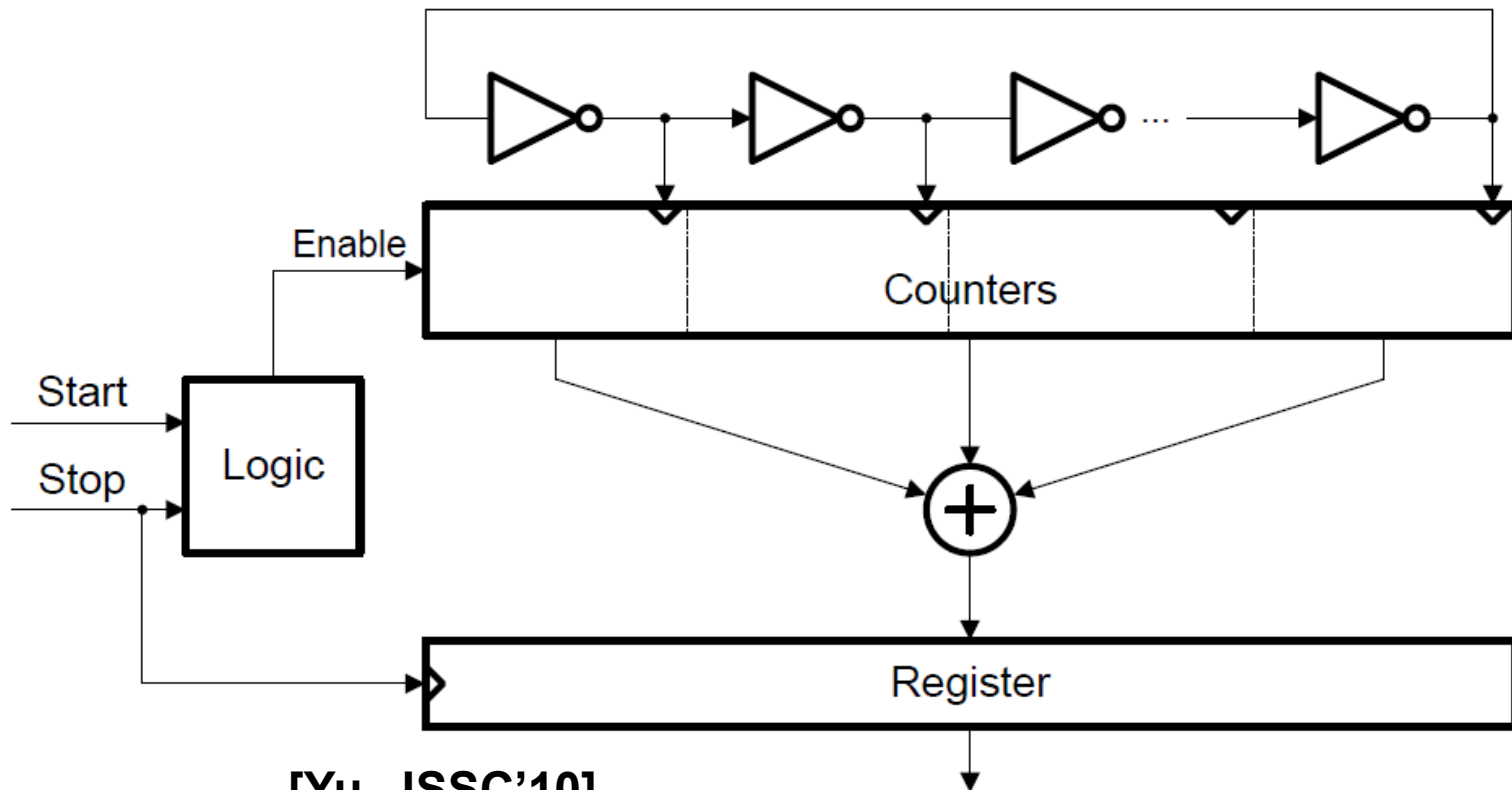
- Interpolative TDC
- Fine resolution using phase interpolation



[Henzler, JSSC'08]

Digital Phase Detector - 5

- Ring oscillator-based TDC
- Wide dynamic range, large power consumption by oscillator



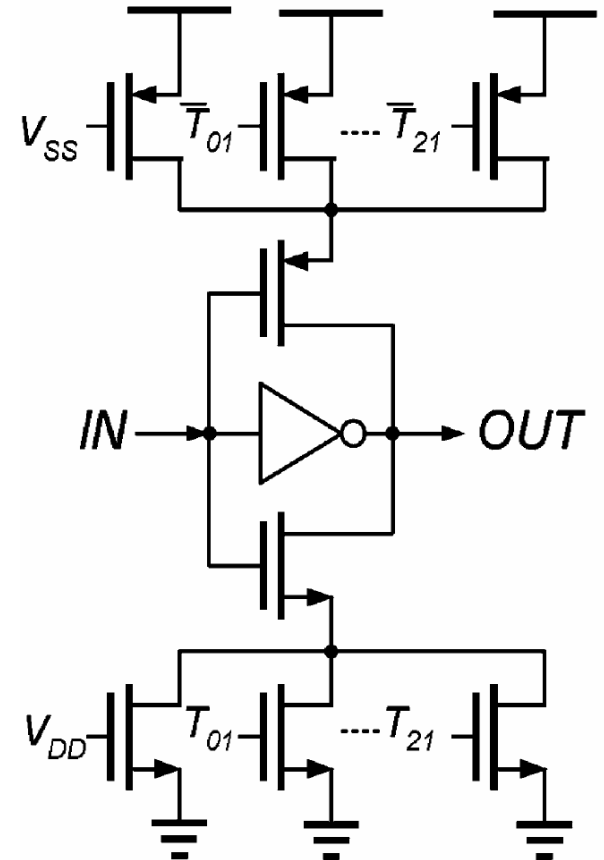
[Yu, JSSC'10]

Digital Delay Element Examples

- **Current-starved inverter**
- **Multiplexer-based delay**
- **Lattice delay line**
- **Synchronous mirror delay**

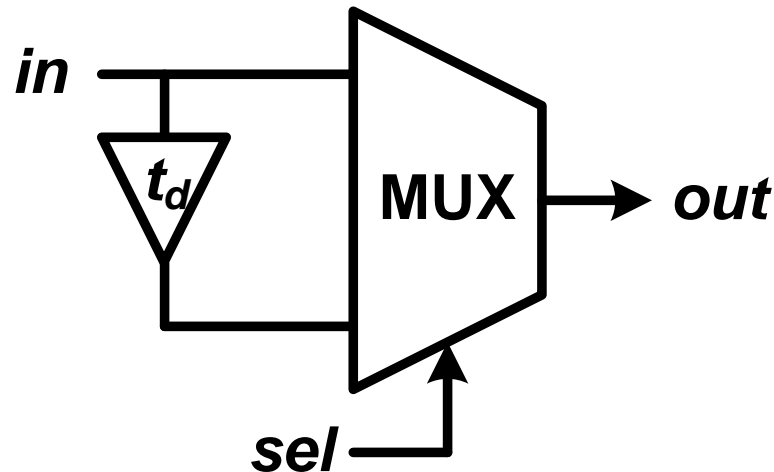
Current-Starved Inverter

- Digitally controlled current-starved inverter
- Fine resolution
- Dynamic range and intrinsic delay depend on clock frequency



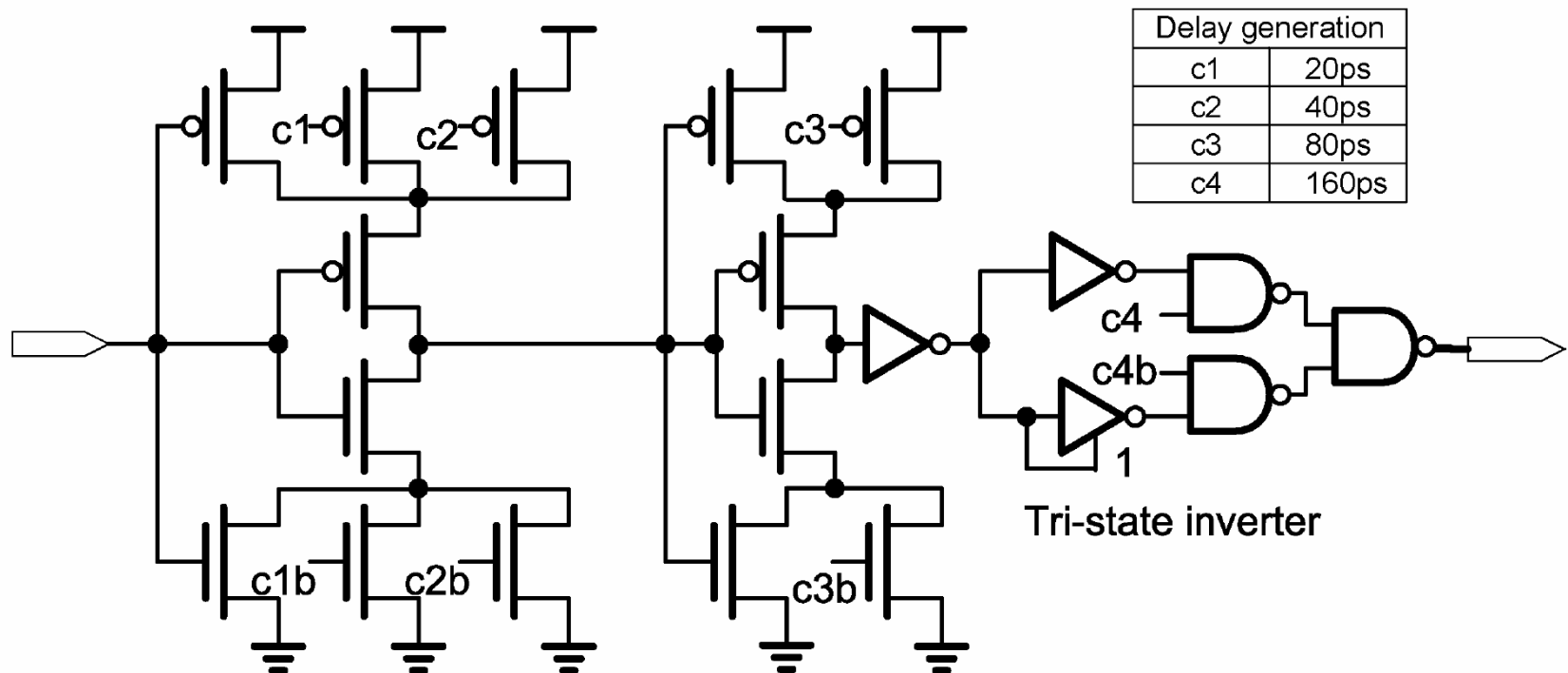
Multiplexer-Based Delay

- Most straightforward
- Tunable delay range increases by cascading the delay units, but the intrinsic delay increases as well



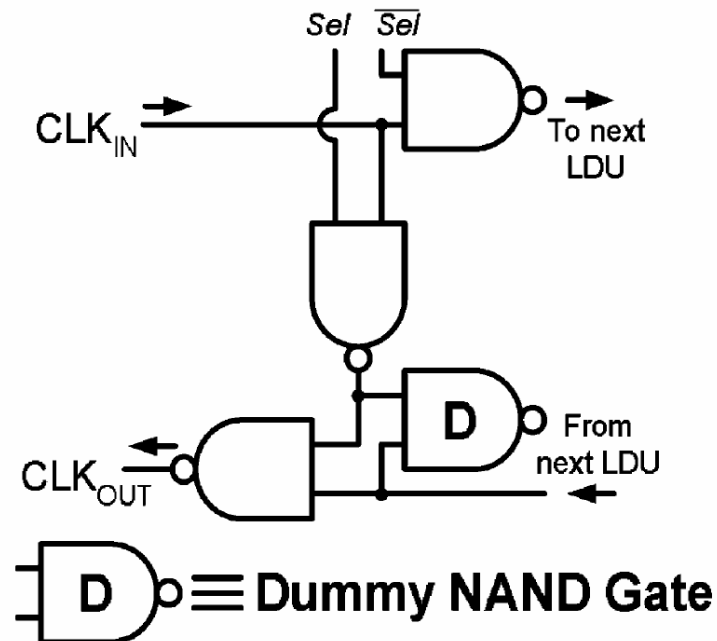
Multiplexer-Based Delay

- Multiplexer-based delay element and current-starved delay element are combined



Lattice Delay Unit

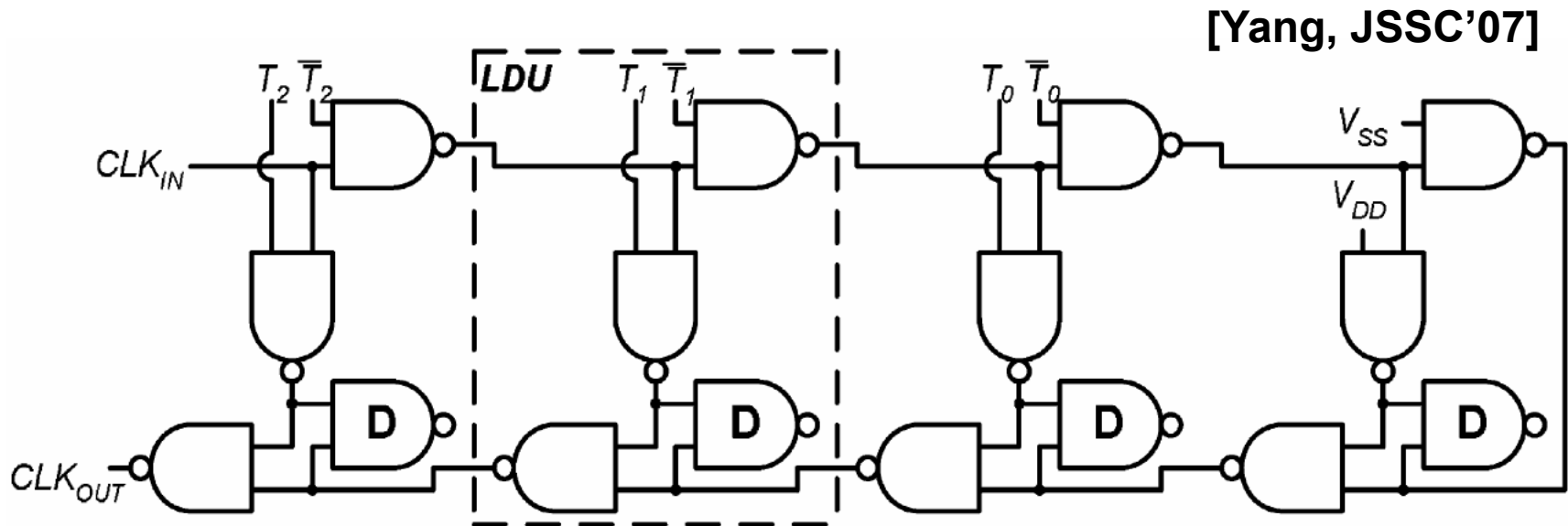
- Intrinsic delay of two NAND delay
- Delay step of two NAND delay
- Dummy NAND for fan-out balancing



