Timing Issues

Outline

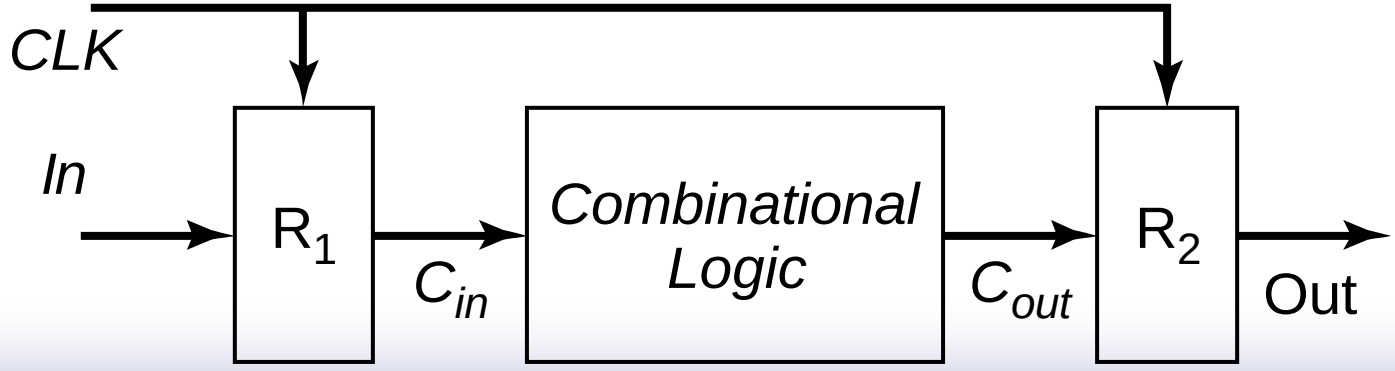
- Timing Classifications: Synchronous vs. Asynchronous
- Skew and Jitter
- Sources of skew and jitter
- Clock distribution techniques

Synchronous Timing

Functionality of synchronous systems is ensured by imposing strict constraints on the generation of the clock signals and their distribution to the memory elements distributed over the chip.

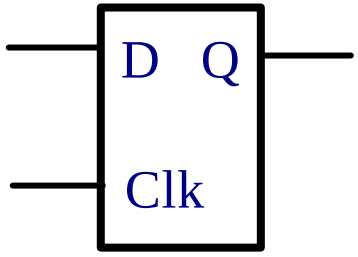
A synchronous signal has the same frequency as the local clock, and maintains a known fixed phase offset to that clock (signal is *synchronized* to clock). The input data signal *In* is sampled with register R_1 to produce signal C_{in}, which is synchronous with the system clock, and is then passed to the combinational logic block. After a suitable setting period, the output C_{out} becomes valid. Its value is sampled by R_2 which synchronizes the *output* with the clock. In a sense, the *certainty period* – period during which data is valid - of signal C_{out} is synchronized with the system clock. This allows register R_2 to sample the data with complete confidence.

The length of the uncertainty period places (period during which data is not valid) places an *upper bound* on how fast a synchronous system can be clocked.

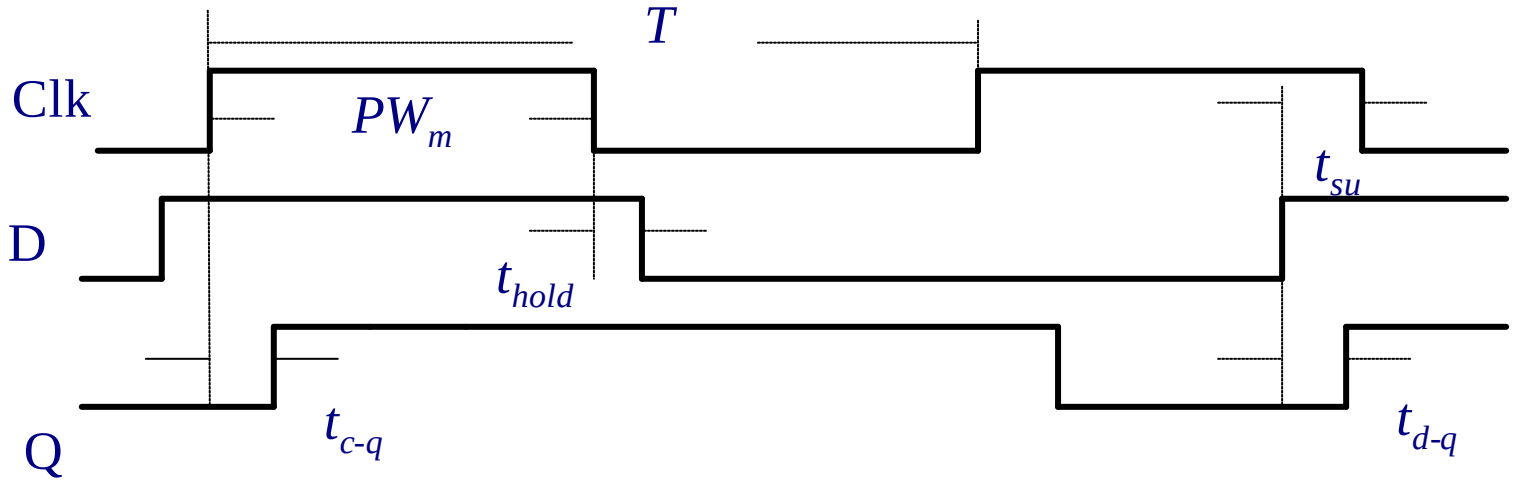


Timing Definitions

Latch Parameters

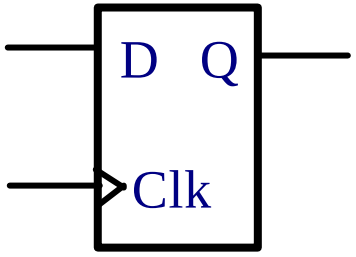


a latch is level sensitive, stores data when clock is low

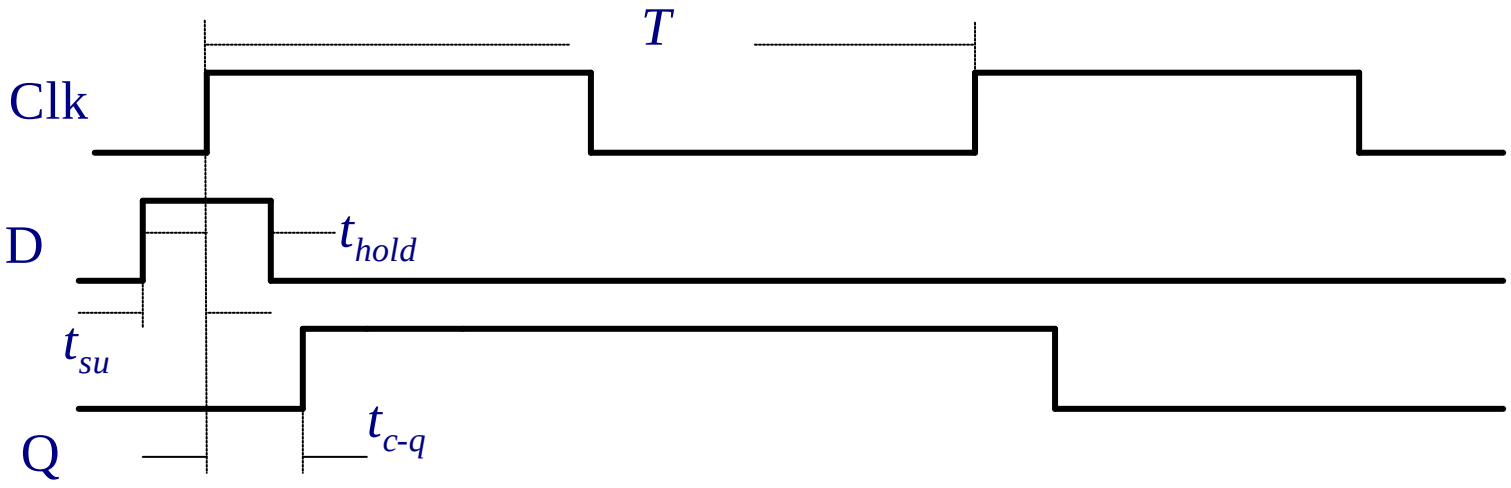


Delays can be different for rising and falling data transitions

Register Parameters

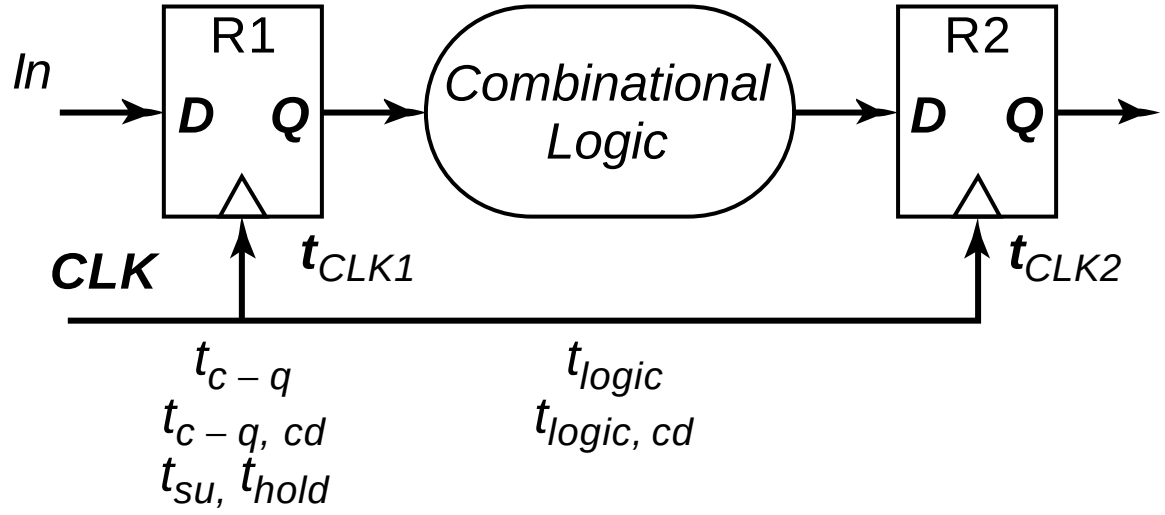


register is edge-triggered, stores data when clock rises



Delays can be different for rising and falling data transitions

Timing Constraints (+ve edge triggered)

