Topic 6

Clocking & Metastability

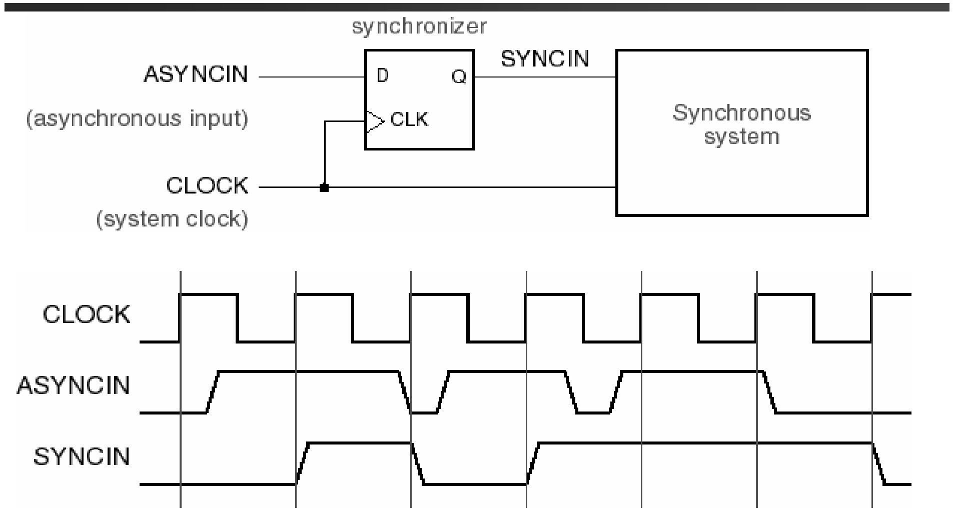
Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

URL: www.ee.imperial.ac.uk/pcheung/
E-mail: p.cheung@imperial.ac.uk

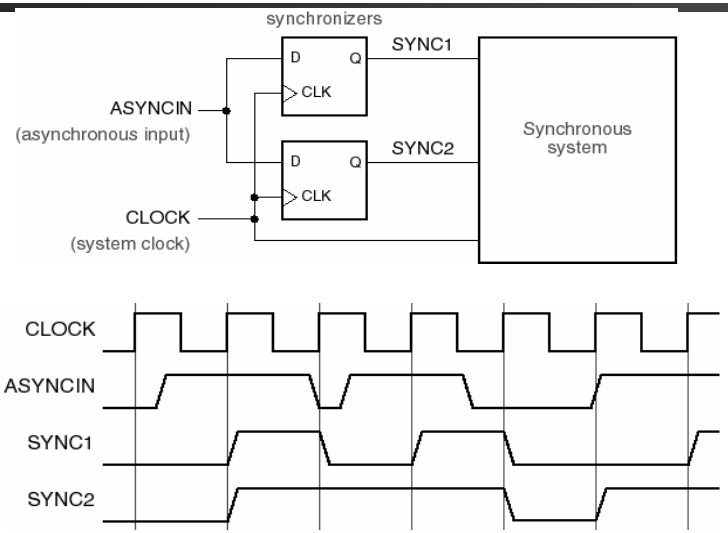
Asynchronous inputs

- Not all inputs are synchronized with the clock
- Examples:
  - Keystrokes
  - Sensor inputs
  - Data received from a network (transmitter has its own clock)
- Inputs must be synchronized with the system clock before being applied to a synchronous system.

