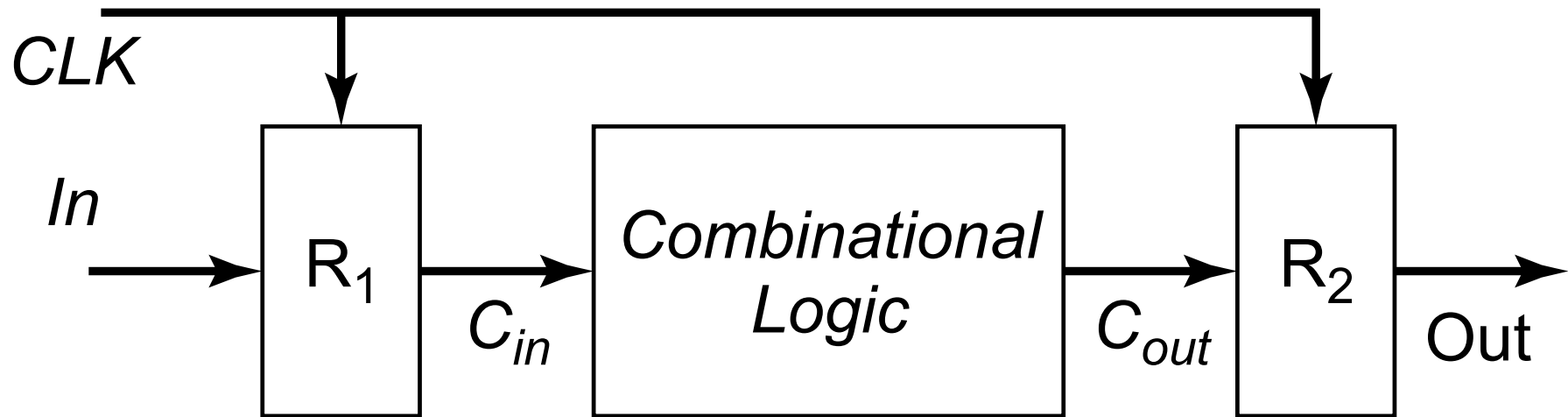




数字集成电路

第十讲 时序问题

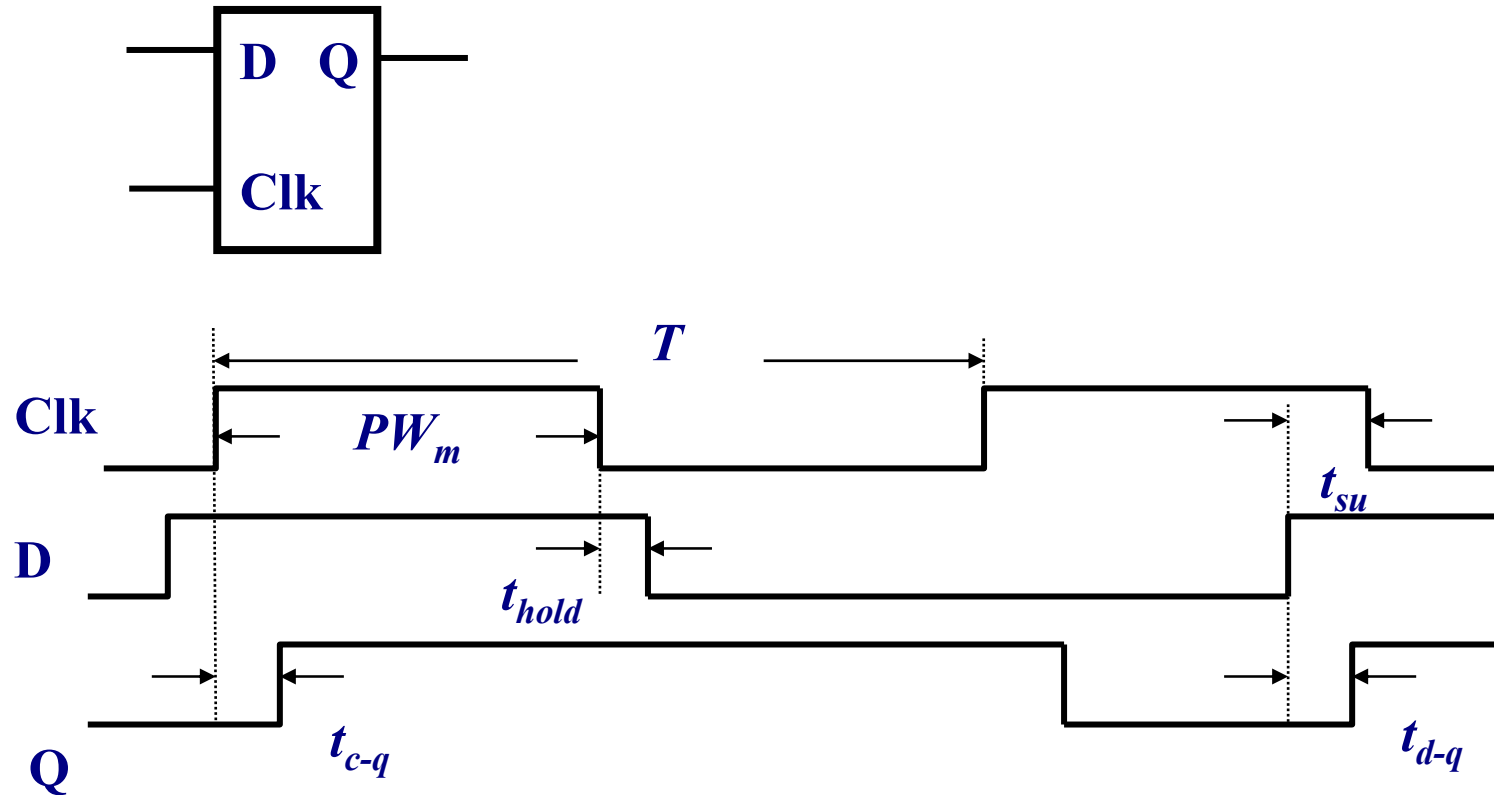
10.3 同步时序





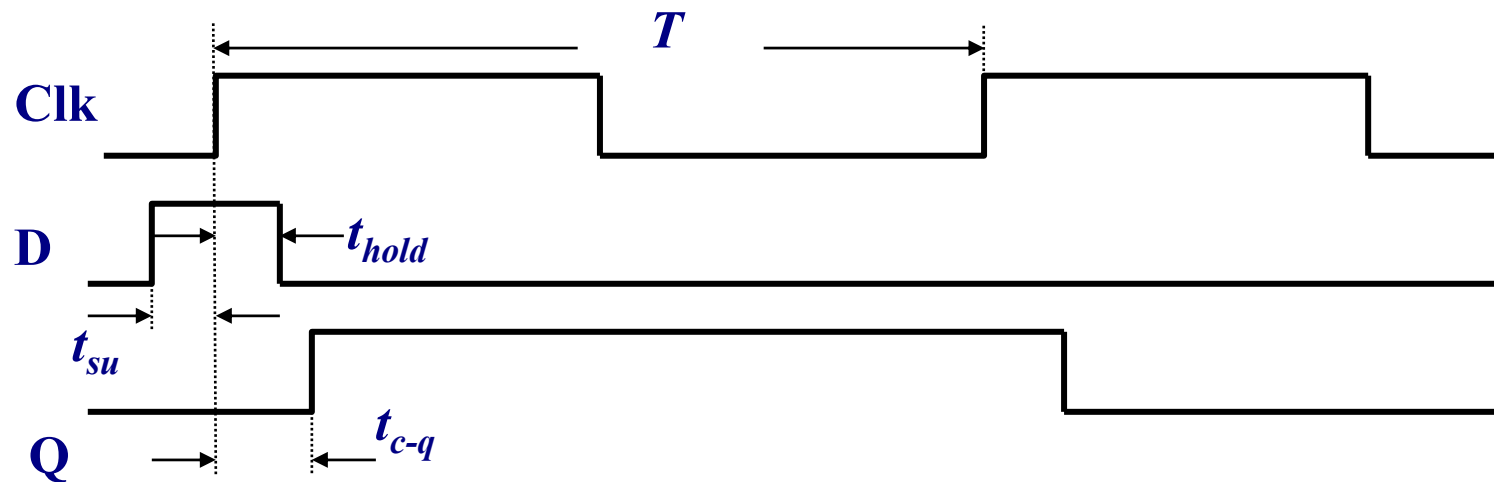
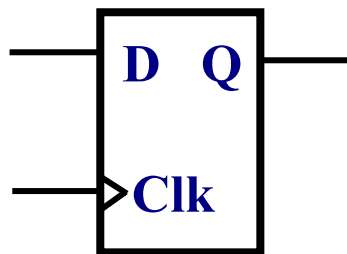
Timing Definitions

锁存器