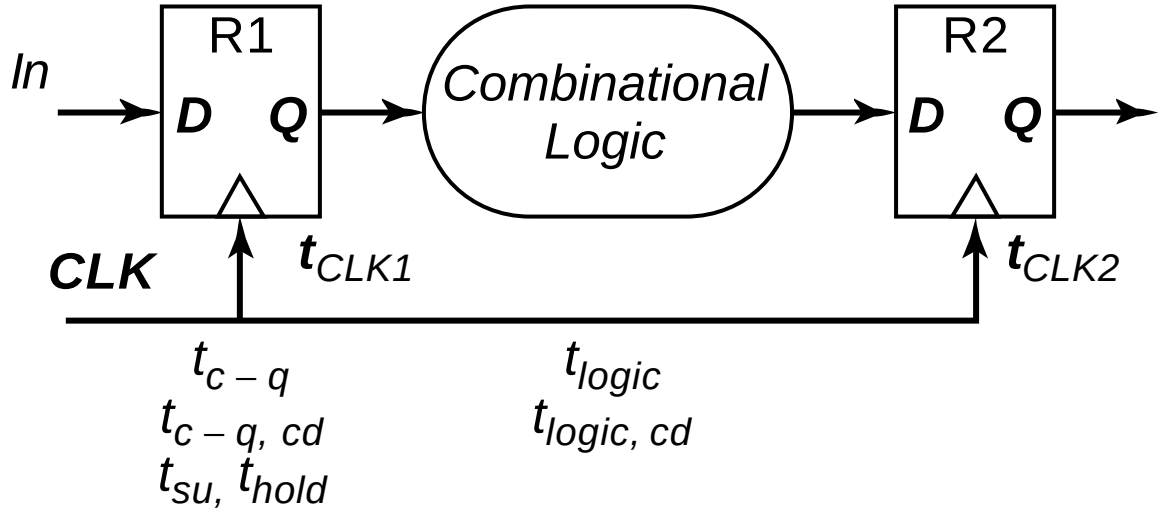


Minimum cycle time:

$T > t_{c\text{-}q} + t_{su} + t_{logic}$

Worst case is when receiving edge arrives early

Timing Constraints



Hold time constraint:

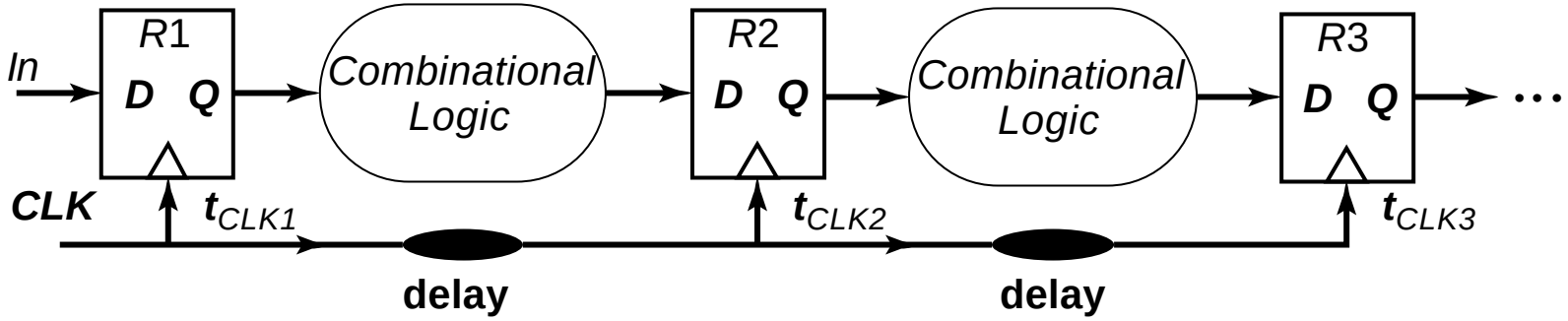
$$t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} > t_{hold}$$

Worst case is when receiving edge arrives late Race between data and clock

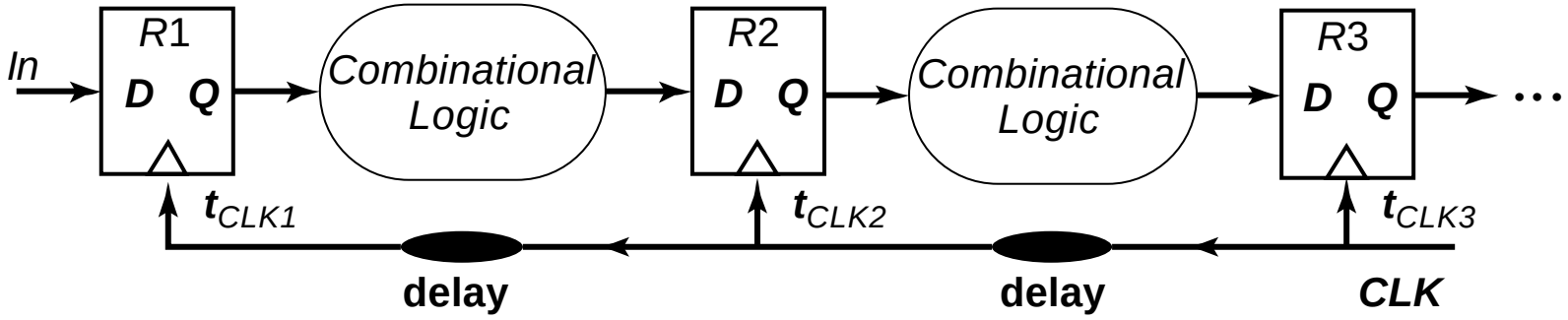
Clock Nonidealities

- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- **Variation of the pulse width**
  - Important for level sensitive clocking

Positive and Negative Skew



(a) Positive skew

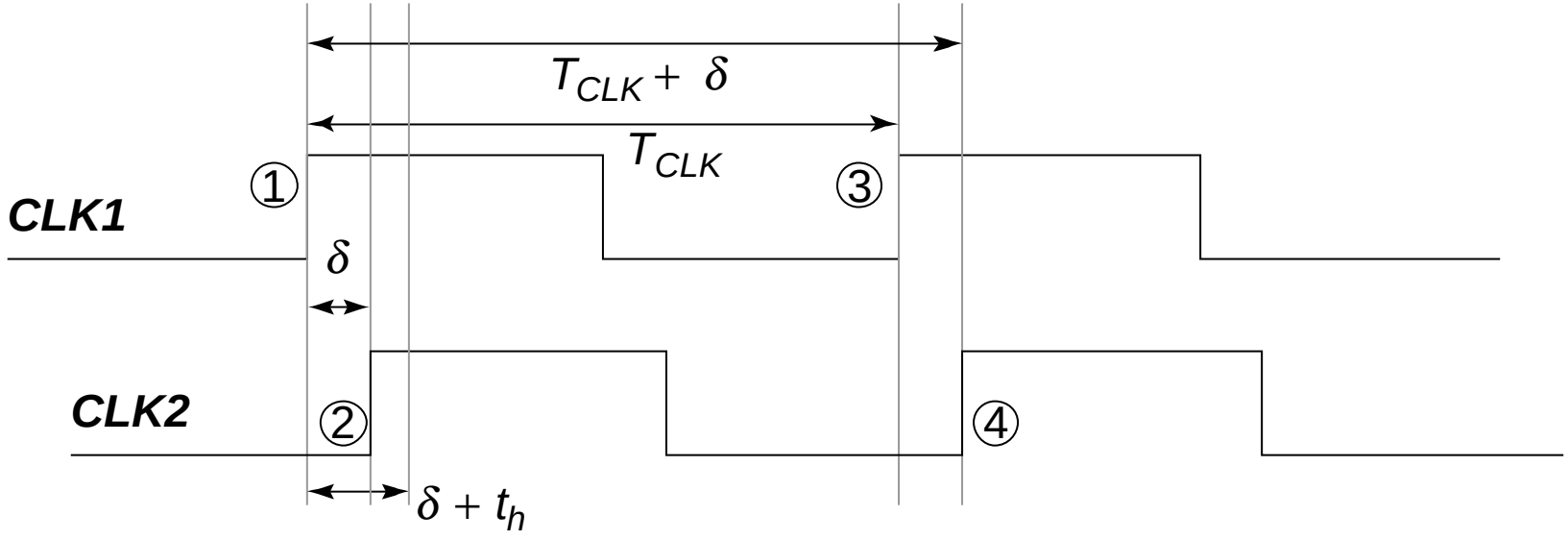


(b) Negative skew

The clock skew can be positive or negative depending upon the routing direction and position of the clock source.

Clock skew is caused by static mismatches in the clock paths and differences in the clock load. By definition, skew is constant from cycle to cycle. That is, if in one cycle CLK_2 lagges CLK_1 by δ, then on the next cycle, it will lag it by the same amount. Skew does not result in clock period variation, but only in phase shift.