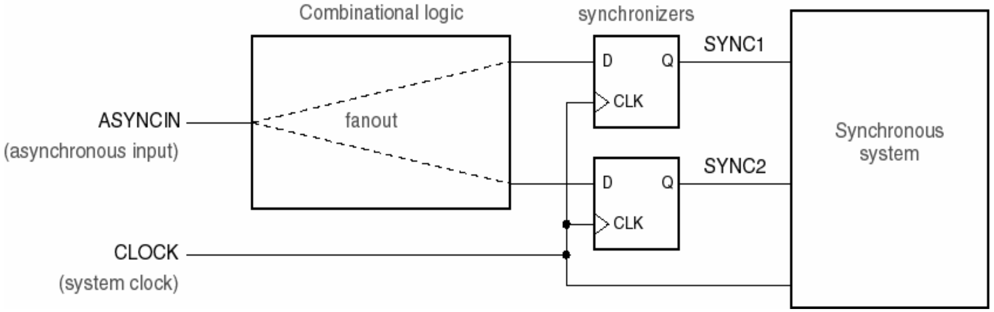
A simple synchronizer



Only one **synchronizer per input**

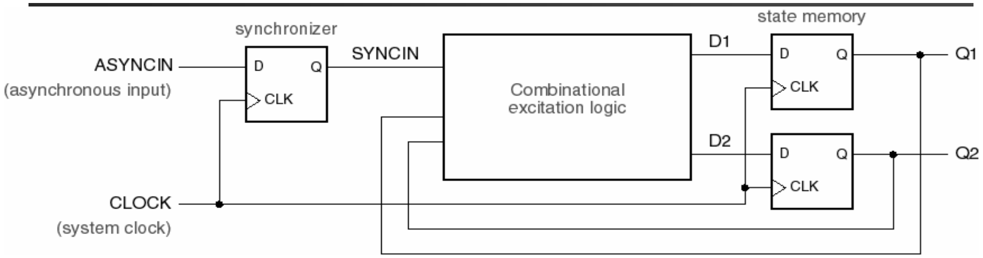


Even worse



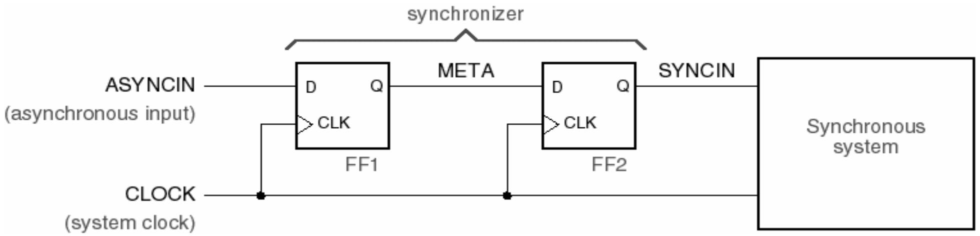
- Combinational delays to the two synchronizers are likely to be different.

The way to do it



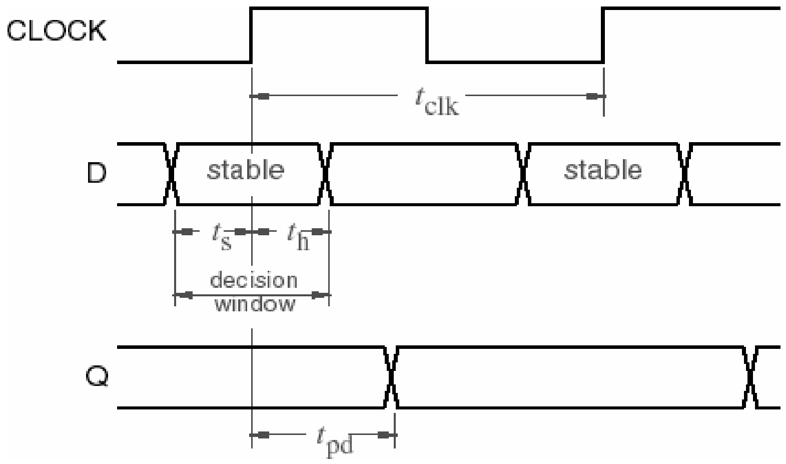
- One synchronizer per input
- Carefully locate the synchronization points in a system.
- But still a problem -- the synchronizer output may become metastable when setup and hold time are not met.

Recommended synchronizer design

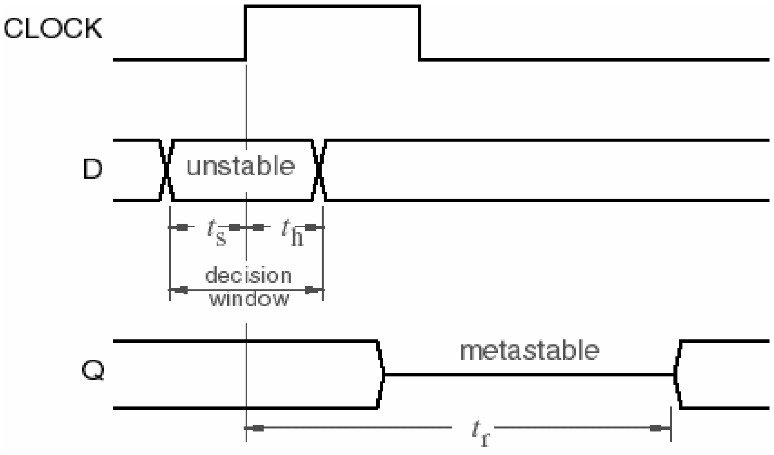


- Hope that FF1 settles down before “META” is sampled.
  - In this case, “SYNCIN” is valid for almost a full clock period.
  - Can calculate the probability of “synchronizer failure” (FF1 still metastable when META sampled)

