Delayed Locked Loop Design Issues

Chulwoo Kim

ckim@korea.ac.kr

Advanced Integrated Systems Lab

Korea University

Outline

- Introduction
- DLL operation and control theory
- DLL building blocks
- DLL design issues
- Multiplying DLL

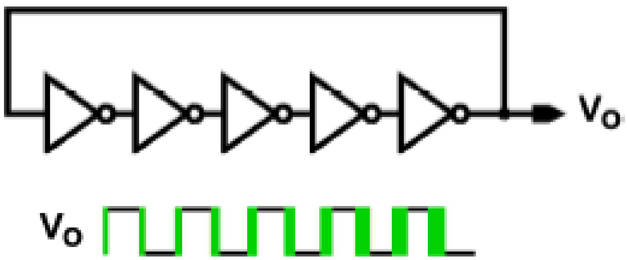
DLL vs PLL

- **PLL**
  - − VCO
    - ✓ jitter accumulation
  - − higher order system
    - ✓ can be unstable
  - − slow locking time
  - − hard to integrate LF
  - − hard to design
  - + less ref. signal dependent
  - + freq. multiplication
  - + no limited locking range

- **DLL**
  - + VCDL
    - ✓ no jitter accumulation
  - + 1st order system
    - ✓ always stable
  - + fast locking time
  - + easy to integrate LF
  - + easier to design
  - - ref. signal dependent
  - - no freq. multiplication
  - - limited locking range

Jitter Accumulation Comparison

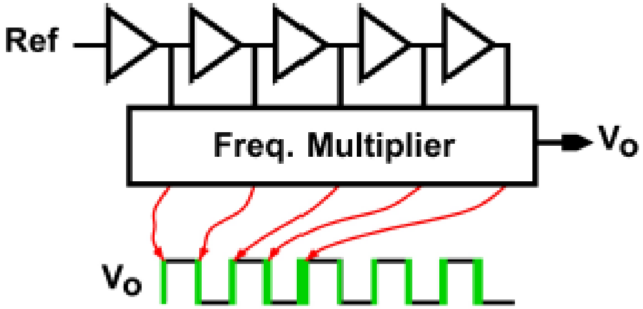
- PLL-based clock generator



Closed loop

- jitter accumulation

- DLL-based clock generator

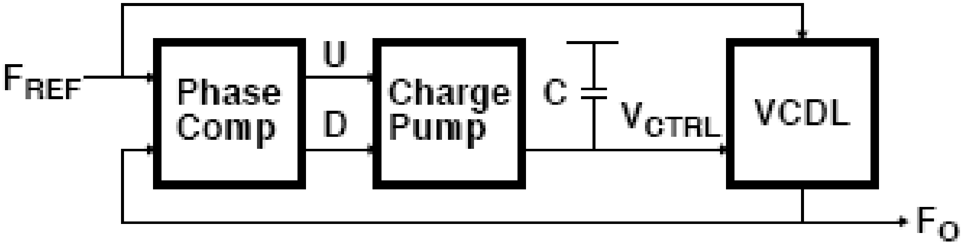


Open loop

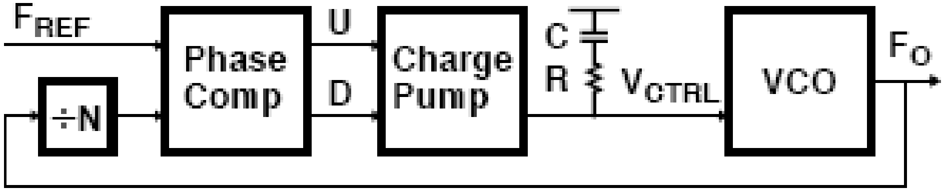
- No jitter accumulation

Basic DLL Architectures

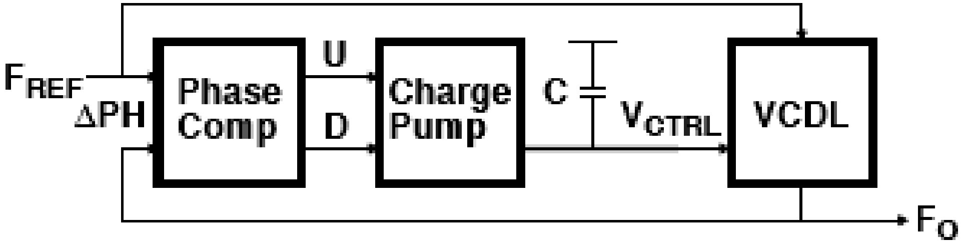
- **Delay-locked Loop (Delay line based first order PLL)**

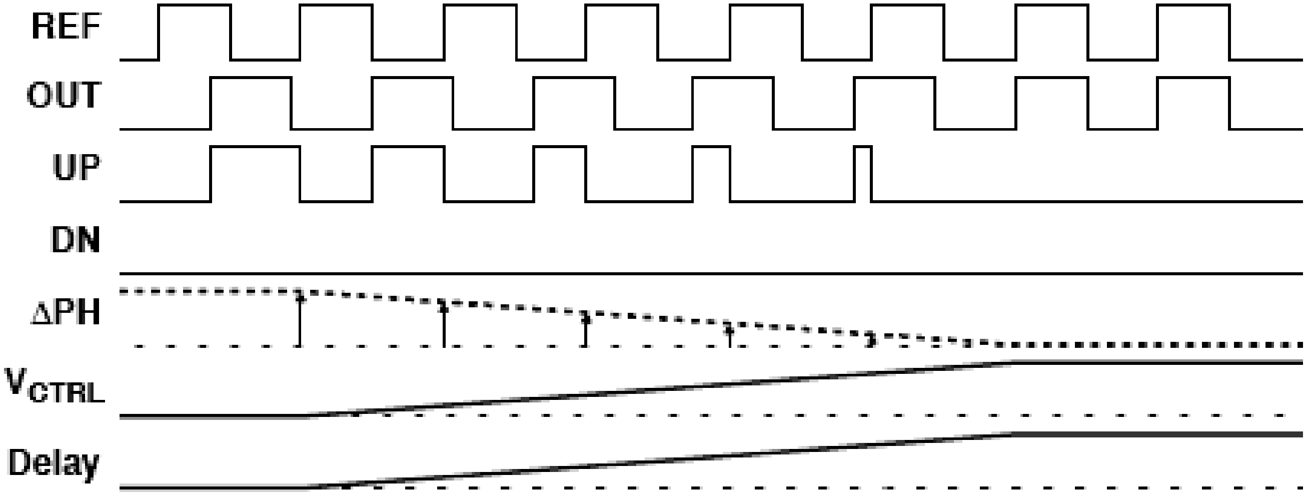


- **Phase-Locked Loop (VCO based second order)**

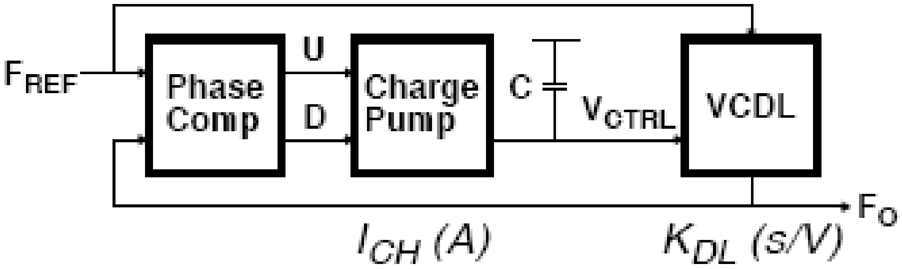


DLL Locking Process





Frequency Response



- **Open loop response**

$$D_O(s) = \left(\frac{D_I(s)}{T_{REF}} \cdot I_{CH}\right) \cdot \frac{1}{s \cdot C} \cdot K_{DL}$$

$$\frac{D_O(s)}{D_I(s)} = \frac{1}{s \cdot C} \cdot I_{CH} \cdot K_{DL} \cdot F_{REF} = H(s)$$

Frequency Response(cnt'd)

- **Closed loop response**

$$D_O(s) = H(s)\cdot(D_I(s) - D_O(s))$$

$$\frac{D_O(s)}{D_I(s)} = \frac{1}{1+\frac{1}{H(s)}}$$

$$\frac{D_O(s)}{D_I(s)} = \frac{1}{1+\frac{s\cdot C}{I_{CH}\cdot K_{DL}\cdot F_{REF}}}$$

$$\frac{D_O(s)}{D_I(s)} = \frac{1}{1+s/\omega_N}$$

where loop bandwidth is

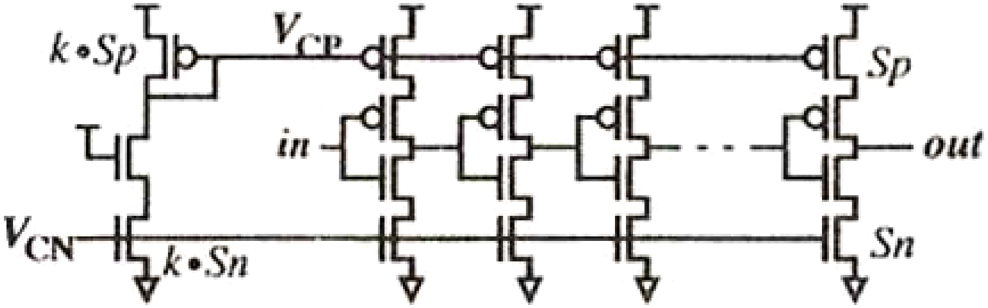
$$\omega_N = I_{CH}\cdot K_{DL}\cdot F_{REF}/C$$

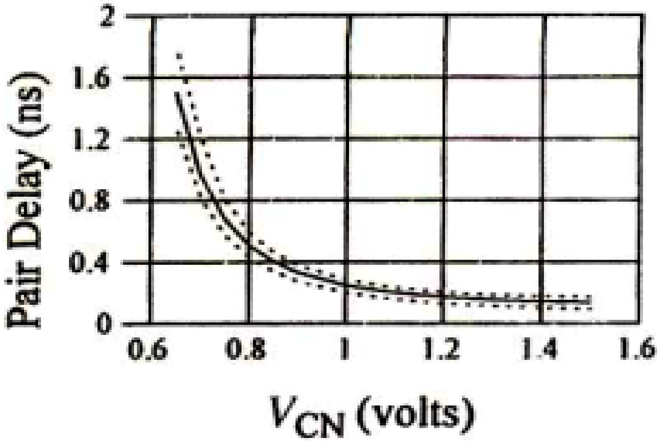
Delay Cell

- Single-ended delay cell
  - Simple
  - Dynamic power only (no static current)
- Differential delay cell
  - Complex biasing
  - Static power consumption
  - Immune to supply noise and thus smaller jitter
- Variables for delay control
  - Current
  - Capacitance
  - Resistance
  - Voltage swing
- Fine delay generation
  - Phase interpolation
  - Vernier delay line

Single-Ended Delay Cell

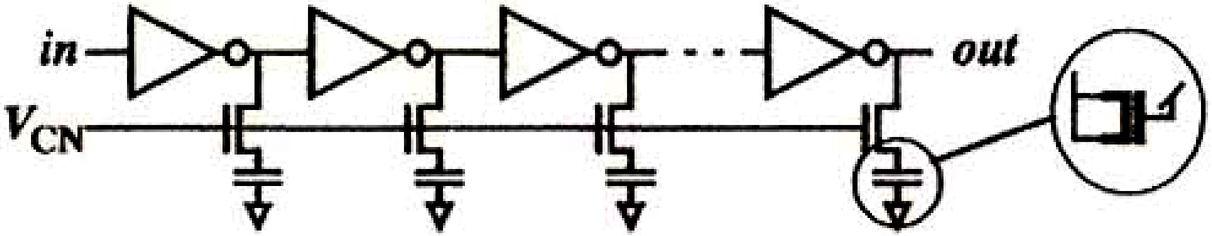
- **Current-starved inverter delay line**

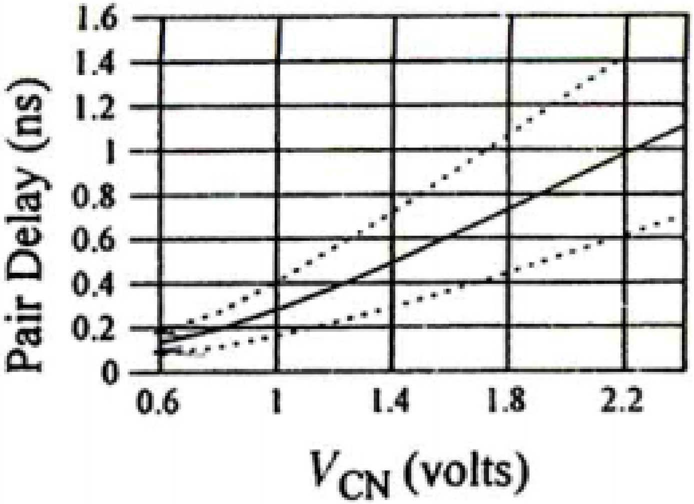




Single-Ended Delay Cell(cnt’d)

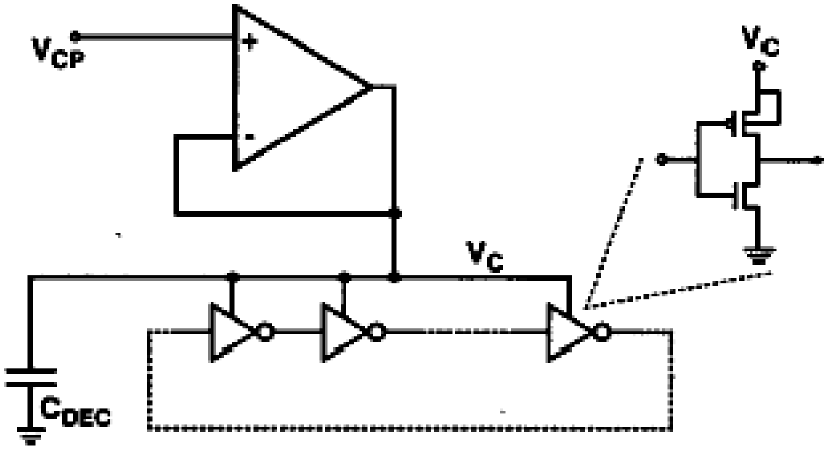
- **Capacitor-loaded inverter delay line**





Single-Ended Delay Cell(cnt'd)