Positive Skew



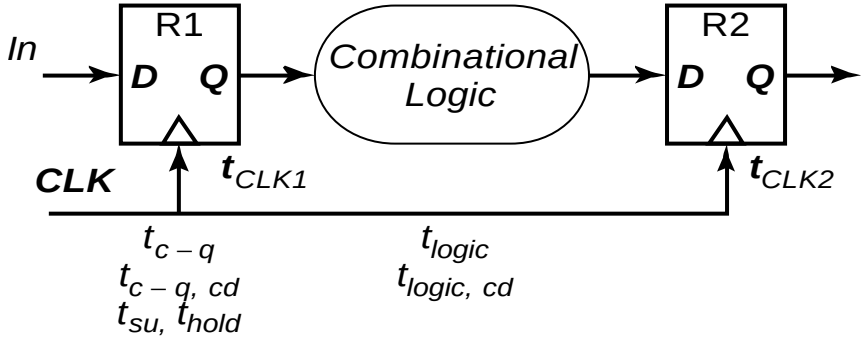
Launching edge arrives before the receiving edge

$$T + d > t_{c\text{-}q} + t_{su} + t_{logic} \quad \textbf{or} \quad T > t_{c\text{-}q} + t_{su} + t_{logic} - d$$

This equation actually suggests that clock skew actually has the potential to improve the performance of the circuit. That is, the minimum clock period required to operate the circuit reliably reduces with increasing clock skew! ***Yes!*** *However, unfortunately, increasing the skew makes the circuit more susceptible to race conditions, which may harm the correct operation of sequential systems.*

Race Conditions

If input In is sampled on the rising edge of CLK_1 at edge 1 into R1. The new value at the output of R1 propagates through the combinational logic and should be valid before edge 4 at CLK_2. However, if the minimum delay of the combinational logic block is small, the inputs to R2 may change before the clock edge 2, resulting in incorrect evaluation. To avoid races, we must ensure that the minimum propagation delay through the register and logic is long enough that the inputs to R2 are valid for a hold time after edge 2.



Hold time constraint:

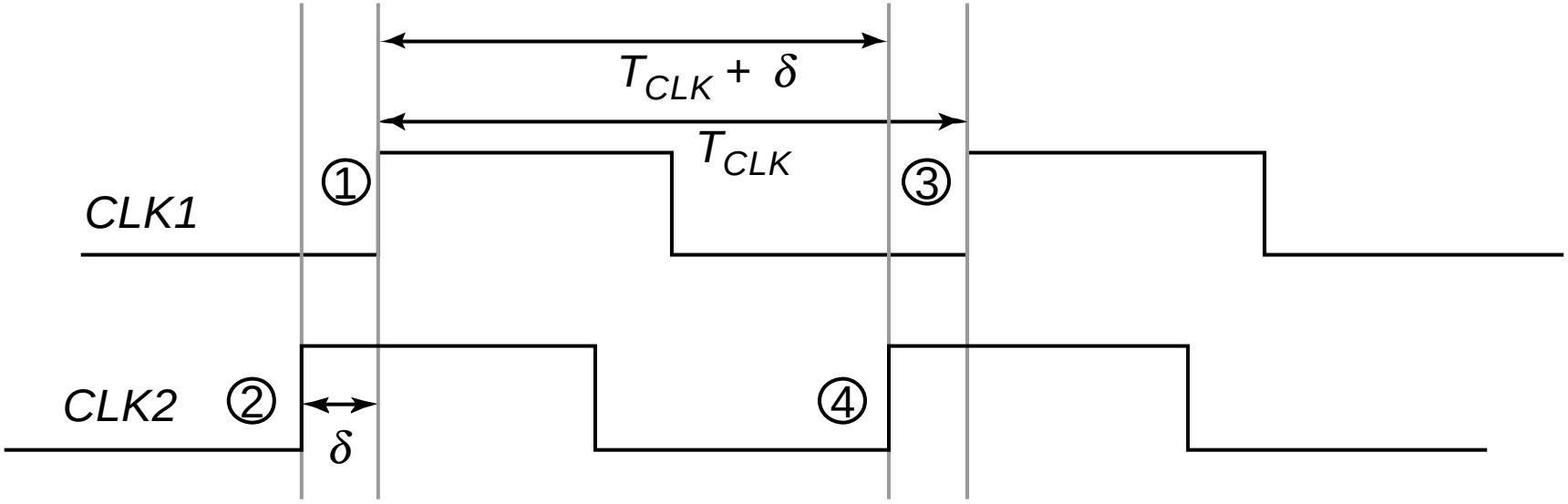
$t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} > t_{hold} + d$ **or** $t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} - t_{hold} > d$

Worst case is when receiving edge arrives late

Race between data and clock

Reducing the clock frequency of an edge-triggered circuit doesn't get around skew problems. It is necessary to satisfy the hold-time constraints at design time. +ve skew increases circuit throughput as the clock period is shortened by d. However, the extent of this improvement is limited, as large d provoke race conditions.

Negative Skew

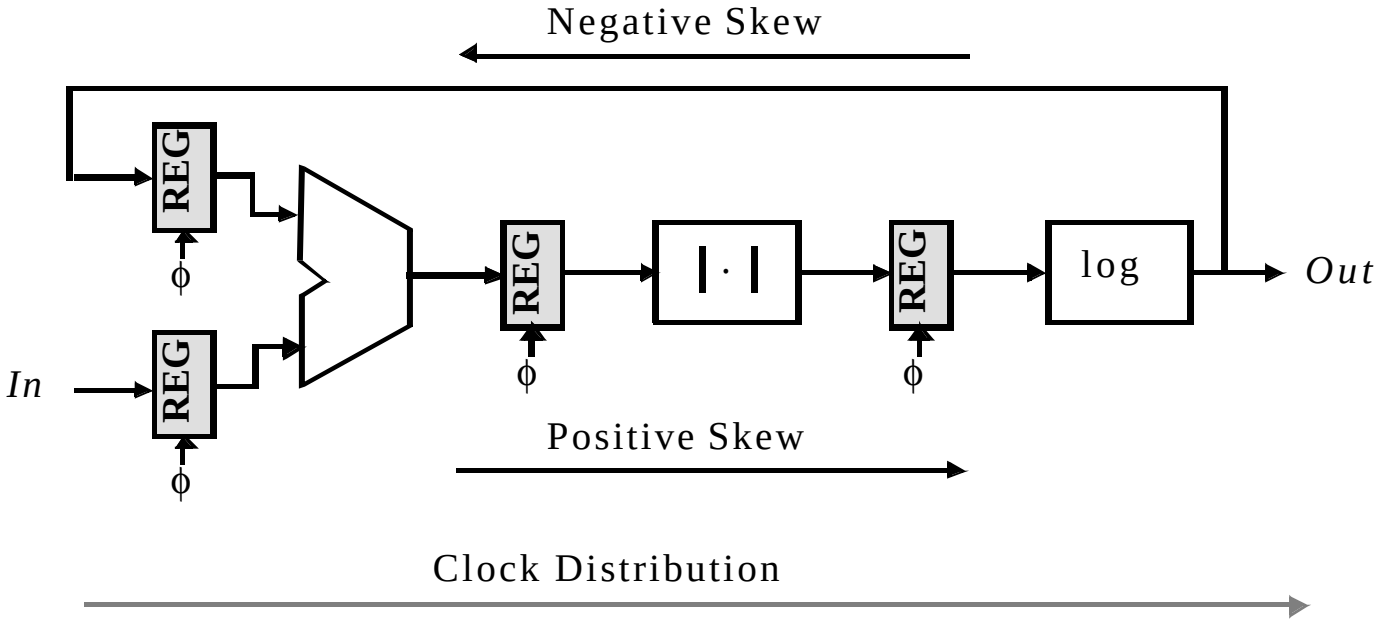


Receiving edge arrives before the launching edge

A negative skew adversely impacts the performance of a sequential system ($T > t_{c\text{-}q} + t_{su} + t_{logic} - d$). However, assuming that $t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} - t_{hold} > d$, a negative skew implies that the system never fails, since edge 2 happens before edge 1.

Significant immunity against race conditions. However, clock period increases by d, harming the circuit's performance

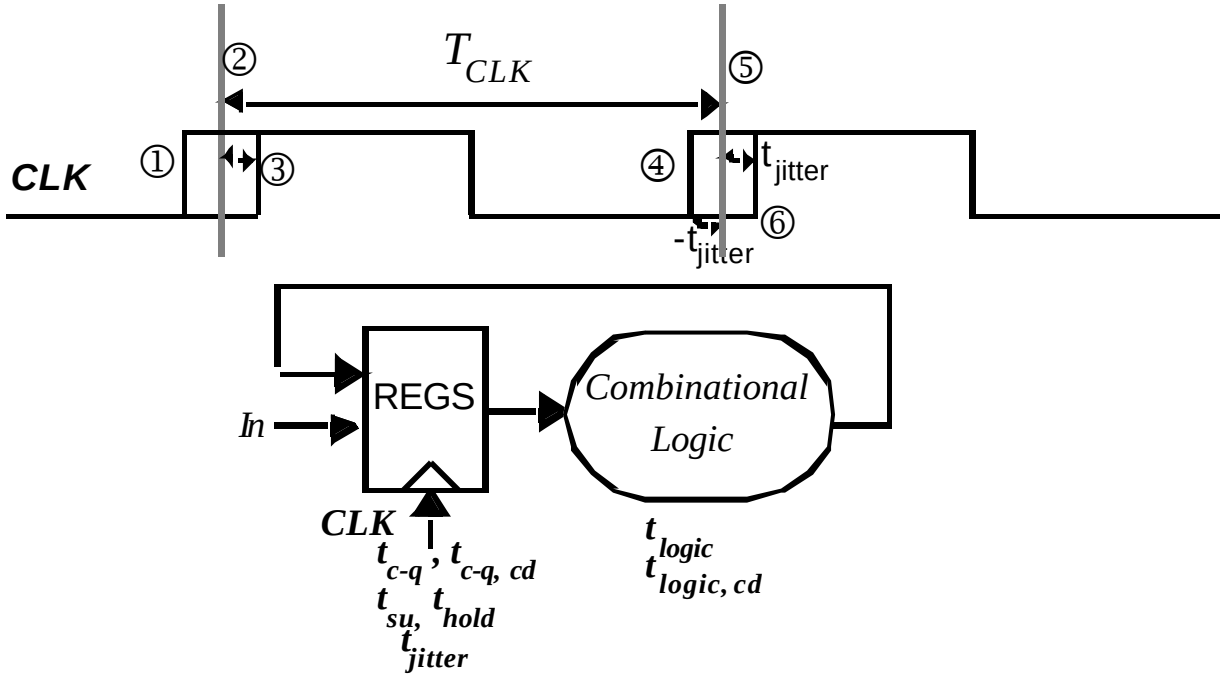
How to counter Clock Skew?