[Yang, JSSC'07]

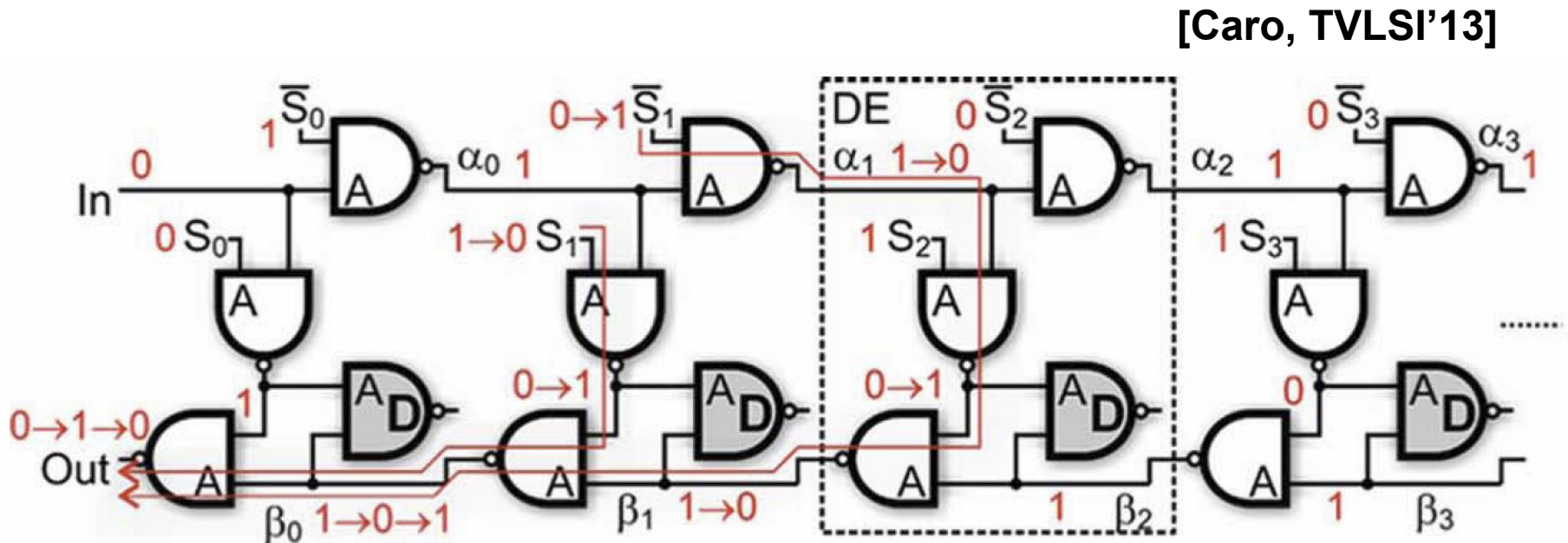
Lattice Delay Line

- Intrinsic delay of two NAND delay
- As the operating frequency increases, the number of activated delay units is reduced and the power remains the same
- Breaking dependency between total delay stages and power



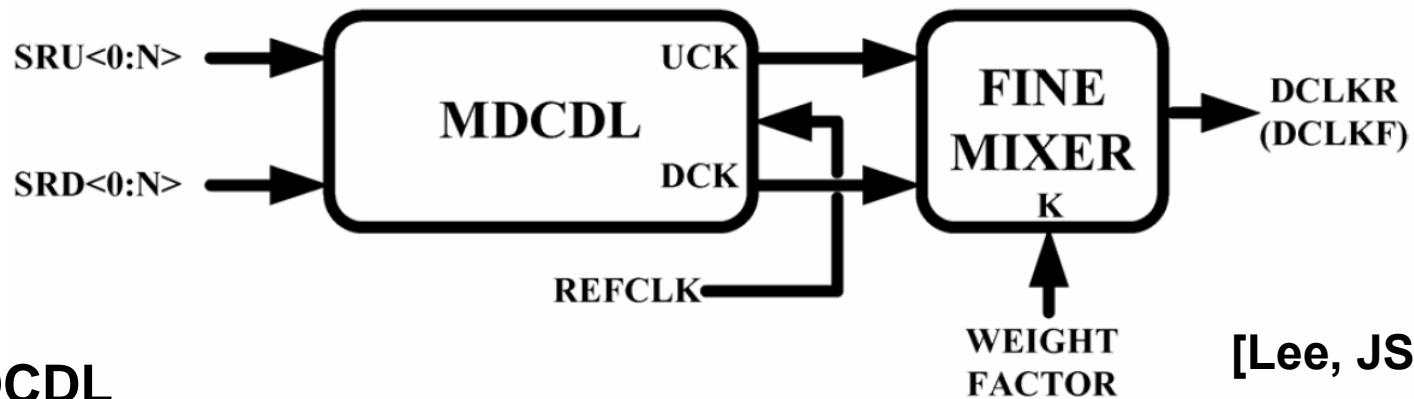
Lattice Delay Line

- Glitch issue
- Switching of S_1 results in two different paths that generates an output glitch

