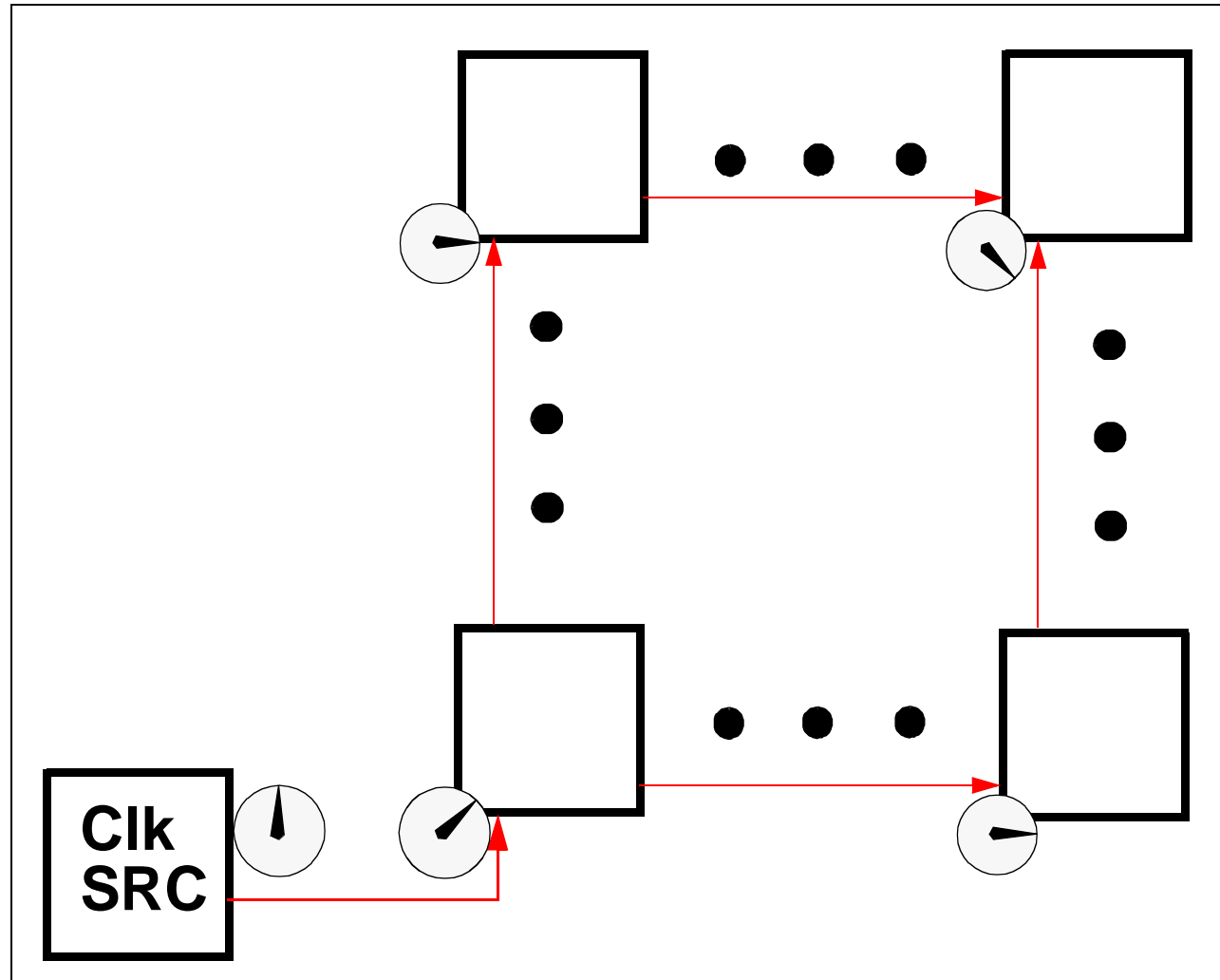
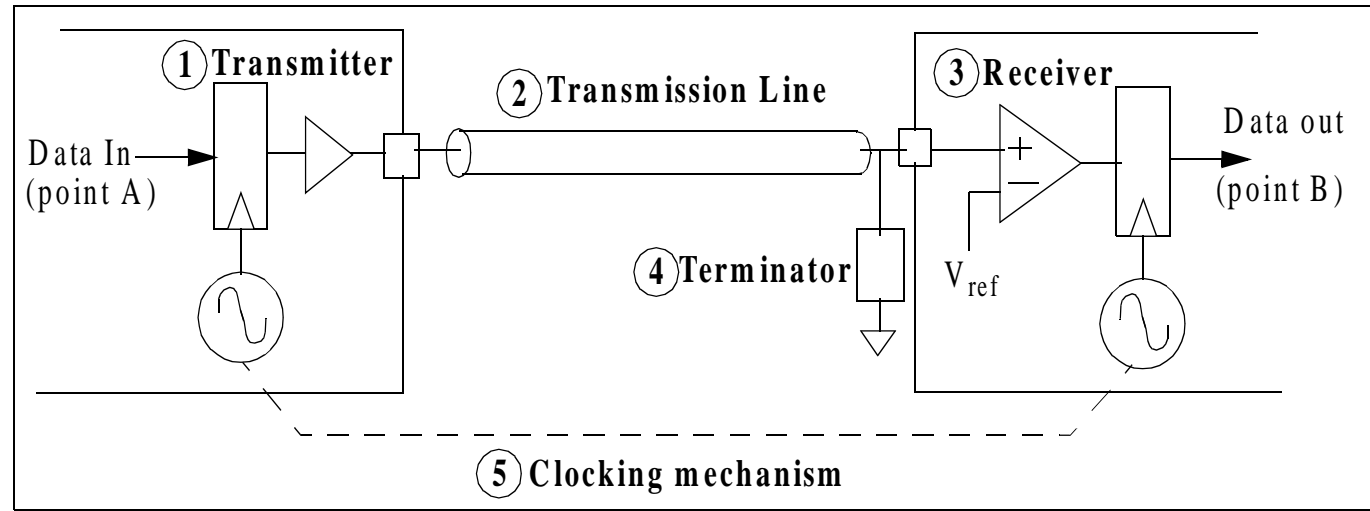