Data and Clock Routing

Since a general logic circuit can have data flowing in both directions (circuits with feedback), this solution to eliminate races does not always work. The skew can assume both positive and negative values, depending on the direction of the data transfer. Under these circumstances, the designer has to account for the worst case skew condition. In general, routing the clock so that only negative skew occurs is not feasible. Therefore, the design of a low-skew clock network is essential.

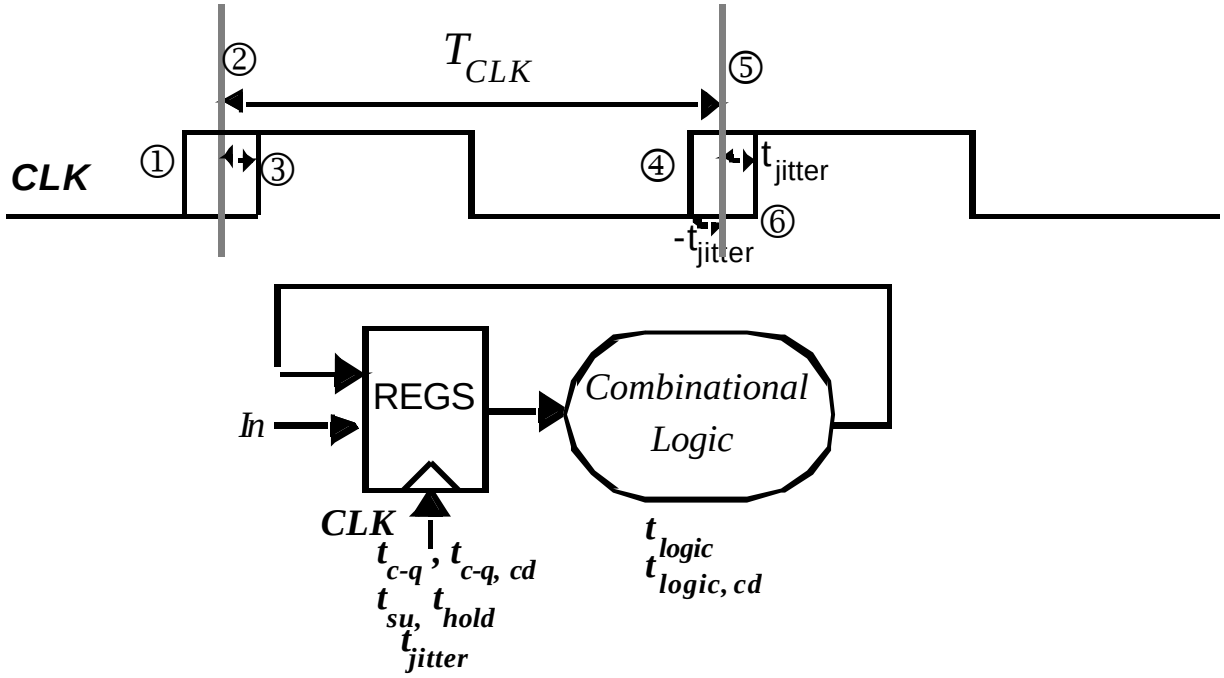
Impact of Jitter



Clock jitter

Temporal variations (uncertainty) of the clock period at a given point on the chip – the clock period can reduce or expand on a cycle-by-cycle basis. It is a *zero mean random variable*, and can be characterized as *absolute jitter* (t_{jitter}) - *worst case* variation of a clock edge with respect to an ideally periodic reference clock edge. The cycle-to-cycle jitter (T_{jitter}) refers to the *time varying* deviations of a single clock period relative to an ideal reference clock.

Jitter



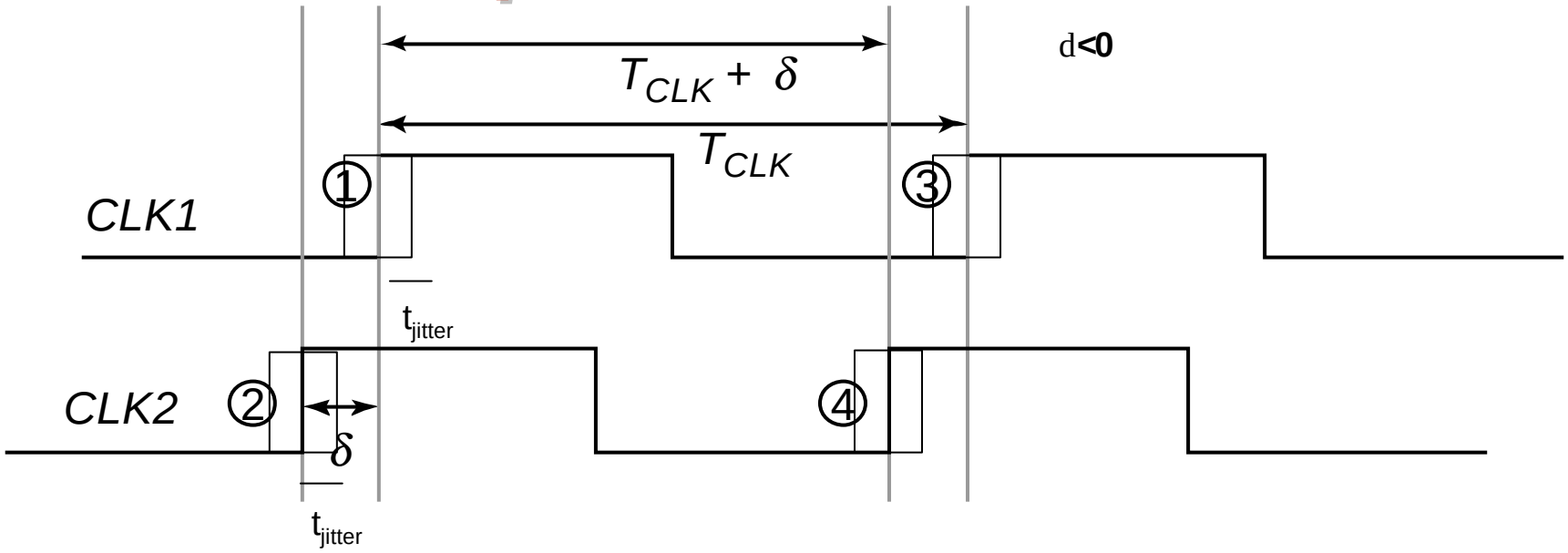
Under the worst case conditions, the magnitude of the cycle-to-cycle jitter equals twice the absolute jitter ($2t_{jitter}$).

Jitter impacts the performance of a sequential system. The total time available to complete the operation is reduced by $2t_{jitter}$ in the worst case and is given by:

$$T_{CLK} - 2t_{jitter} > t_{c\text{-}q} + t_{su} + t_{logic} \quad or \quad T_{CLK} > t_{c\text{-}q} + t_{su} + t_{logic} + 2t_{jitter}$$

This happens when the leading edge of the current clock period is delayed by t_{jitter}, while the leading edge of the next clock period is delayed by t_{jitter} .. Jitter directly reduces the performance of a sequential circuit. Keeping jitter within strict bounds is essential if one is concerned about performance.

Combined Impact of Skew and Jitter



$T_{CLK}+d - 2t_{jitter} > t_{c\text{-}q} + t_{su} + t_{logic}$ *or* $T_{CLK} > t_{c\text{-}q} + t_{su} + t_{logic} - d + 2t_{jitter}$

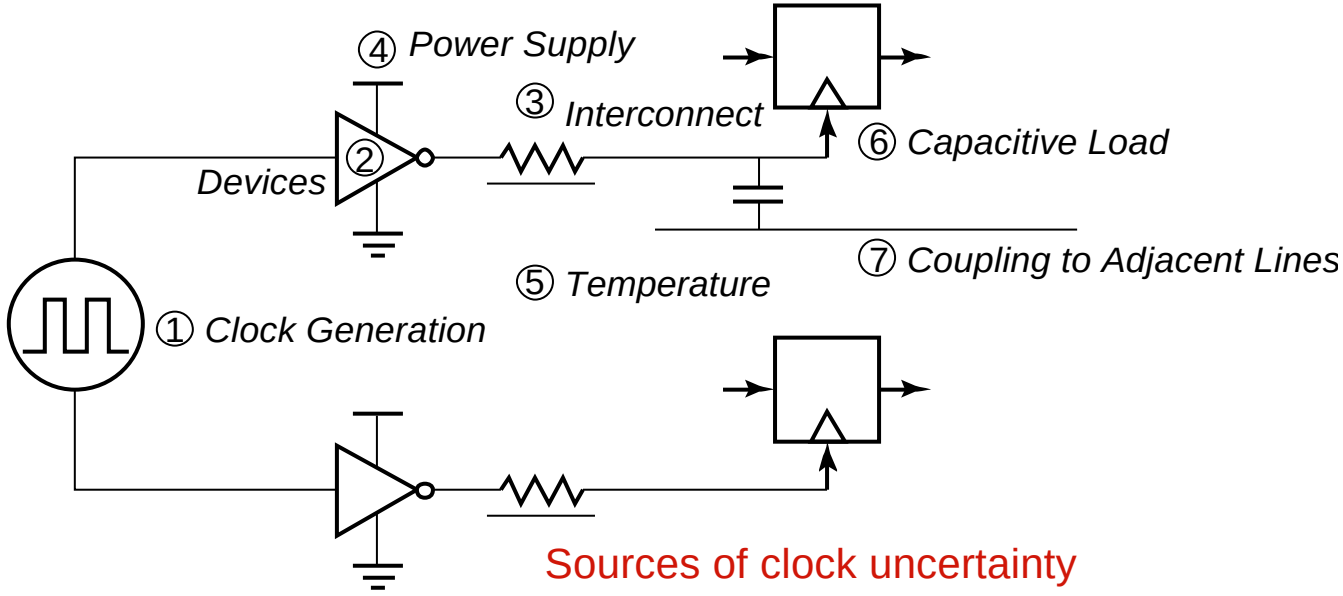
If $\delta>0$ (+ve skew), this would provide a performance advantage. On the other hand, jitter always has a negative impact on the minimum clock period. To formulate the minimum delay constraint:

$t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} > t_{hold} + d + 2t_{jitter}$ *or* $t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} - t_{hold} - 2t_{jitter} > d$

This analysis is for worst case. It assumes that the jitter values at the source and destination nodes are independent statistical variables. In reality, the clock edges involved in the hold-time analysis are derived from the same clock edge and statistically dependent. Taking this dependence into account reduces the timing constraints substantially.