Metastability decision window



Metastability resolution time



Flip-flop metastable behavior

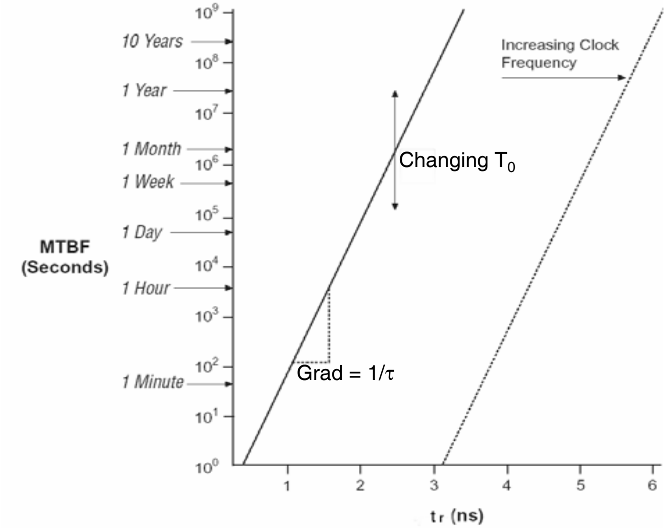
- Probability of flip-flop output being in the metastable state is an exponentially decreasing function of t_r (time since clock edge, i.e. "**resolution time**").
- Stated another way,

$$MTBF\ (t_r) = \frac{\exp\ (t_r / \tau)}{T_o f\ a}$$

where

τ and T_0 are parameters for a particular flip-flop,
f is the clock frequency, and
a is the number of asynchronous transitions / sec

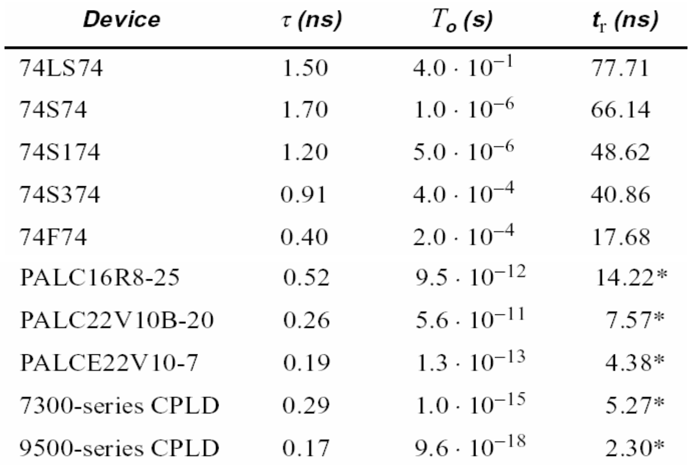
MTBF versus Resolution Time (t_r)



Typical flip-flop metastability parameters

$$MTBF\ (t_r) = \frac{\exp\ (t_r / \tau)}{T_o f\ a}$$

MTBF = 1000 yrs.
F = 25 MHz
a = 100 KHz
t_r = ?