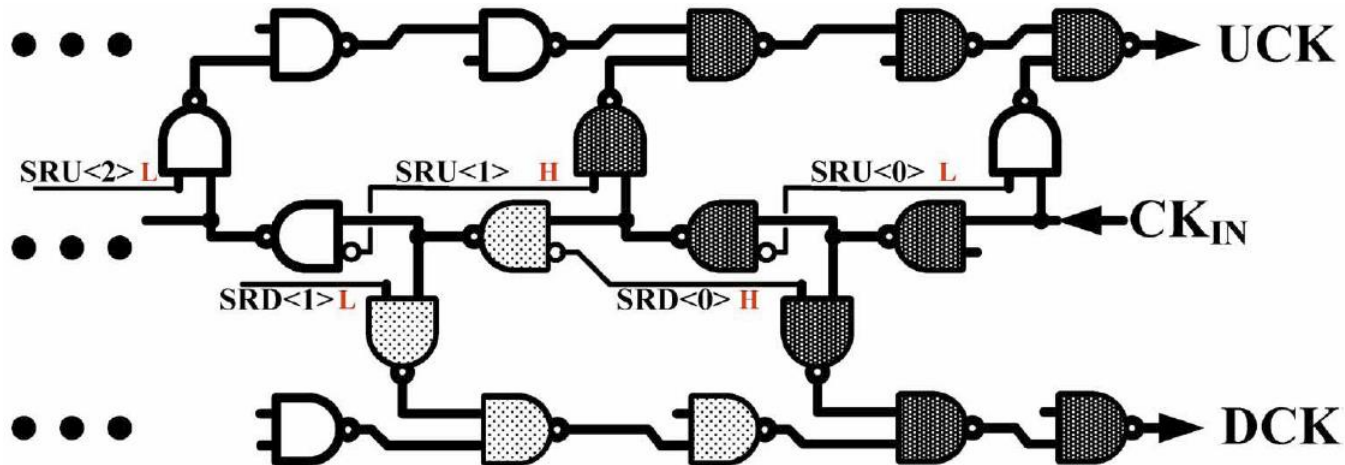


Modified Lattice Delay Line

- Dual lattice delay line followed by fine phase mixer

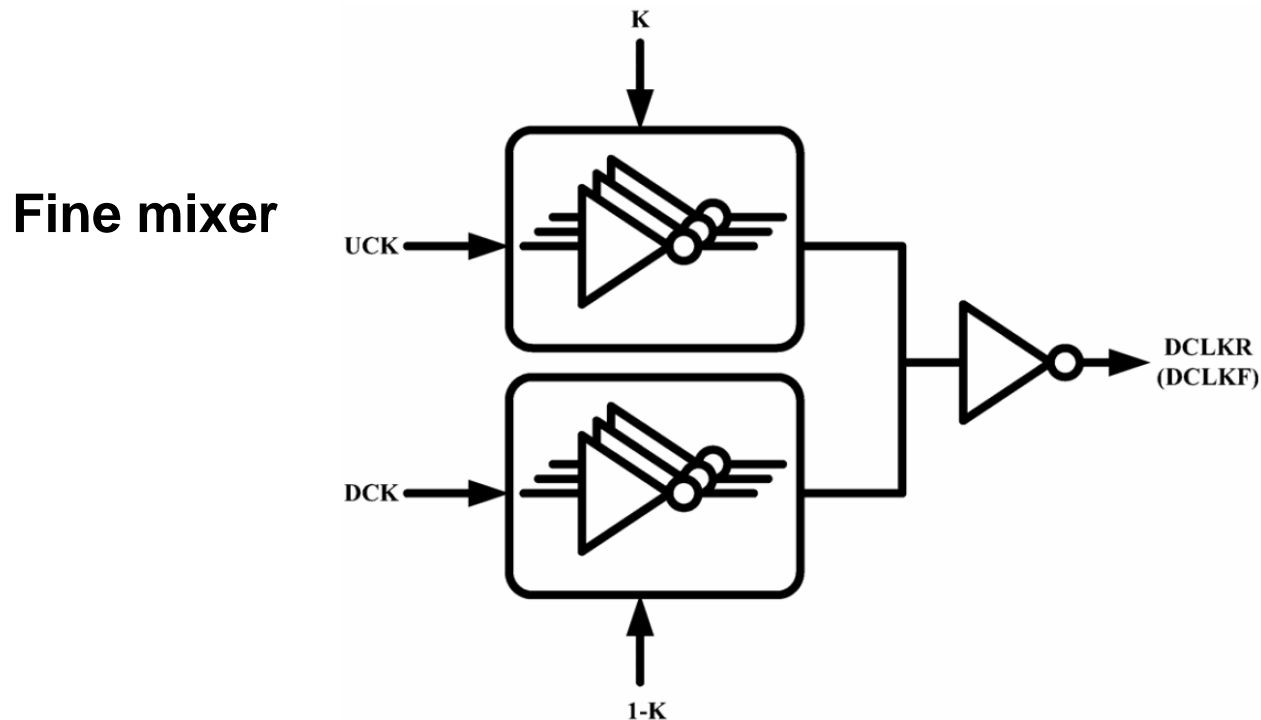


MDCDL

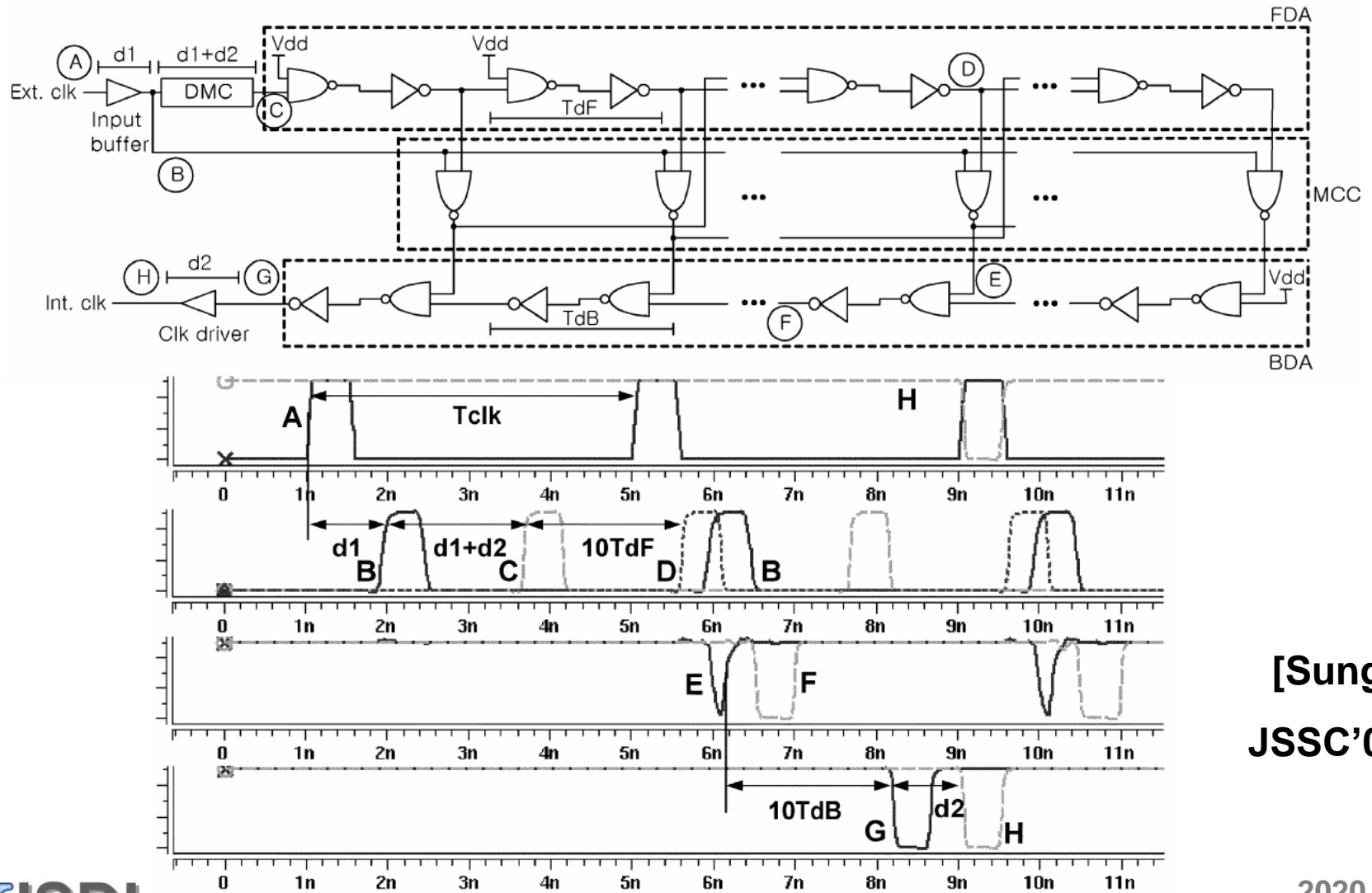


Modified Lattice Delay Line

- Dual lattice delay line followed by fine phase mixer
- Seamless operation between coarse and fine delay line
- Power saving by shared delay line



Synchronous Mirror Delay



[Sung,
JSSC'04]

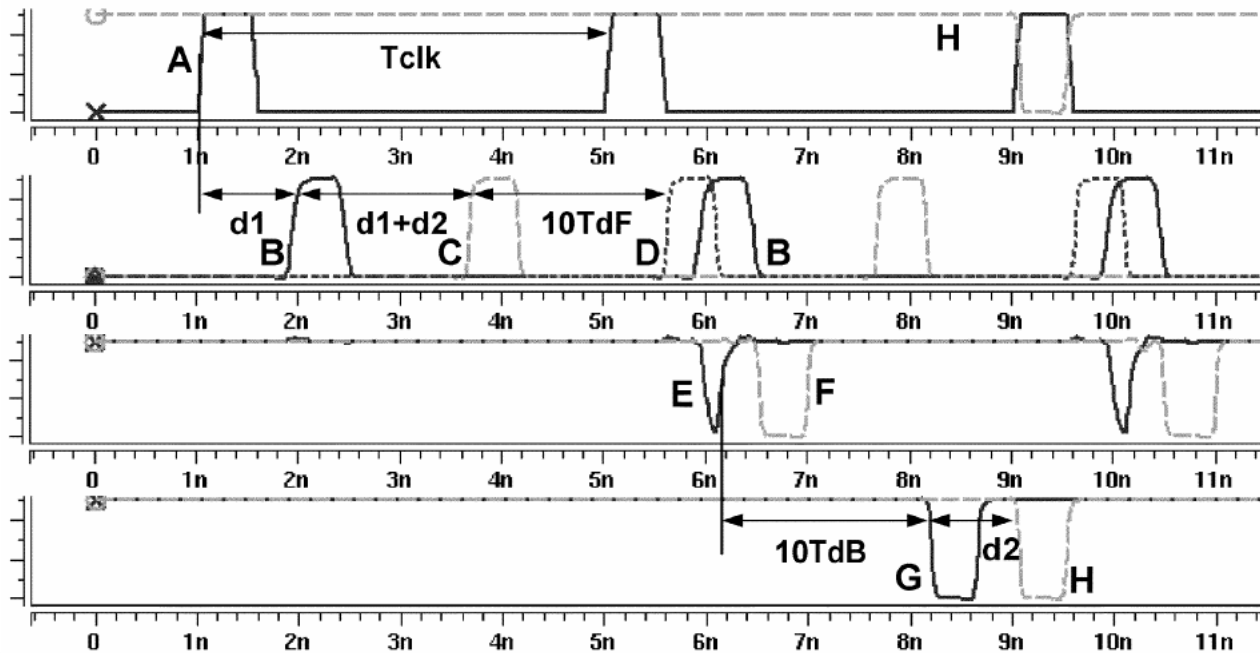
Synchronous Mirror Delay

$$T_{clk} = d1 + (d1 + d2) + 10TdF - d1$$

$$\therefore 10TdF = T_{clk} - (d1 + d2)$$

$$T_{out} = d1 + (d1 + d2) + 10TdF + 10TdF + d2$$

$$= 2 T_{clk}$$



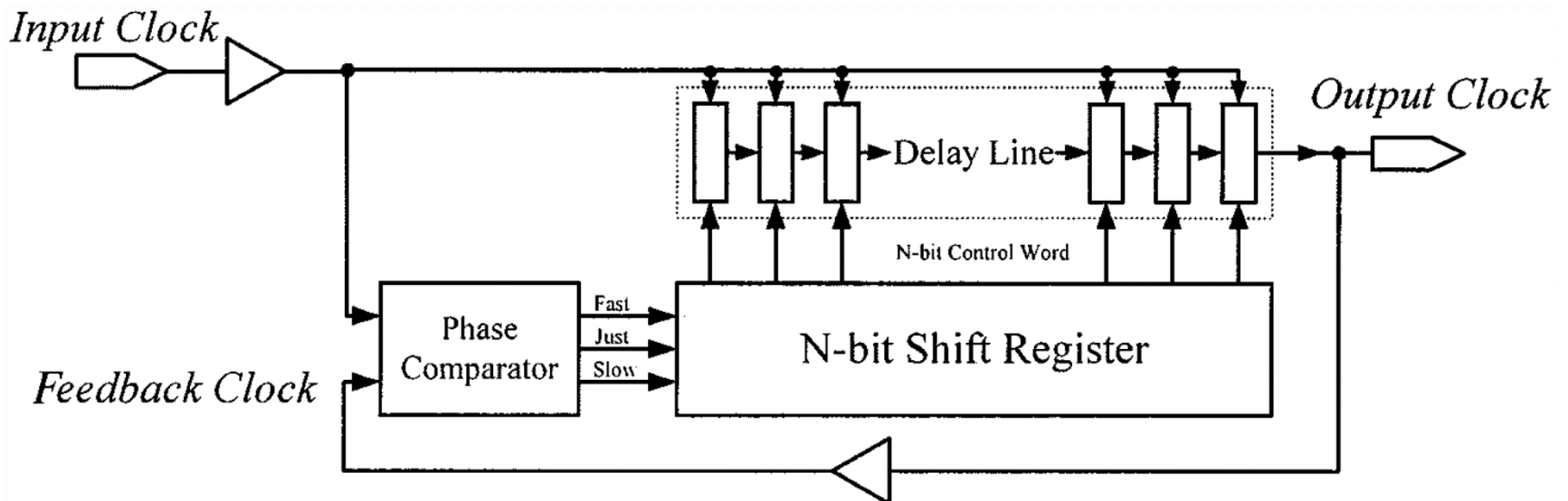
[Sung,
JSSC'04]

Synchronous Mirror Delay

- **Forward delay array (FDA) measures timing information of lock**
- **Backward delay array (BDA) with a mirror control circuit (MCC)**
- **Clock pulse is propagated backward through BDA as it is propagated forward through FDA**
- **Total delay is two clock cycle: clock skew is suppressed in two clock cycle**
- **Device mismatch and dynamic noise make the skew between the clocks**

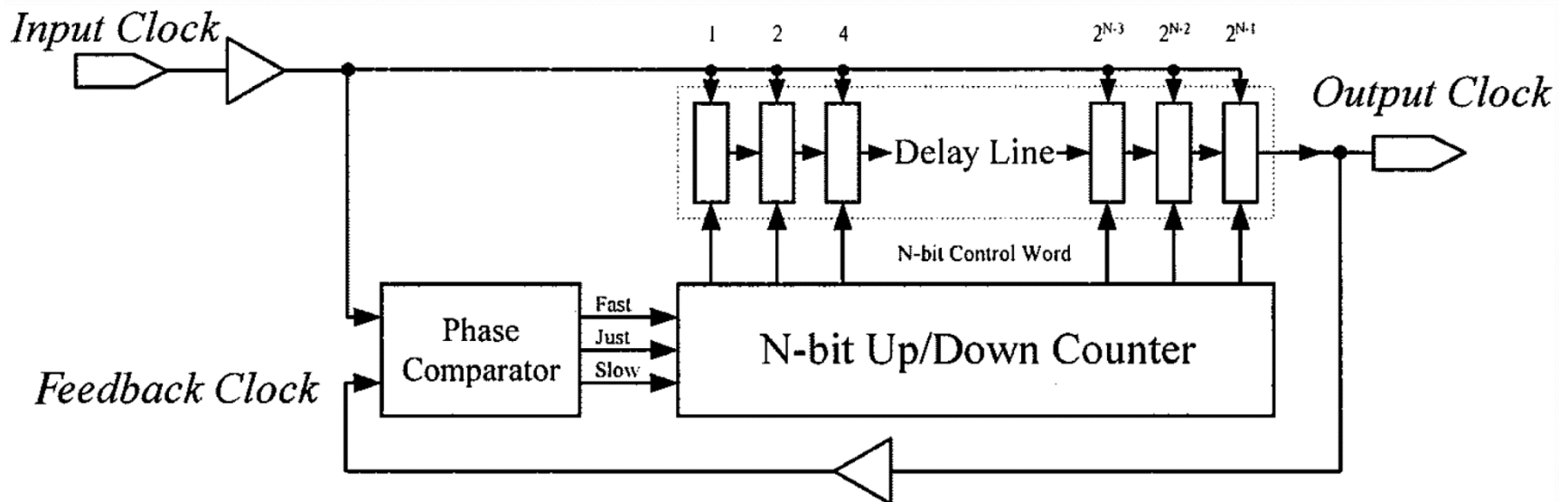
DLL Architecture - 1

- Register-controlled DLL
- Only one bit of the shift register is active to select a point of entry of the delay line (One-hot coded)
- Wider range achieved by adding more delay stages: large area



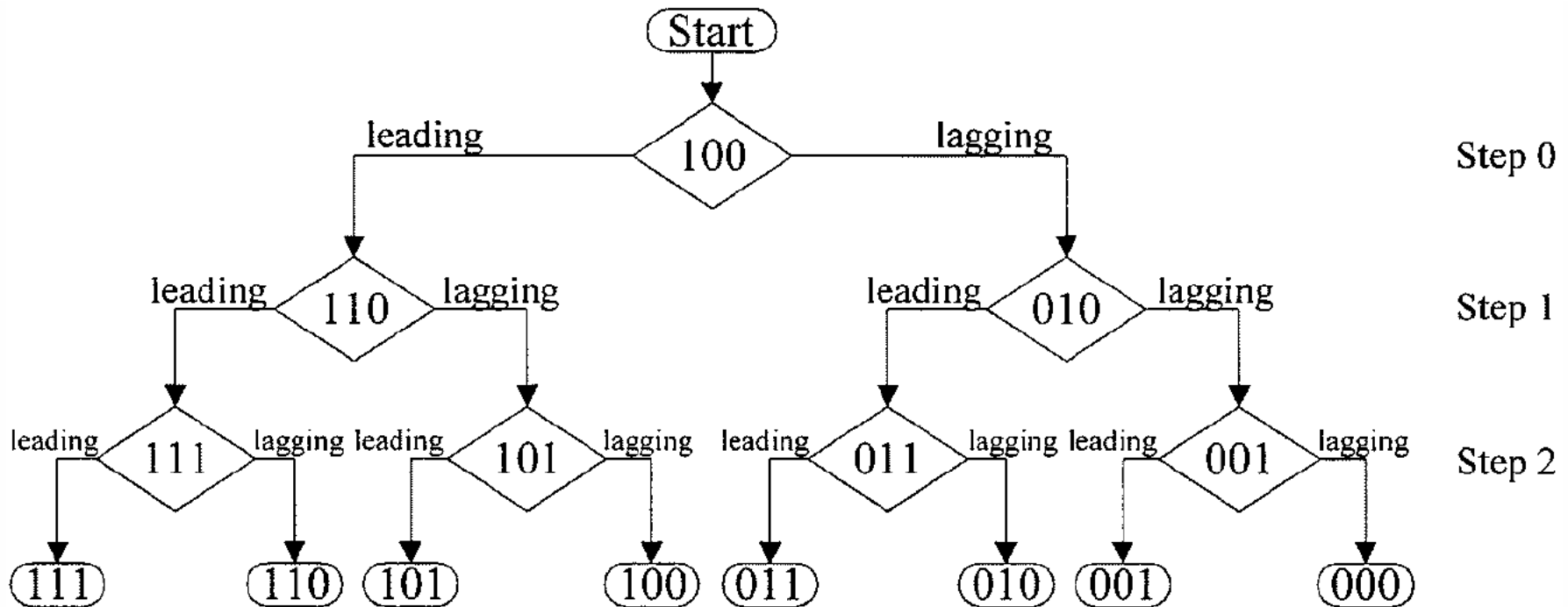
DLL Architecture - 2

- Counter-controlled DLL
- Binary-weighted delay line
- 64-bit shift register in a RDLL can be replaced by 6-bit counter
- Long lock time: 32 clock periods for 6-bit counter