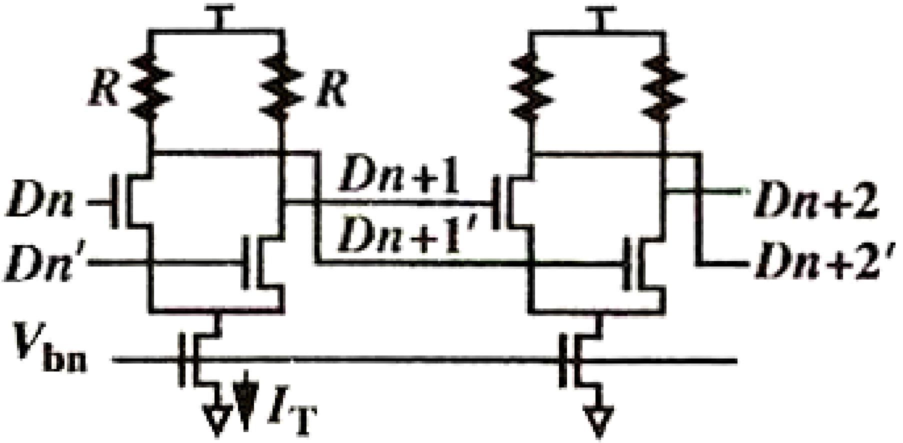
- **Inverters with regulated supply voltage**



[S. Sidiropoulos, SOVC00]

Differential Delay Cell

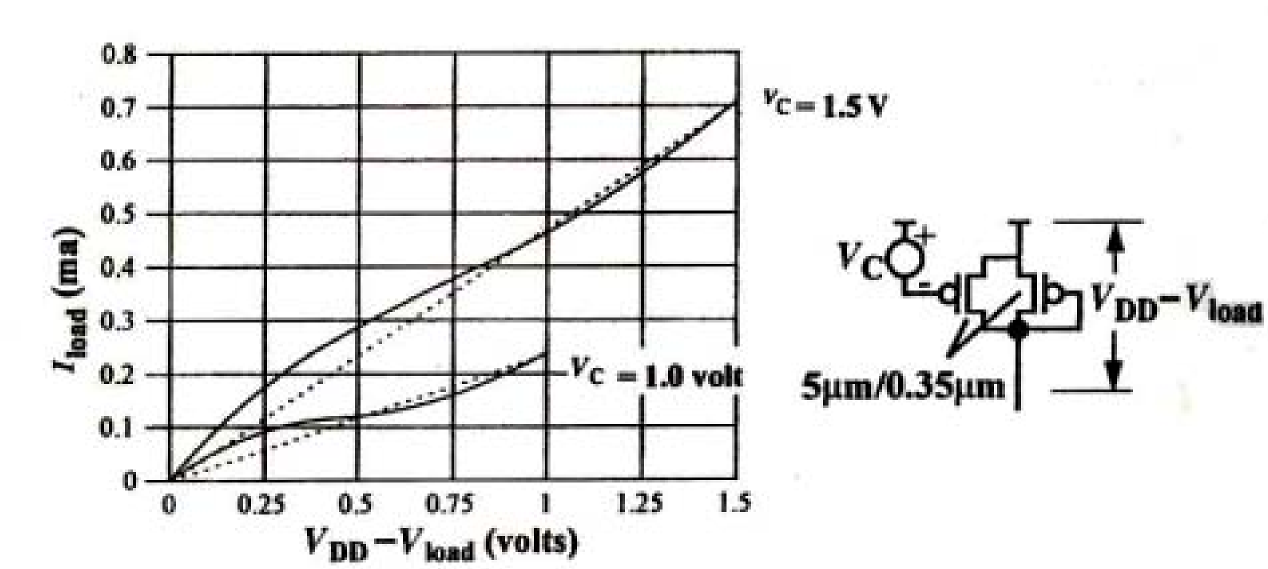
- Differential delay element with resistive loads



  - High power supply rejection ratio
  - Requirements
    - ✓ Adjustable loads to control the delay and resistive loads to reject power supply noise

Differential Delay Cell(cnt'd)

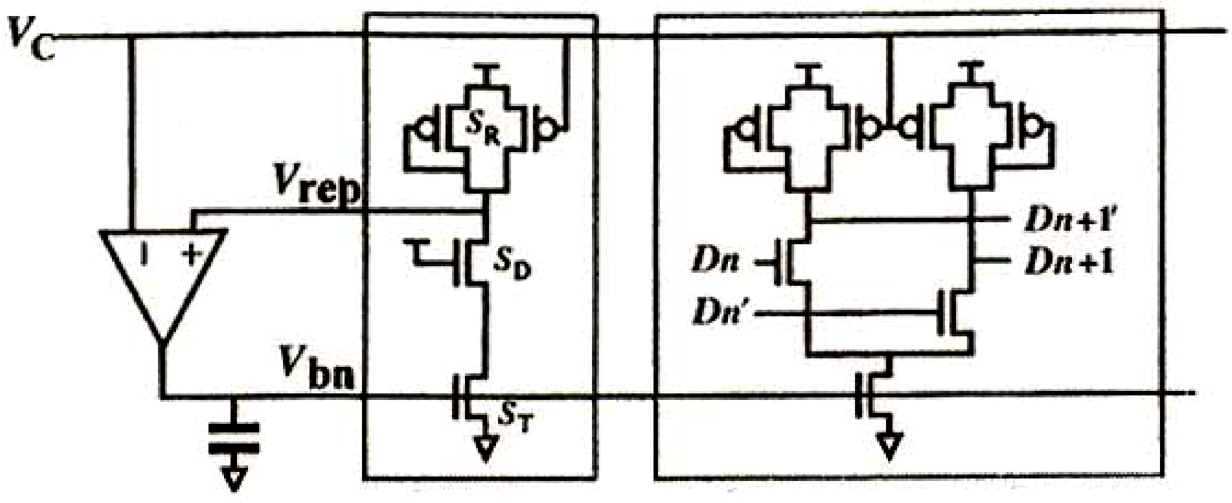
- Voltage-controlled two-element PFET "Resistor"



  - Adjustable load : I_{load}=$B_p(V_c$-$V_{tp})^2$
  - Resistive load : S-shaped, nearly resistive

Differential Delay Cell(cnt'd)

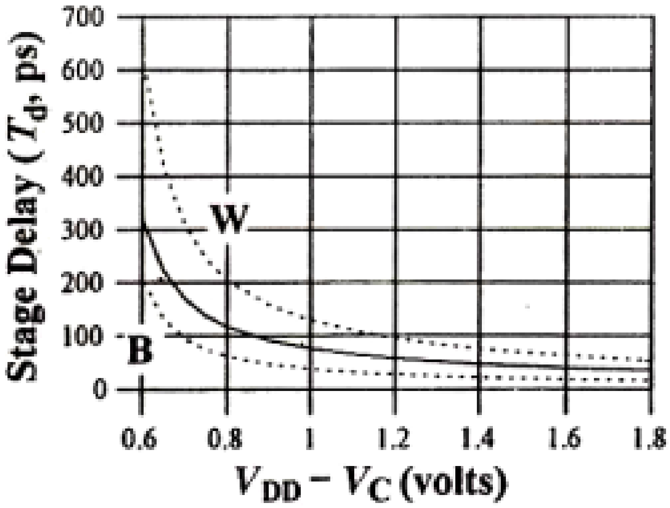
- **Replica-biased differential delay line circuitry**

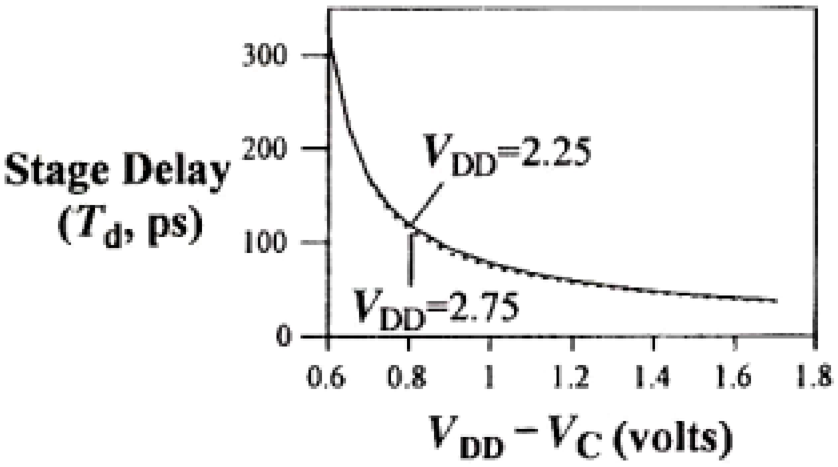


  - **The low end of the signal swing can be set by controlling the bias current with a replica bias circuit**

Differential Delay Cell(cnt'd)

- **Replica-biased delay line**

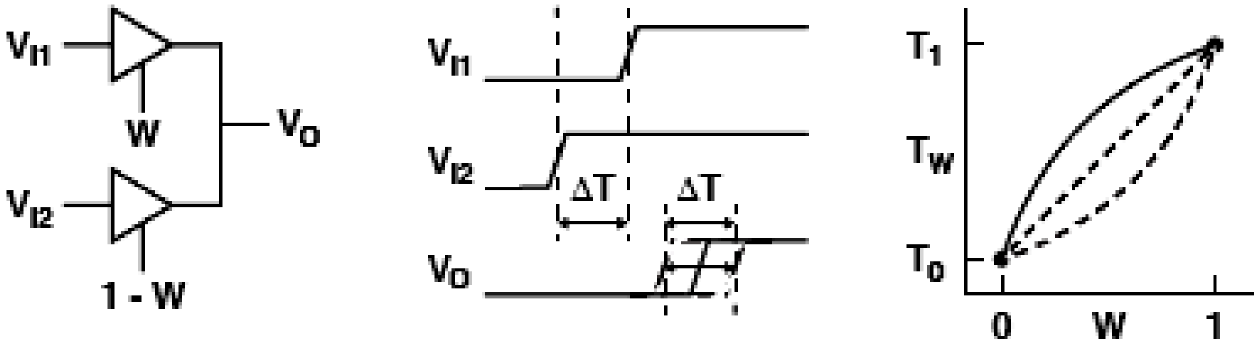




(a) Delay adjustment range for replica-biased delay element

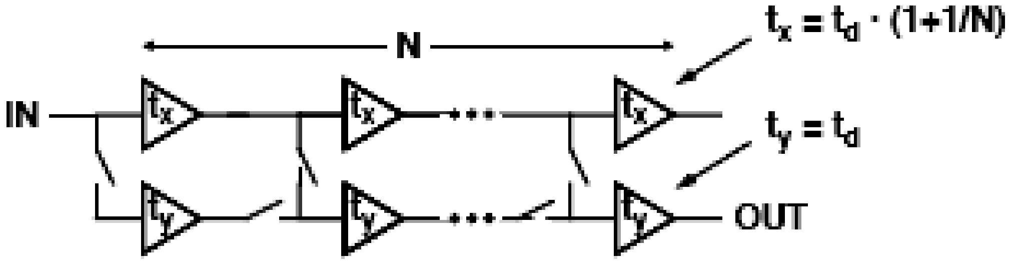
(b) Static supply sensitivity for replica-biased delay element

Phase Interpolation



- **Can interpolate between two edges through a weighted sum**
  - **Control over delay is guaranteed to be monotonic, but not necessarily linear**
    - ✓ **Resolution can be arbitrarily high**
    - ✓ **Precision is limited by linearity**

Delay Line Vernier

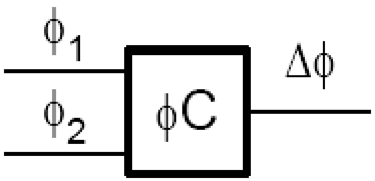


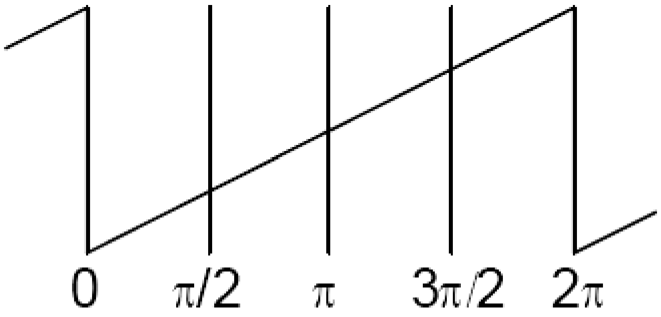
$$Delay = n \cdot t_d \cdot (1 + 1/N) + (N - n) \cdot t_d = (N + n/N) \cdot t_d$$

- **Can use two delay lines with switches to use part of one and remainder of the other with fractionally larger delays** (1 + 1/*N*)
  - **Delay resolution is a buffer delay** / *N*
  - **Relative precision is limited by control over** *tx* / *ty*

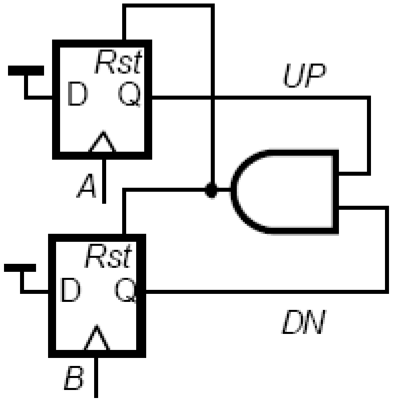
Phase Detector

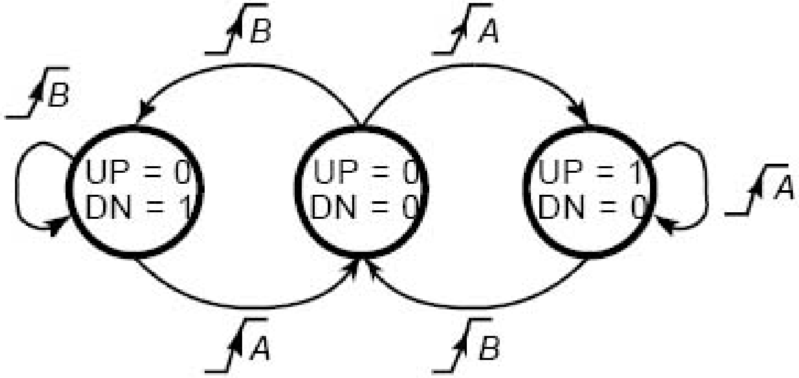
- **Output describes phase difference between two inputs**
  - may be analog or digital
  - may linearly cover a wide range, or just a narrow phase difference
  - “Dead zone” may occur