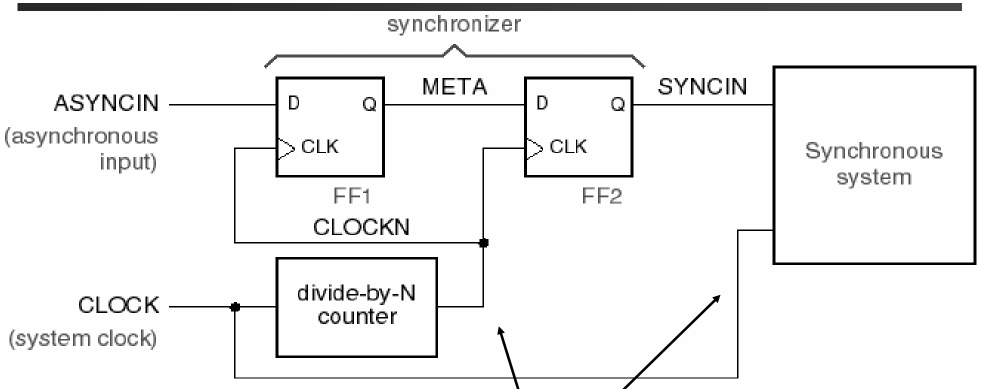


Device	*τ (ns)*	*T_o (s)*	*t_r (ns)*
74LS74	1.50	$4.0 \cdot 10^{-1}$	77.71
74S74	1.70	$1.0 \cdot 10^{-6}$	66.14
74S174	1.20	$5.0 \cdot 10^{-6}$	48.62
74S374	0.91	$4.0 \cdot 10^{-4}$	40.86
74F74	0.40	$2.0 \cdot 10^{-4}$	17.68
PALC16R8-25	0.52	$9.5 \cdot 10^{-12}$	14.22*
PALC22V10B-20	0.26	$5.6 \cdot 10^{-11}$	7.57*
PALCE22V10-7	0.19	$1.3 \cdot 10^{-13}$	4.38*
7300-series CPLD	0.29	$1.0 \cdot 10^{-15}$	5.27*
9500-series CPLD	0.17	$9.6 \cdot 10^{-18}$	2.30*

Is 1000 years enough?

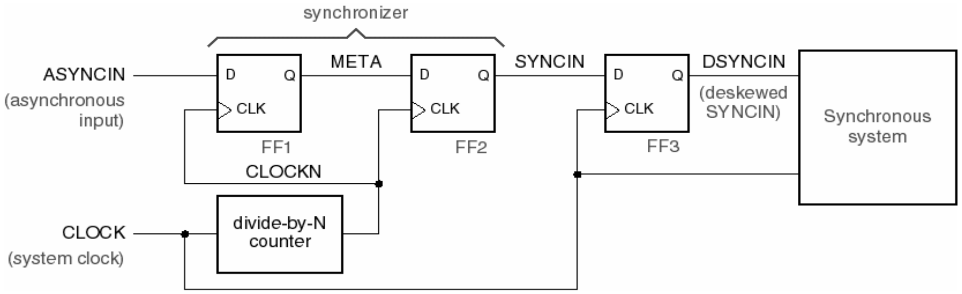
- If MTBF = 1000 years and you ship 52,000 copies of the product, then some system experiences a mysterious failure every week.
- Real-world MTBFs must be much higher.
- How to get better MTBFs?
  - Use faster flip-flops
    - ↗ But clock speeds keep getting faster, thwarting this approach.
  - Wait for multiple clock ticks to get a longer metastabilty resolution time
    - ↗ Waiting longer usually doesn't hurt performance
    - ↗ …unless there is a critical "round-trip" handshake.

Multiple-cycle synchronizer



- Clock-skew problem

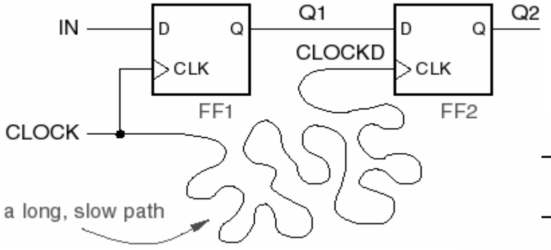
Deskewed multiple-cycle synchronizer

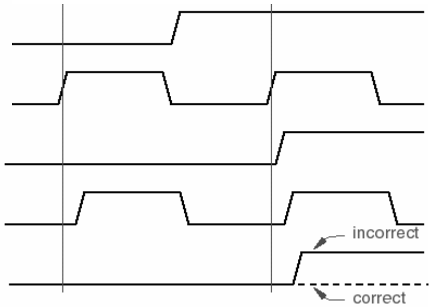


- Necessary in really high-speed systems
- DSYNCIN is valid for almost an entire clock period.

Clock Skew

- Clock signal may not reach all flip-flops simultaneously.
- Output changes of flip-flops receiving "early" clock may reach D inputs of flip-flops with "late" clock too soon.