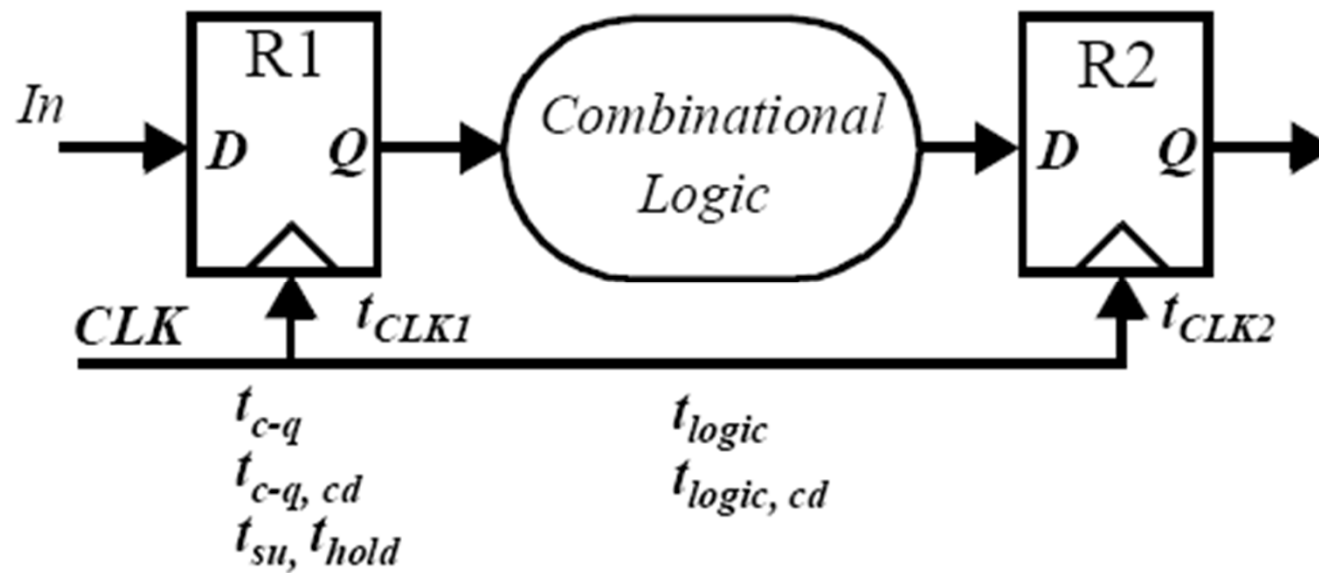
Delays can be different for rising and falling data transitions

寄存器



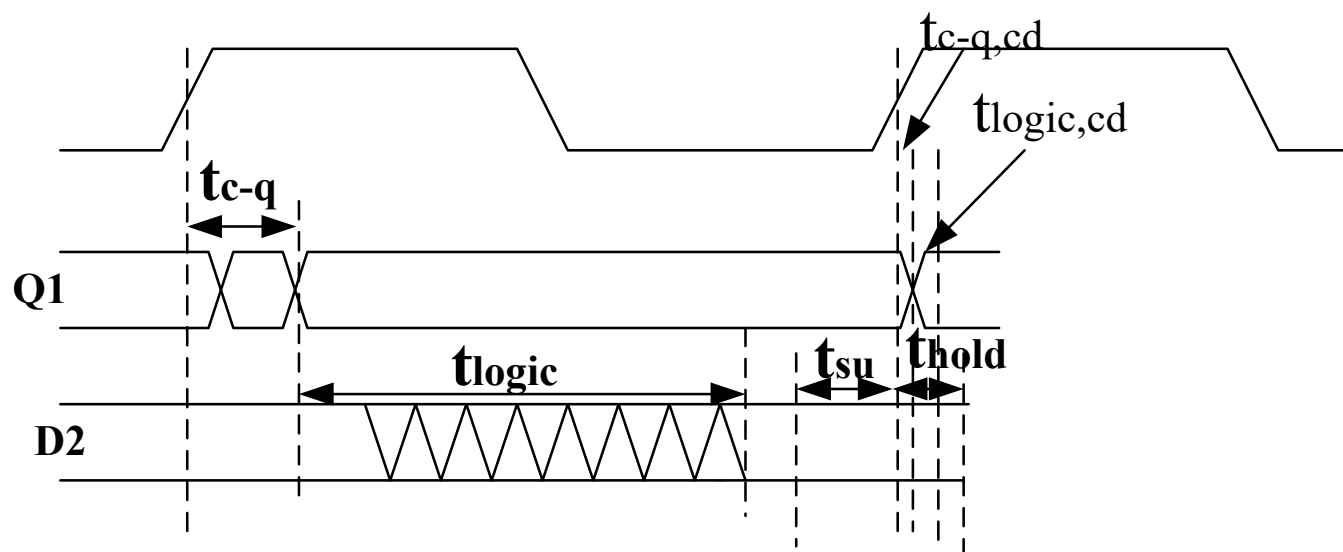
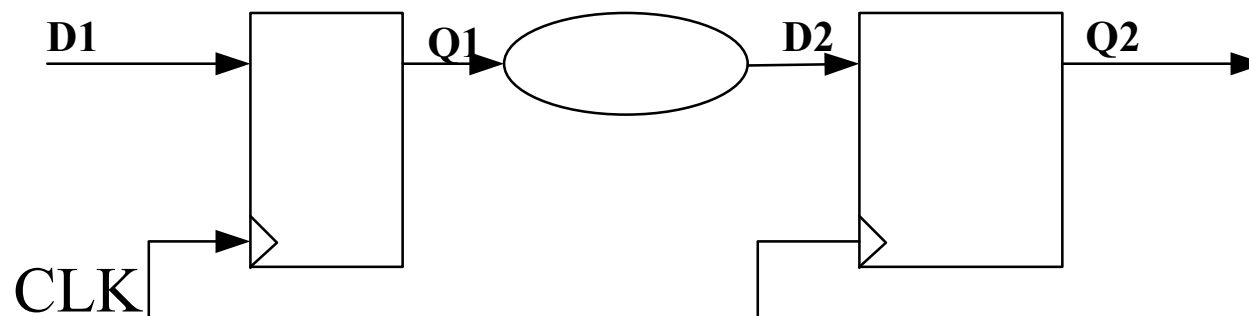
Delays can be different for rising and falling data transitions