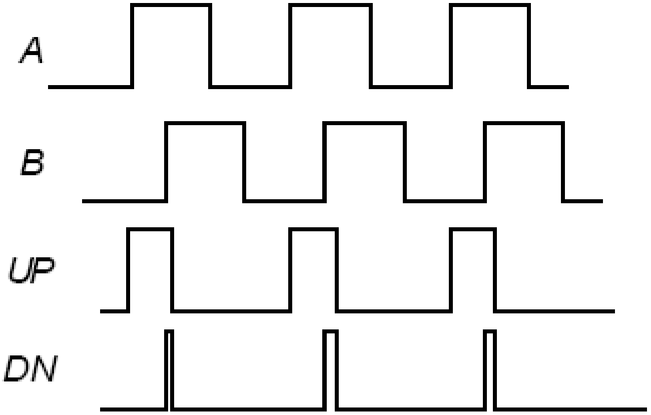


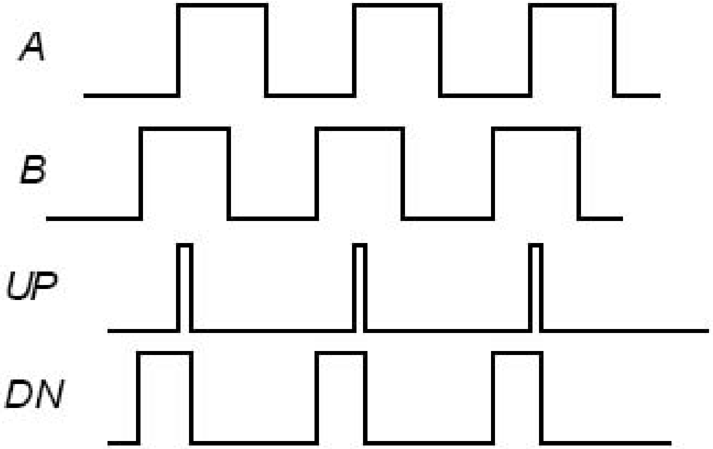


Phase-Frequency Detector



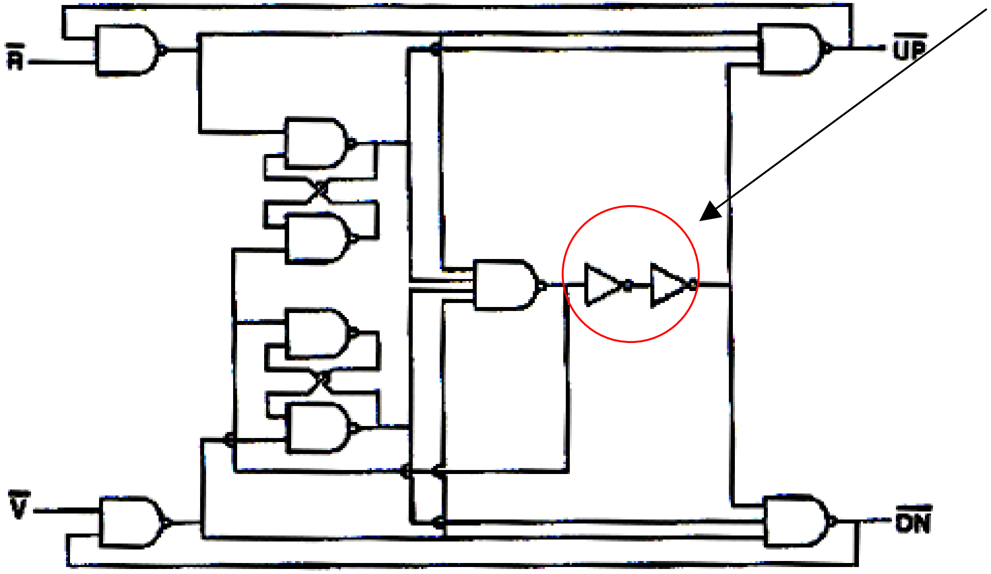




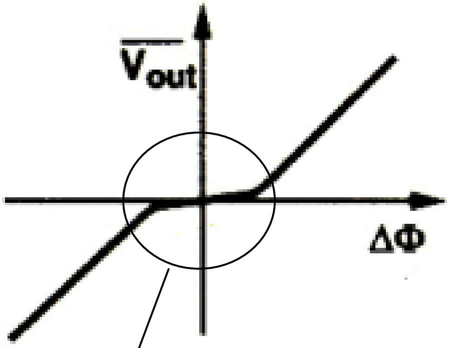


Dead-Zone in PFD

- "Dead-zone" occurs when the loop doesn't respond to small phase errors - e.g. 10 ps phase error at PFD inputs:
  - Solution: delay reset to guarantee min. pulse width (typically > 100 ps)



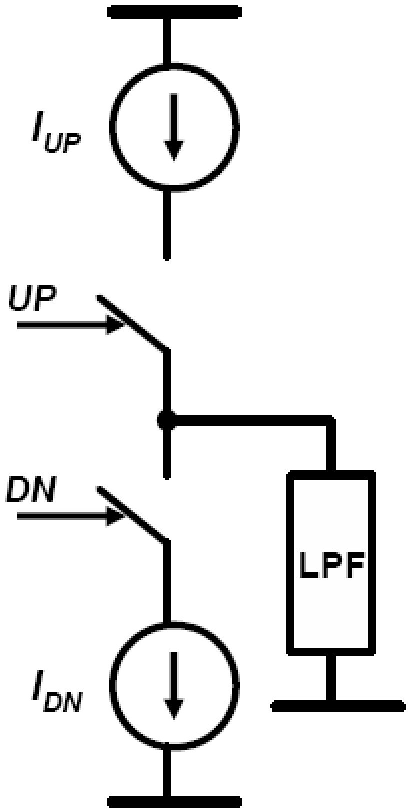
Remove dead zone



dead zone

Charge Pump

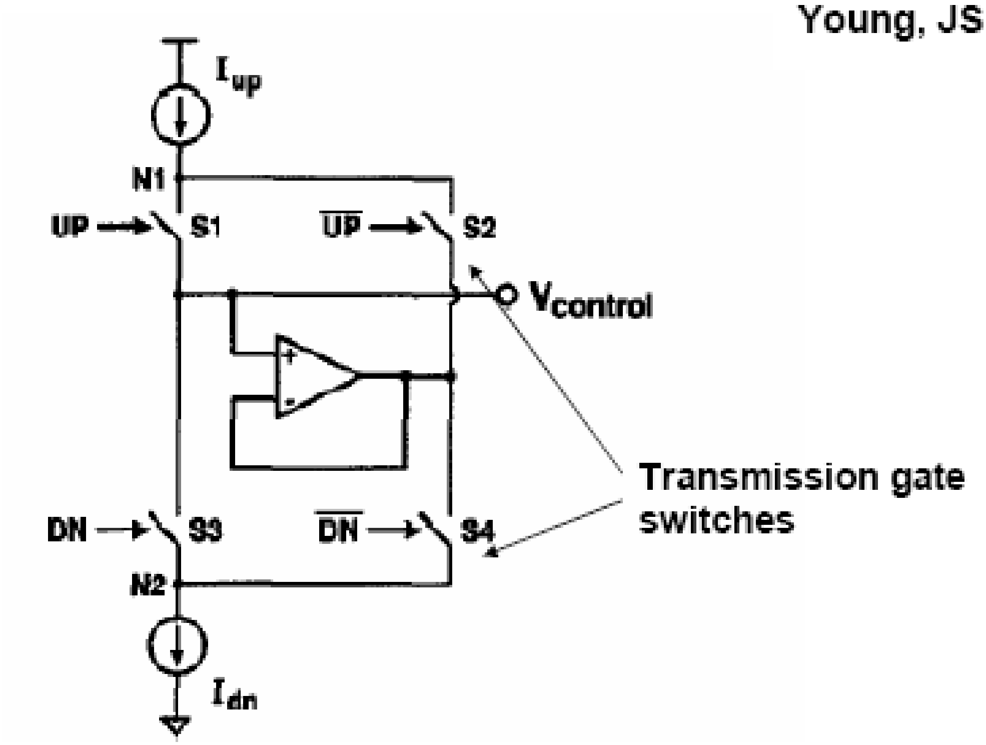
- Converts PFD digital *UP*/*DN* signals into charge
- Charge is proportional to duration of *UP*/*DN* signals
- $Q_{cp} = I_{UP}*t_{UP} - I_{DN}*t_{DN}$
- The LPF converts integrates currents
- Charge pump requirements:
  - Match currents *IUP* and *IDN*
  - Reduce control voltage coupling
  - Supply noise rejection, PVT insensitivity (Simple or bandgap biased)



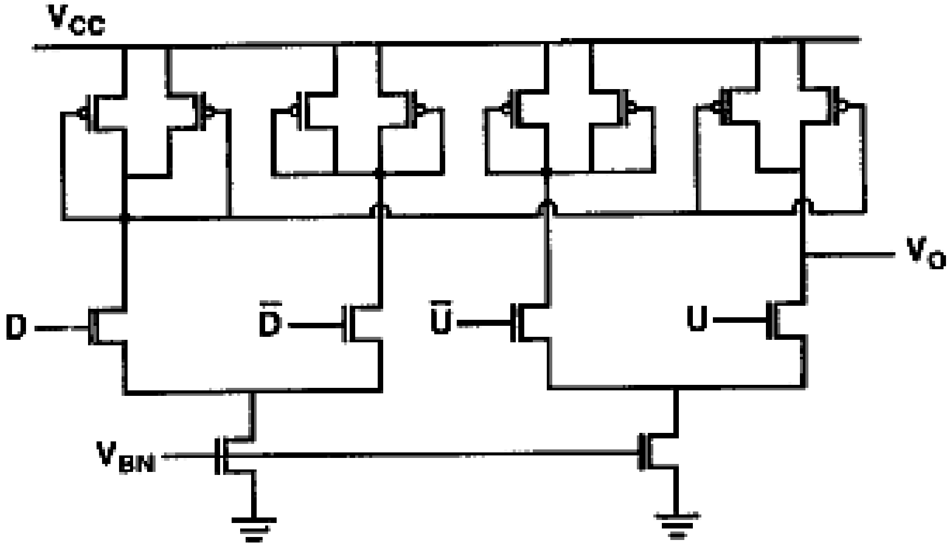
Charge Pump: Better Switches

- Unity-gain buffer controls the voltage over switches
- Current mirrored into I_{up}/I_{dn}

Young, JSSC 12/92

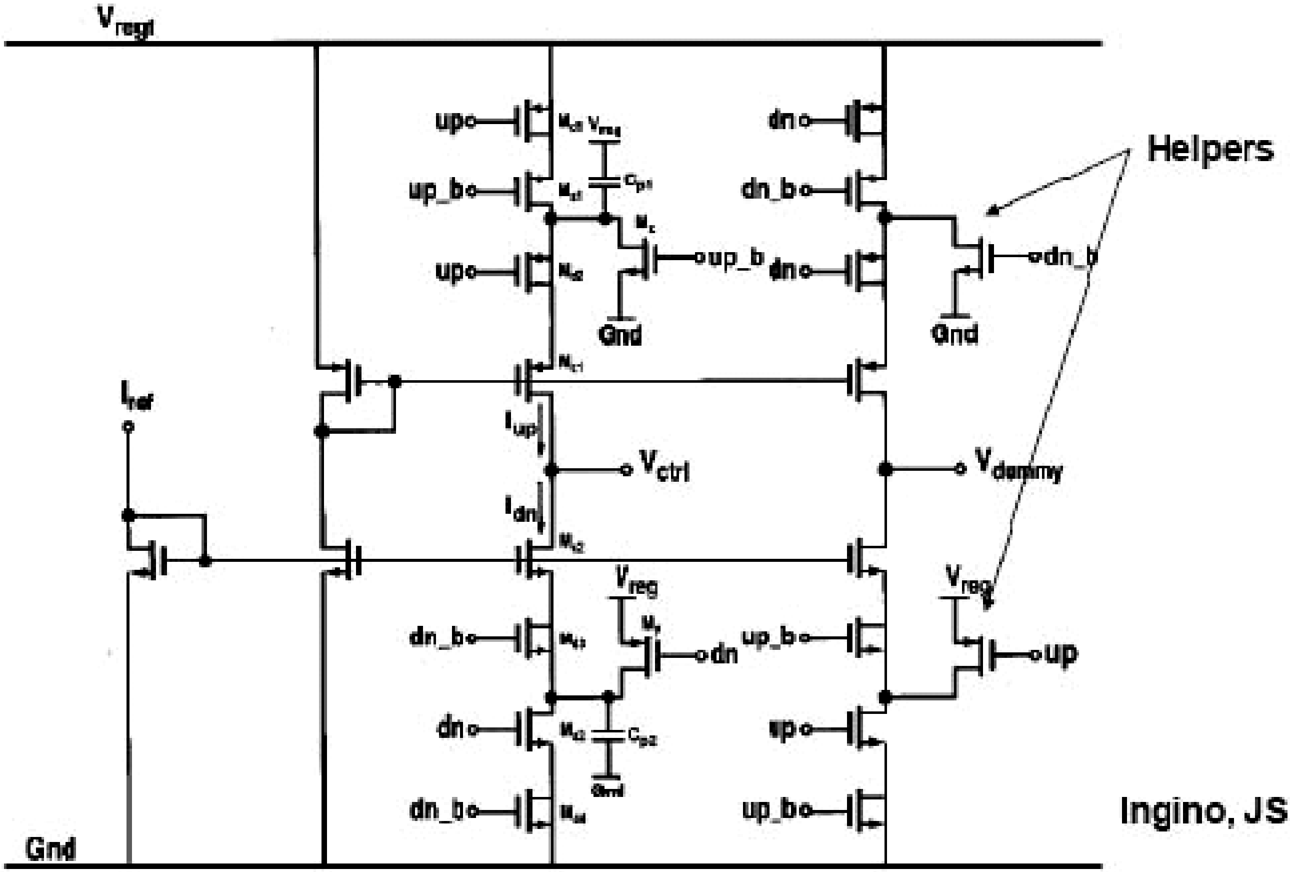


Charge Pump: Zero-Offset



- Up and down nodes track with each other thanks to the self-bias scheme.