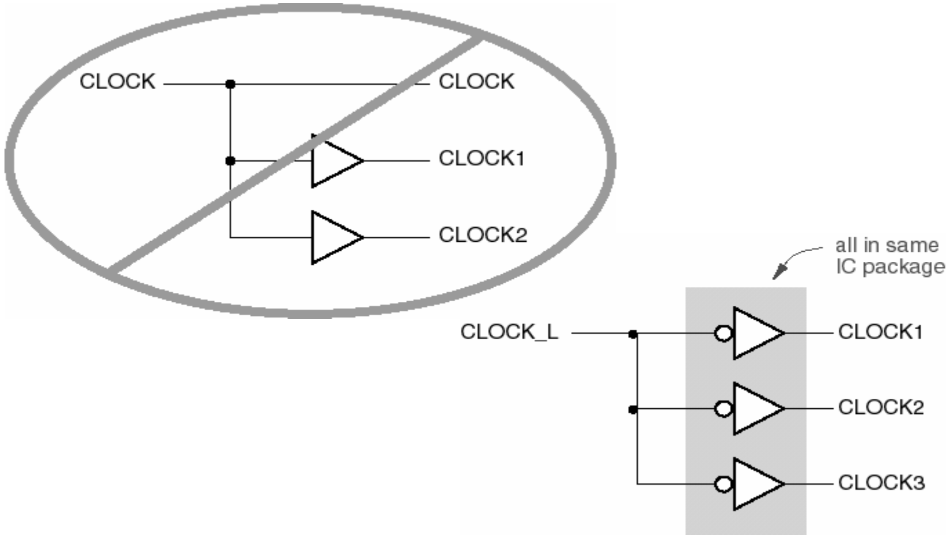
Reasons for slowness:
(a) wiring delays
(b) capacitance
(c) incorrect design

Clock-skew calculation

$$t_{\mathrm{ffpd(min)}} + t_{\mathrm{comb(min)}} - t_{\mathrm{hold}} - t_{\mathrm{skew(max)}} > 0$$

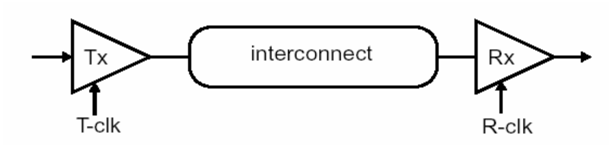
- First two terms are minimum time after clock edge that a D input changes
- Hold time is earliest time that the input may change
- Clock skew subtracts from the available hold-time margin
- Compensating for clock skew:
  - Longer flip-flop propagation delay
  - Explicit combinational delays
  - Shorter (even negative) flip-flop hold times

Example of bad clock distribution



Multiple Clock Domains

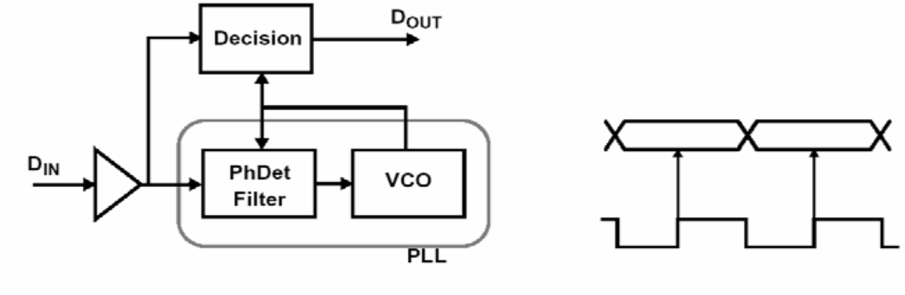
- Many digital systems have more than one clock domains:-



- Needs to synchronise the two clock domains using two basic building blocks:
  - Phase-locked loop (PLL)
  - Delay-locked loop (DLL)