同步时序原理



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

同步时序原理



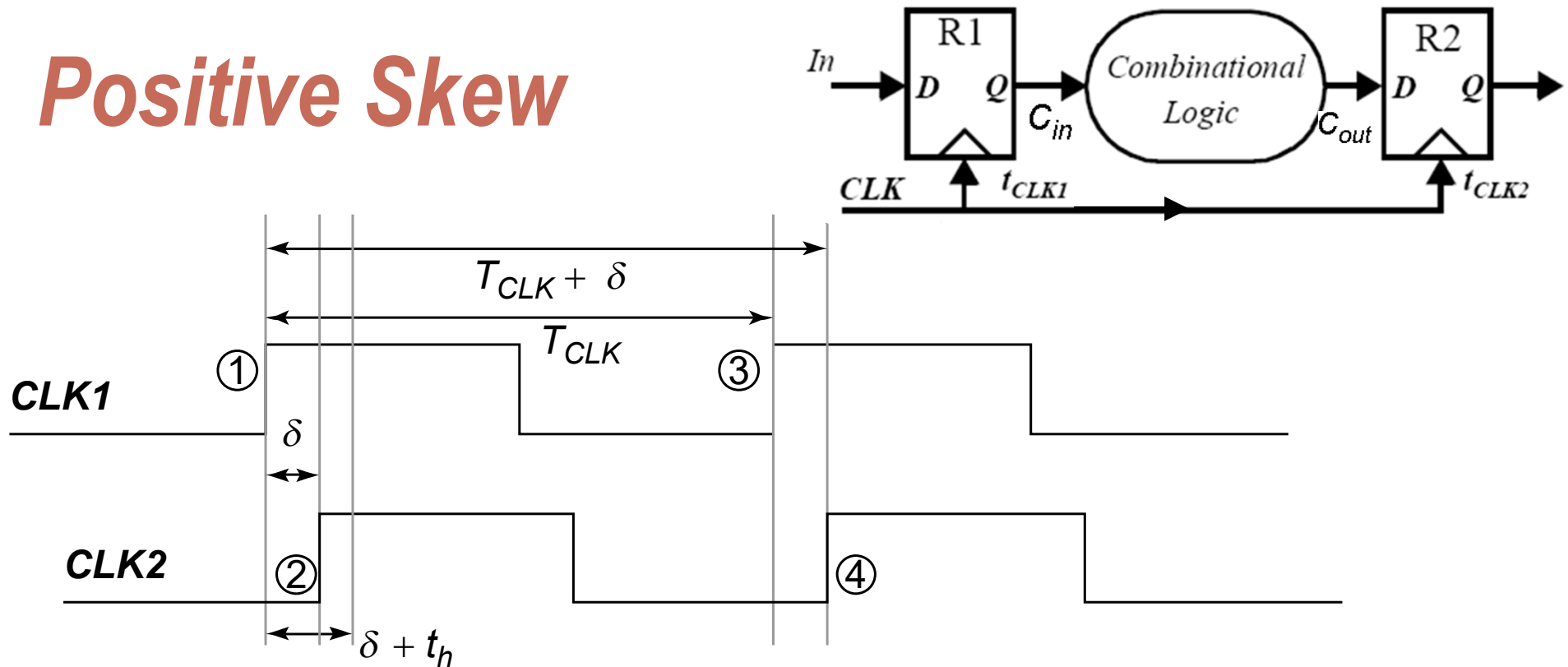
$$T > t_{c-q} + t_{logic} + t_{su}$$

$$t_{hold} < t_{c-q,cd} + t_{logic,cd}$$

时钟偏差 (Clock Skew)

- ❑ 时钟偏差：集成电路中一个时钟翻转的到达时刻在空间上的差别。
- ❑ 时钟偏差是由时钟路径的静态不匹配以及时钟在负载上的差异造成的。
- ❑ 时钟偏差并不造成时钟周期的变化，只是相位的偏移，各个周期的偏差是相同的。
- ❑ 时钟偏差对时序电路的性能和功能会产生很大的影响。

Positive Skew



Launching edge arrives before the receiving edge

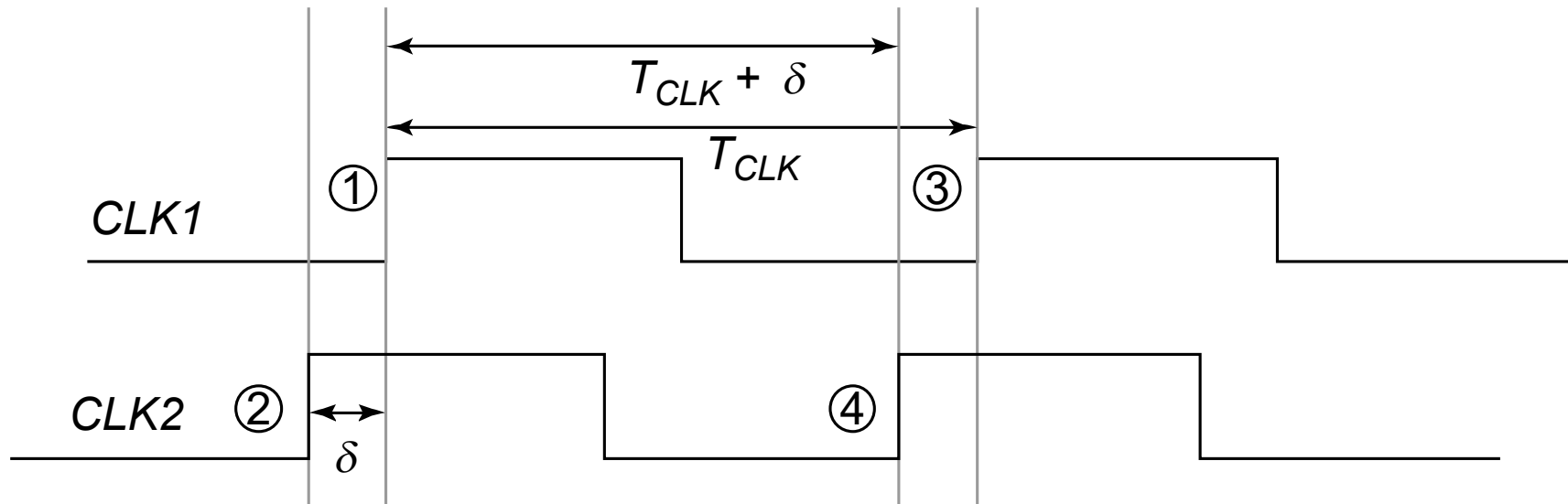
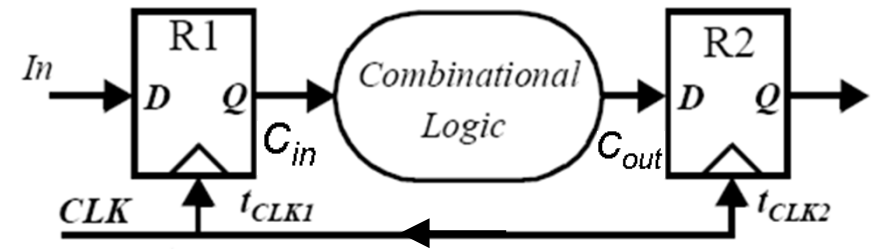
$$T + \delta \geq t_{c-q} + t_{logic} + t_{su} \quad T \geq t_{c-q} + t_{logic} + t_{su} - \delta$$