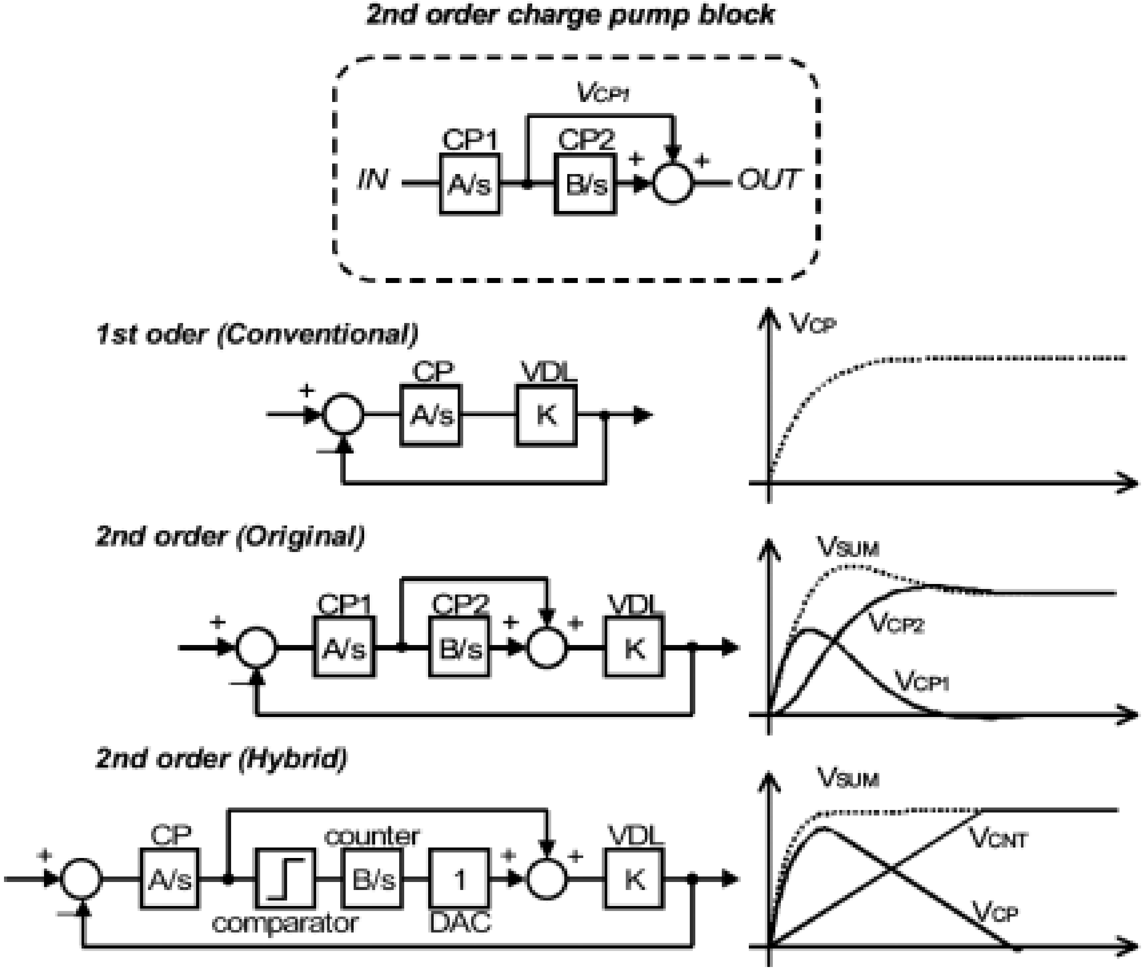
Charge Pump : Reversed Switches



Ingino, JSSC 11/01

2nd Order Charge-Pump Scheme : Mismatch Cancellation

[K. Kim, ISSCC 04]



Design Issues

- Bandwidth
- Limited lock range
- Lock in time
- Static phase offset
- Power dissipation limits
- Area limits
- Peak output jitter

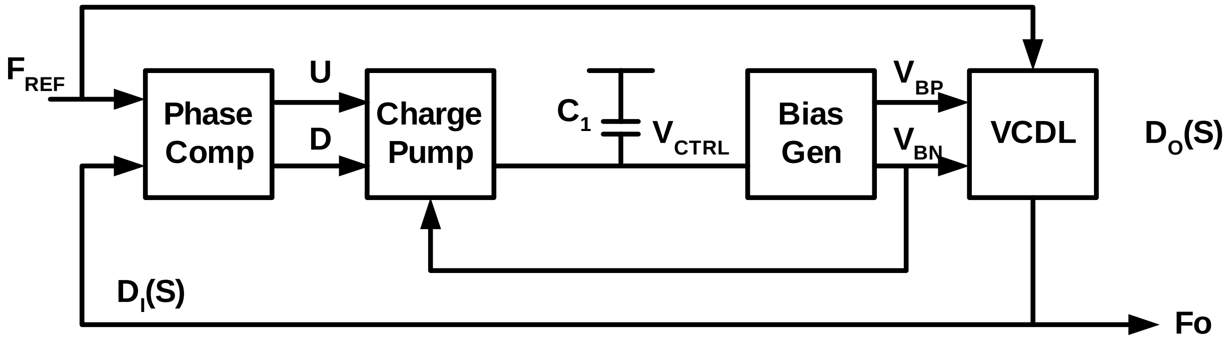
Bandwidth

- **Bandwidth**
  - **A wider loop bandwidth**
    - ✓ **Fast acquisition time but degraded jitter performance**
  - $$\frac{\omega_N}{F_{REF}} = \frac{I_{CH} \cdot K_{DL}}{C} \leq \frac{\pi}{5}$$ [A. Chandrakasan, IEEE Press, 2001]
  - **I_{CH}, K_{DL}, and C are process technology dependent.**
  - **According to the design target, the loop bandwidth varies.**

Adaptive Bandwidth

- **Self-biased DLL**

[J. Maneatis,JSSC 96]



Current and gain of VCDL

$$I_D = \frac{k}{2} \bullet (V_{CTRL} - V_T)^2 \quad K_{DL} = \frac{C_B}{4 \cdot I_D}$$

Current of Charge pump

$$I_{CH} = x \bullet (2 \bullet I_D)$$

$$\frac{\omega_N}{\omega_{REF}} = \frac{1}{\omega_{REF}} \cdot I_{CH} \cdot K_{DL} \cdot F_{REF} \cdot \frac{1}{C_1}$$

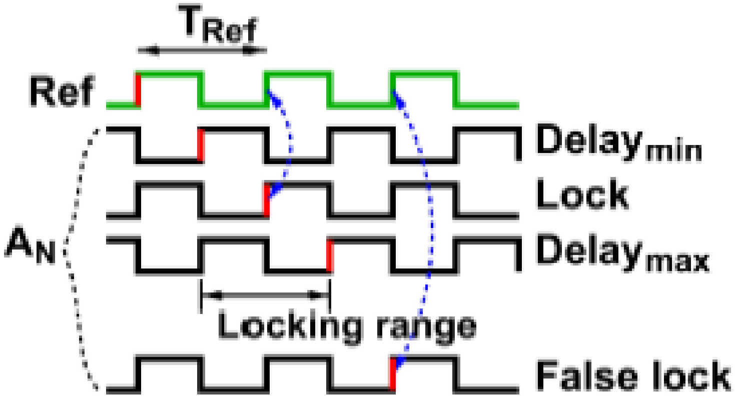
$$= \frac{1}{2\pi} \cdot I_{CH} \cdot K_{DL} \cdot F_{REF} \cdot \frac{1}{C_1}$$

$$= \frac{1}{2\pi} \cdot x \cdot (2 \cdot I_D) \cdot \frac{C_B}{4 \cdot I_D} \cdot \frac{1}{C_1}$$

$$= \frac{1}{4\pi} \cdot \frac{C_B}{C_1}$$

Locking Range

- **Locking range**



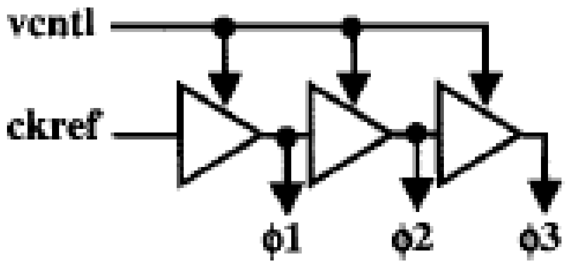
$$0.5 \times T_{CLK} < T_{VCDL,\min} < T_{CLK}$$

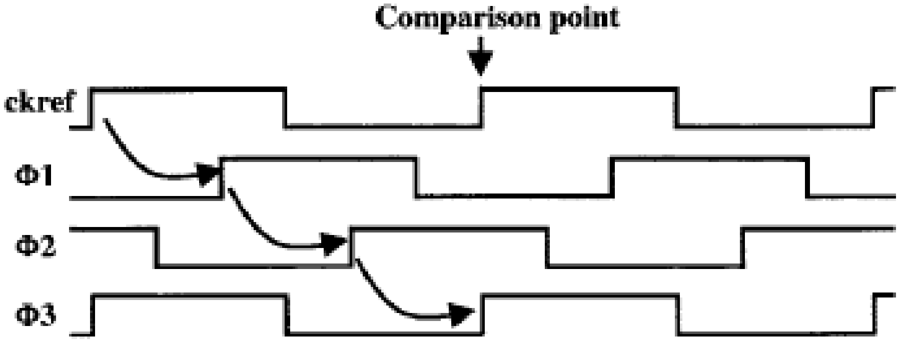
$$T_{CLK} < T_{VCDL,\max} < 1.5 \times T_{CLK}$$

$$Max(T_{VCDL,\min}, 2/3 \times T_{VCDL,\max}) < T_{CLK} < Min(2 \times T_{VCDL,\min}, T_{VCDL,\max})$$

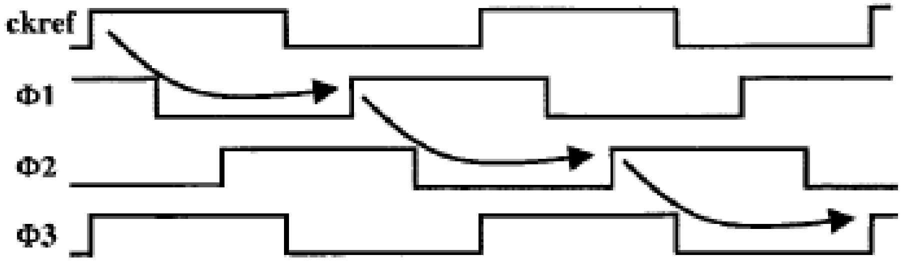
Harmonic Lock Problem

- **Correct and false locking**



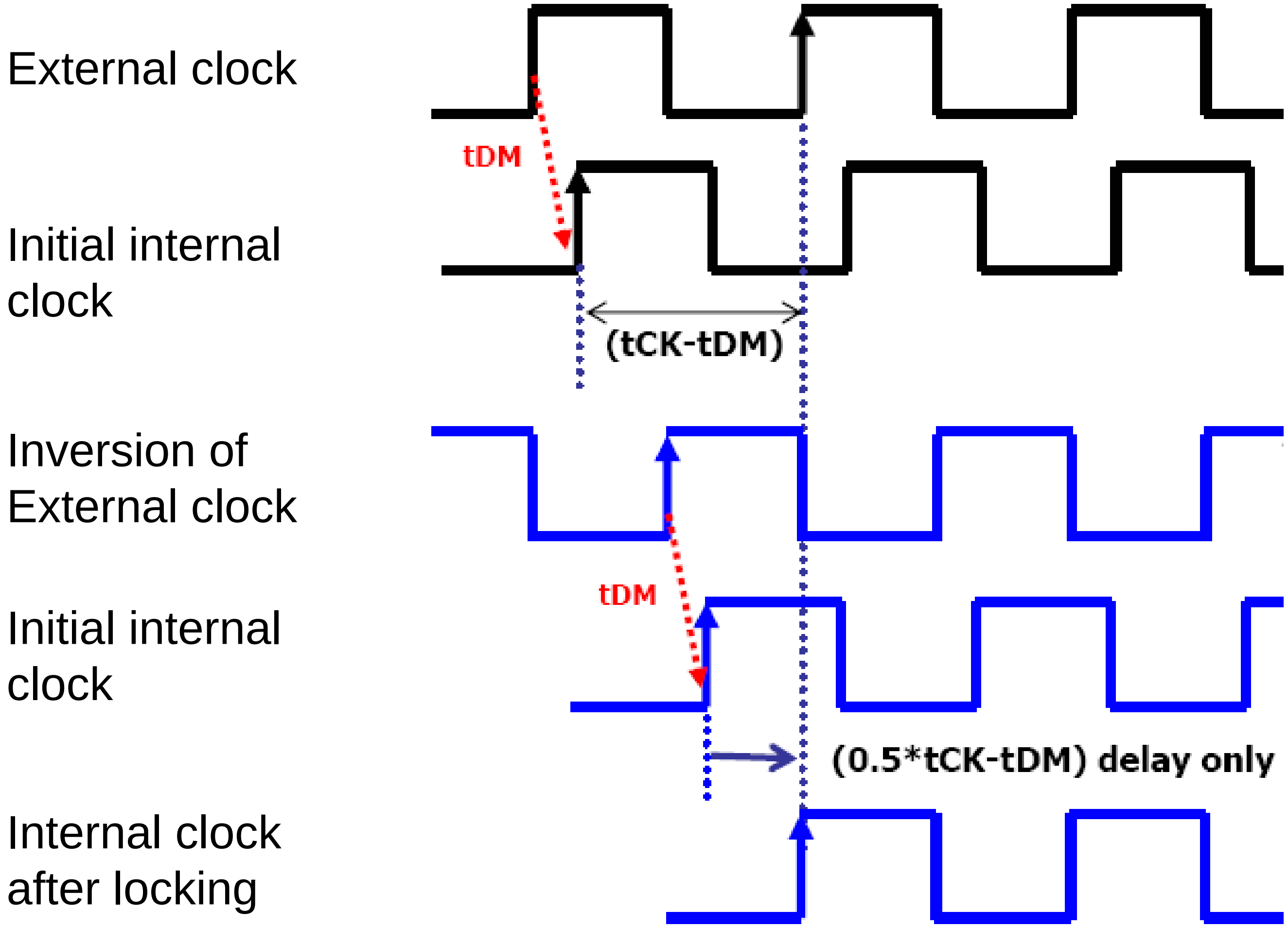


Correct locking



False locking

DLL Locking Using Inversion



External clock

Initial internal clock

Inversion of External clock

Initial internal clock

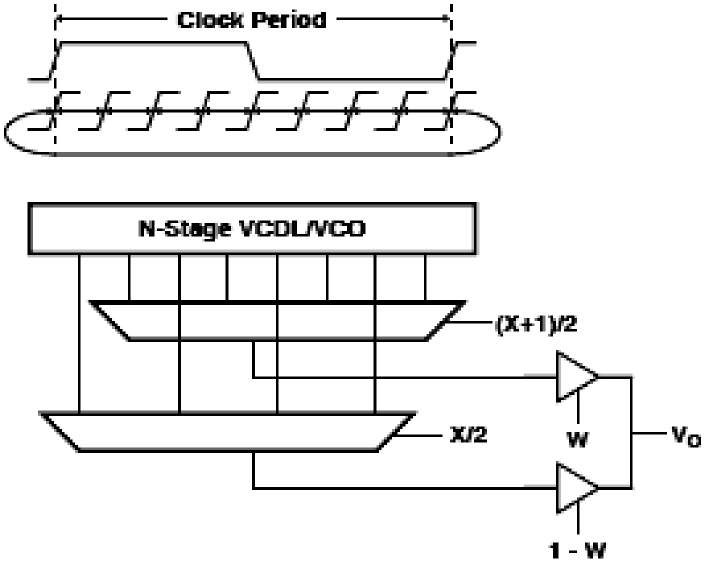
Internal clock after locking

Wide Range DLL

- Overcome the false locking problem
  - Rotating phase DLL
  - Phase detector which can detect harmonic locking
  - Initial locking starts within delay range