Clock Uncertainties

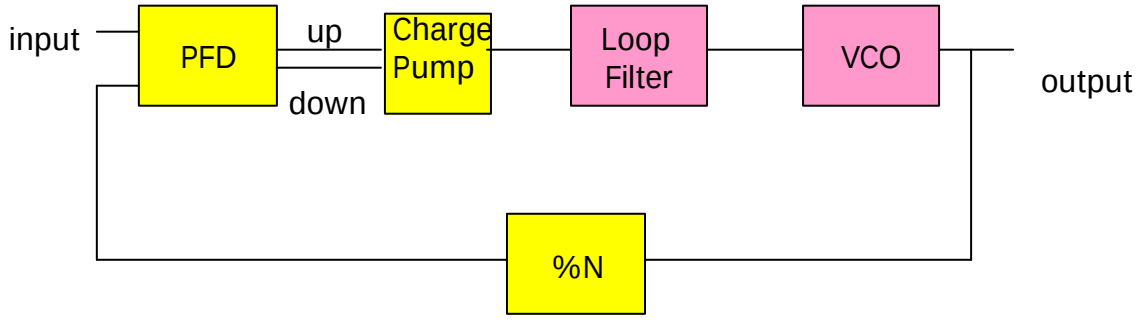
The clock is never ideal. The different clock events turn out to be neither periodic nor perfectly simultaneous. As a result, process and environmental variations, the clock signal can have both spatial and temporal variations, which lead to performance degradation or circuit malfunction.



Sources of clock uncertainty are classified in several ways. Firstly, errors can either be systematic or random. Systematic errors are normally identical from chip to chip and are predictable (variation in total load capacitance of each block path). These errors can be modeled and corrected/compensated at design time. Random errors are due to manufacturing variations that are difficult to model and eliminate (V_t variations due to dopant fluctuations).

1) Clock Signal Generation

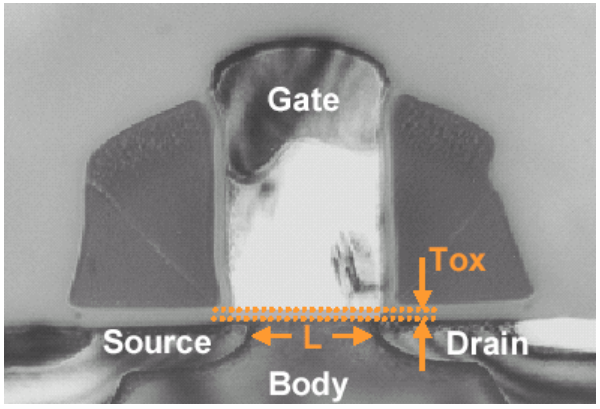
The generation of the clock signal itself causes **jitter**. A VCO is a key block to generate a high-frequency global reference for the processor from a low-frequency reference clock signal. VCO's are sensitive to intrinsic device noise and V_{dd} variations, and coupling from surrounding noisy digital circuits through the substrate. These noise sources cause temporal variations in the clock signal that propagate unfiltered through the clock drivers to the flip-flops, and result in cycle-to-cycle period variations.



2) Manufacturing Device Variations

Distributed buffers are required to drive both the register loads and global and local interconnects. The matching of devices in those buffers along multiple clock path is critical to minimize timing uncertainty. As a result of **process variations, device parameters** in the buffers vary along different paths, resulting in ***static skew***. Process variations include T_{ox}, dopant variations, lateral dimensions (W,L). **Spatial variation** usually consists of a wafer-level (inter-die) and die-level (intra-die) variations. Part of these variations are systematic which can be modeled and compensated for (Will be discussed in depth in the “Variations Section”). Random variations are more difficult to model.

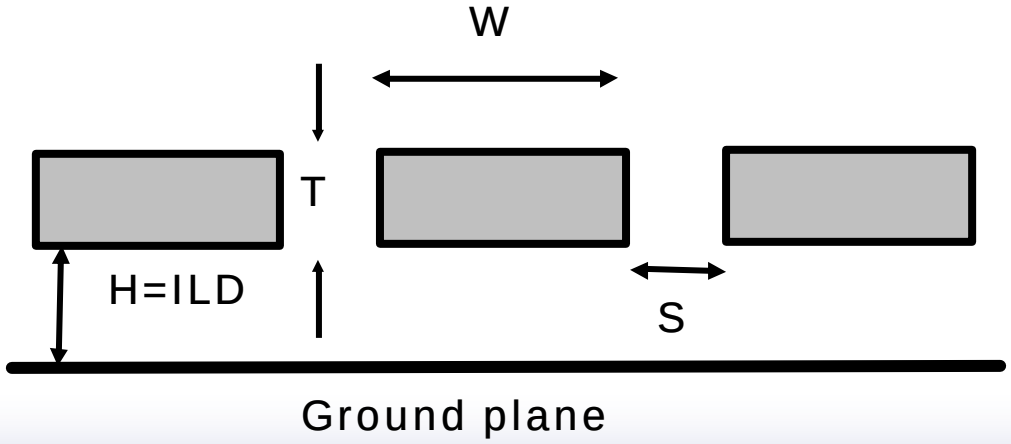
In addition the **orientation** of polysilicon can have an impact on the device parameters. Thus keeping the orientation the same across the chip for clock drivers is critical.



3) Interconnect Variations

Vertical and lateral dimensions variations cause the interconnect capacitance and resistance to vary across a chip. ***Static skew*** is produced. Sources of variations include Inter Layer Dielectric (ILD) thickness due to variations in Chemical Mechanical Polishing (CMP) process. Significant advances to develop analytical models for *estimating the ILD thickness variations based on spatial density*. Since this component is often predictable from the layout (CMP polish rate is dependent on circuit layout density), it is possible to actually correct for the systematic component at design time. Since there is a clear correlation between the density and the thickness of the dielectric, clock distribution networks must exploit such information in order to reduce clock skew.

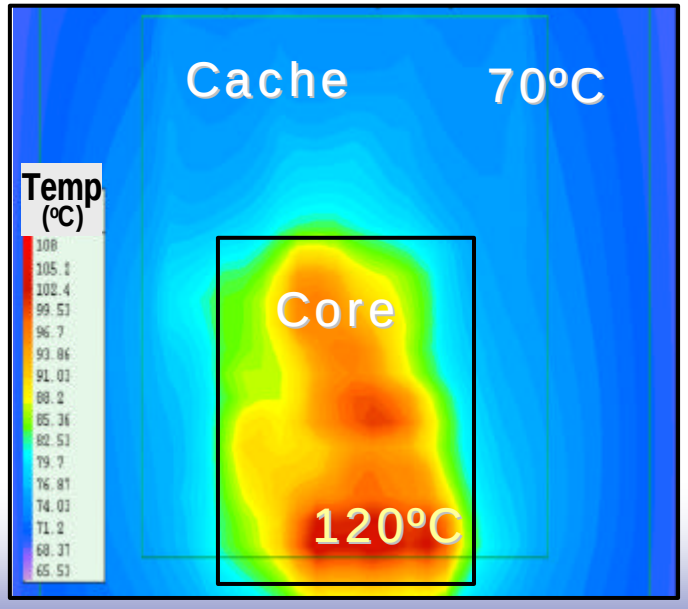
Variations of width of wires and line spacing, which result from lithography and etch dependencies. The width is a critical parameter because it directly impacts the resistance of the line, and the wire spacing affects the wire-to-wire capacitance.



4) Temperature Variations

Temperature and V_{dd} variations are the most significant contributors to skew and jitter. Temperature gradients across the chip result from variations in power dissipation across the die. Temperature variation is an important issue with clock gating, where some parts of the chip may be idle, where other parts are fully active (i.e., difference in temperature). Since device parameters - such as threshold and mobility - depend strongly on temperature, the buffer delay for a clock distribution network varies drastically from path to path. More importantly, this component is time varying, since the temperature changes as the logic activity of the circuit varies. Thus, the worst case variation in temperature must be simulated (statistical analysis is also done).

The difference in temperature is time varying, but the changes are relatively slow (time constants for temperature change on the order of milliseconds). Thus, it is considered as a skew component. By using feedback, temperature is calibrated and compensated for its effect.



5) V_{dd} Variations

V_{dd} variations are the major source of **jitter** in clock distribution networks. Buffer delays are a strong function in V_{dd}. As with temperature, V_{dd} is a strong function of the switching activity. Therefore, the buffer delay varies strongly from path to path.

Power supply variations can be classified into slow- (or **static**) and **high-frequency variations**. Static power-supply variations result from fixed currents drawn from various modules, while high-frequency result from instantaneous IR drops along the power grid due to fluctuations in switching activity. Inductive effects on the power supply also are a major concern since they cause voltage fluctuations. **Clock gating has exacerbated this problem**, because the logic transitions between the idle and active states can cause major changes in current drawn from the supply. Since V_{dd} can change rapidly, the period of the clock signal is modulated on a cycle by cycle basis, resulting in jitter. The jitter on two different clock points may be correlated or uncorrelated, depending on how the power network is configured and the profile of switching patterns.