$$\delta + t_{hold} < t_{(c-q, cd)} + t_{(logic, cd)}$$

or

$$\delta < t_{(c-q, cd)} + t_{(logic, cd)} - t_{hold}$$

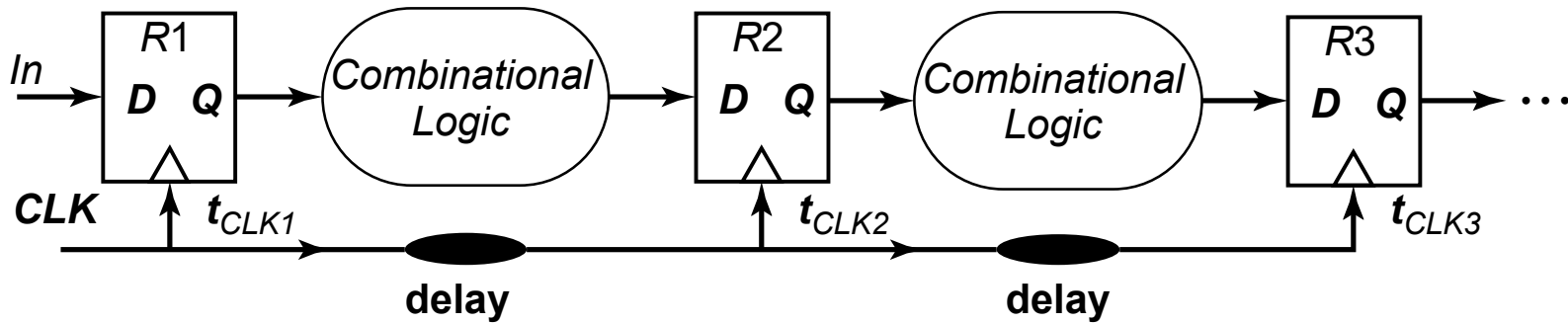
Negative Skew



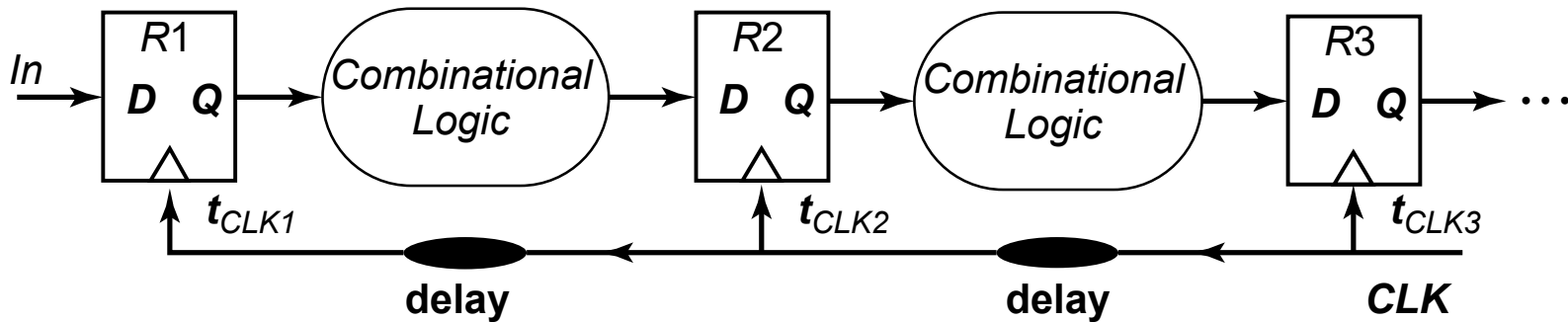
Receiving edge arrives before the launching edge

竞争现象永远不可能出现
但是影响电路性能

一、偏差 *Positive and Negative Skew*

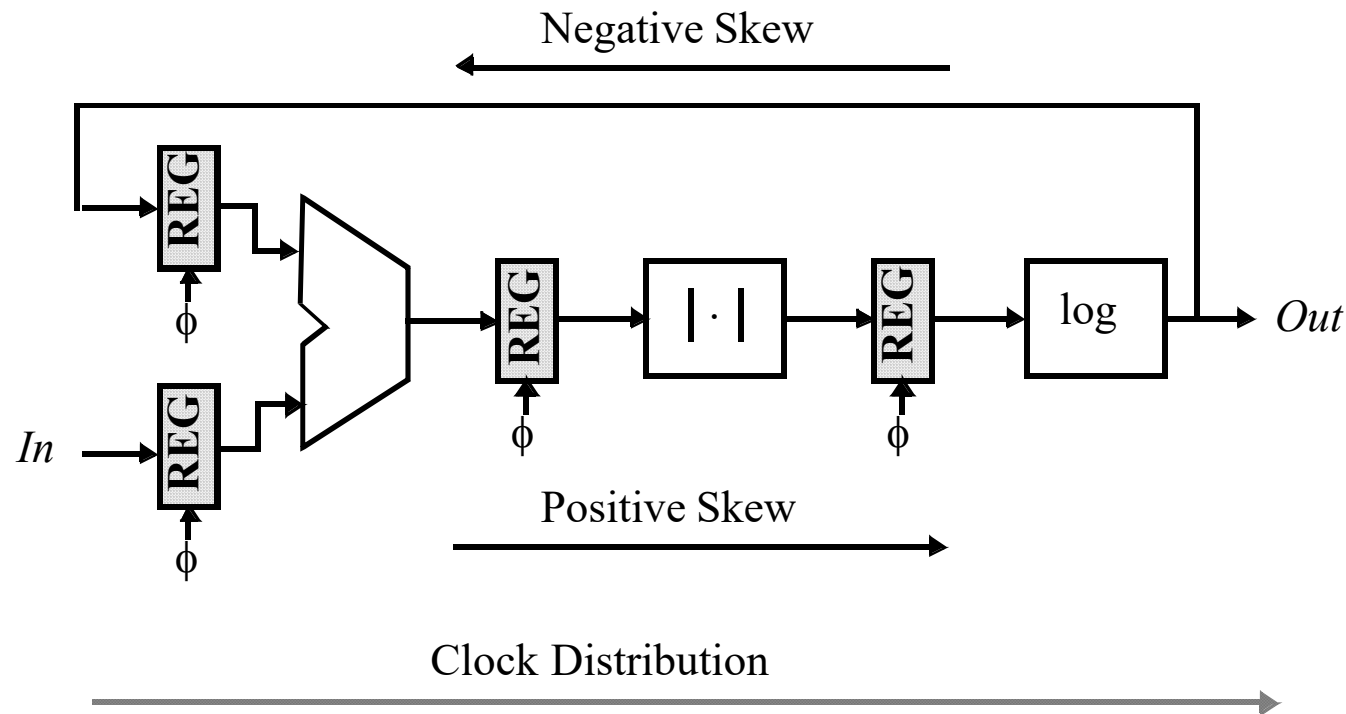


(a) Positive skew



(b) Negative skew

How to counter Clock Skew?