- Widen operating frequency
  - Using multiple phases

Rotating Phase DLL

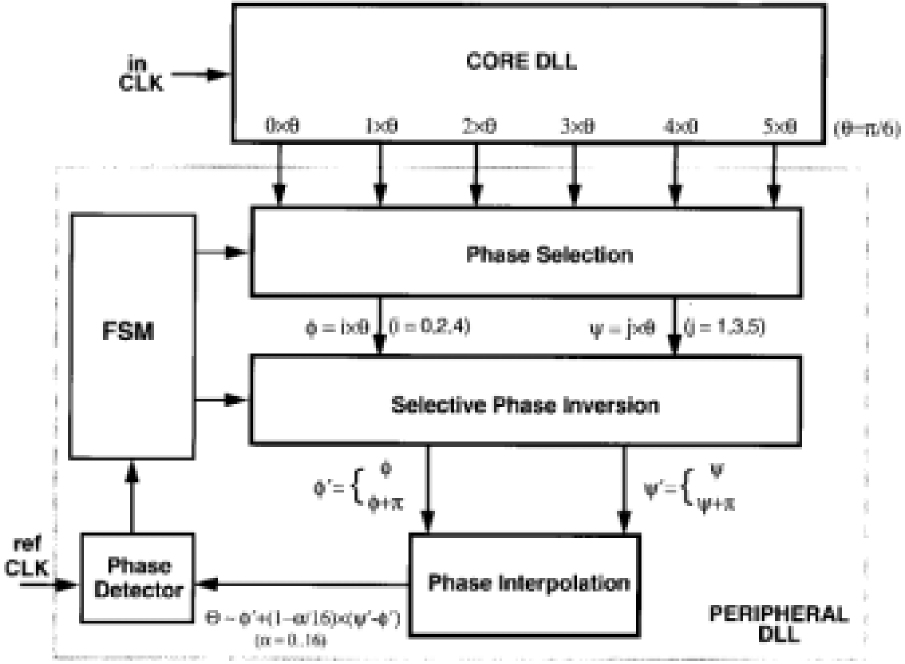


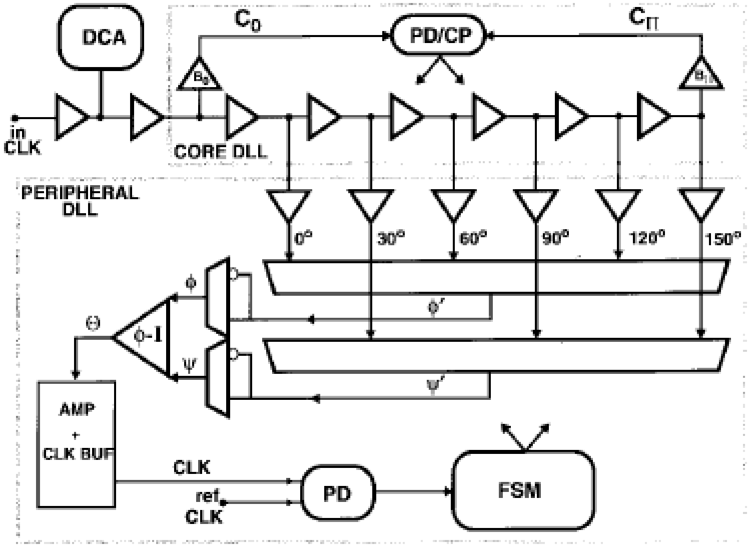
- Uses N-stage VCDL or VCO phase-locked to clock period as timing reference to supply output phases that are uniformly distributed over clock period
- Selects or interpolates output phases from delay reference
- Unlimited output phase range (modulo clock period)

Rotating Phase DLL (cnt'd)

Semi-digital DLL

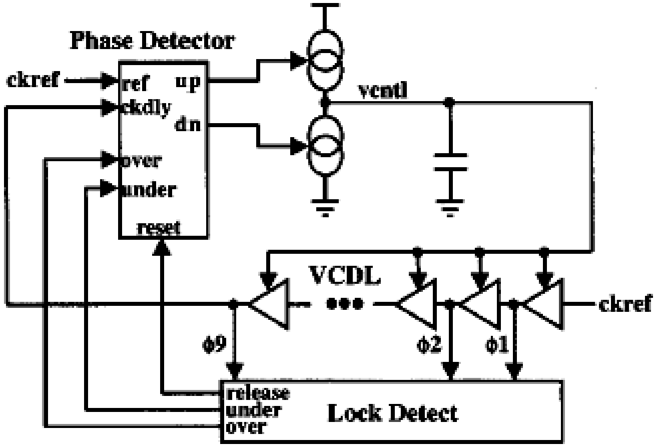
[S. Sidiropoulos, JSSC97]

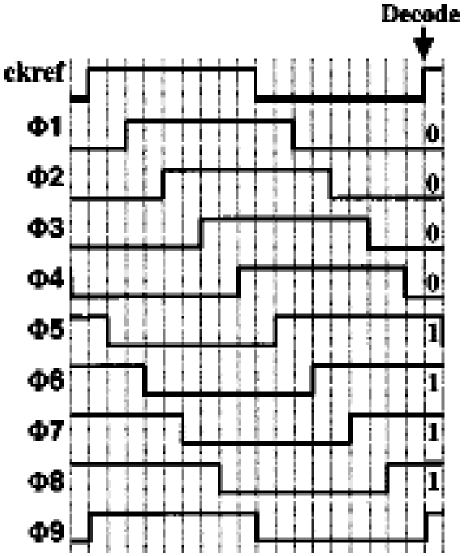




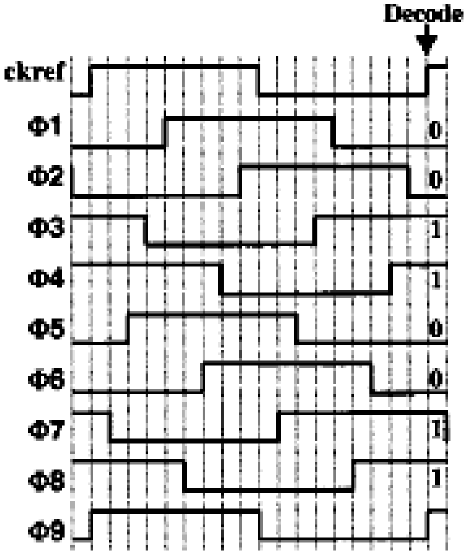
Self-Correction DLL

[D.J.Foley, JSSC01]





Correct locking

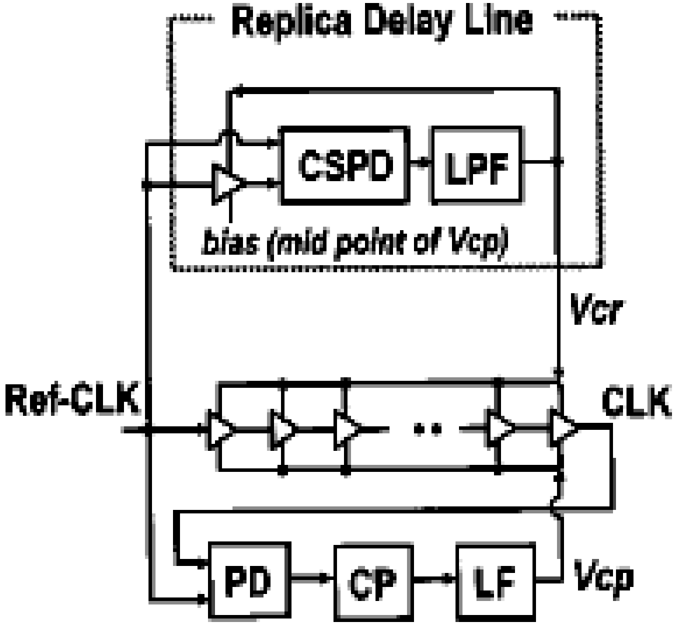


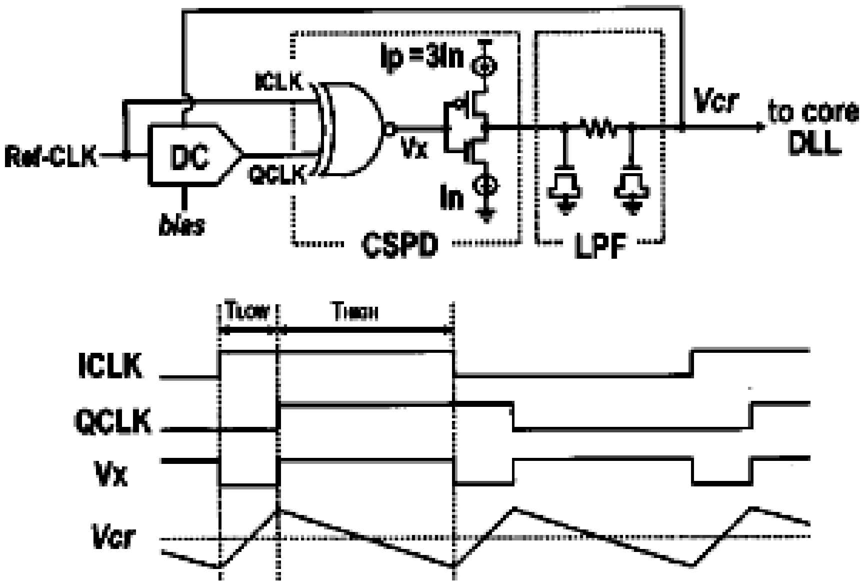
False locking

- **Phase detector gains the control of loop according to release, under and over**

DLL Using A Replica Delay Line

[Y.Moon, JSSC00]

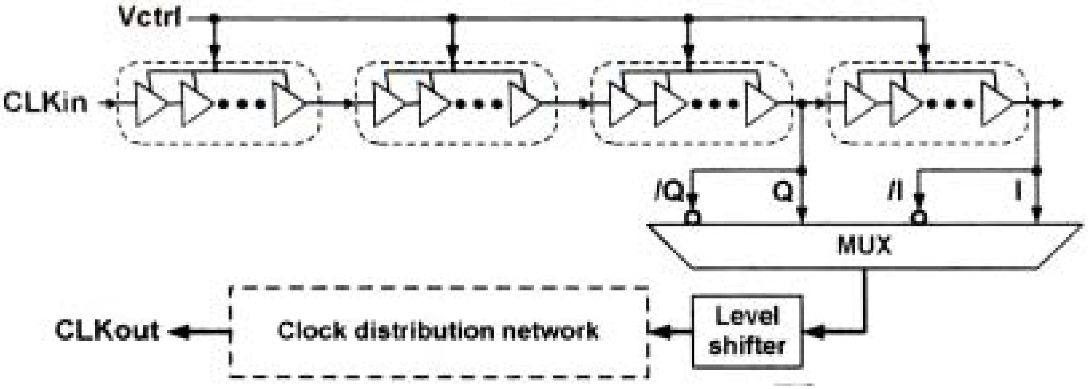


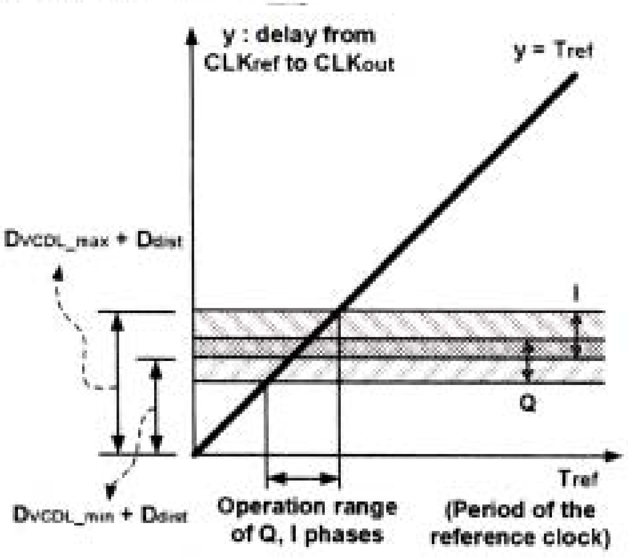


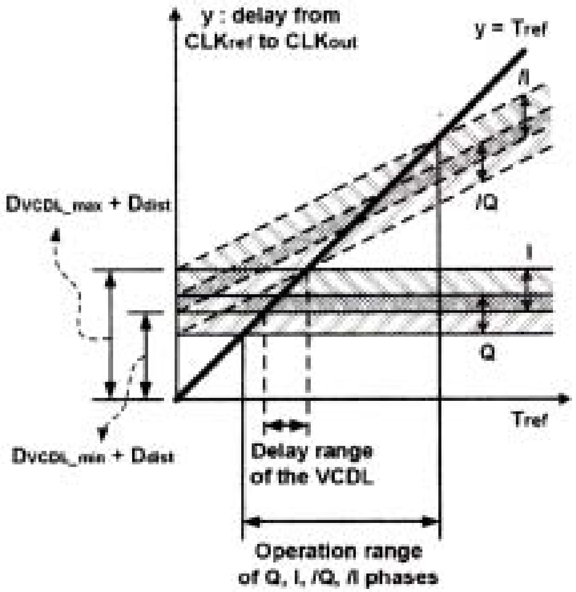
- The control voltage of RDL protect false locking

DLL Using Multiple Phases

[B.Kim, CICC04]



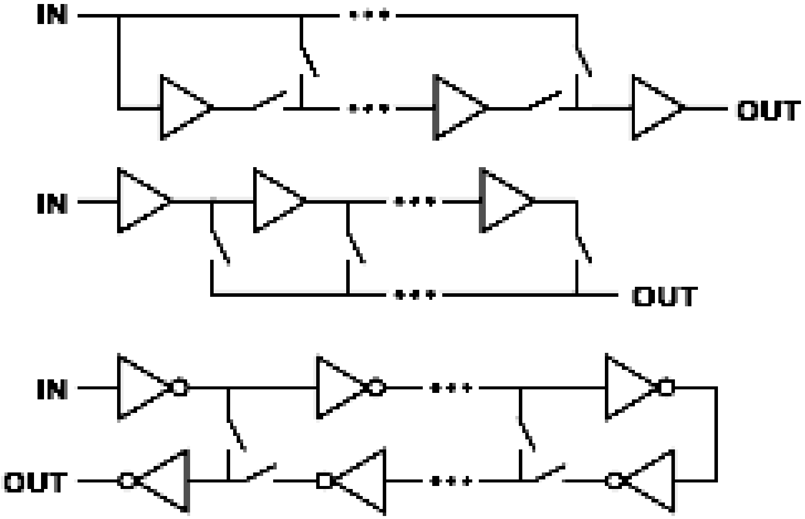




Lock Time

- Lock time limits (< 100 cycles for DDR)
  - Need high tracking bandwidth (self-biased DLLs)
  - Digital DLLs can use “non-linear” techniques
  - Open loop