# Timing and Synchronization



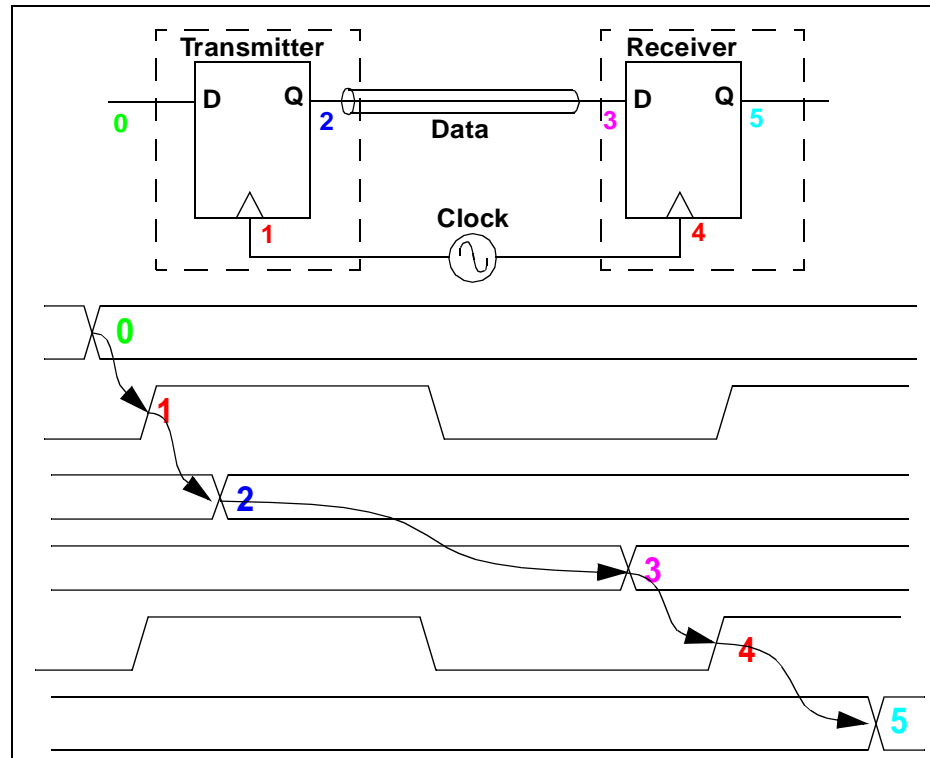
# A Signaling System



- 1. Transmitter: Encodes data as current/voltage level onto the line**
- 2. Transmission Line: Deliver data from transmitter to receiver**
- 3. Receiver: Compare against reference to extract data**
- 4. Terminator: Remove signal from line, once they're received**
- 5. Clock: Tells transmitter when to send, receiver when to sample signal**

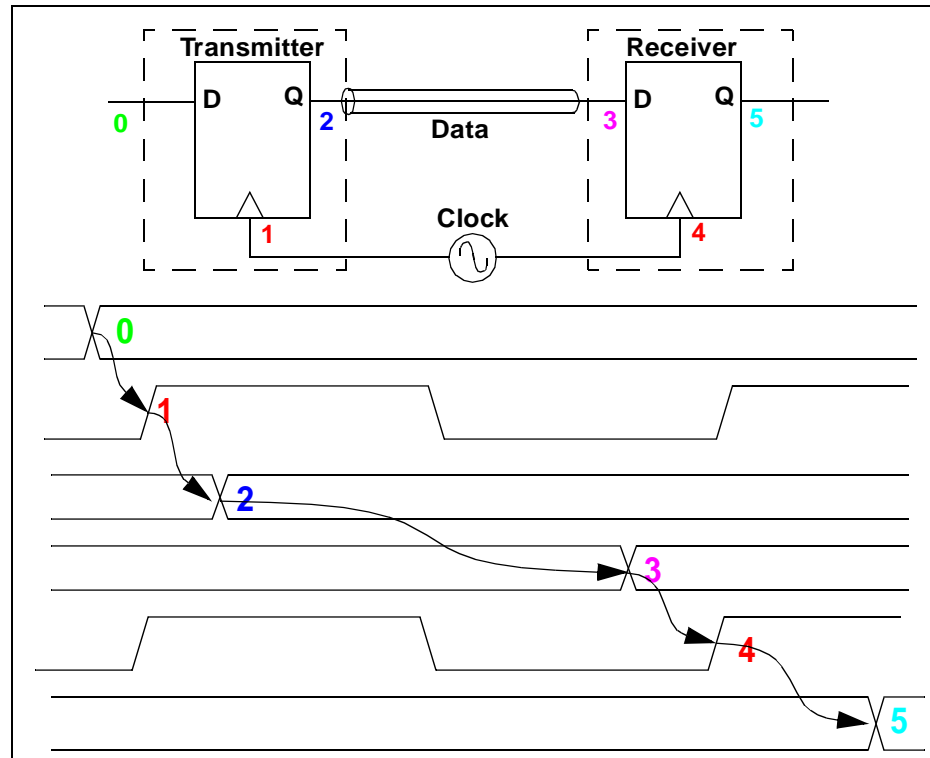
\* Poulton ISSCC 1999 Signaling Tutorial