High-frequency V_{dd} changes are difficult to compensate for, even with feedback techniques. **Thus, V_{dd} noise fundamentally limits the performance of clock networks.** Decoupling capacitors are usually added around major clock drivers to minimize V_{dd} variations.

6) Capacitive Coupling

Changes in capacitive load contribute to timing uncertainty. These changes are due to: 1) coupling between clock lines and adjacent signal wires, and 2) variation in capacitances (gate caps. of latches and registers, and wire caps. for interconnect lines). The load capacitance is highly nonlinear and depends on the applied voltage.

Since the adjacent signal can transition in arbitrary directions and at arbitrary times, the exact coupling to the clock network is not fixed from cycle to cycle, causing jitter. For many latches and registers, the clock load is a function of the current state of the latch/register, as well as the next state. This causes the delay through the clock buffers to vary from cycle to cycle, which causes jitter.

Clock Distribution

The design of a clock network must minimize both skew and jitter. In addition, the majority of the power is dissipated in the clock network. Clock conditioning (gating) is an effective technique to reduce power, however, it produces additional clock uncertainty (as mentioned previously).

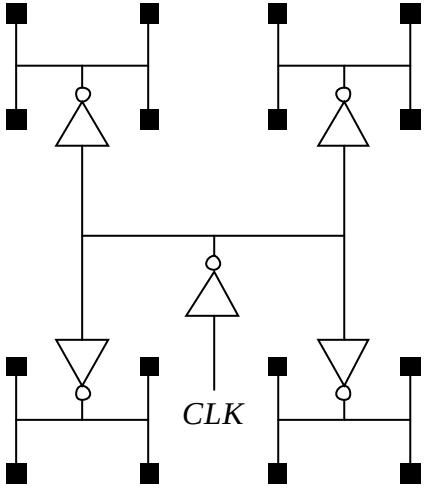
The design of the clock network depends on 1) the type of material used for wires, 2) the basic topology and hierarchy, 3) the sizing of wires and buffers, 4) the rise and fall times, and 5) the partitioning of load capacitances.

Fabrics for Clocking

Clock networks include a network that is used to distribute a **global reference** to various parts of the chip, and a final stage that is responsible for **local distribution** of the clock.

A common approach to distributing a clock is to use balanced paths (**tree**) to reduce the relative phase between two clocking points. **H-tree** is a popular clock distribution. Although it may take multiple clock cycles for a signal to propagate from the central point to the leaf nodes, the arrival times are identical at every leak node (skew=0). In reality, variations cause skew and jitter.

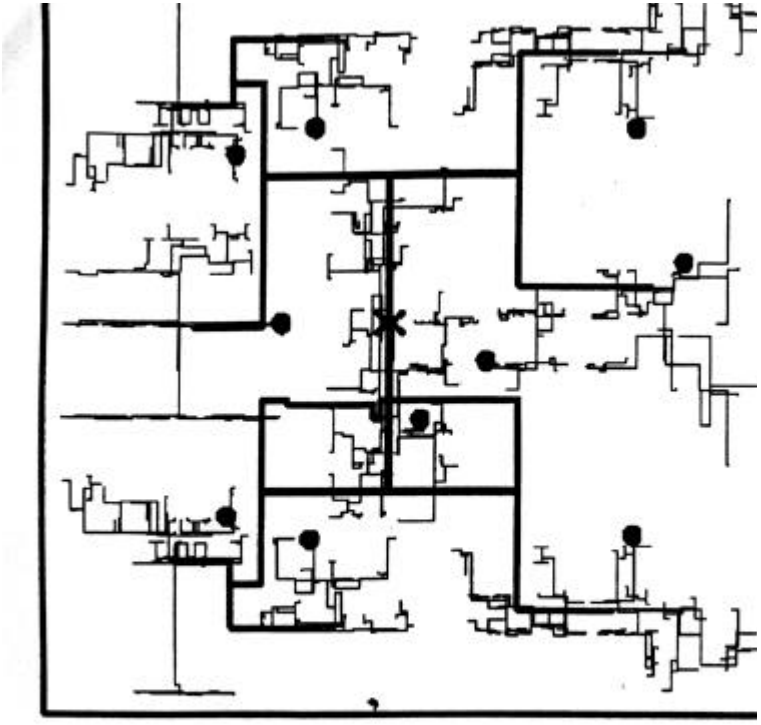
The H-tree configuration is particularly useful for regular array networks in which all elements are identical. The more general approach, referred to as matched RC trees, represents a floorplan that distributes the clock signal so that the interconnections carrying the clock signals to the functional sub-blocks are of equal length (i.e., not a regular physical structure).



H-tree

More realistic H-tree – Matched RC

The chip is partitioned into 10 balanced load segments (tiles). The global clock driver distributes the clock to the tile drivers. A lower level RC-matched tree is used to drive 580 additional drivers inside each tile.



RC-matched distribution for an IBM microprocessor

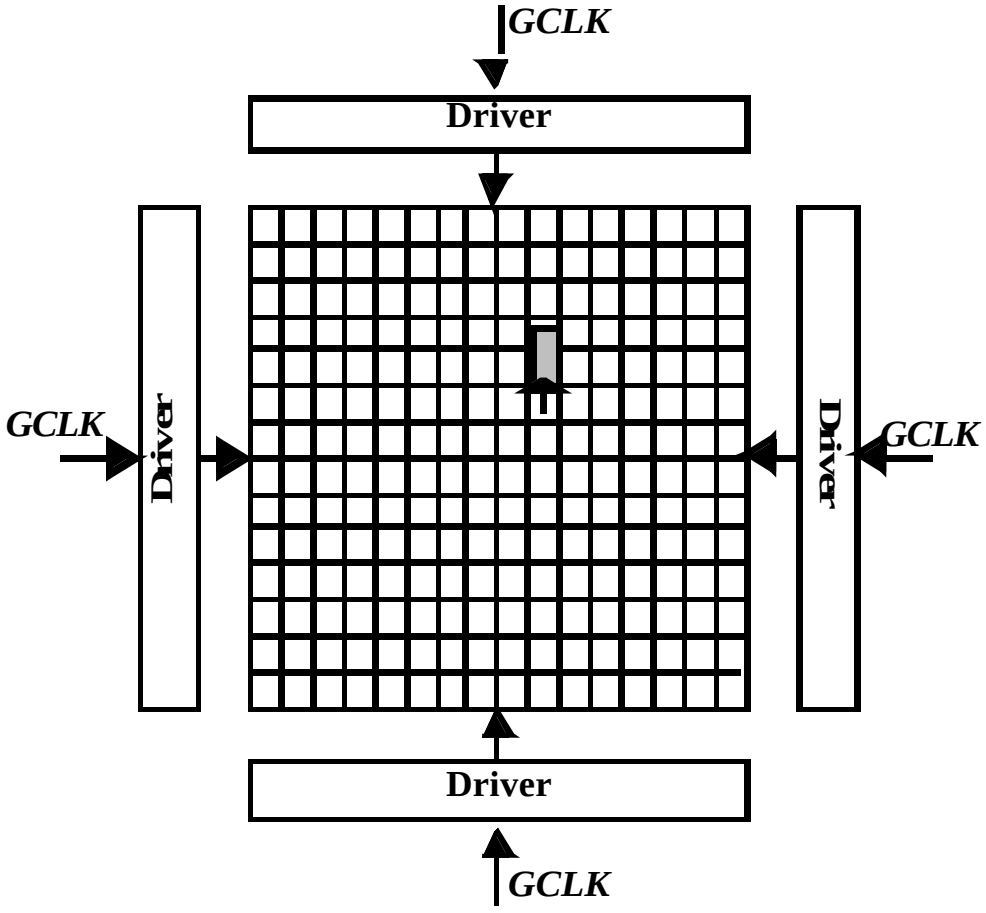
The Grid System

Grids are typically used in the final stage of a clock network to distribute the clock to the clocking element loads. Here, the main difference with the balanced RC approach is that the delay from the final driver to each load is not matched. Rather, the absolute delay from the final driver is minimized.

The major advantage of this grid structure is that it allows for late design changes, since the clock is easily accessible at various points on the die.

The penalty is large power dissipation since the structure has a lot of excess interconnect.

In general, it is important to consider clock distribution in the early phases of the design of a complex circuit, as it influences the shape of the floorplan. If considered after chip layout has been done, this would result in unwieldy clock networks and multiple timing constraints that hamper performance and the operation of the final circuit.



- *No rc-matching*
- *Low skew distribution*
- *Large power*

Example: DEC Alpha 21164

Clock Frequency: 300 MHz - 9.3 Million Transistors (0.5μm CMOS tech.)

Total Clock Load: 3.75 nF (extensive use of dynamic logic)

Power in Clock Distribution network : 20 W (40% of P_{total})

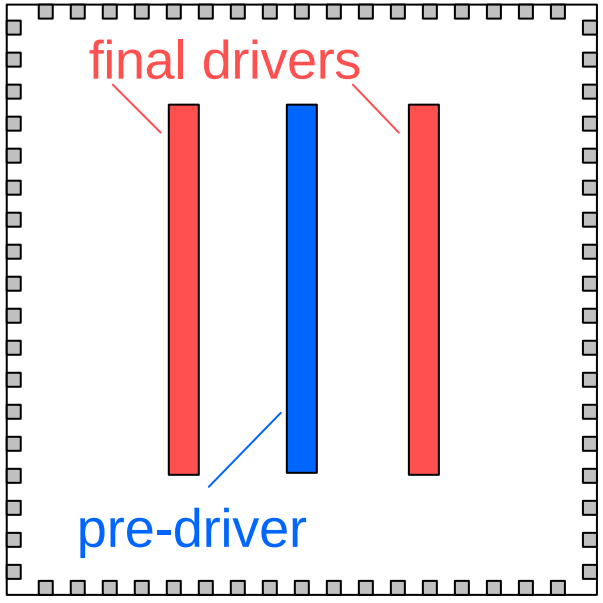
Uses Two Level Clock Distribution:

- **Incoming clock signal is routed through a single 6-stage driver at center of chip**
- **Secondary buffers drive left and right side between the secondary cache memory and the outside edge of the execution unit. The produced clock signal is driven into a grid of Metal3 and Metal4.**

Total driver size: 58 cm!