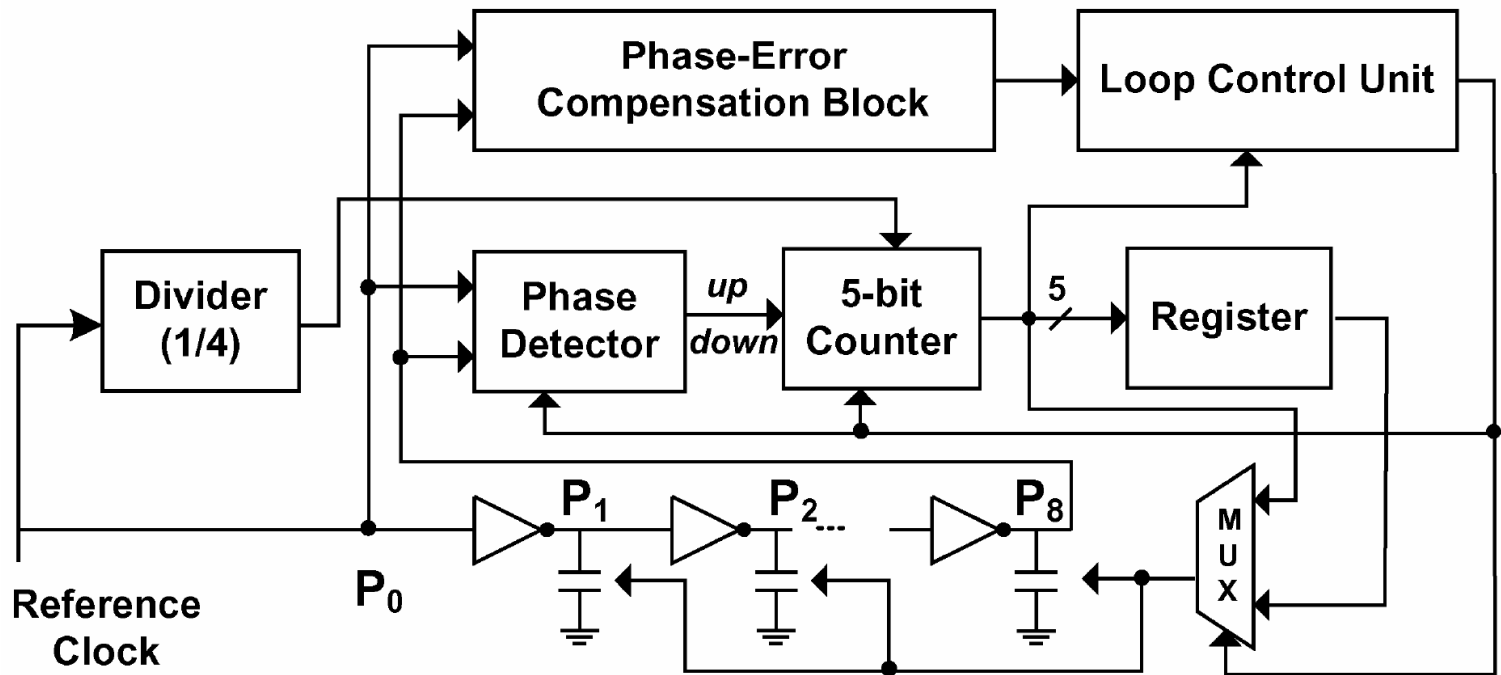
DLL Architecture - 3

- SAR-controlled DLL
- 6 clock periods for 6-bit binary-weighted delay line

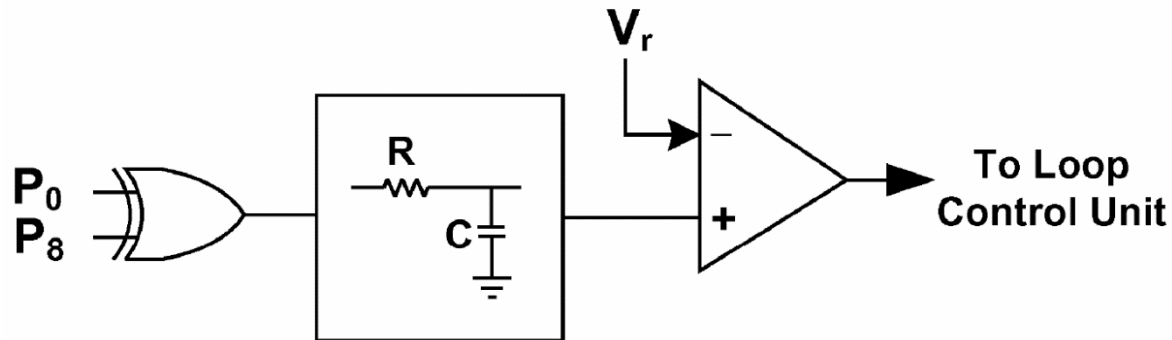


Low Power with Open-Loop Mode

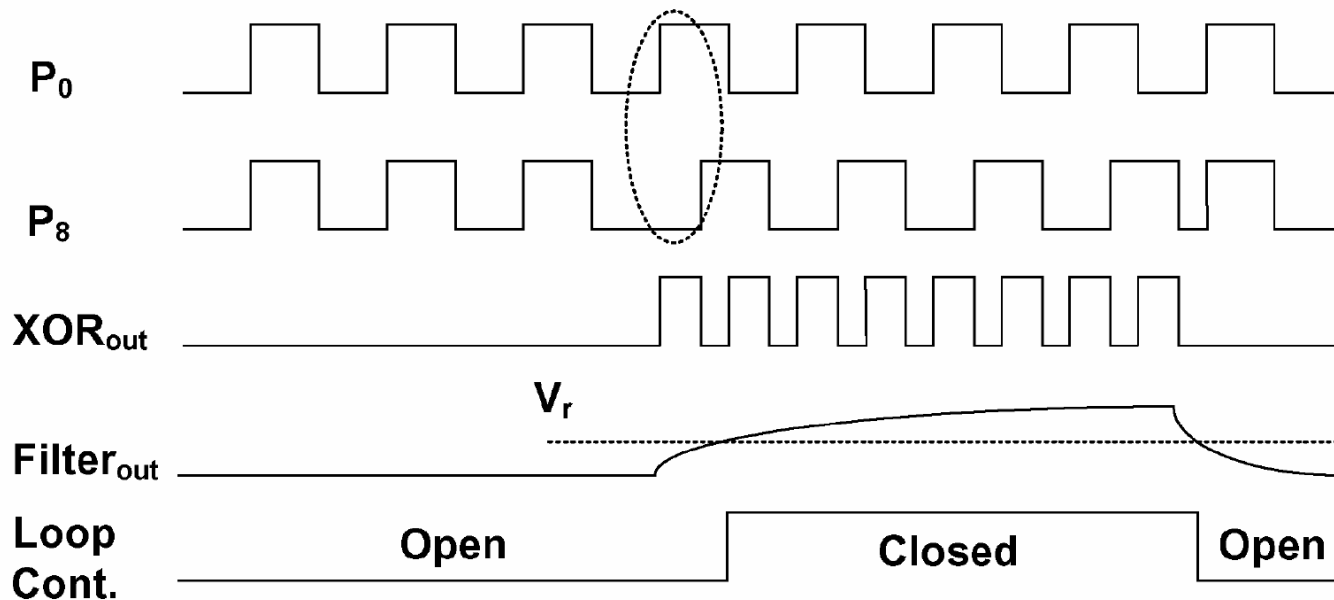
- Once DLL is locked, the feedback loop is opened by LCU
- If any phase error is detected, PEC block sends a closed-loop request to LCU



Phase-Error Compensation Block



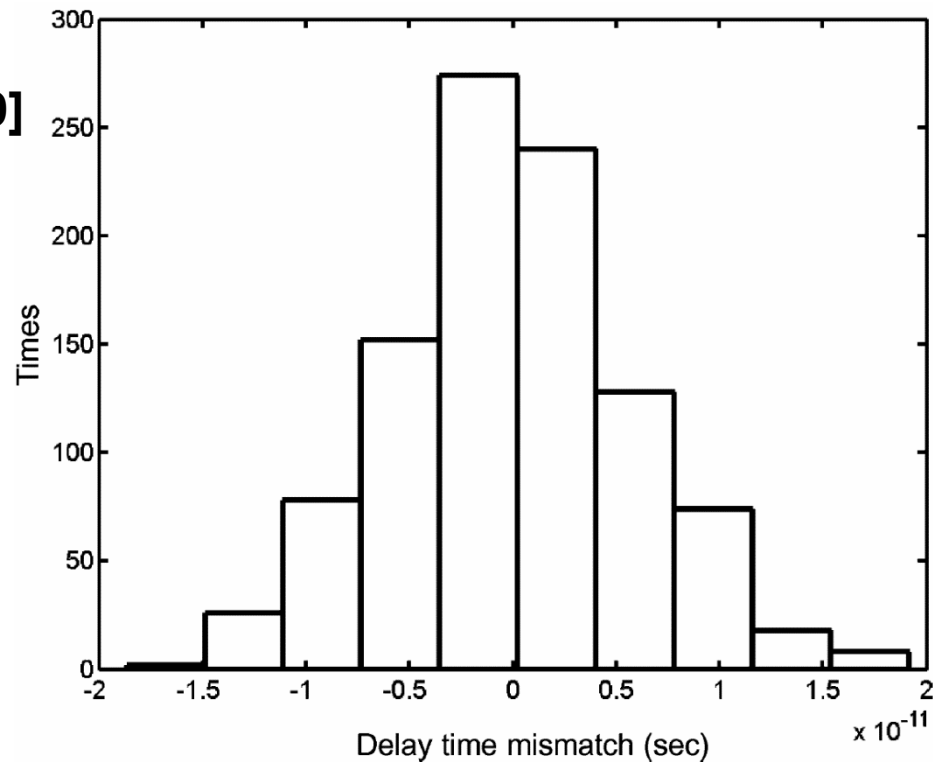
Out of Lock



DLL with Low Supply

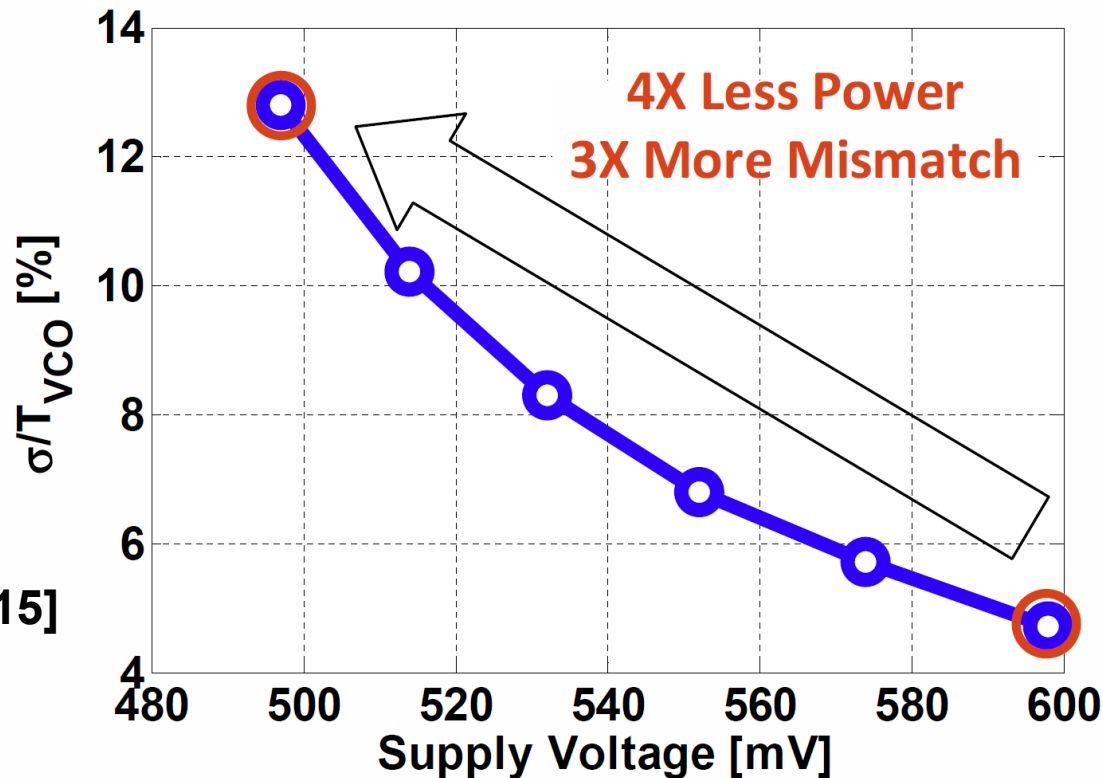
- Delay time mismatch due to the threshold voltage mismatch
- Unequal phase spacing