# Global Clock I



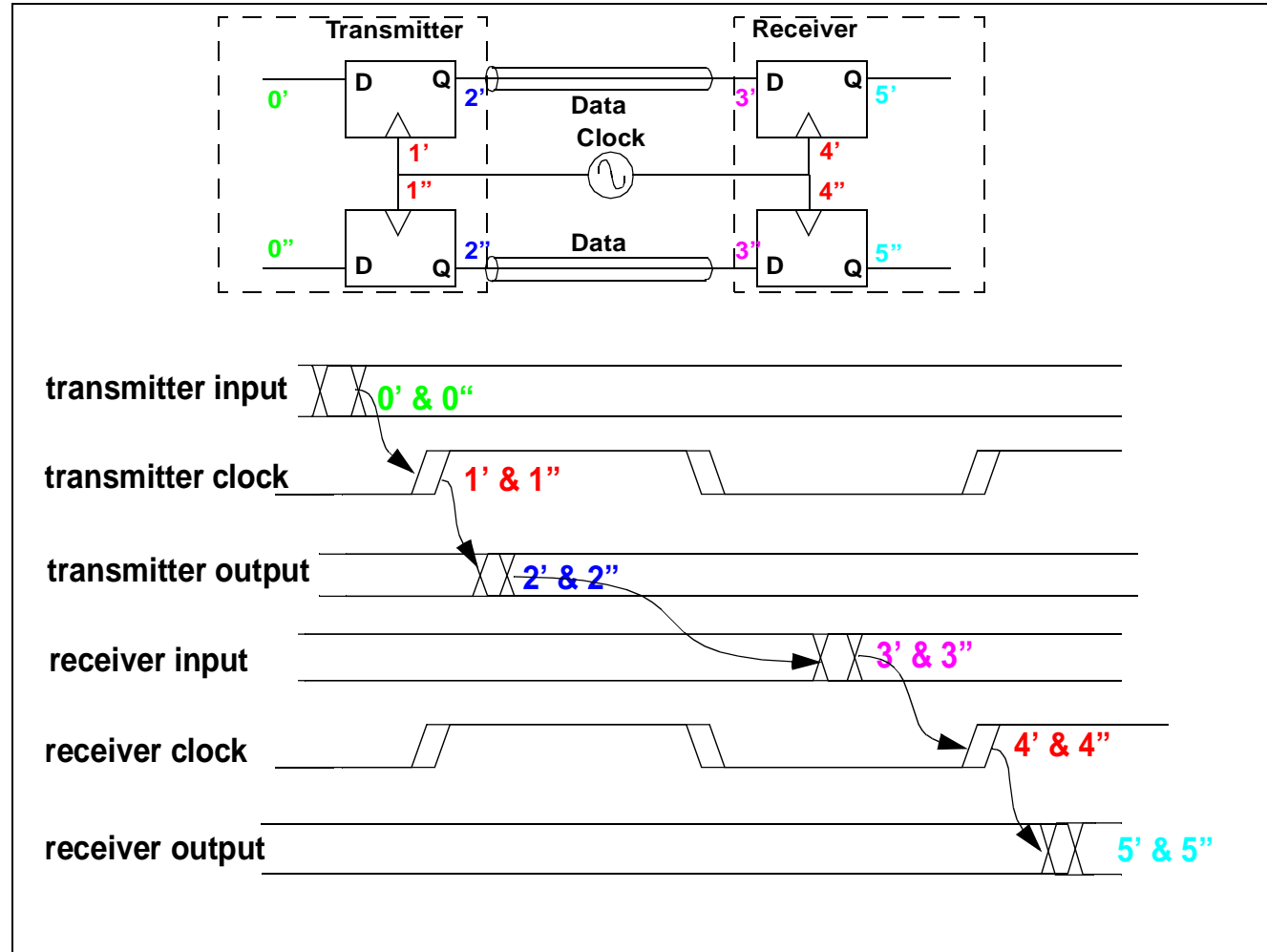
- 0:** Assume data is stable for setup time before clock edge
- 1:** Rising edge of transmitter clock
- 2:** Transmitter begins to drive data
- 3:** Signal reaches input of receiver.
- 4:** Rising edge of receiver clock
- 5:** Receiver latches data and drives internal signal lines

# Global Clock II



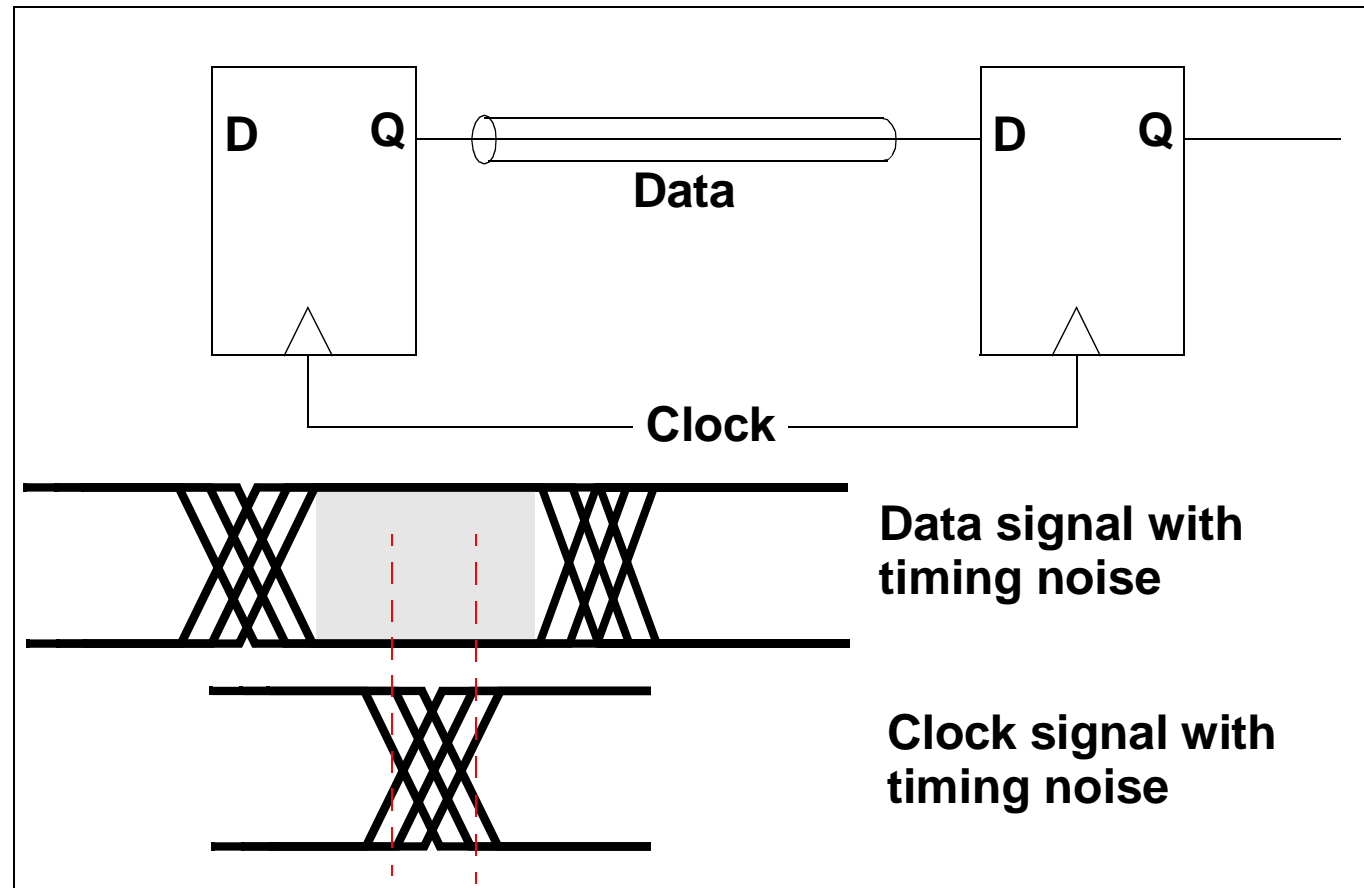
- “Clock cycle” occurs between steps 1 and 4.
- Assume signal transported cleanly with one way wire delay time between steps 2 and 3.
- If wire delay time exceeds cycle time, multiple bits will be “in flight” simultaneously on the transmission medium.