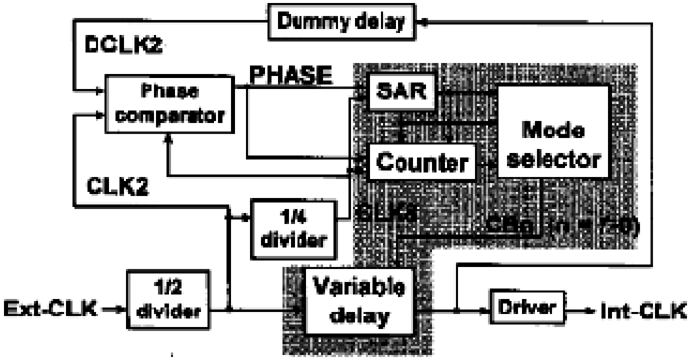
Digital Delay Line DLL



- Uses digital delay line with fixed delay as timing reference
- Selects output phases from delay reference (digital control)
- Correction step size is typically fixed
  - Large locking time (clock cycle · number of steps)
  - Can use exponentially decreasing steps (SADLL)
- Digital delay control provides more flexibility

Digital Delay Line DLL with SAR

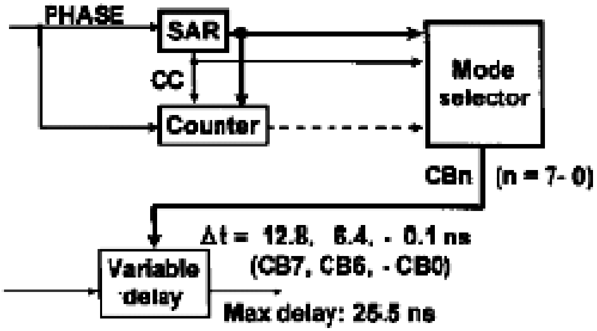
- **Successive Approximation Register DLL** [M.Hasegawa, ISSCC98]



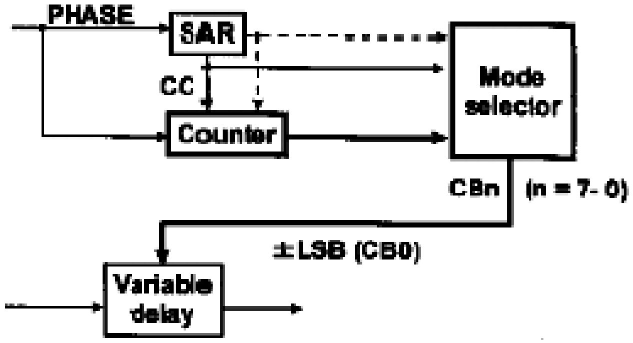
SAR = Successive approximation register

- Fast-lock by successive approximation < 64 cycles
- Counter-mode operation during normal cycle

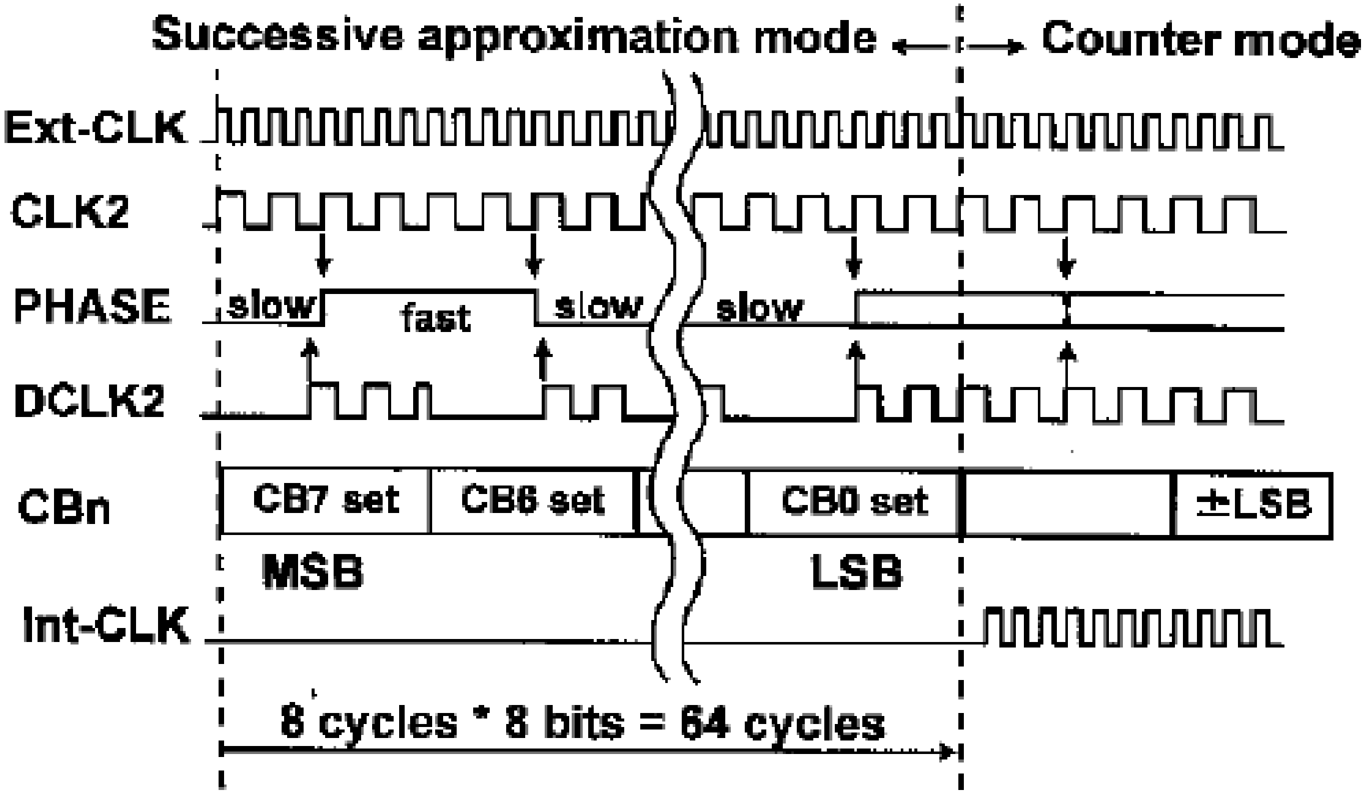
During lock-in



After lock-in

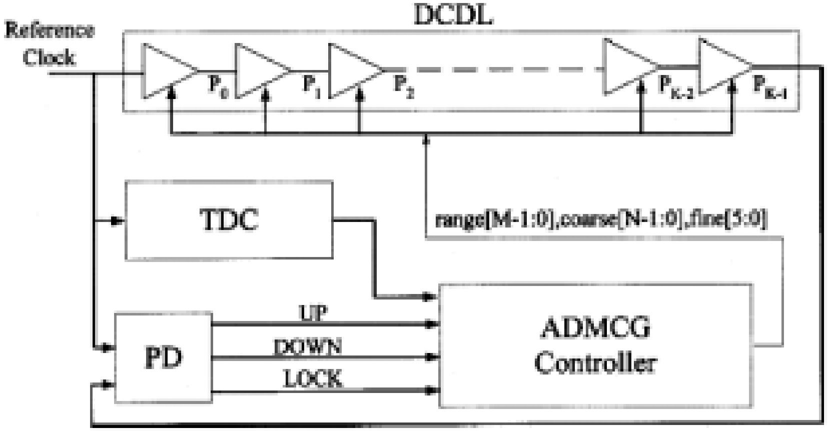


Digital Delay Line DLL with SAR(cnt'd)

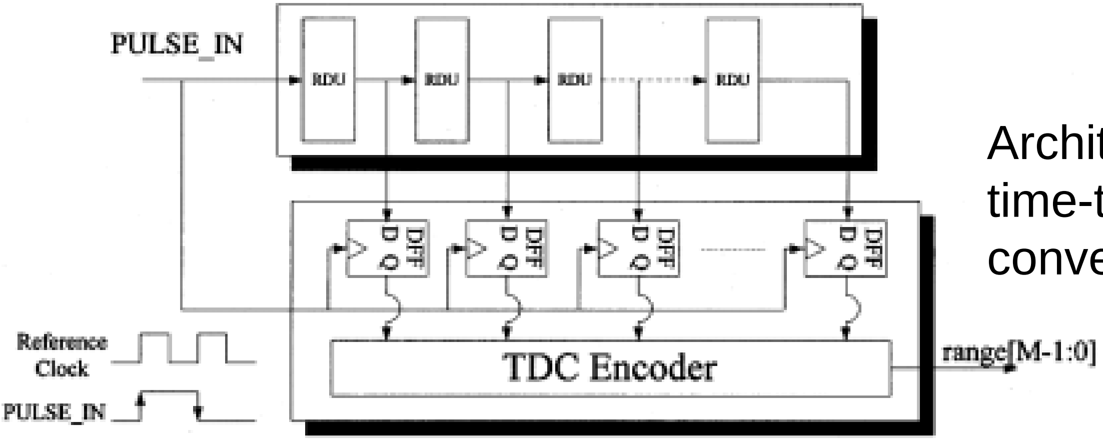


All-Digital DLL Using a TDC

[C.Chung, JSSC04]



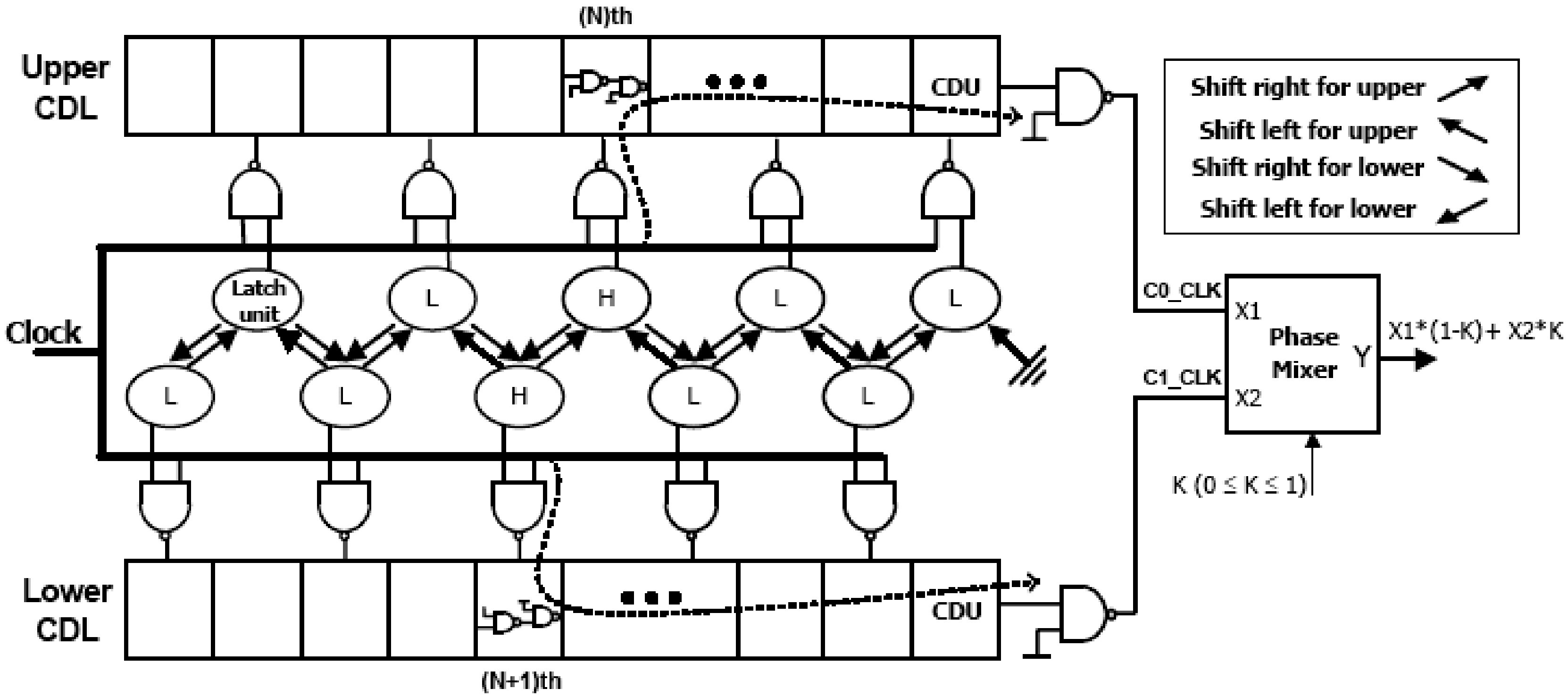
Overall architecture



Architecture of the time-to-digital converter (TDC)

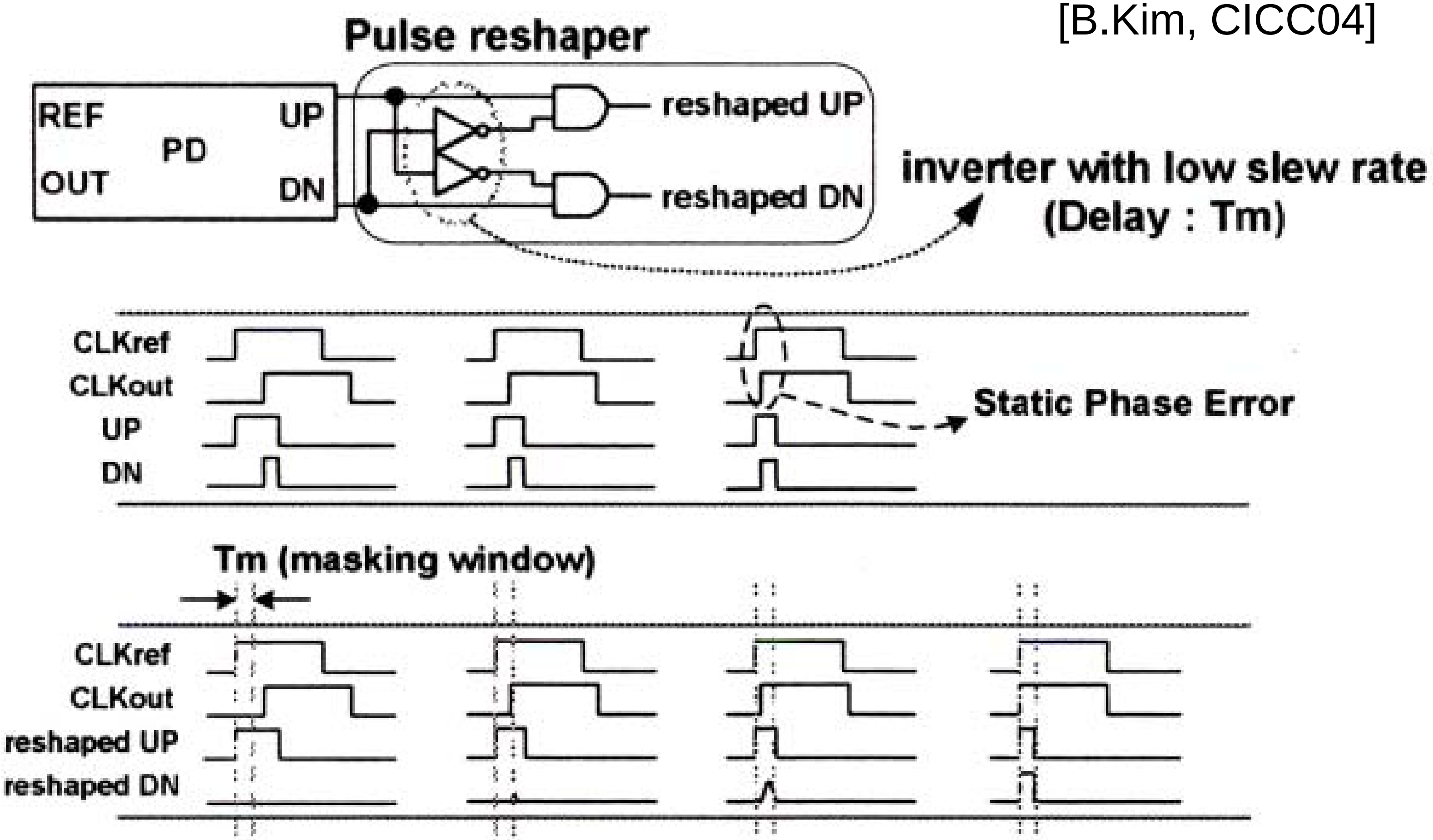
Schematic of Dual Coarse Delay Line

[J. Kwak, SOVC03]