[Chang, JSSC'09]



DLL with Low Supply

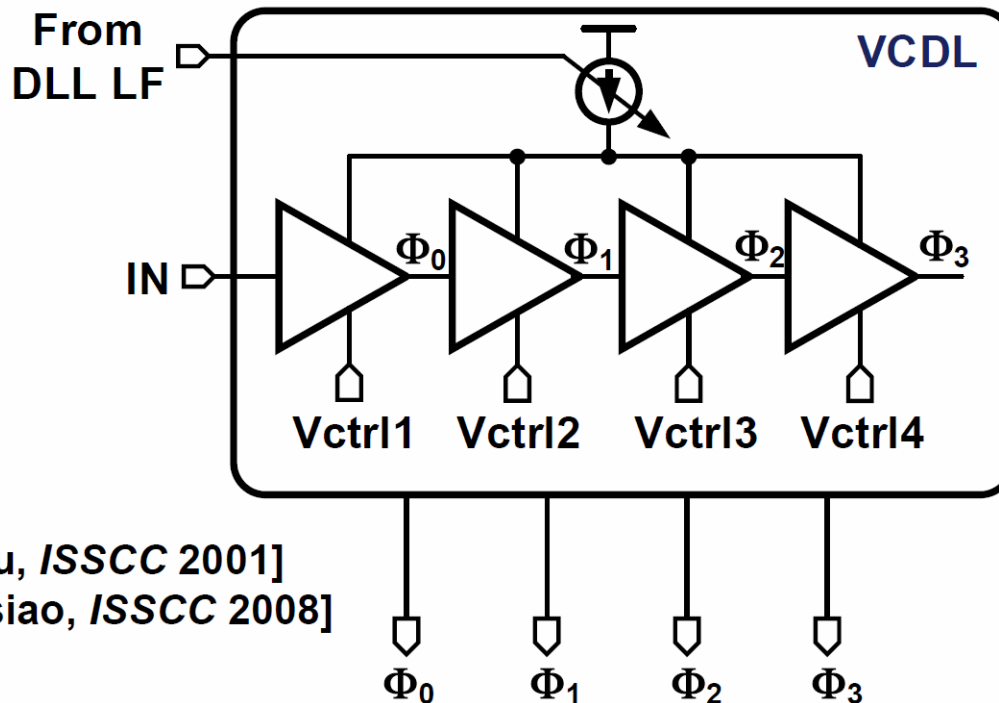
- Mismatch becomes severe with low supply voltage
- Mismatch calibration circuit required



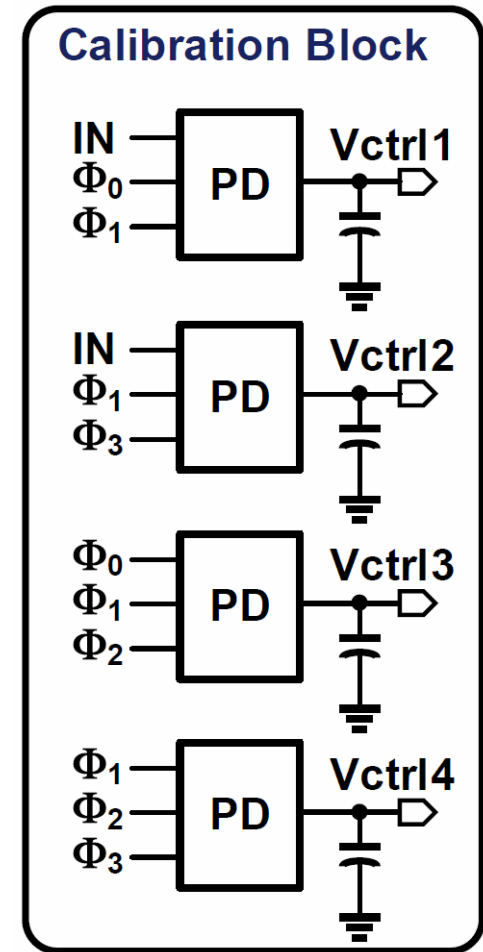
[Choi, ISSCC'15]

Mismatch Calibration Example

- Multiple DLLs
- Area overhead

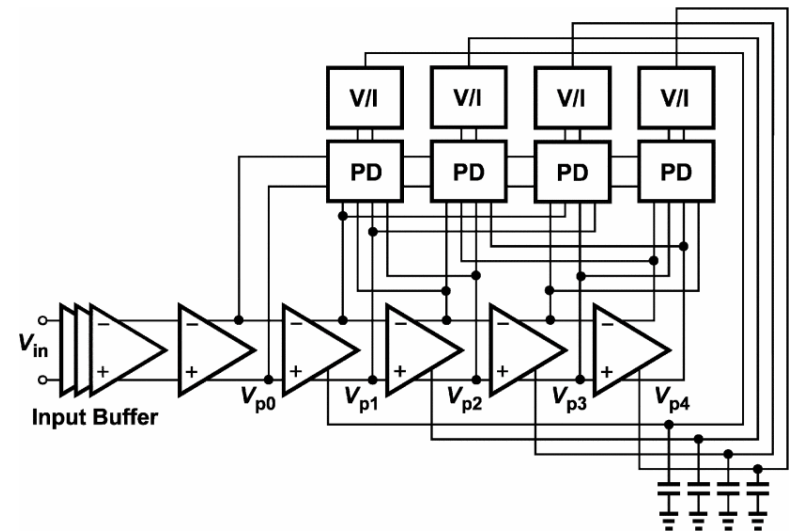
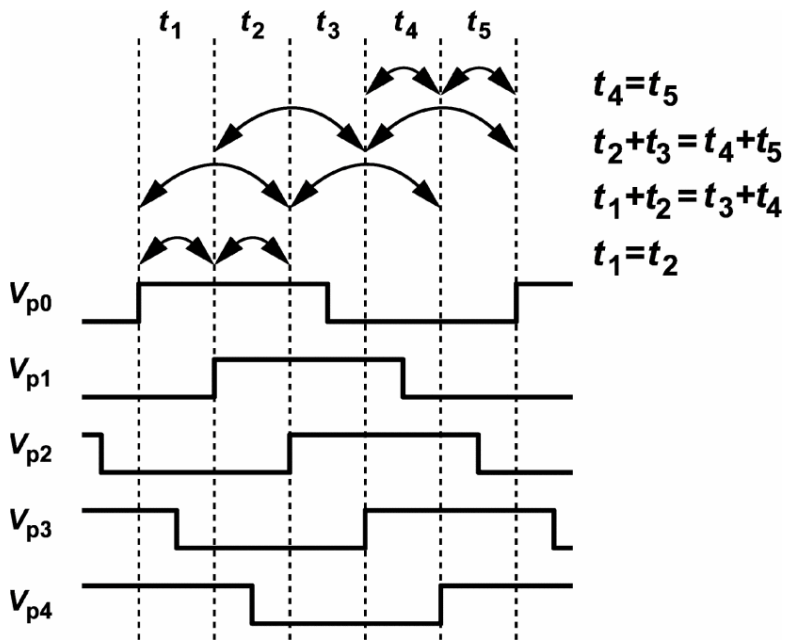


[Wu, ISSCC 2001]
[Hsiao, ISSCC 2008]



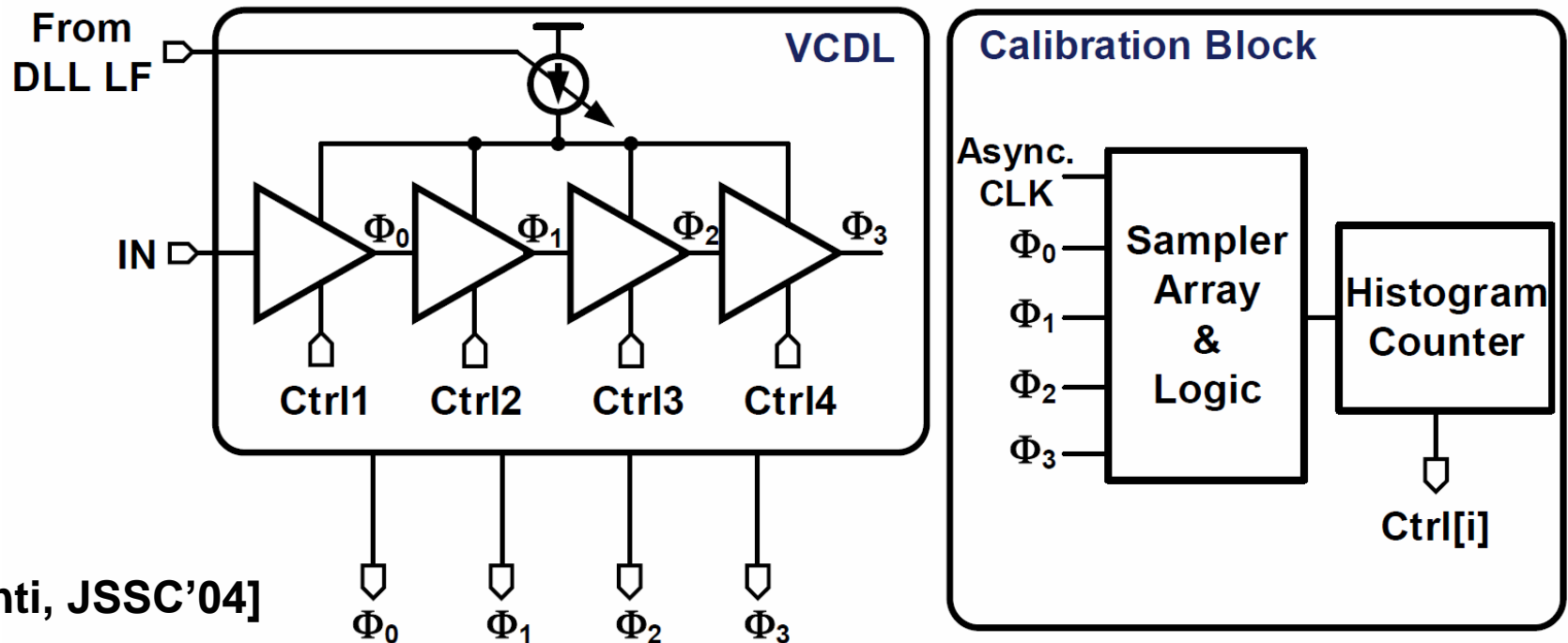
Mismatch Calibration Example

- Multiple DLLs
- Area overhead



Mismatch Calibration Example

- Code Density Test
- Asynchronous clock required



[Baronti, JSSC'04]