# Global Clock III - Parallel Data



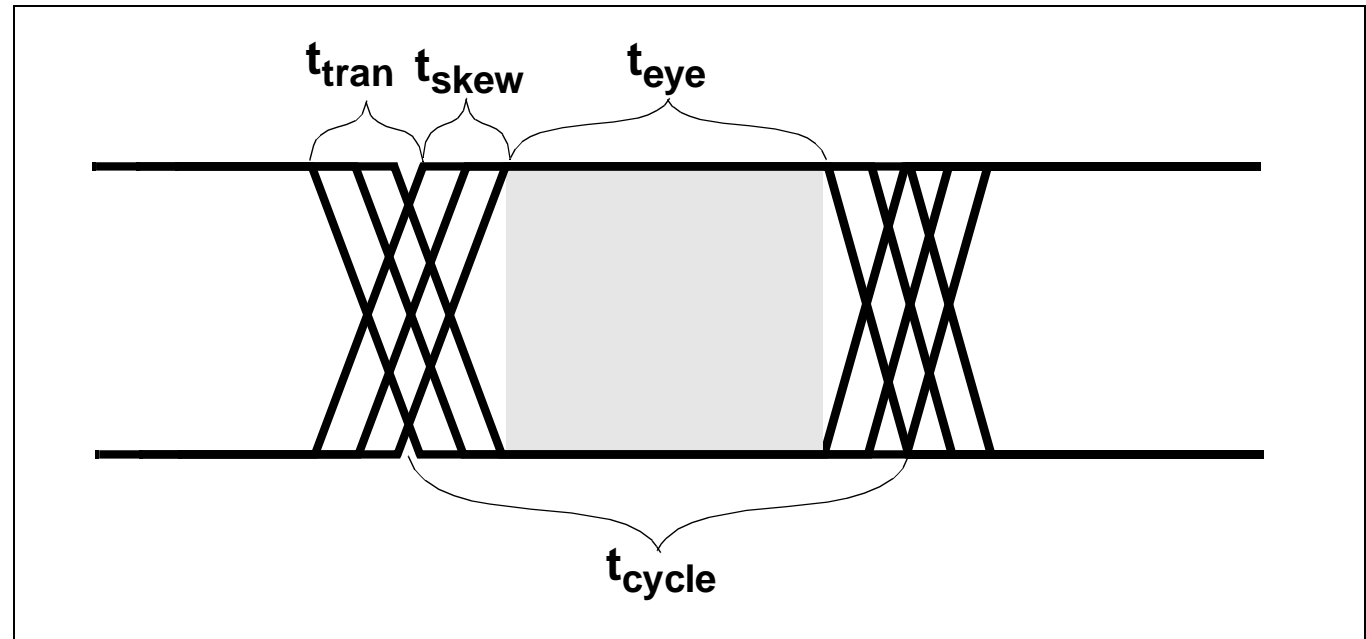
- Skew and jitter eats into timing budget

# Basic Problem



- Clock signal is just another signal subject to same constraints of voltage noise, skew and jitter as data signals

# Cycle Budget

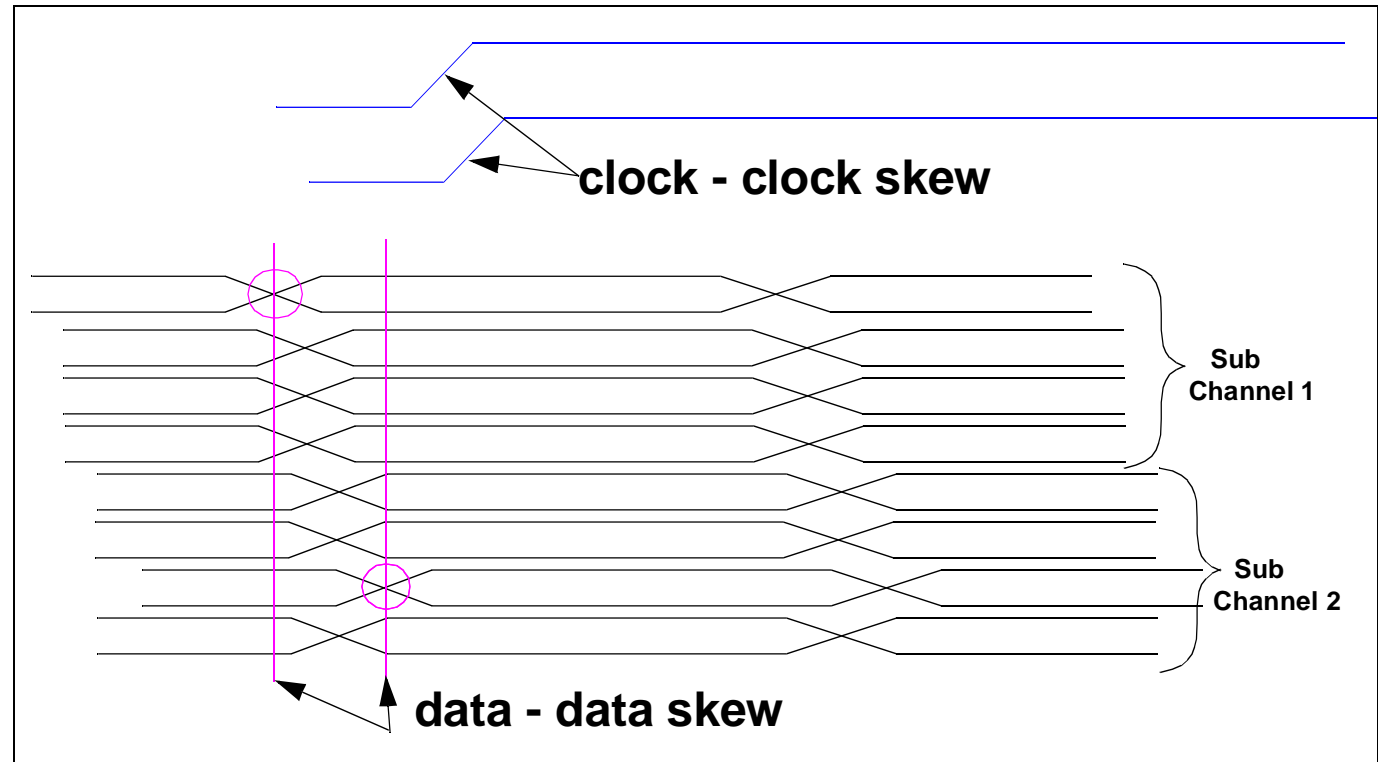


$$t_{skew} = \max(\text{skew}) + \max(\text{jitter})$$

$$t_{tran} = \text{Edge transition time} = \max(\text{rise\_time}, \text{fall\_time})$$