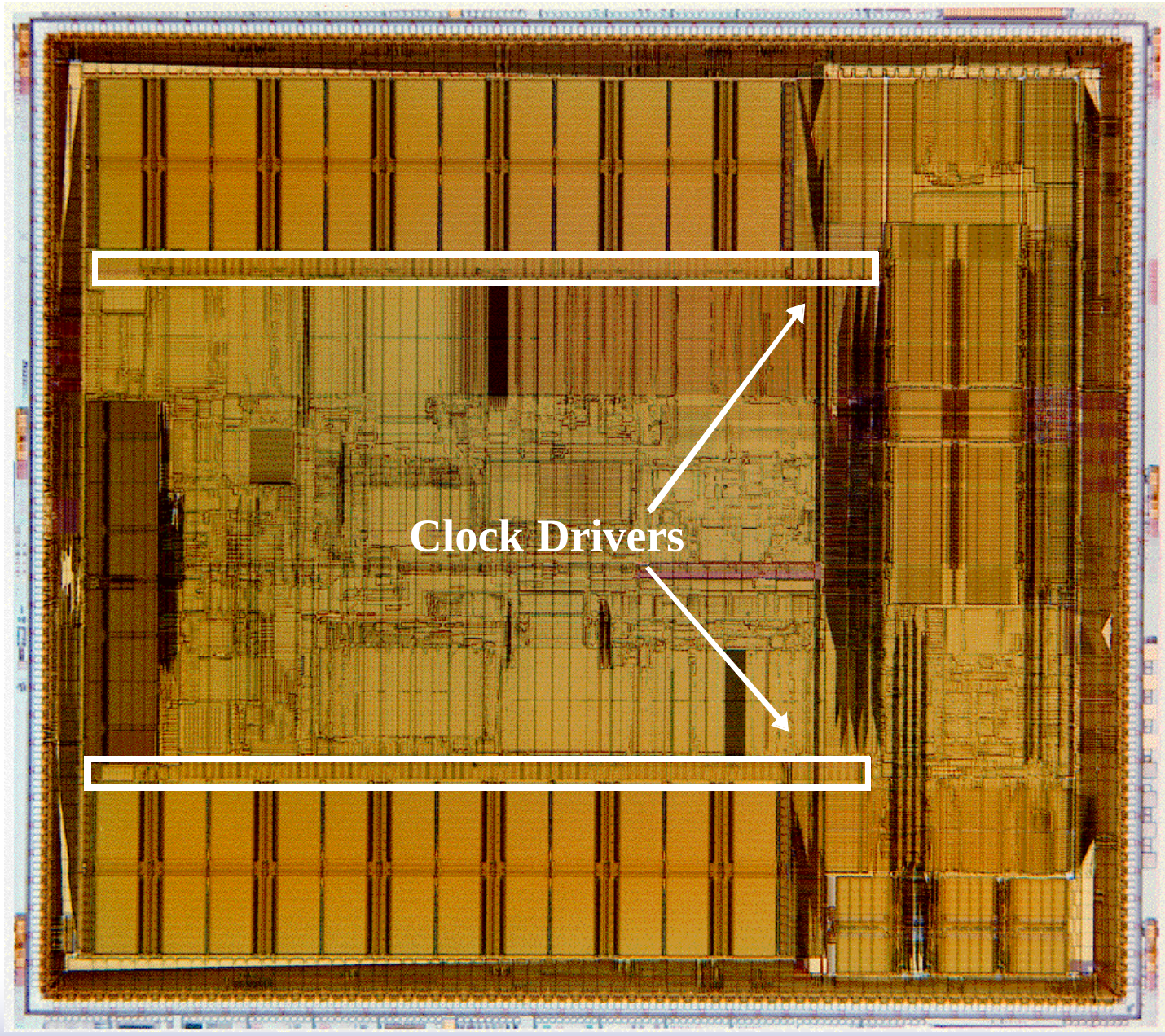
21164 Clocking

Clock waveform

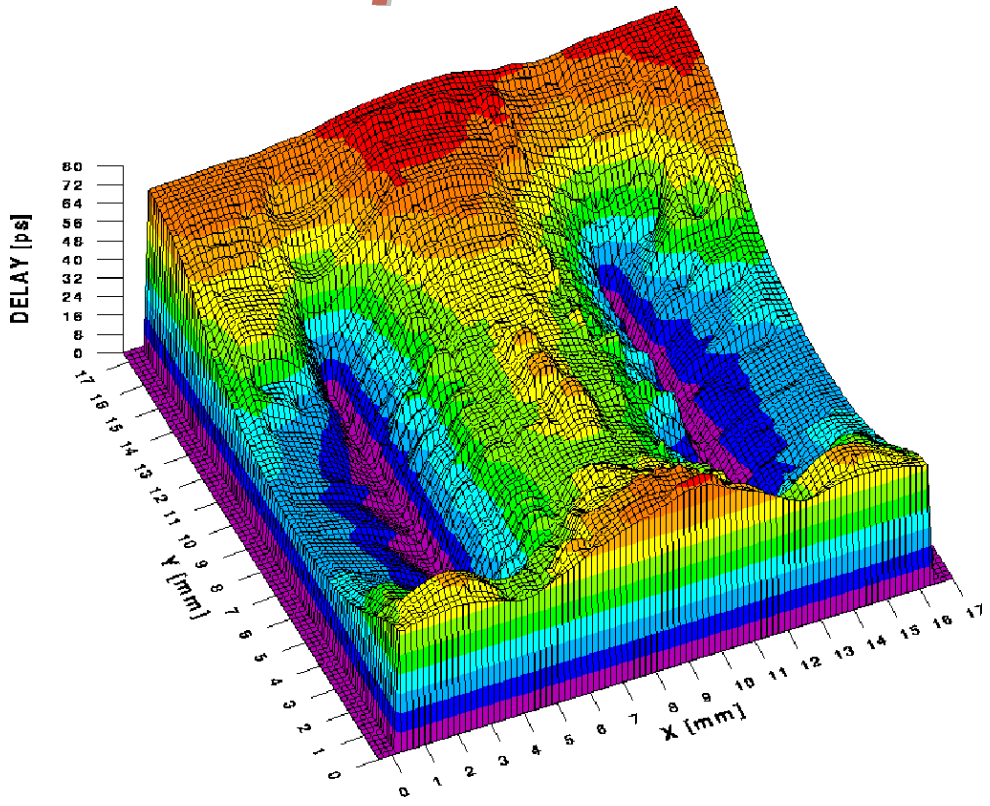


Location of clock driver on die

- 2 phase single wire clock, distributed globally
- 2 distributed driver channels
  - Reduced RC delay/skew
  - Improved thermal distribution
  - 3.75nF clock load
  - 58 cm final driver width
- Local inverters for latching
- Conditional clocks in caches to reduce power

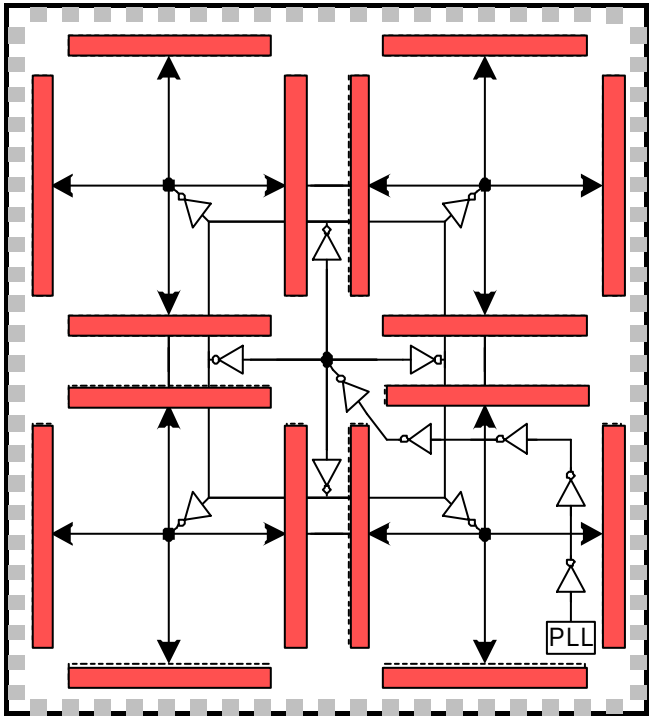

Clock Drivers

Clock Skew in Alpha Processor



A 3-D representation of the simulated skew. Max value of absolute skew is 90ps. The critical instruction and execution units all see a clock within 65ps. To avoid race-through conditions: 1) careful sizing of local clock buffers with minimal skew, 2) at least one gate is inserted between connecting latches to ensure a minimum contamination delay.

EV6 (Alpha 21264) Clocking
600 MHz – 0.35 micron CMOS



- **Hierarchical clocking** approach to better tradeoff power and skew management. Power is reduced because the clocking networks for individual clocks can be gated. Also, the flexibility of having local clocks provides the designer with more freedom w.r.t. circuit styles at the module level. Drawback of hierarchical design is the difficulty to reduce skew, since clocks to various local registers may go through very different paths, contributing to skew. This skew can be reduced, however, by tweaking the clock drivers with the aid of verification tools.