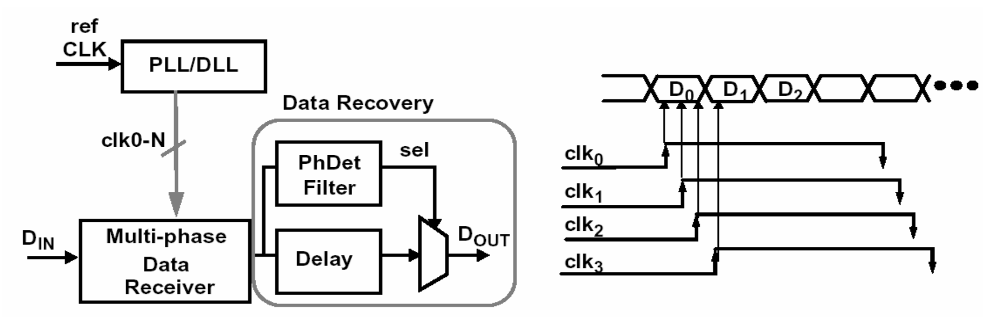
Example: Classical clock recovery

- Clocking information embedded in data stream
- Use PLL to recover the clock
- State of system is stored in analog loop filter



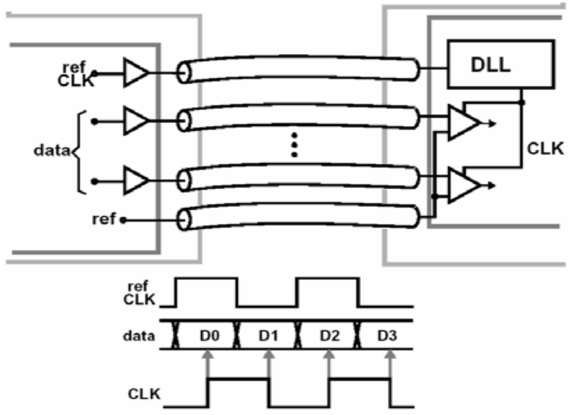
Oversampled Clock/Data Recovery

- ◆ Oversample the data and perform phase alignment digitally
- ◆ De-couples clock generation from tracking of data
- ◆ Data must guarantee transitions to ensure tracking

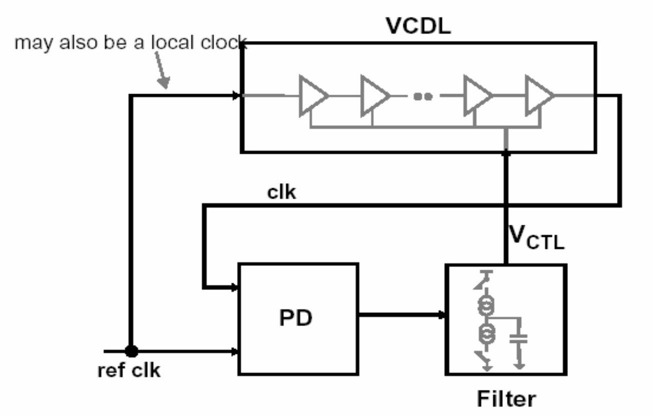


Phase Alignment in Source Synchronous Systems

- ◆ Timing information carried by reference clock
- ◆ Use DLL to ensure proper clock phase for sampling

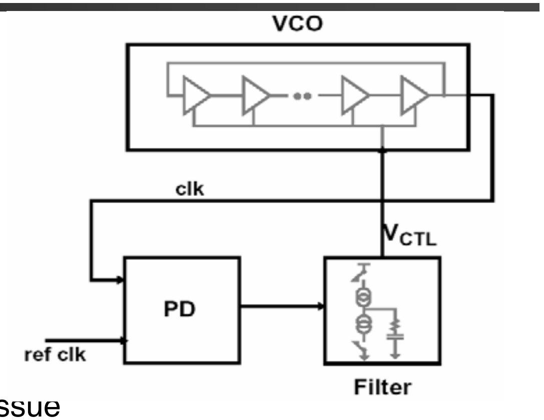


What is a Delay locked loop?



- ◆ First order loop:
  - easily stabilized
  - frequency synthesis is difficult
  - reference clock jitter passes to output
  - no phase error accumulation

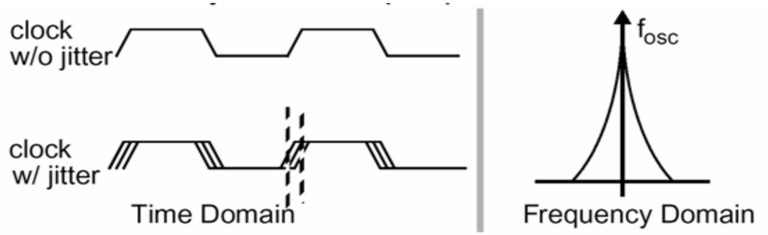
What is Phase locked loop?