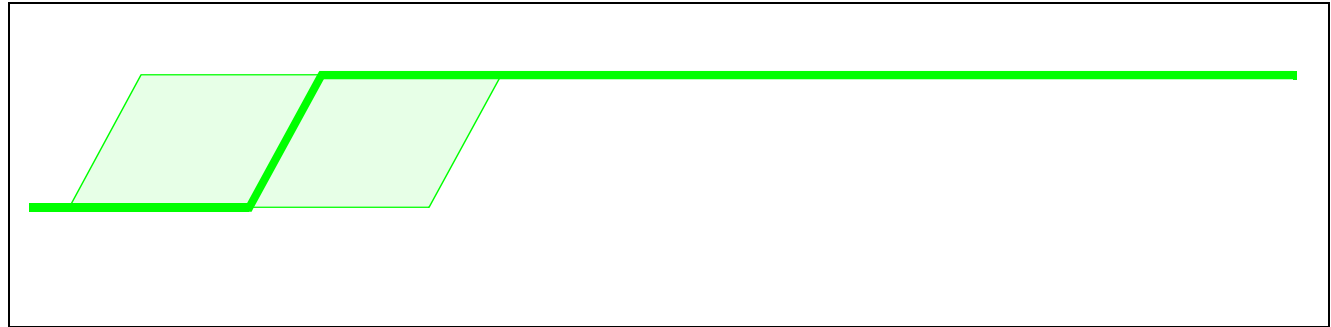
$$t_{eye} = t_{cycle} - t_{tran} - t_{skew}$$

# Skew



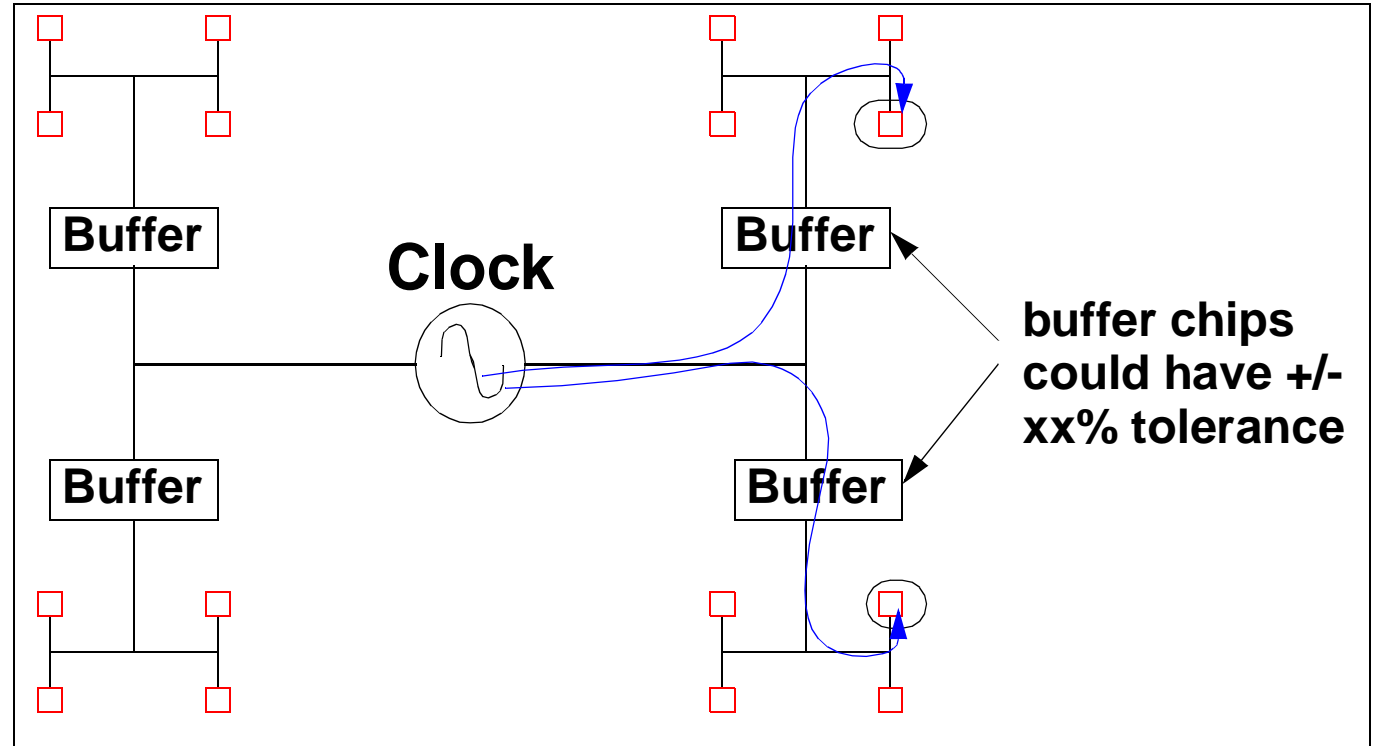
- **Static timing displacement from ideal design**
- **Caused by differences in signal path characteristics**
- **Total timing budget must take data-data skew, data-clock skew as well as clock-clock skew into cycle budget consideration**