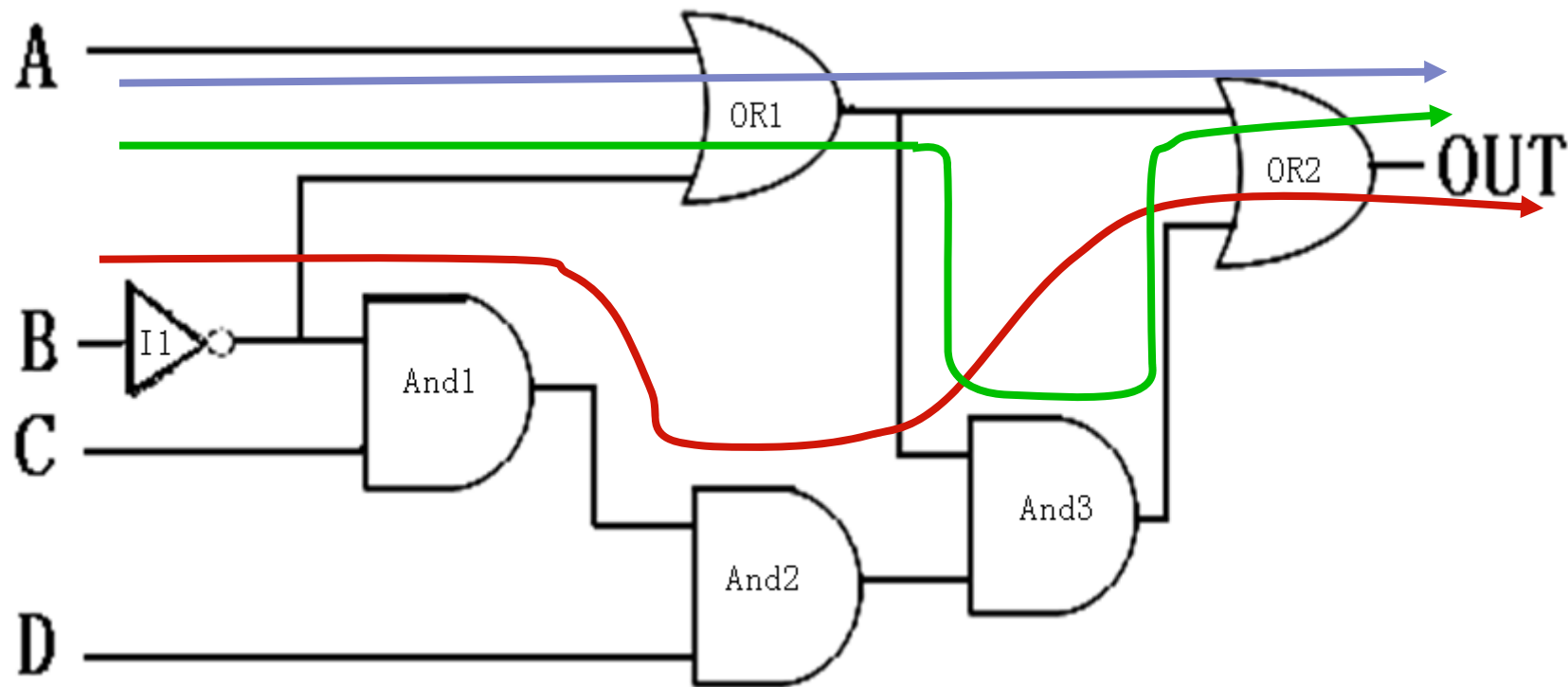
Data and Clock Routing

设计一个偏差小的时钟网络是非常重要的!!!

P12

估计该逻辑电路的污染延时和最大延时

P365 例10.1

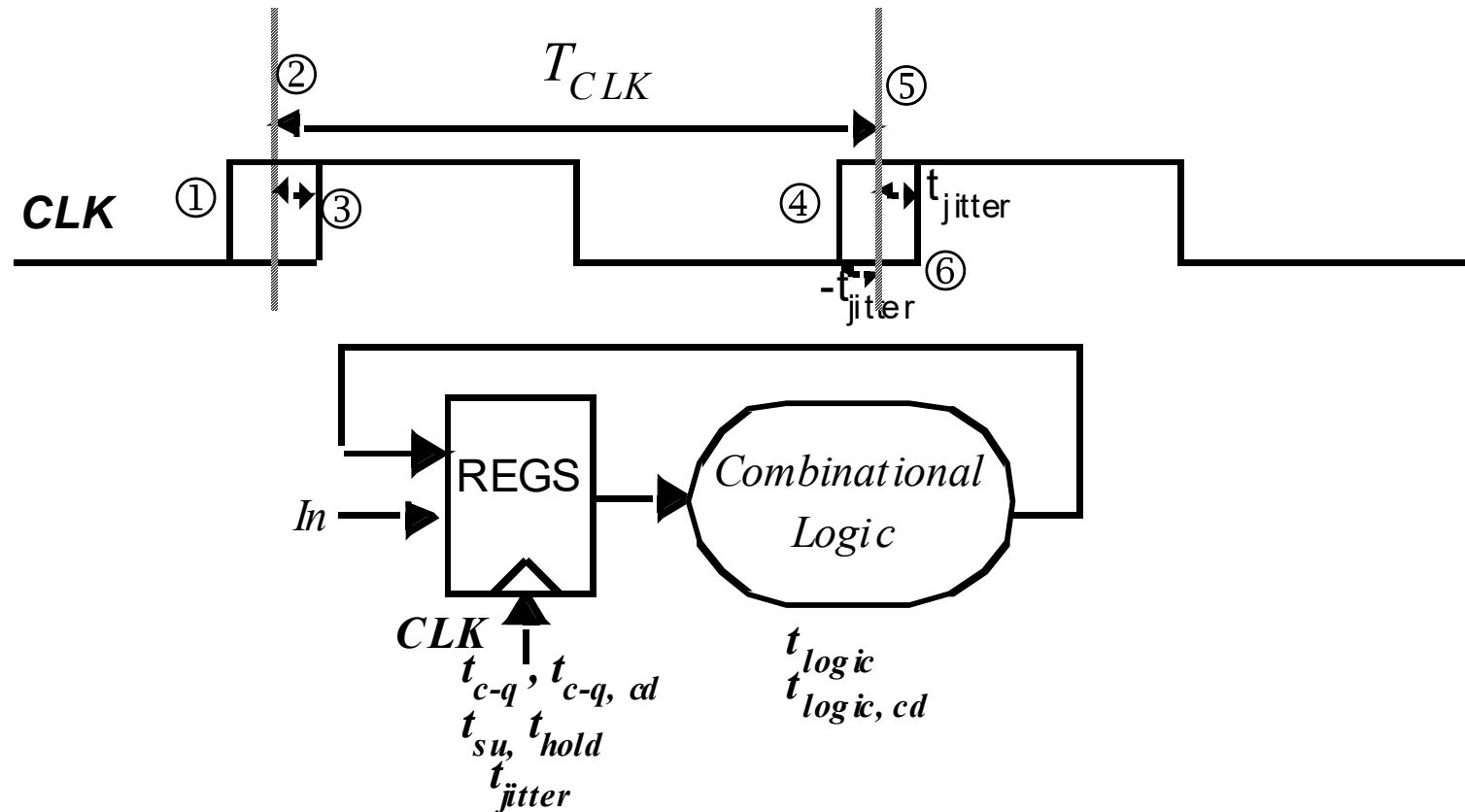


二、时钟抖动 (clock jitter)