- ◆ 2nd/3rd order loop:
  - stability could be an issue
  - frequency multiplication is easy
  - reference clock jitter reduced by filtering
  - phase error accumulation

Timing Loop Performance Parameters

- Phase Jitter:



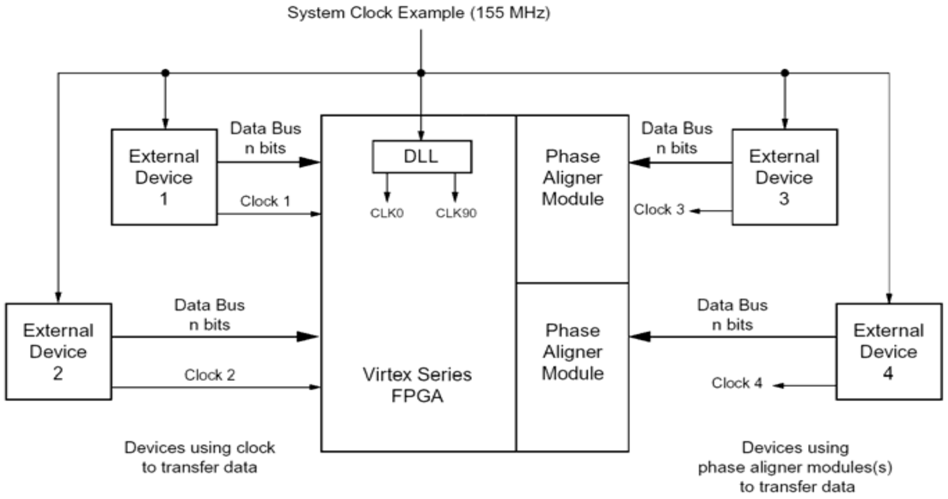
- Phase Offset
  - Error between output phase and reference phase
- Bandwidth
  - rate at which output phase tracks reference
- Acquisition time (to lock)
- Frequency range (lock range)

Clock Management with DLL

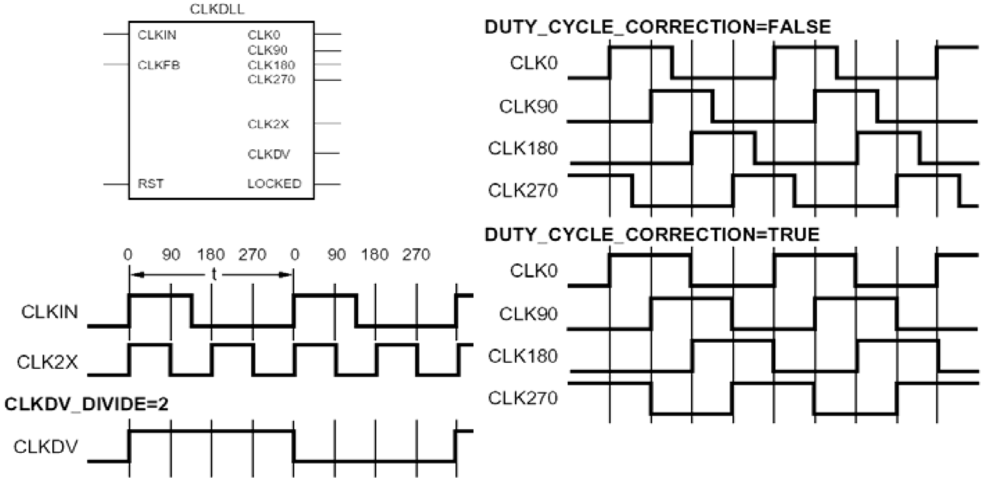
- Can eliminate on-chip clock delay
  - can also eliminate on-board clock delay
- 4 fixed-phase outputs (0°, 90 °, 180 °, 270 °)
- Selectable phase shift (n / 256 of the period)
  - through configuration
  - or through increment/decrement
  - 1/256 of clock period or 50 picosecond granularity
- Frequency synthesis (division and multiplication)
- Outputs are always phase-coherent