# Jitter



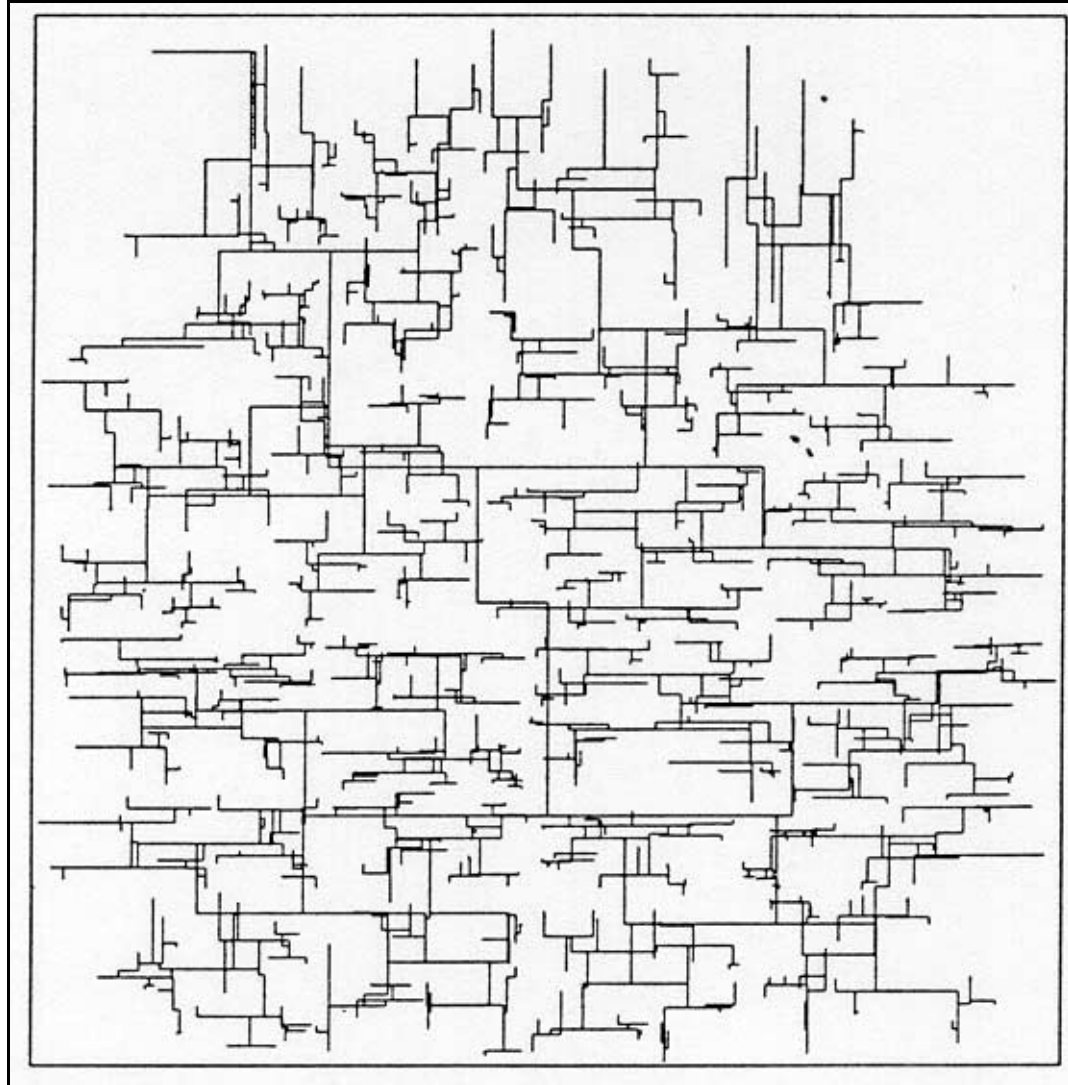
- **Dynamic timing displacement from nominal timing characteristics**
- **Magnitude and offset of timing displacement could depend on: previous signal state(s), current signal state(s), supply voltage level(s), crosstalk, variations in thermal characteristics. Perhaps even phases of the moon (not proven).**

# Clock Tree I



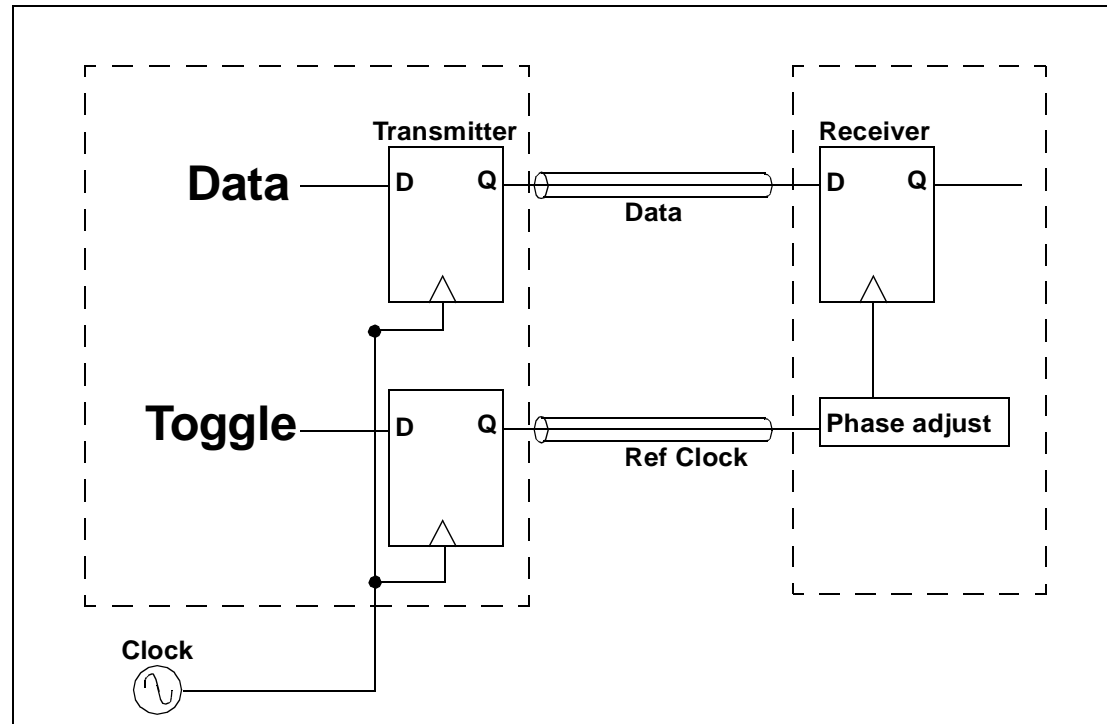
- Large synchronous systems require all chips to be driven by clock signal.
- Clock signal paths and buffer chips could introduce both skew and jitter at each stage
- Jitter and skew are additive with larger systems. More buffering, more skew and jitter.

# Clock Tree II



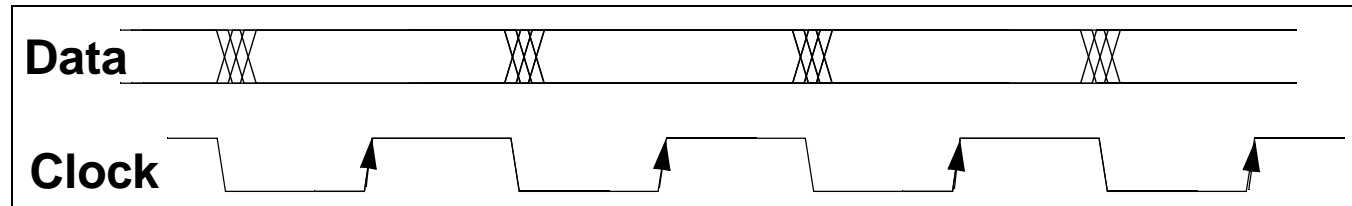
**An on chip clock tree**

# Clock Forwarding

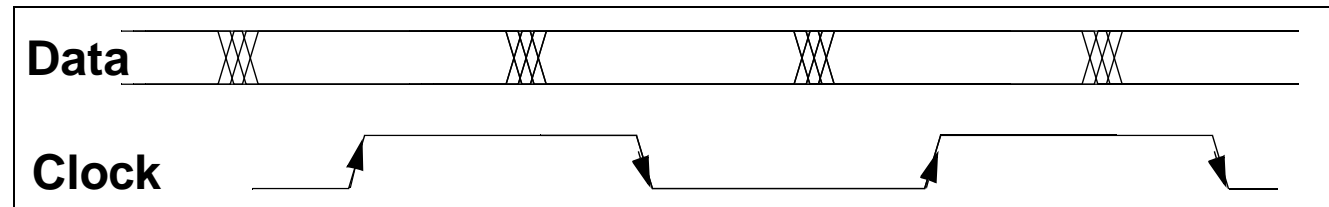


- Reference clock experiences same electrical loading condition and signal discontinuity characteristics as data signals.
- clock skew with respect to data signal may be minimized but jitter remains.
- Also known as source synchronous clocking