- ❑ 时钟抖动是指在芯片上的某一个给定点上时钟周期发生暂时性的变化，即时钟周期在每个不同的周期上可以缩短或加长。
- ❑ 时钟抖动是严格衡量时钟暂时不确定性的一项指标，并且经常针对某一给定点进行说明。
- ❑ 抖动可以使用许多方法来衡量和表征，它是一个平均值为零的随机变量。
- ❑ 绝对抖动(jitter)指某一给定位置处的一个时钟边沿相对于理想的周期性参照时钟边沿在最坏情形下的变化值。
- ❑ 周期至周期的抖动 (T_{jitter}) 指单个时钟周期相对于理想参照时钟的偏离。

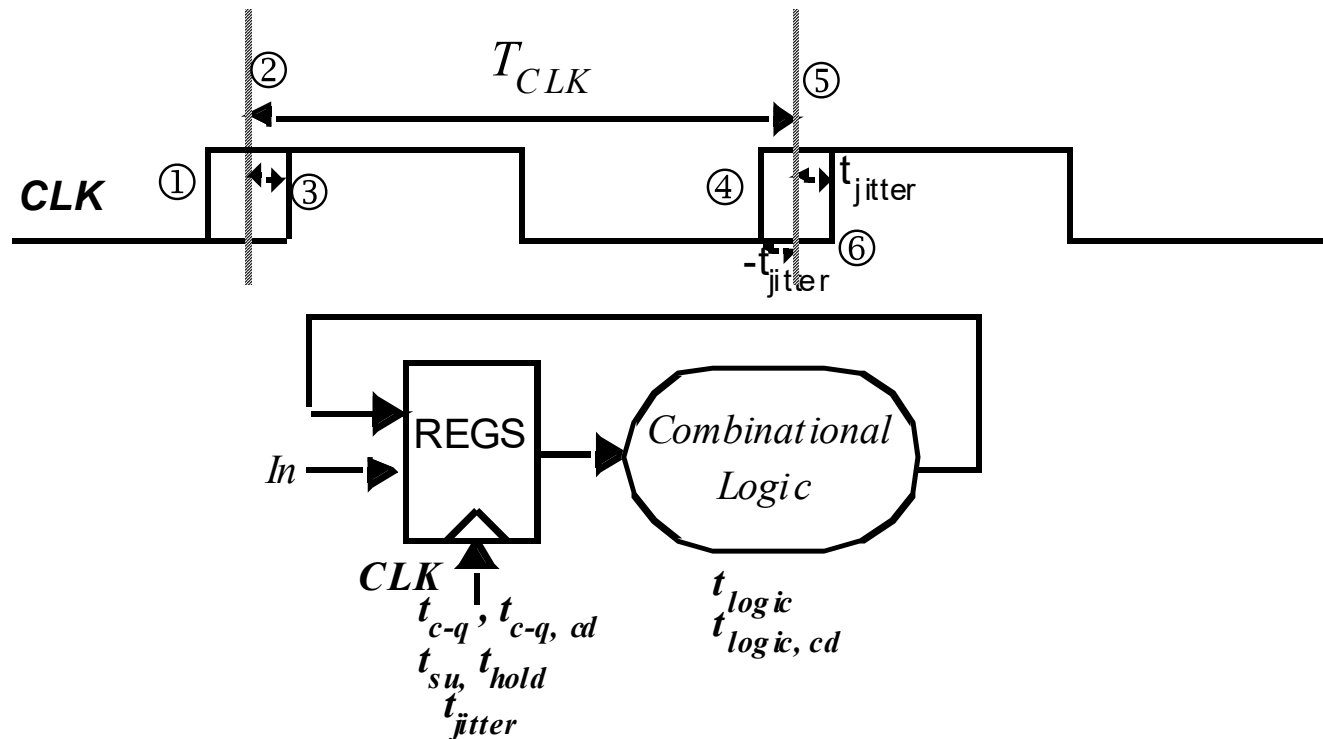
$$T_{jitter}(n) = t_{clk,n+1} - t_{clk,n} - T_{CLK}$$

时钟抖动的影响



最坏情况下，周期至周期抖动绝对值等于绝对抖动的2倍 ($2t_{jitter}$)