Delay-Locked Loops

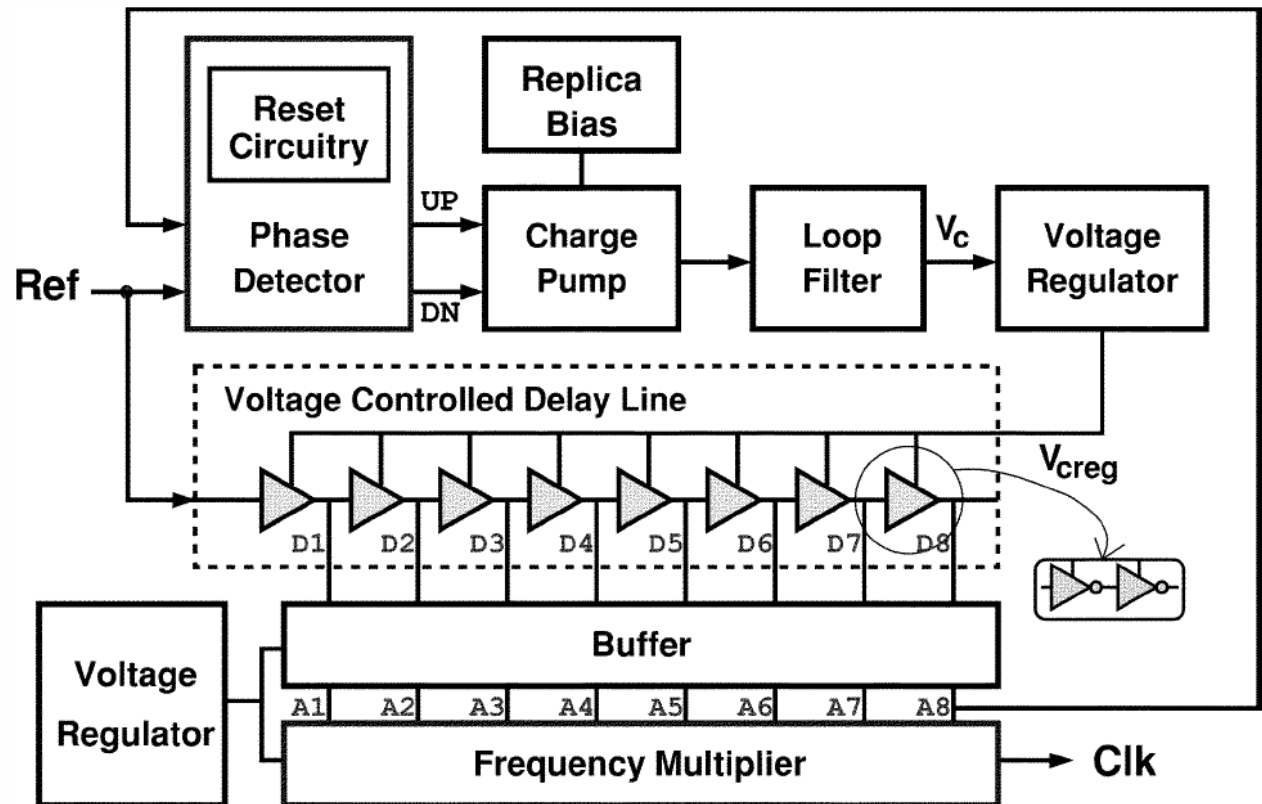
5.7 Multiplying DLL

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

October 27, 2020

DLL with a Edge-Combining Logic

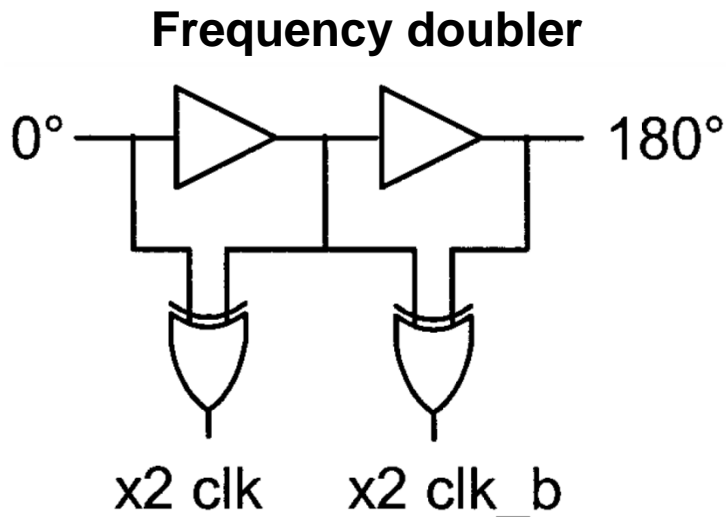
- Equally spaced phases of the reference clock are processed through an edge-combining logic



[Kim, JSSC'02]

DLL with a Edge-Combining Logic

- Any mismatch in the delay element or the edge-combining logic translates directly into duty cycle error and deterministic jitter
- Programmable clock multiplication ratio is difficult

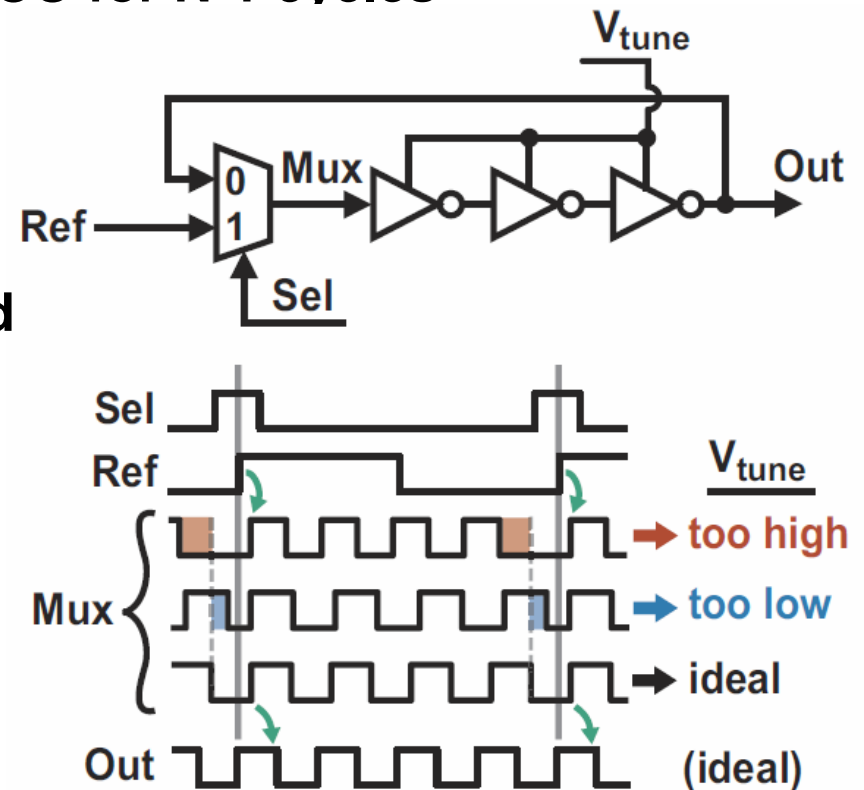


[Farjad-Rad, JSSC'02]

Multiplying DLL (MDLL)

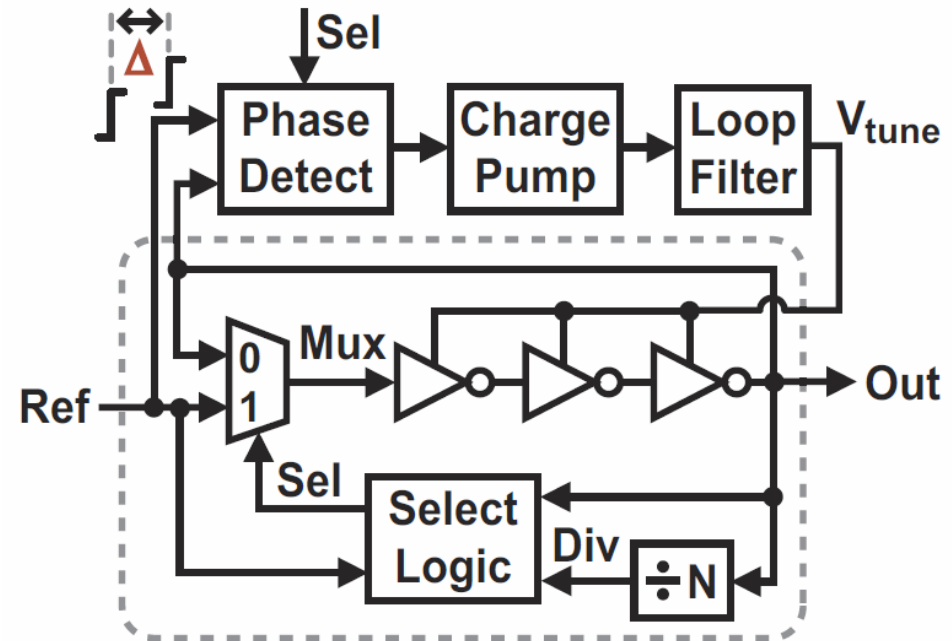
- MDLL or Recirculating DLL
- The VCDL is configured as an VCO for $N-1$ cycles
- Reset by the reference clock for one cycle
- Incorrect V_{tune} leads to undesired deterministic jitter

[Helal, Ph.D Dissertation, MIT]



Multiplying DLL (MDLL)

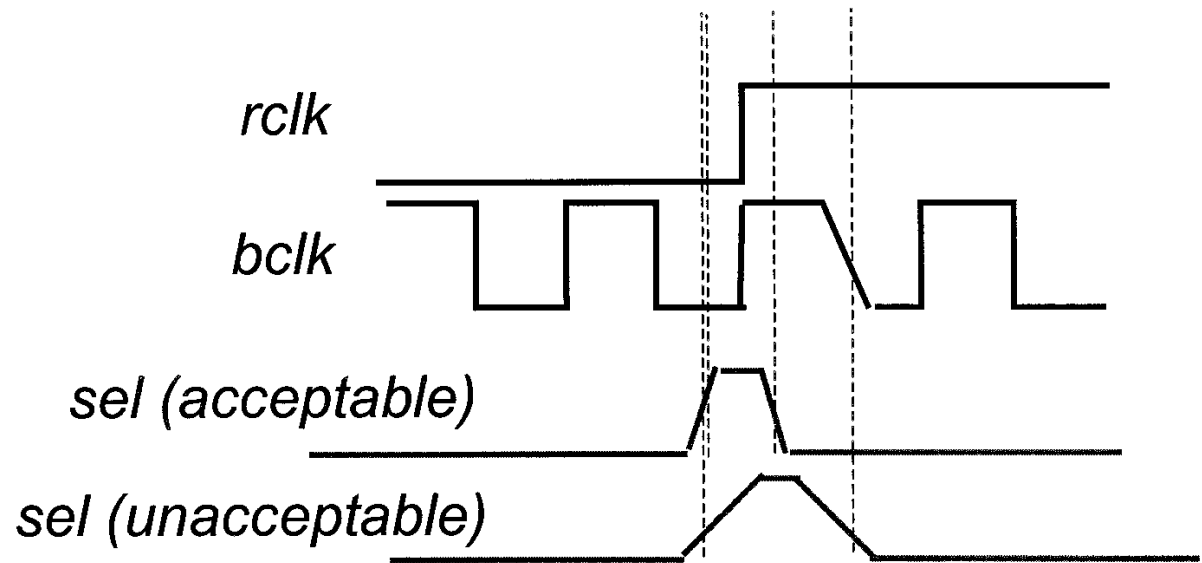
- A PLL-like feedback loop to set proper V_{tune}
- Path mismatch in the MUX and phase detector causes non-ideality



[Helal, Ph.D Dissertation, MIT]

MUX Select Signal for MDLL

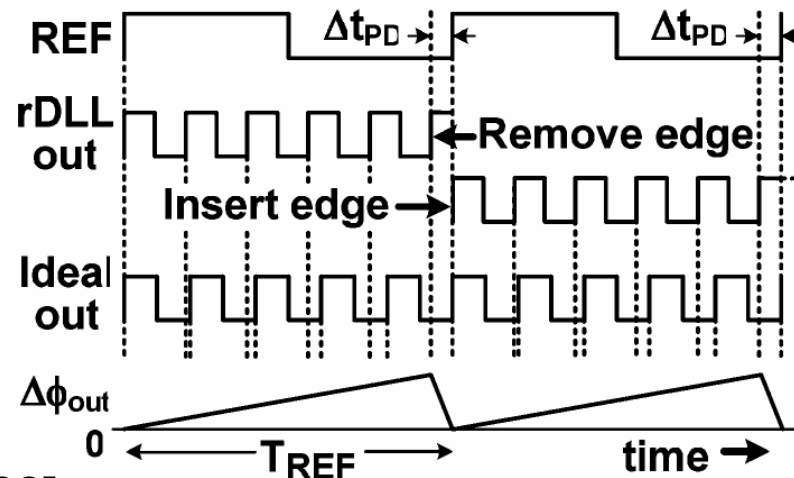
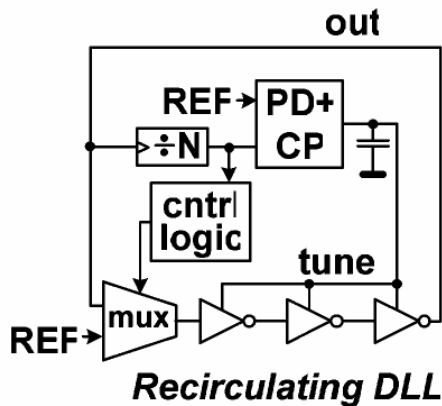
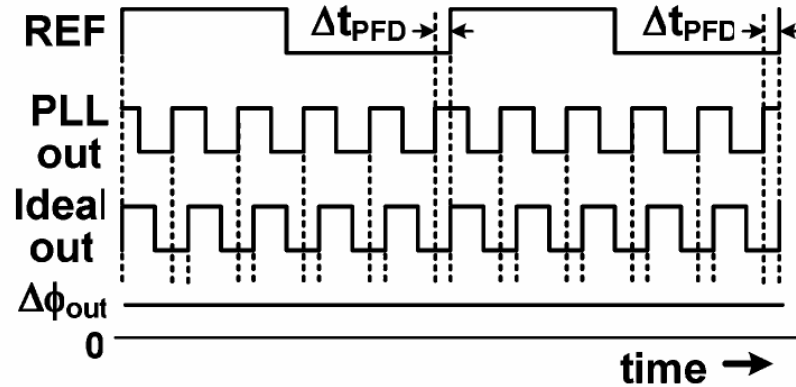
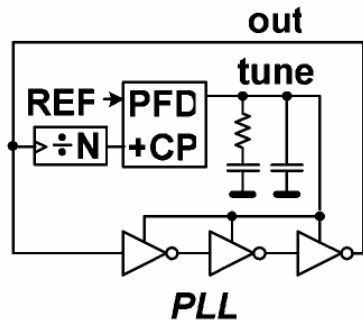
- Timing of the select logic is critical at very high frequency
- If the select signal is too slow, the clock edge experiences a distortion and therefore a phase error in that cycle
- Select signal should have fast transitions