# Dual Edge Clocking

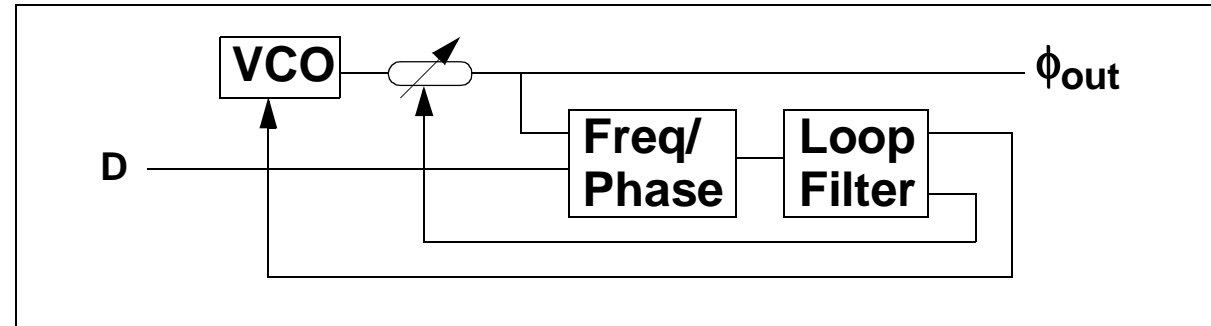


- Only one edge of clock latches data
- Duty cycle of clock signal is not relevant
- Clock signal operating at 2X switching rate of data



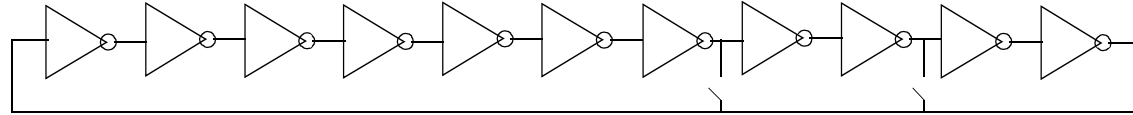
- Both edges of clock used to latch in data
- Duty cycle of clock signal must be even
- Clock signal must be phase shifted by 90 degrees relative to phase of data signal
- How do you get 90 degrees ?

# Phase Locked Loop



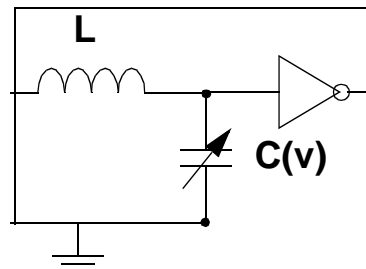
- Given a data signal, recover the frequency and phase of the data signal, generate local reference clock  $\phi_{out}$
- Local reference clock may be frequency multiple of input clock
- PLL depends on data input to provide “enough” signal transitions to lock onto, else PLL could lose coherency.
- Modern processors utilize PLL's for frequency multiplication

# Voltage Controlled Oscillator



**Ring Oscillator**

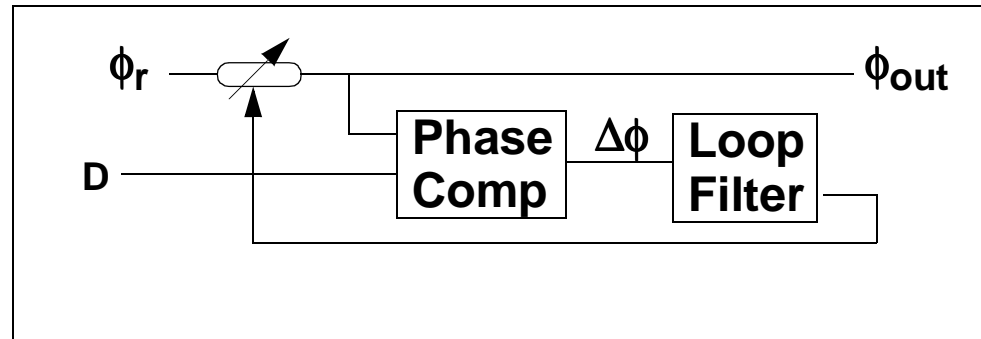
- VCO may be designed from ring oscillator where voltage controls the number of (odd) stages of inverters in the feedback ring