考虑时钟抖动时的时序要求

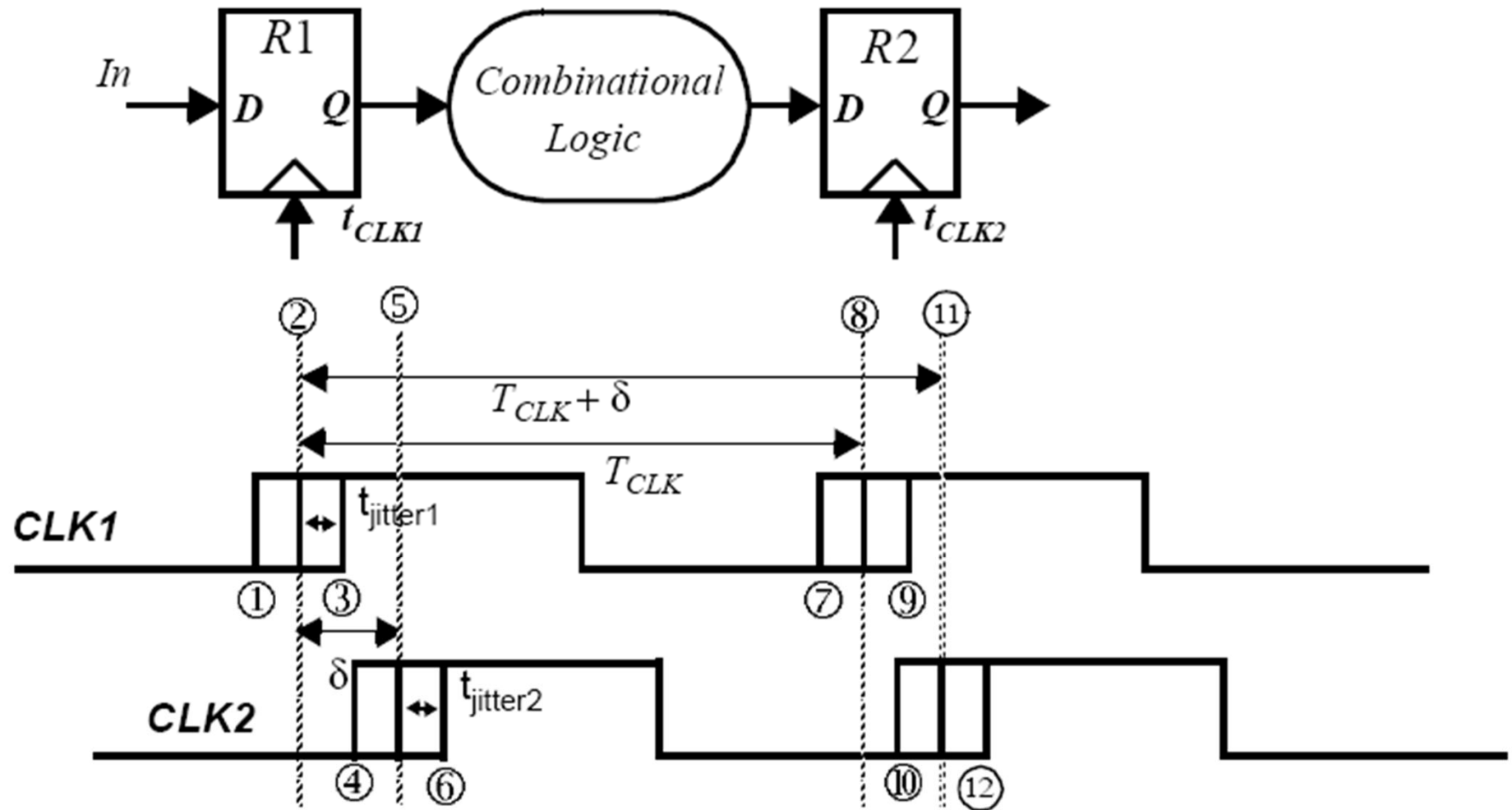


$$T_{CLK} - 2t_{jitter} \geq t_{c-q} + t_{logic} + t_{su}$$

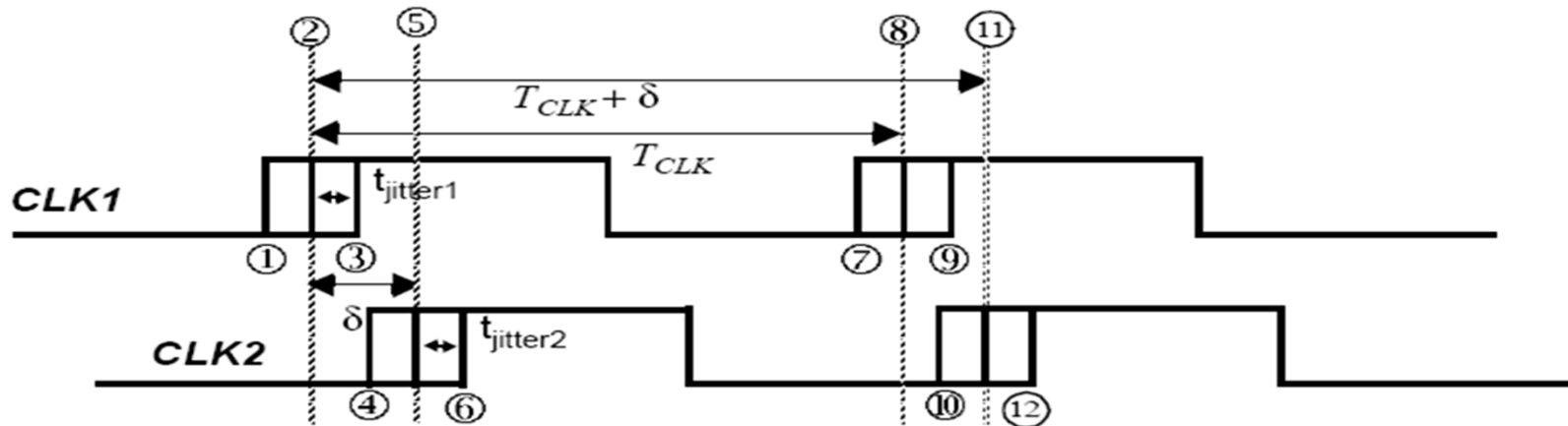
$$T \geq t_{c-q} + t_{logic} + t_{su} + 2t_{jitter}$$

时钟抖动直接降低了一个时序电路的性能。如果性能是一个电路的关键因素，应该严格把抖动限定在一定的范围内。

偏差和抖动的共同影响



偏差和抖动的共同影响



Clock skew > 0 时的约束条件:

$$T_{CLK} + \delta - t_{jitter1} - t_{jitter2} \geq t_{c-q} + t_{logic} + t_{su}$$

$$\text{or } T \geq t_{c-q} + t_{logic} + t_{su} - \delta + t_{jitter1} + t_{jitter2}$$

Clock skew > 0 时的维持时间约束条件:

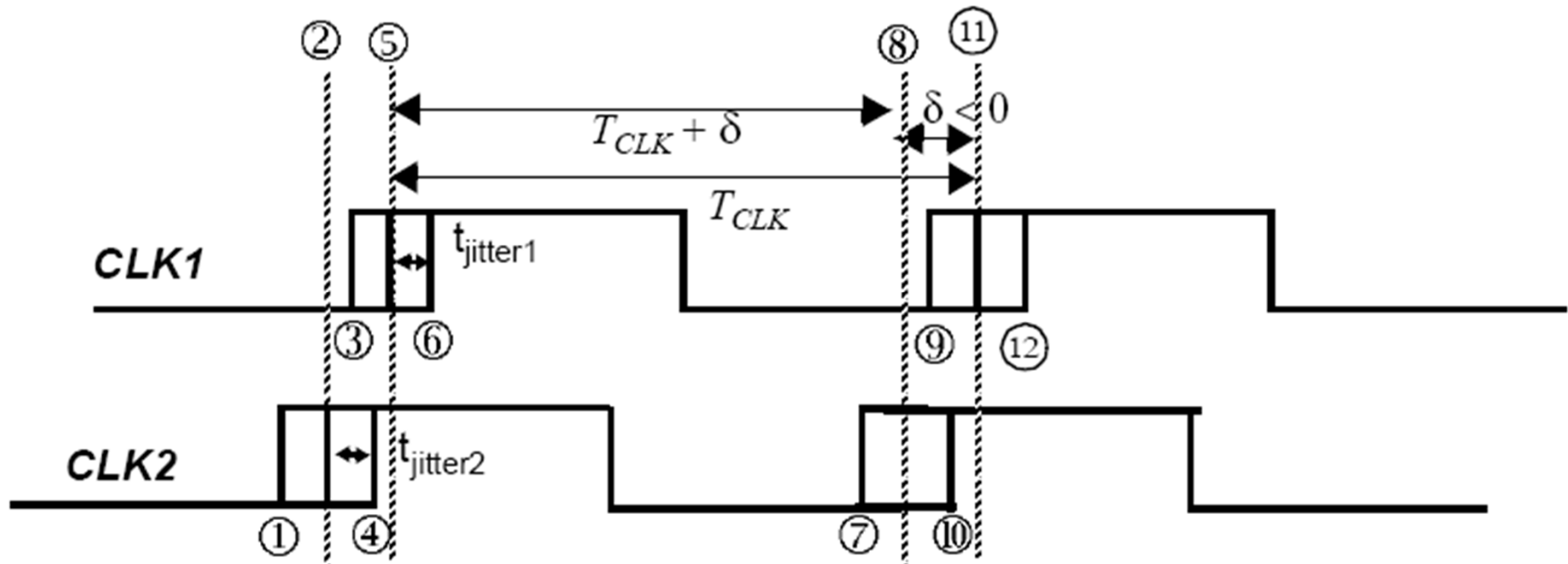
$$\delta + t_{hold} + t_{jitter1} + t_{jitter2} < t_{(c-q, cd)} + t_{(logic, cd)}$$

or

$$\delta < t_{(c-q, cd)} + t_{(logic, cd)} - t_{hold} - t_{jitter1} - t_{jitter2}$$