- 2 Phase, with multiple conditional buffered clocks
  - 2.8 nF clock load
  - 40 cm final driver width
- Local clocks can be gated “off” to save power
- Reduced load/skew
- Reduced thermal issues
- Multiple clocks complicate race checking

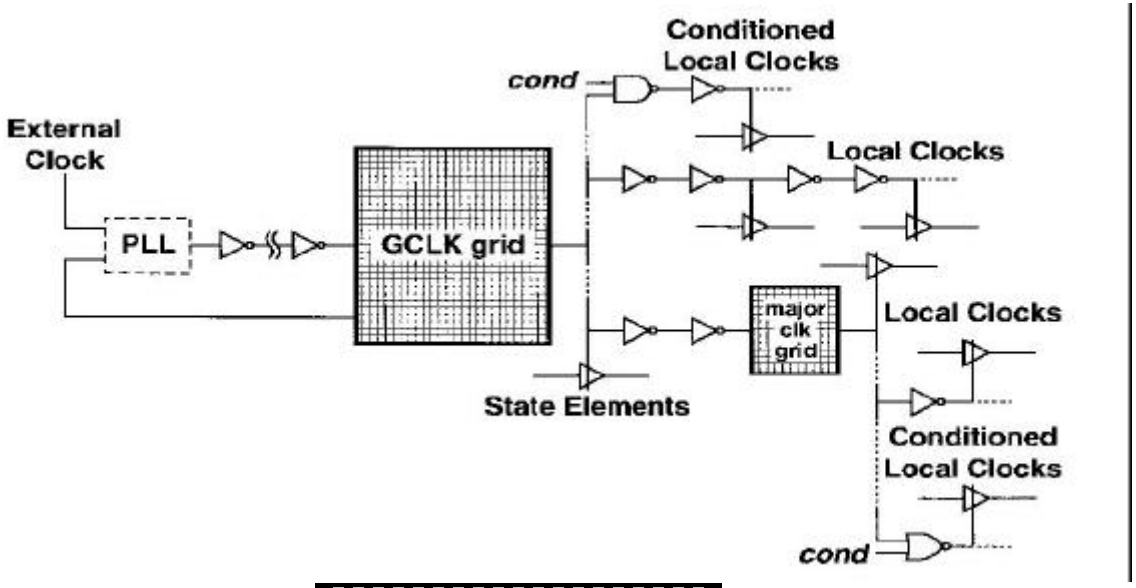
21264 Clocking

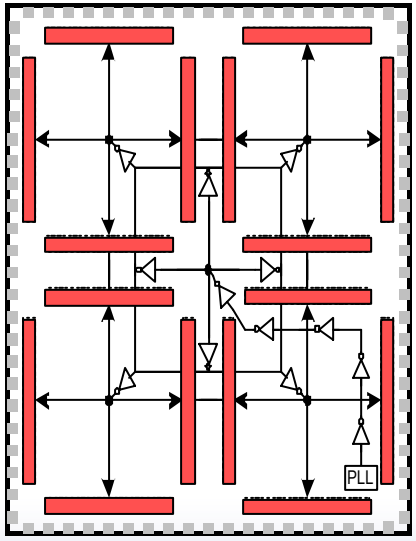
It consists of a global clock grid (GCLK), that covers the entire die. The generated clock is routed to the center of the die and distributed using tree structures to 16 clock drivers. The global CLK network utilizes a windowpane configuration, which achieves low skew by dividing the clock into 4 regions, to reduce the distance between from drivers to loads.

Each grid pane is driven from 4 sides, reducing the dependence on process variations. This also helps the Vdd and thermal problems, as the drivers are distributed through the chip.

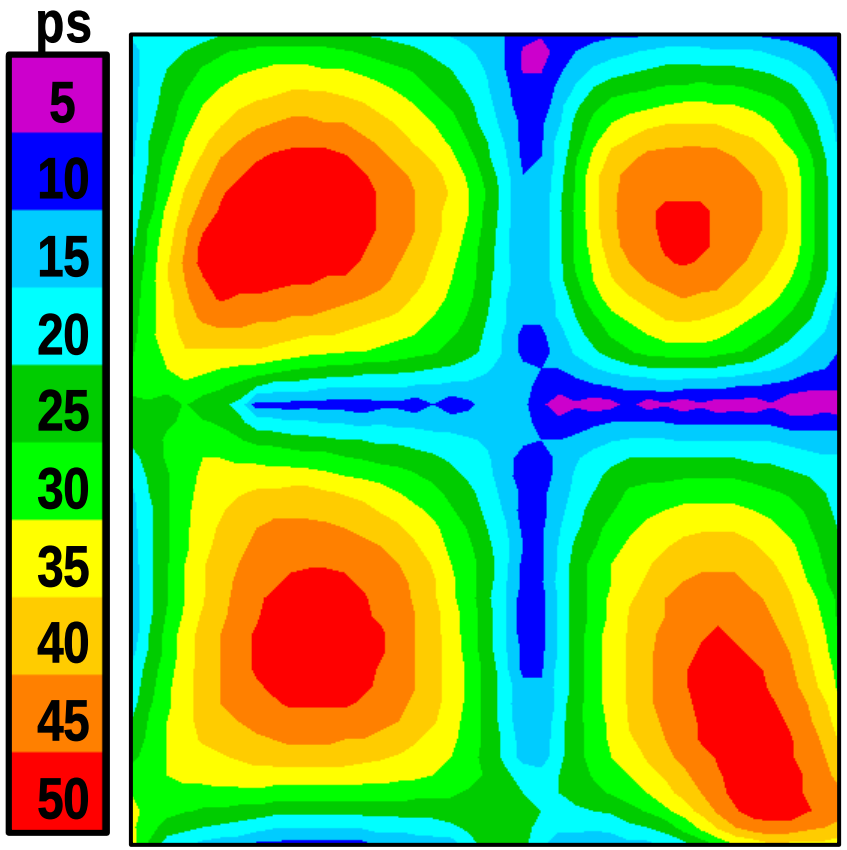
Adv: reduces clock skew, while providing universal availability of clock signals.
Disadv: increased capacitance of GCLK compared to a tree-grid

Next level of CLK hierarchy is “major CLK grid”: Major clocks are introduced to reduce power: they have localized loads, and they can be sized to meet skew and edge requirements for local loading at timing-critical units. “Local clocks” are generated as needed and designed to meet local timing requirements. The local clocks provide flexibility in the design of logic blocks, but more difficult to manage skew. Local clocks are also more susceptible to coupling from data lines, because they are not shielded as global gridded clocks. Thus, local clock distribution is highly dependent on its local interconnection and thus has to be designed very carefully.





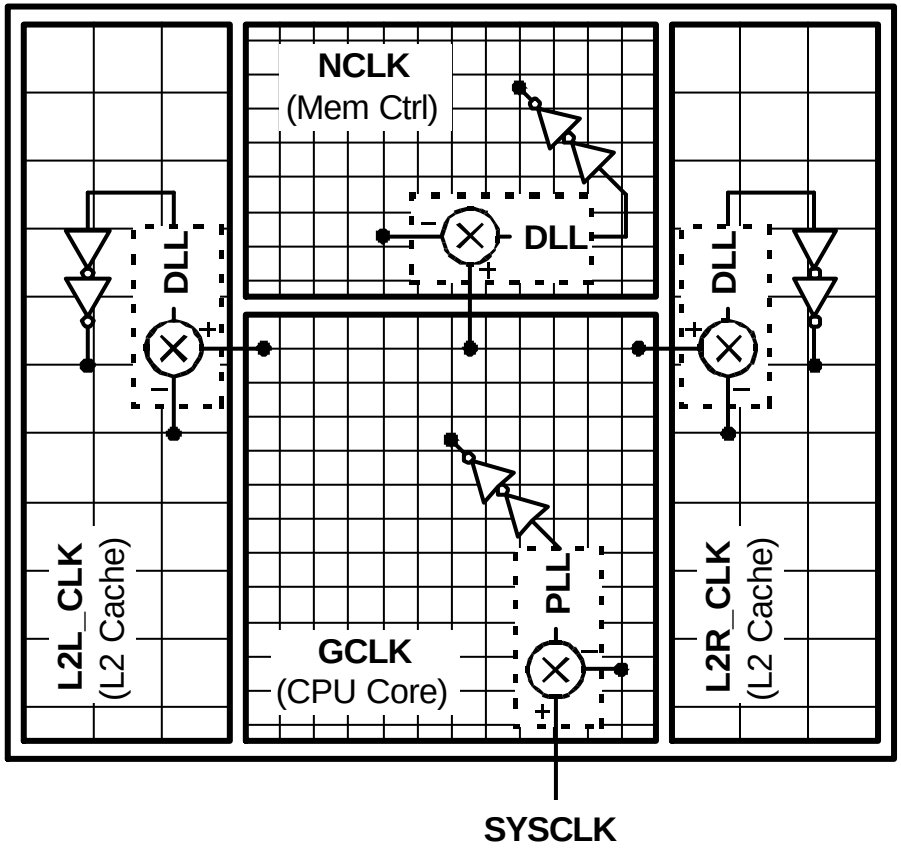
EV6 Clock Results



GCLK Skew
(at Vdd/2 Crossings)

EV7 Clock Hierarchy

Active Skew Management and Multiple Clock Domains



+ widely dispersed drivers

+ DLLs compensate static and low-frequency variation

+ divides design and verification effort

- DLL design and verification is added work

+ tailored clocks

Design Techniques – Dealing with Clock Skew and Jitter

- To minimize skew, balanced clock paths from a central distribution source to individual clocking elements, using H-tree structures or routed matched-tree structures. When using routed clock trees, the effective clock load of each path including that of wires and devices, must be equalized.
- The use of local clock grids (instead of routed trees) can reduce skew at the cost of increased capacitive load and power dissipation
- If data flow in one direction, route the data and the clock in opposite directions. This eliminates races at the cost of performance.
- Avoid data-dependent noise by shielding clock wires from adjacent signal wires. By placing power lines (Vdd and GND) next to the clock wires, coupling from neighboring signal nets can be minimized.
- Variations in chip temperature across the die causes variations in clock buffer delay. The use of feed-back circuits based on DLLs can compensate for temperature variations.
- Vdd variations is a significant component of jitter, as it impacts the cycle-to-cycle delay through clock buffers. High-frequency Vdd variations can be reduced by adding on-chip decoupling capacitors. Unfortunately, decoupling capacitors require significant amount of area, and may cause gate leakage for nanometer designs