Solves the speed problem of large chips

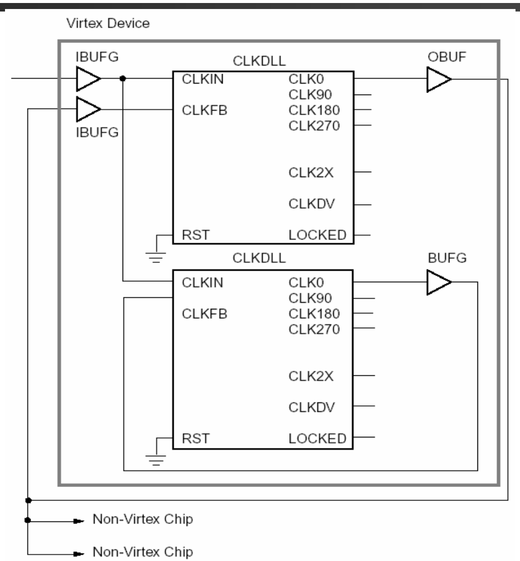
DLL in Xilinx Virtex data/clock alignment



Xilinx DLL with various phase outputs

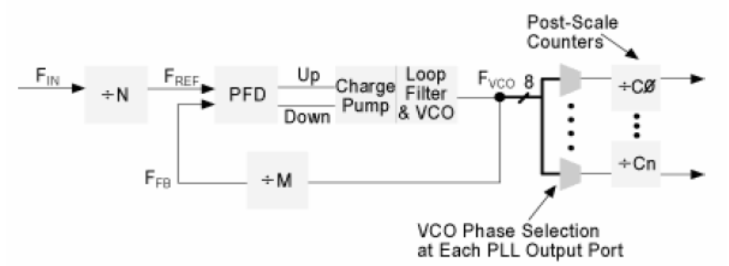


Using DLL to de-skew onboard clock signals



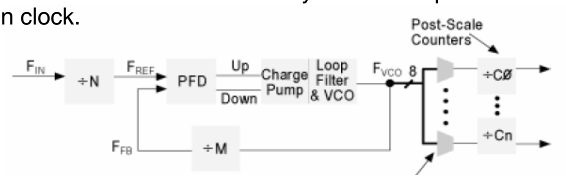
Altera Cyclone II PLL (1)

- Phase-locked loop (PLL) is a closed-loop frequency-control system based on the phase difference between the input clock signal and the feedback clock signal of a controlled oscillator.
- Main components:
  - Phase frequency detector (PFD)
  - Charge pump & loop filter
  - Voltage controlled oscillator (VCO)
  - Counters (N – pre-scale, M – feedback, C – post-scale)



Altera Cyclone II PLL (2)

- PLL aligns the rising edge of reference input clock to feedback clock using the PFD.
- PFD detects difference in phase and frequency between reference clock and feedback clock and generates an “up” or “down” control signal based on whether the feedback frequency is lagging or leading the reference frequency.
- If the charge pump receives an up signal, current is driven into the loop filter, otherwise, current is drawn from the loop filter.
- Loop filter converts these “up” “down” signals to a control voltage to control the oscillation frequency of the voltage controlled oscillator (VCO).
- Feedback loop counter (M) is used to increase VCO frequency above input reference frequency.
- Pre-scale counter (N) is used to produce the reference frequency from F_{IN}.
- The post-scale counters (C) allows a number of harmonically related frequencies be generated from one common clock.



Altera Cyclone II PLL (3)

- The output frequency is given by:
  - $F_{REF} = F_{IN} / N$
  - $F_{VCO} = F_{REF} \times M = F_{IN} \times M/N$
  - $F_{OUT} = F_{VCO} / C = (F_{REF} \times M) / C = (F_{IN} \times M) / (N \times C)$

where:

- F_{VCO} = VCO frequency
- F_{IN} = input frequency
- F_{REF} = reference frequency
- F_{OUT} = output frequency
- M = counter (multiplier), part of the clock feedback path
- N = counter (divider), part of the input clock reference path
- C = post-scale counter (divider)

References for this topic

- Chapter 8, pp757-773, Digital Design Principles & Practices, John Wakerly.
- “Metastability in Altera Devices”, Altera App Note 42.
- “Using the ClockLock &ClockBoost PLL Features in APEX Devices”, Altera App Note 115.
- “Using the Virtex Delay-Locked Loop”, XAPP-132.
- “Advantages of APEX PLLs Over Virtex DLLs”, Altera TB60.