[Farjad-Rad, JSSC'02]

MDLL Compared to PLL

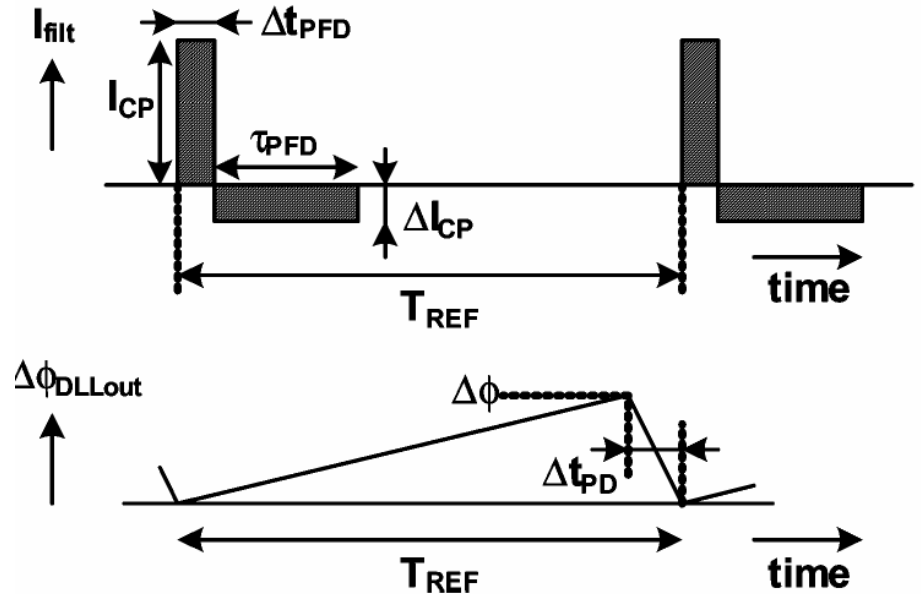
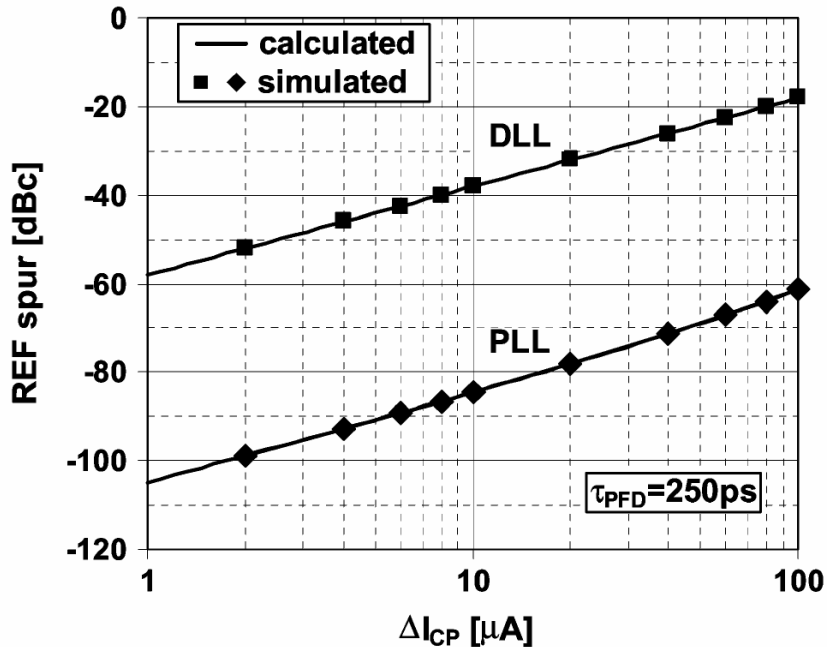
- In case of static phase offset



[Gierkink, JSSC'08]

MDLL Compared to PLL

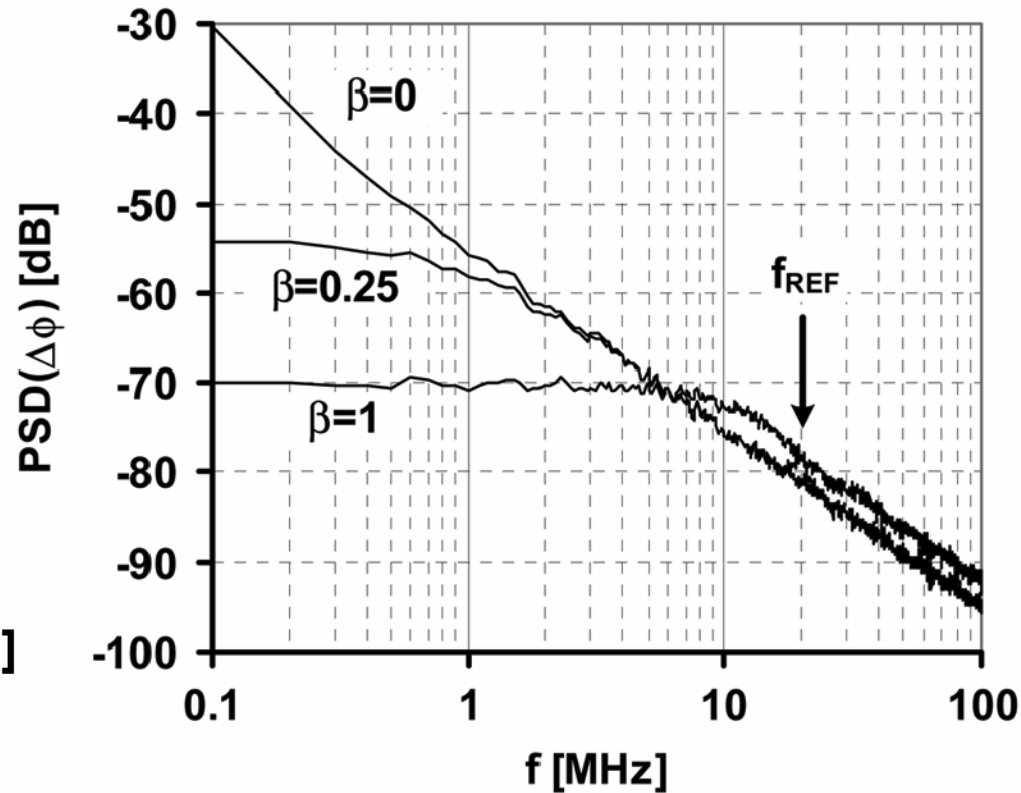
- Reference spur
 - PLL: Periodic ripple on V_{ctrl} due to CP mismatch
 - DLL: Phase offset itself



[Gierkink, JSSC'08]

Phase Noise of MDLL

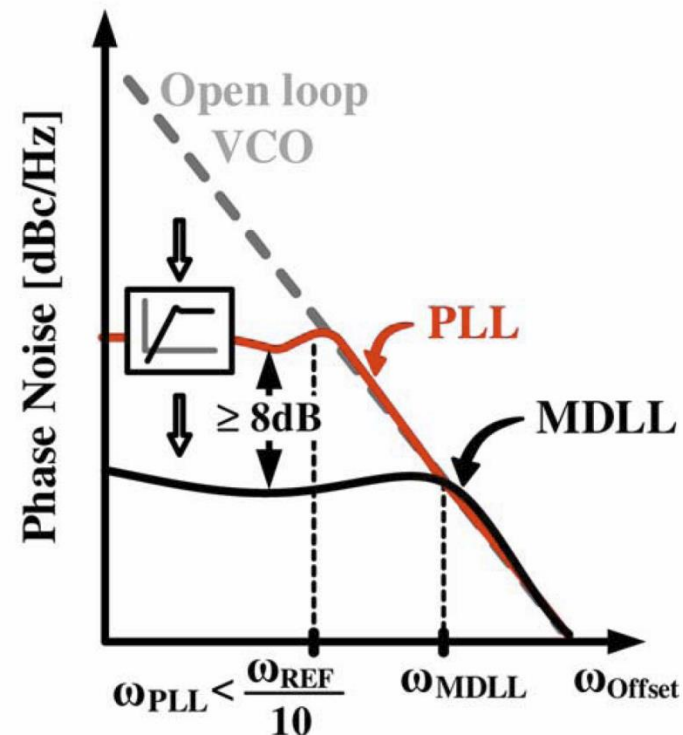
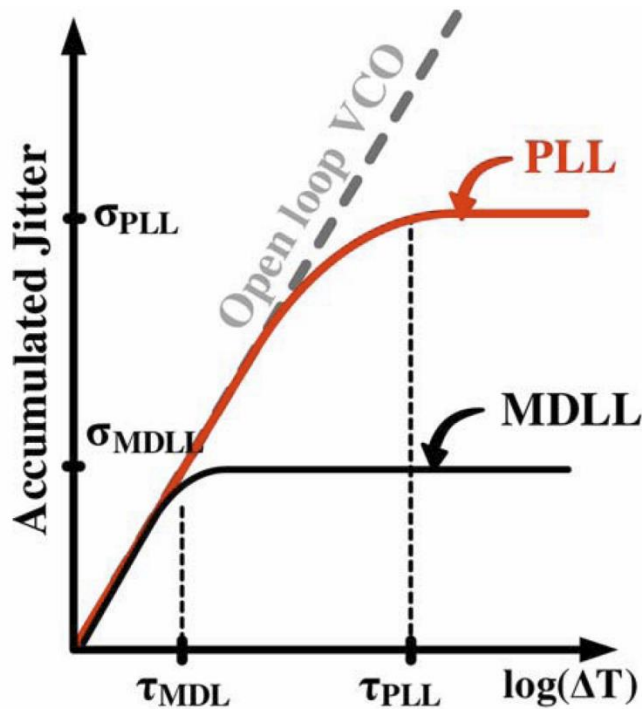
- Realignment strength β
- $\beta=0$: no phase realignment, $\beta=1$: full phase realignment



[Gierkink, JSSC'08]

MDLL Phase Noise Compared to PLL

- PLL bandwidth is at most $F_{REF}/10$
- MDLL bandwidth is at least 2.5x the PLL bandwidth



[Elshazly, JSSC'13]

Delay-Locked Loops

5.8 Duty Cycle Correction

Deog-Kyoon Jeong
dkjeong@snu.ac.kr

October 27, 2020

DCC Circuits

- **Level Type**
 - High time and low time are compared
- **Edge Type**
 - Rising edge and falling edge are compared

DCC Level Type I (US7,705,647)

- Rising and falling edges are adjusted in separate paths
- Wide-range operation possible