Compensation of Static Phase Offset

[B.Kim, CICC04]

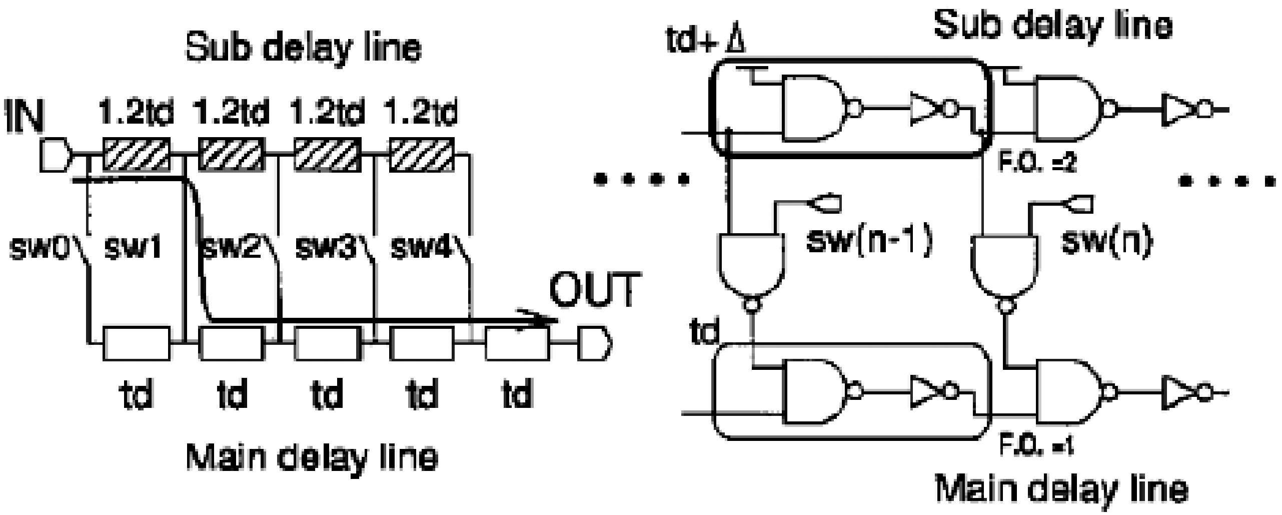


Power Dissipation and Area

- **Power dissipation limits**
  - **Operating power**
    - ✓ **Analog DLLs can use less power than digital DLLs**
  - **Stand-by power**
    - ✓ **Analog DLLS cannot be turned off for long without relocking (charge in loop filter cap. Will leak away)**
    - ✓ **Digital DLLs store locked state in registers**
- **Area limits**
  - **Analog DLLs can have smaller area than digital DLLs**
    - ✓ **Digital delay line DLLs that support low operating frequencies can be very large**

Register Controlled DLL

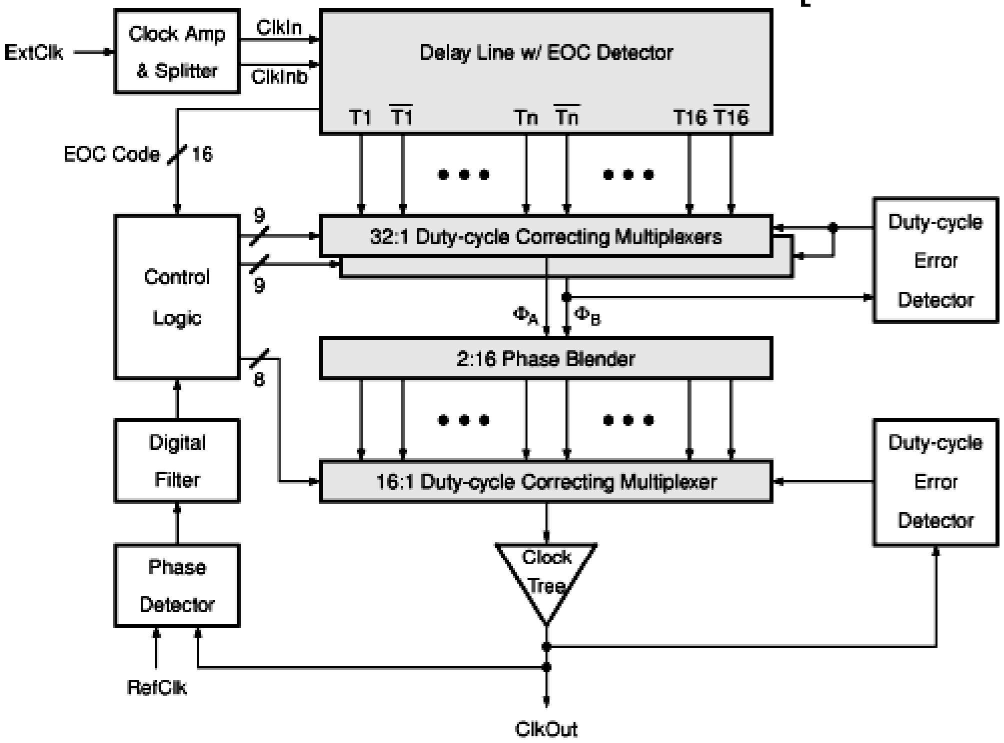
[A.Hatakeyama, ISSCC97]



- **Locking information is stored as a digital code.**
- **High resolution because of vernier type delay line.**

Portable DLL

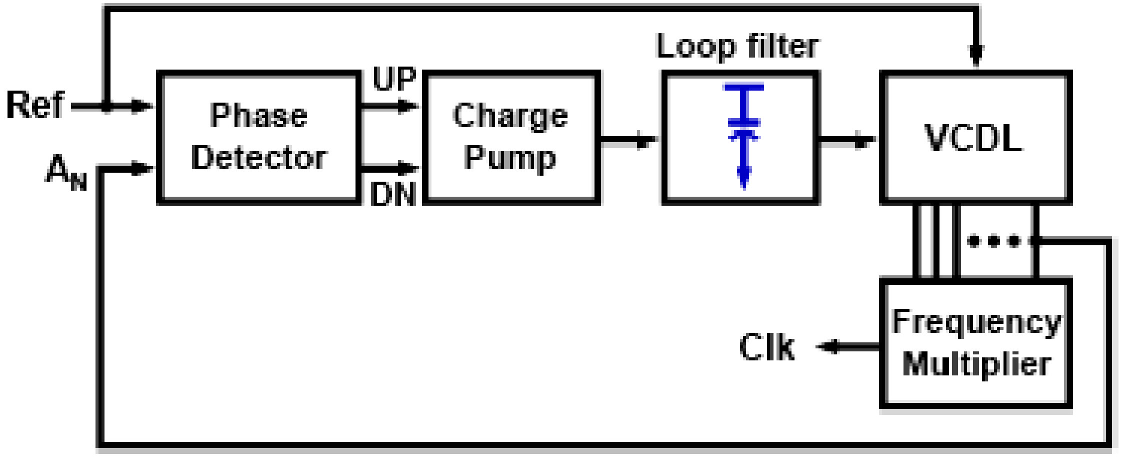
[B.W.Garlepp, JSSC99]


ExtClk
Clock Amp & Splitter
ClkIn
ClkInb
Delay Line w/ EOC Detector
T1 $\overline{T1}$ Tn $\overline{Tn}$ T16 $\overline{T16}$
EOC Code
16
9
9
8
32:1 Duty-cycle Correcting Multiplexers
Duty-cycle Error Detector
Control Logic
Φ_A
Φ_B
2:16 Phase Blender
Digital Filter
16:1 Duty-cycle Correcting Multiplexer
Duty-cycle Error Detector
Phase Detector
Clock Tree
RefClk
ClkOut

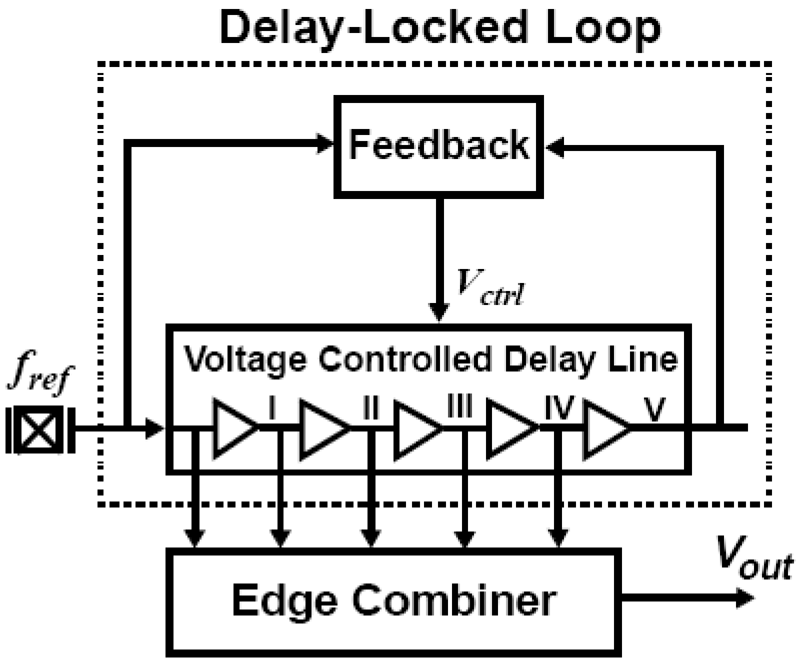
- **Phase information is stored in control logic.**

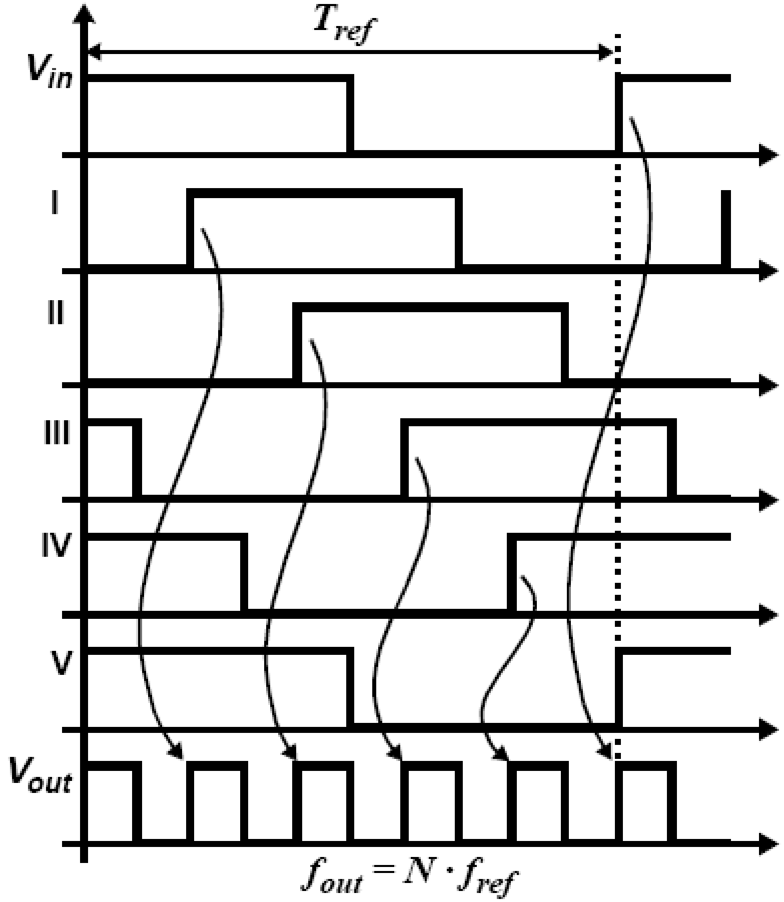
Multiplying DLL

- Avoid jitter accumulation problem of PLL without VCO
- 1^{st}-order system
  - Stable and easier to design
- Block diagram



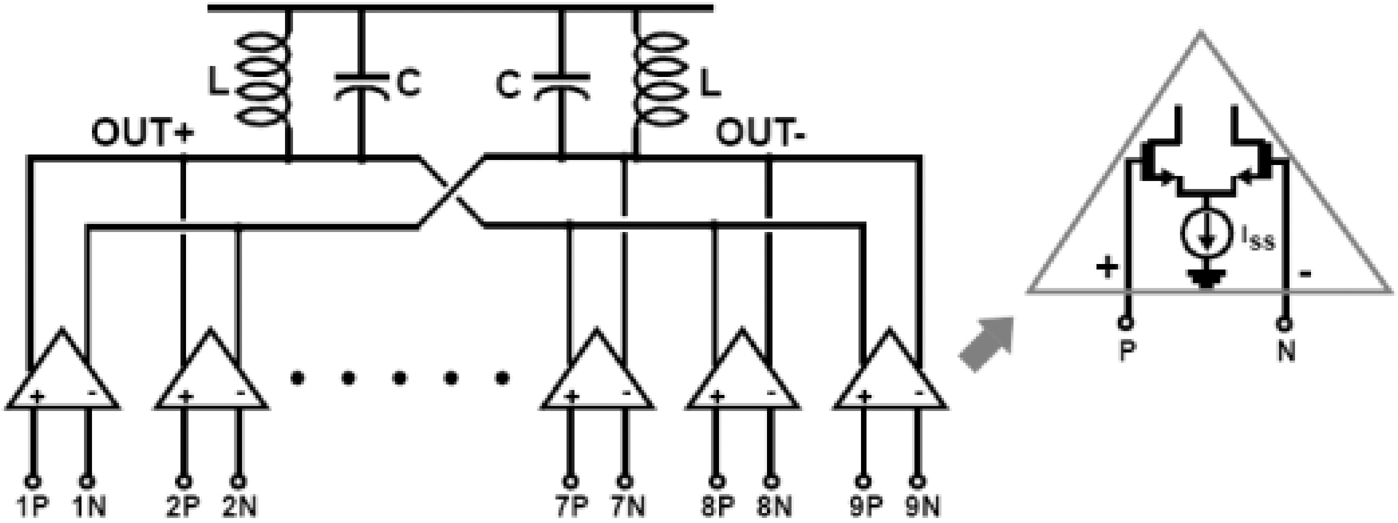
Multiplying DLL With LC tank





[G. Chien, JSSC 2000]