**LC Oscillator (conceptual illustration)**

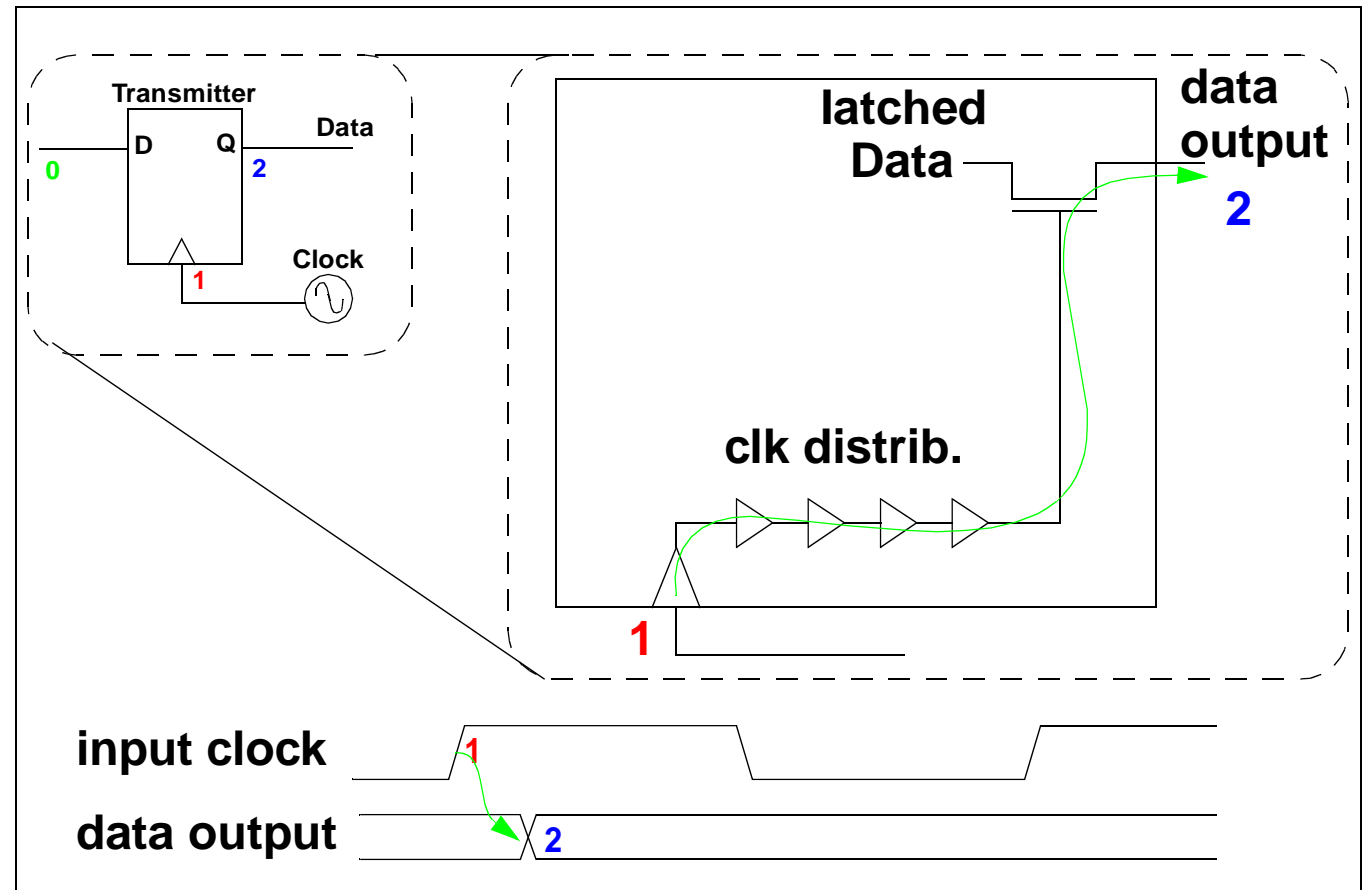
- VCO may be designed from resonant oscillator where voltage controls capacitance in LC circuit.

# Delay Locked Loop



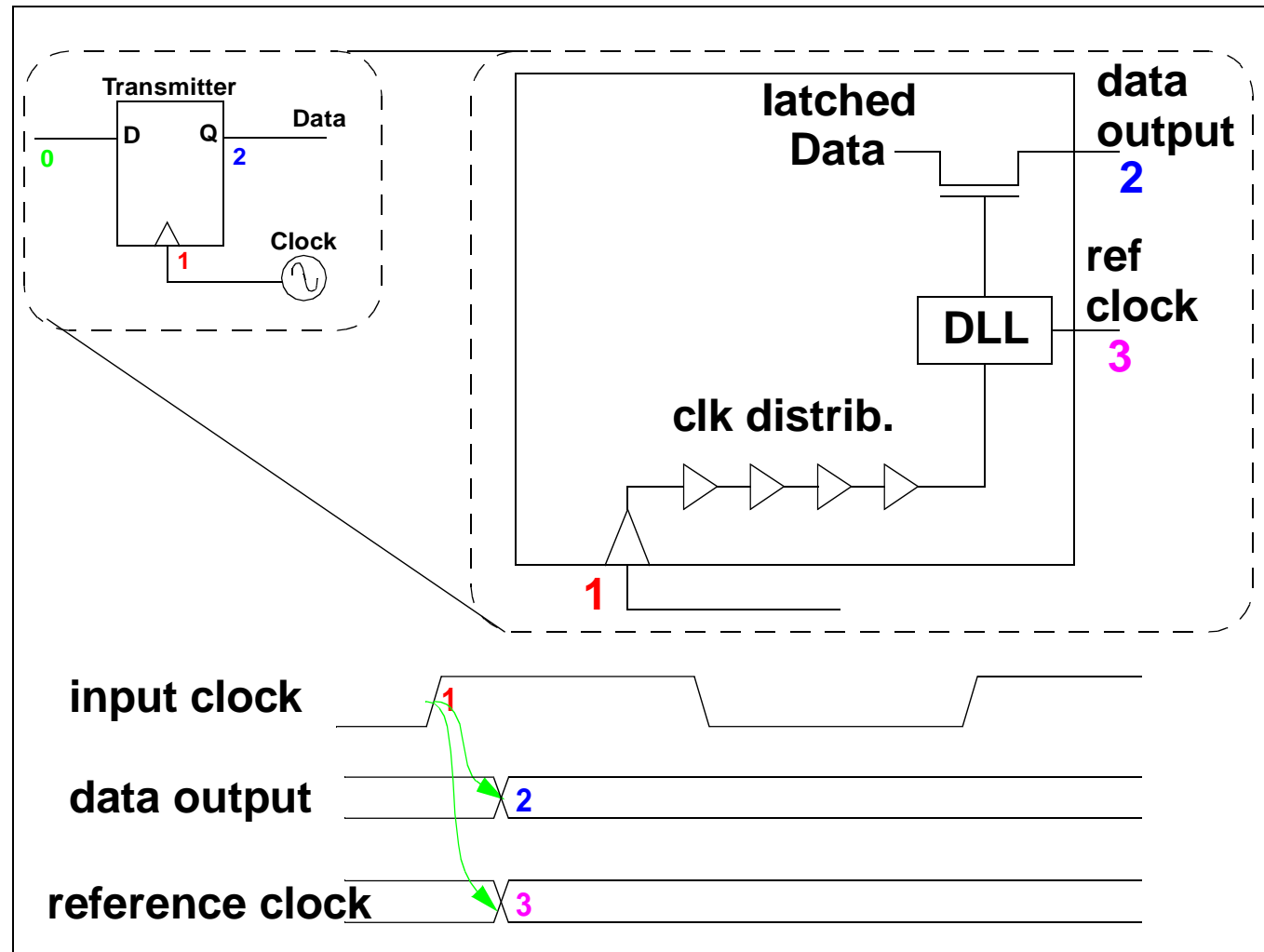
- Given a data signal and reference clock, compare and adjust phase of local clock signal by  $\Delta\phi$
- Unlike PLL, requires reference clock
- Hence, no need to “recover” clock signal with VCO
- Modern DRAM with dual edged clocking utilizes DLL's for phase compensation. (gets you 90 degrees)

# DLL in DRAM I



- Step 1 to step 2 is the **transmitter delay** from clock edge to output of data onto signal wire
- Data slightly skewed (delayed) with respect to clock edge. Deskew with DLL to recover lost phase.

# DLL in DRAM II



- reference clock has same phase (zero skew) with respect to data output. (DDR SDRAM DQS signal)