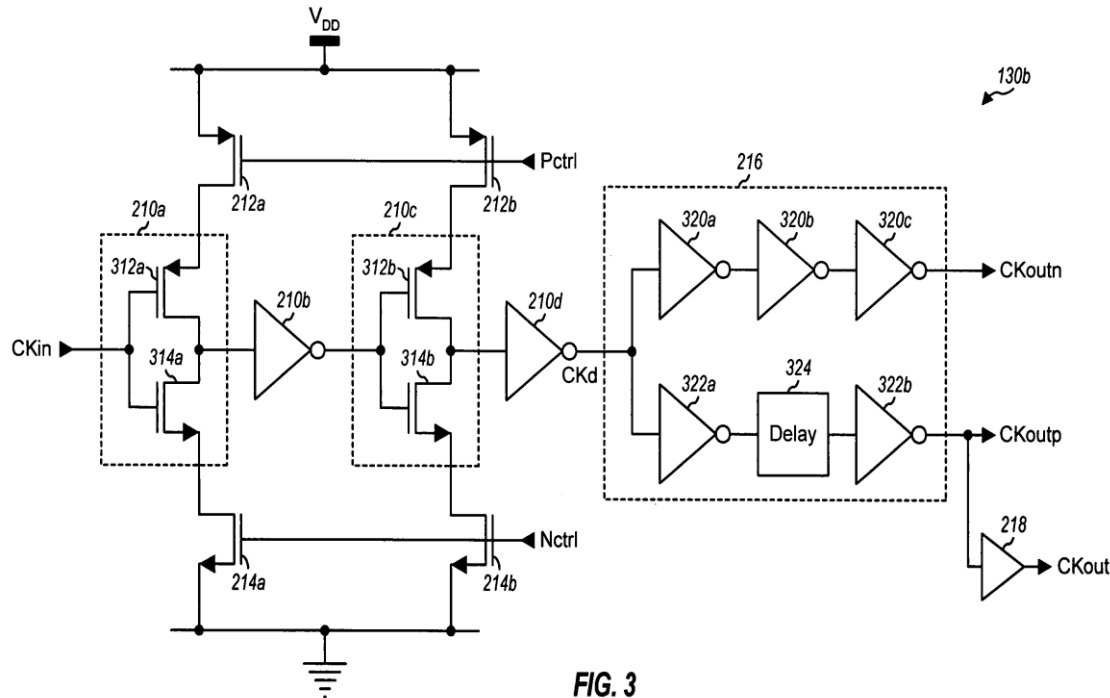
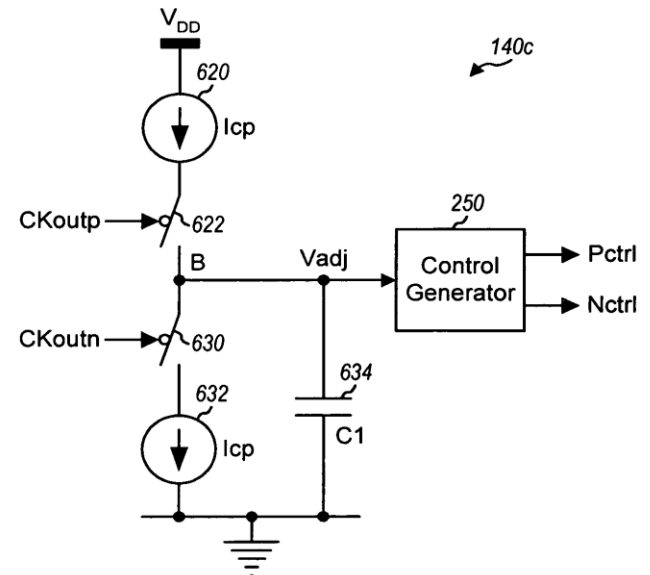
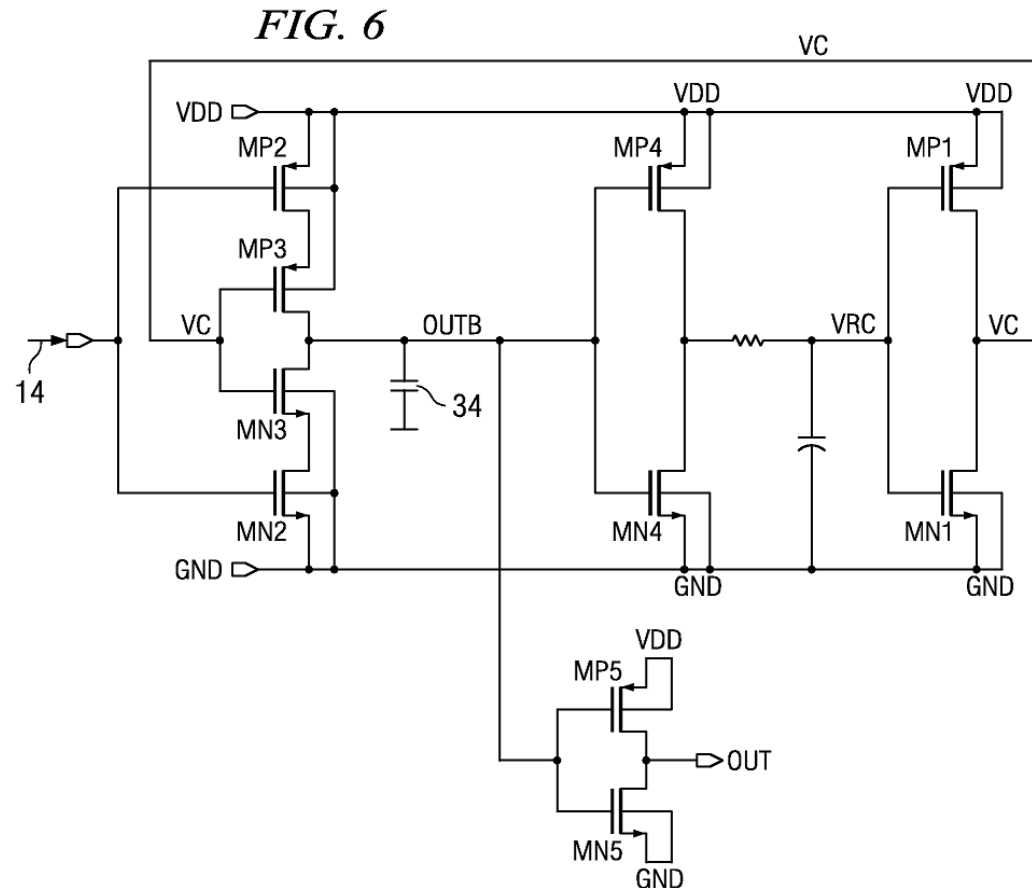


FIG. 3



Level Type – II

- High and low levels are averaged
- No charge pump



Level Type -III

- High loop gain is required unless charge pump is used.
- Otherwise offset remains.

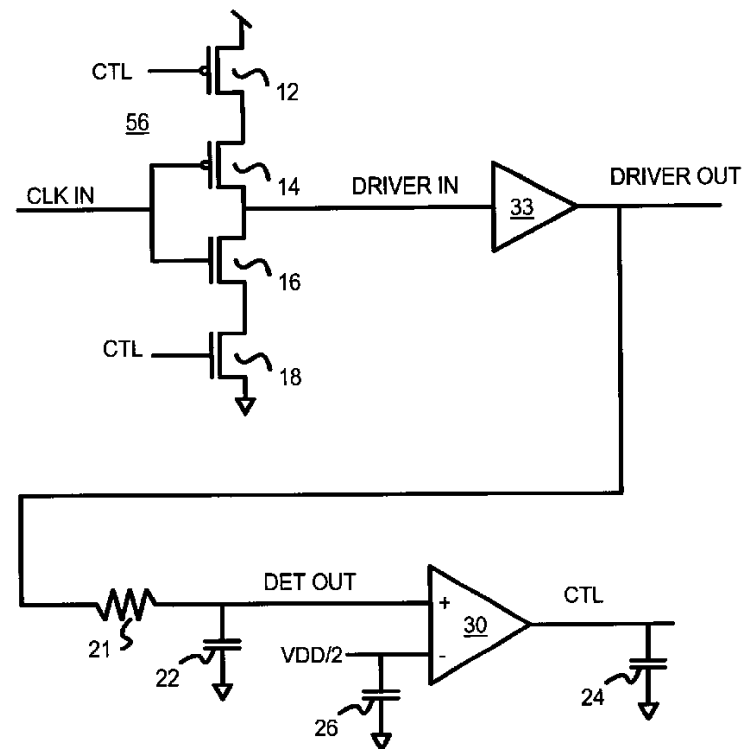
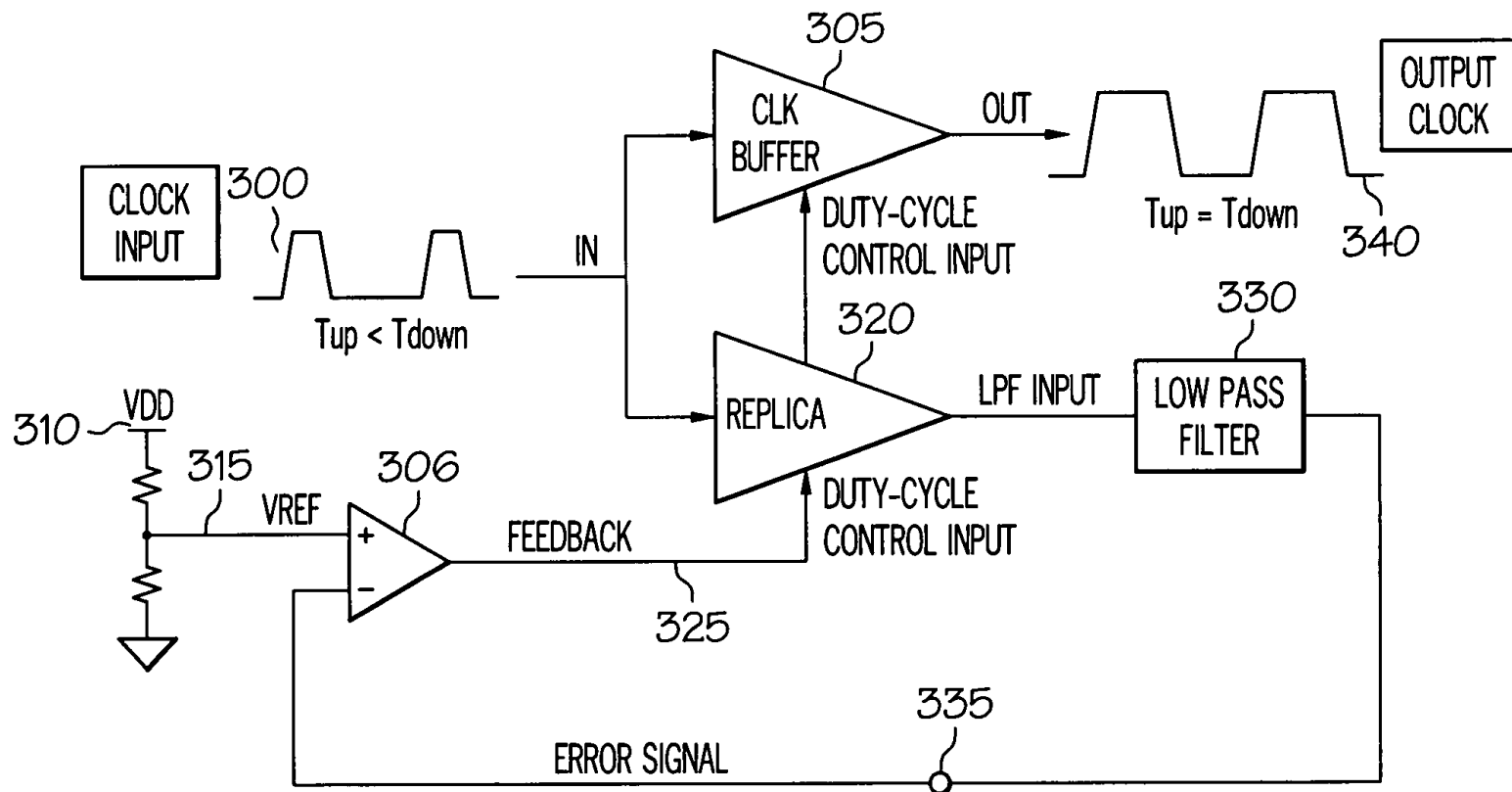


FIG. 2

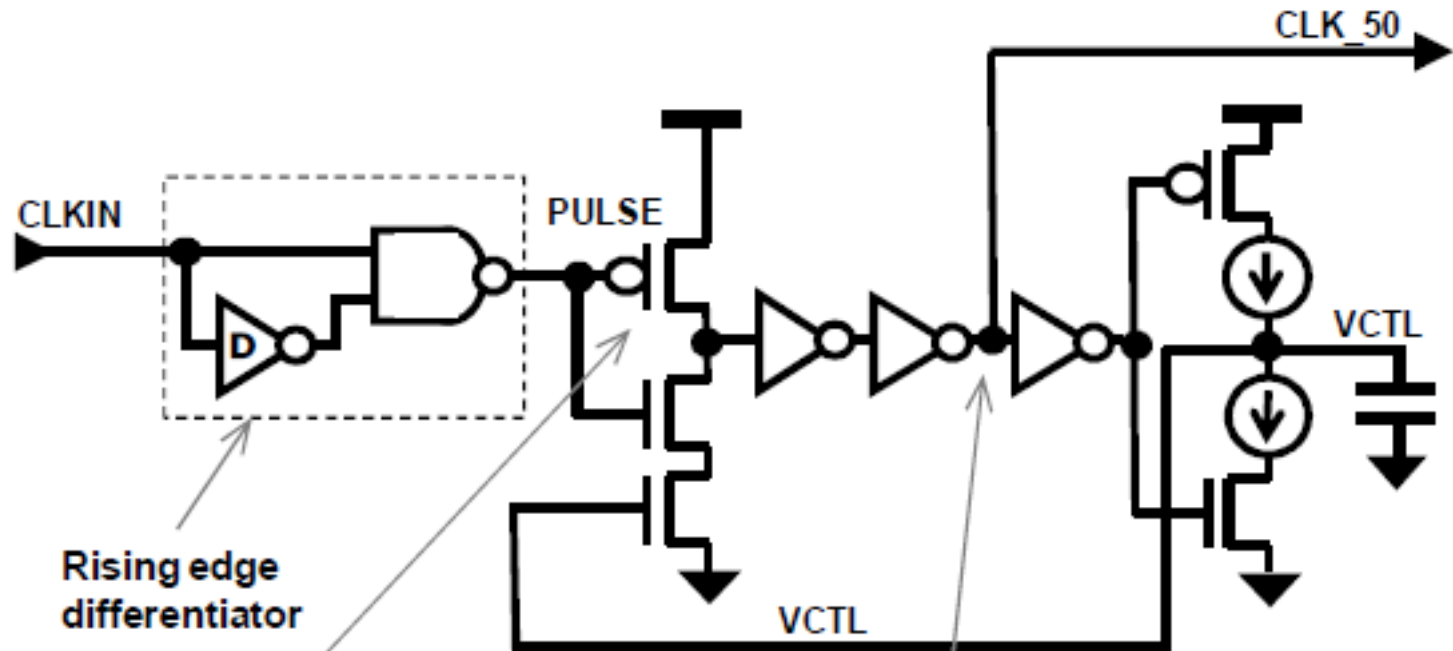
Level Type -IV

- Replica circuit for reduced loading



Synchronous DCC

- Falling edge is modulated.



No need to control the rising edge delay

The clock is taken to output one stage earlier to compensate for the inverting stage at the input.