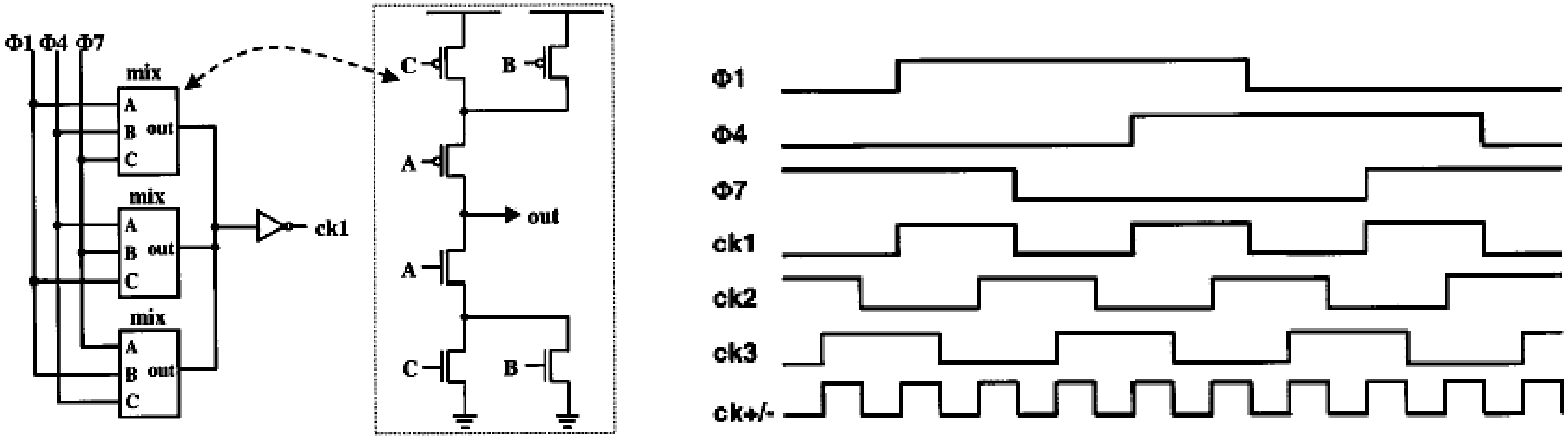
Multiplying DLL With LC tank (cnt'd)



- Diff. pair modulates tail currents into LC-tank circuit
- LC-Tank enhances load impedance : large area & fixed multiplication ratio
- Large current necessary for large voltage swing

Multiplying DLL With AND/OR Gate

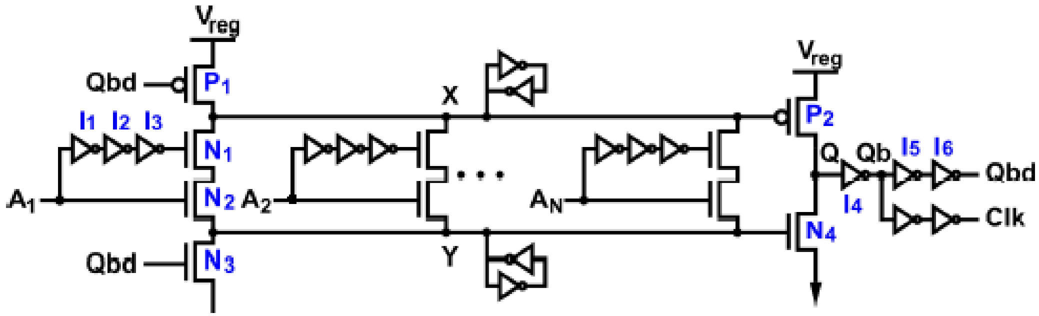
[D.J.Foley, JSSC 2001]

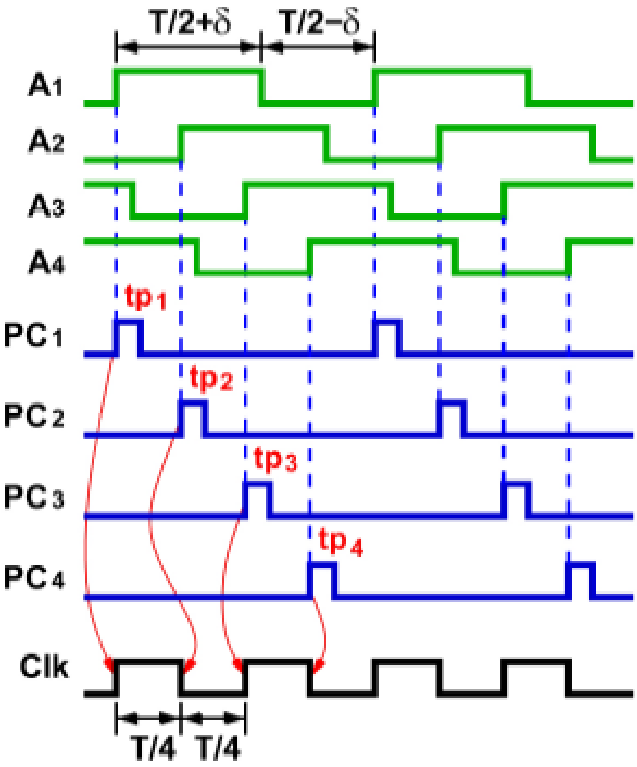


- AND/OR gate: 9 times freq. multiplication
- Need analog OR I/O buffer & 50 Ω pull-up resistor: off-chip clock signal

Multiplying DLL With Digital Controls

[C.Kim, JSSC 2002]

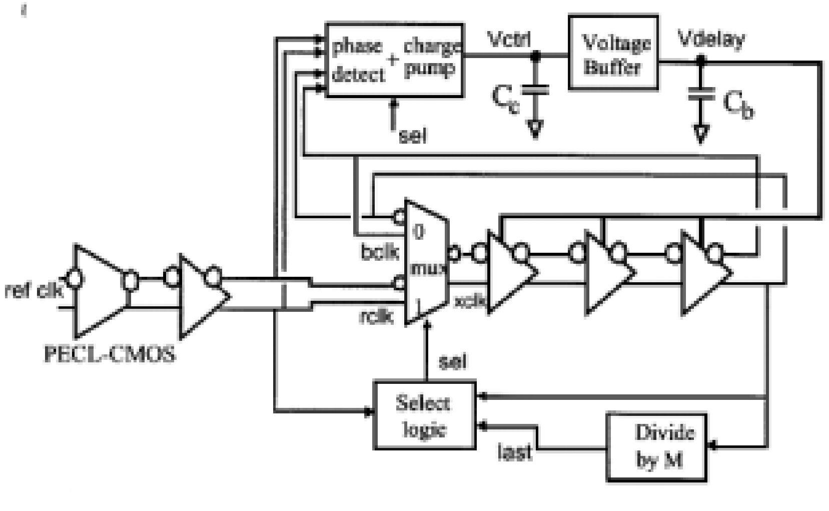


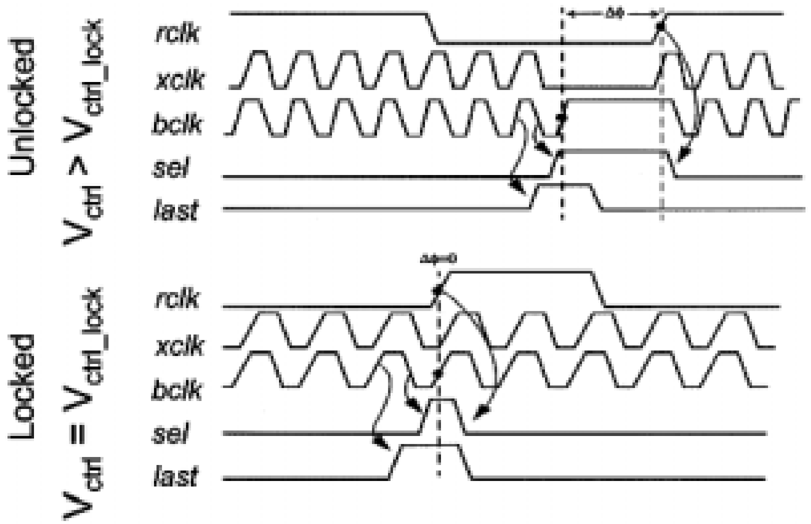


- Low power and small area
- Easier to integrate
- Multiplication factor can be easily programmable

Multiplying DLL for Low-jitter Clock Generation

[R. Farjad-Rad, JSSC 2002]





Conclusions

- **As data rates increase, DLLs will become essential to relax system timing constraints in each direction of data transfer.**
- **DLL consists of phase detector, voltage controlled delay cell, loop filter and charge pump.**
- **DLL should be designed as considering below issues.**
  - **Bandwidth**
  - **Limited lock range**
  - **Lock in time**
  - **Power dissipation limits**
  - **Area limits**
  - **Peak output jitter**

References

[S.Sidiropoulos SOVC 2000] S. Sidiropoulos, D. Liu, J. Kim, G. Wei, and M. Horowitz, “Adaptivebandwidth DLL’s and PLL’s using regulated supply CMOS buffers,” in*VLSI Symp. Dig. Tech. Papers*, June 2000, pp. 124–127.

[Young JSSC 1992] I.A. Young; J.K Greason; K.L Wong, “A PLL clock generator with 5 to 10MHz of lock range for microprocessors,” IEEE J. Solid-state Circuits, vol. 27, no. 11, pp1599-1607

[Ingino JSSC 2001] J.M. Ingino, V.R von Kaenel, “A 4-GHz clock system for a high-performance system-on-a-chip design” IEEE J. Solid-state Circuits, vol. 36, no. 11, pp1693-1698

[K. Kim ISSCC 04] K. Kim *et al* , “1.4Gb/s DLL using 2nd order charge-pump scheme with low phase/duty error for high-sped DRAM application” ISSCC 2004 Dig. Tech. Papers, pp 212-213, Feb 2004

[A. Chandrakasan, IEEE Press, 2001] A. Chandrakasan, W. J. Bowhill, and F. Fox, Design of High-Performance Microprocessor Circuit. New York: IEEE Press, 2001, p. 240.

[J. Maneatis, JSSC Nov 96] J. Maneatis, “Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques,” IEEE J. Solid-State Circuits, vol. 31, no. 11, pp. 1723-1732, Nov. 1996.

[S. Sidiropoulos, JSSC Nov 97] S. Sidiropoulos, and M. Horowitz, “A semidigital dual delay-locked loop,” IEEE J. Solid-State Circuits, vol. 32, no. 11, pp. 1683-1692, Nov. 1997.

[D.J. Foley JSSC] D.J.Foley and M.P.Flynn, “CMOS DLL-based 2-V 3.2-ps jitter 1-GHz clock synthesizer and temperature-compensated tunable oscillator,” IEEE J. Solid-State , vol. 36, pp.417-423, Mar. 2001

[Y.Moon JSSC 2000] Y. Moon, J. Choi, K. Lee, D. K. Jeong, and M. K. Kim, “An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance,” IEEE J. Solid-State Circuits, vol.35, pp. 377–384, Mar. 2000.

[B.Kim, CICC04] B.Kim and L. Kim, “A 250MHz-2GHz wide range delay-locked loop,” In *Proc.IEEE CICC* 2003, pp.139-142, Oct 2004.

[M. Hasegawa, ISSCC98] M. Hasegawa, et al., “A 256Mb SDRAM with subthreshold leakage current suppression,” ISSCC 1998 Dig. Tech. Papers, pp. 80-81, Feb. 1998.

[C.Chung, JSSC04] C. Chung and C. Lee “A New DLL-Based Approach for All-Digital Multiphase Clock Generation” IEEE J. Solid-State Circuits,vol. 39, pp 469-475, MAR. 2004

[J. Kwak, SOVC03] J.Kwak *et al*.“A low cost high performance register-controlled digital DLL for 1 Gbps_spl times_32 DDR SDRAM” Symp. VLSI Circuits Dig. Tech. Papers, pp283-284, 2003

[A. Hatakeyama ISSCC 1997] A. Hatakeyama, H. Mochizuki, T. Aikawa, M. Takita, Y. Ishii, H. Tsuboi, S. Fujioka, S. Yamaguchi, M. Koga, Y. Serizawa, K. Nishimura, K. Kawabata, Y. Okajima, M. Kawano, H. Kojima, K. Mizutani, T. Anezaki, M. Hasegawa, and M. Taguchi, “A 256 Mb SDRAM using register-controlled digital DLL,” in ISSCC 1997 Dig. Tech. Papers, Feb.1997, pp. 72–73.

[B.W Garlepp JSSC 99] B.W. Garlepp, K. S. Donnely, J. Kim, P. S. Chau, J. L. Zerbe, C. Haung, C. V. Tran, C. L. Pourtman, D. Stark, Y. Chan, T. H. Lee, and M. A. Horowitz, “A portable digital DLL for high-Speed CMOS interface circuits,”IEEE J. Solid-State Circuits, vol. 34, pp. 632–644, May 1999.

[V. von Kaenel, ISSCC98] V. von Kaenel, et al., “A 600MHz CMOS PLL microprocessor clock generator with a 1.2GHz VCO,” ISSCC 1998 Dig. Tech. Papers, pp. 396-397, Feb. 1998.

[G. Chien , JSSC 2000] G. Chien and P. R. Gray, “A900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications,” *IEEE J. Solid- State Circuits*, vol. 35, pp. 1996–1999, Dec. 2000.

[C. Kim, JSSC 2002] C. Kim, I.-C. Hwang, and S.-M. Kang, "A low-power small-area ±7.28-ps-jitter 1-GHz DLL-based clock generator," IEEE J. Solid-State Circuits, vol. 37, no. 11, pp. 1414-1420, Nov. 2002.

[R. Farjad-Rad, JSSC 2002] R. Farjad-Rad *et al.*, “Alow-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *IEEE J.Solid-State Circuits*, vol. 37, pp. 1804–1812, Dec. 2002.[J. Kwak, SOVC03] J.Kwak *et al*.“A low cost high performance register-controlled digital DLL for 1 Gbps_spl times_32 DDR SDRAM” Symp. VLSI Circuits Dig. Tech. Papers, pp283-284, 2003