偏差和抖动的共同影响

Clock skew < 0 的情况

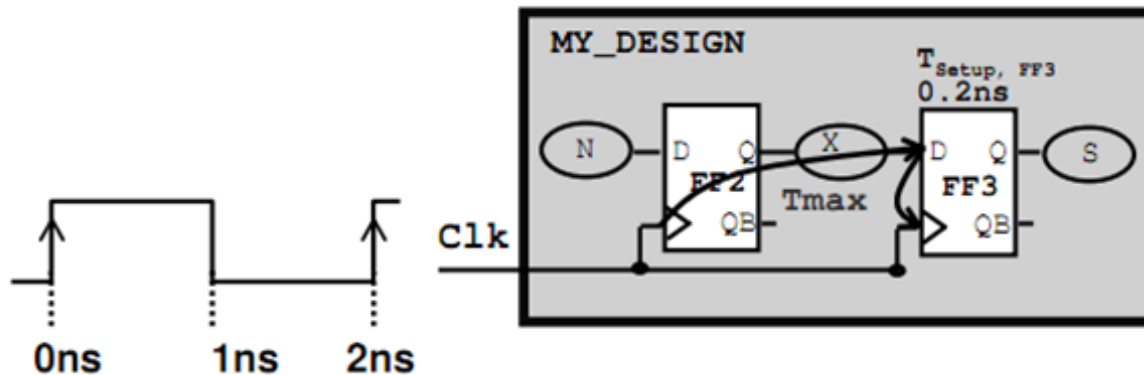


时序分析与前面分析的情况一致
负偏差会降低性能

Constraints Reg-to-Reg Paths

Spec:

Clock Period = 2ns



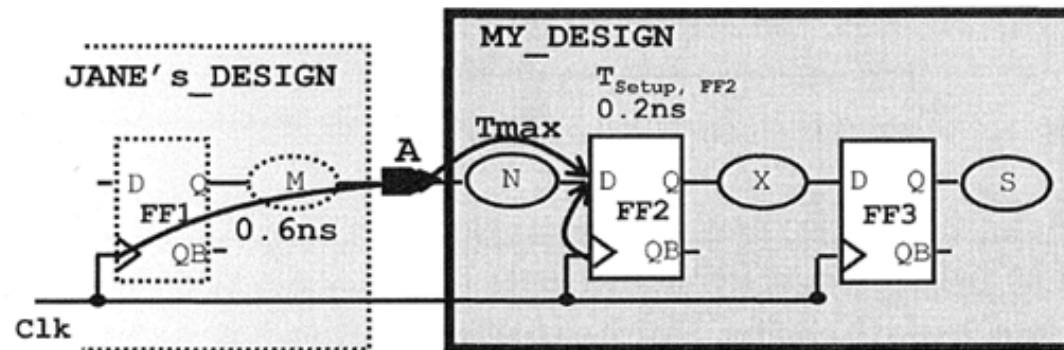
? What is the maximum delay requirement T_{max} for the register-to-register path through X in the MY_DESIGN? _____

$$T_{max} = \text{Period} - \text{Setup_Time (assuming 0 clock skew)} = 2 - 0.2 = 1.8 \text{ ns}$$

Constraining Input Paths

Spec:

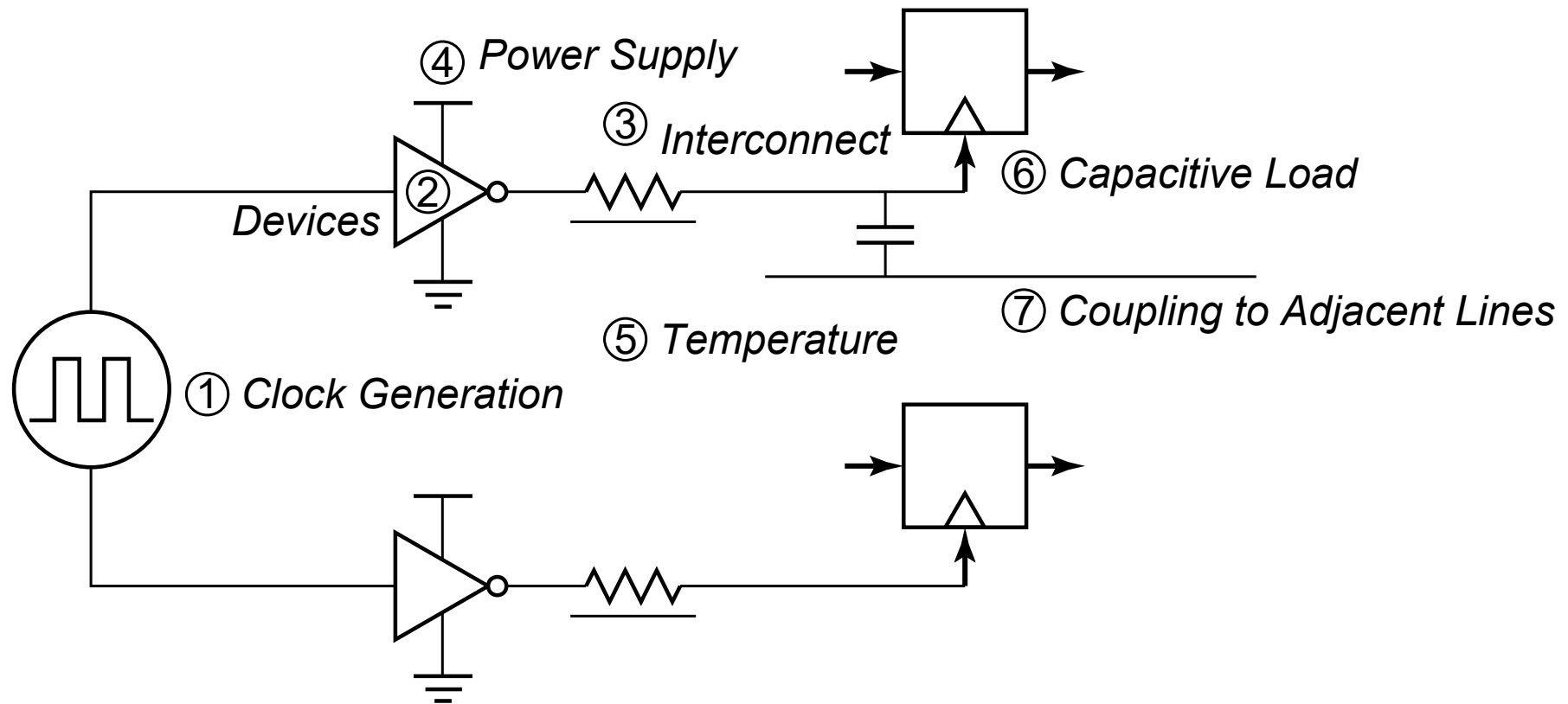
Latest Data Arrival Time at Port A, after Jane's launching clock edge = 0.6ns



? What is the maximum delay T_{max} for the input path N in MY DESIGN? _____

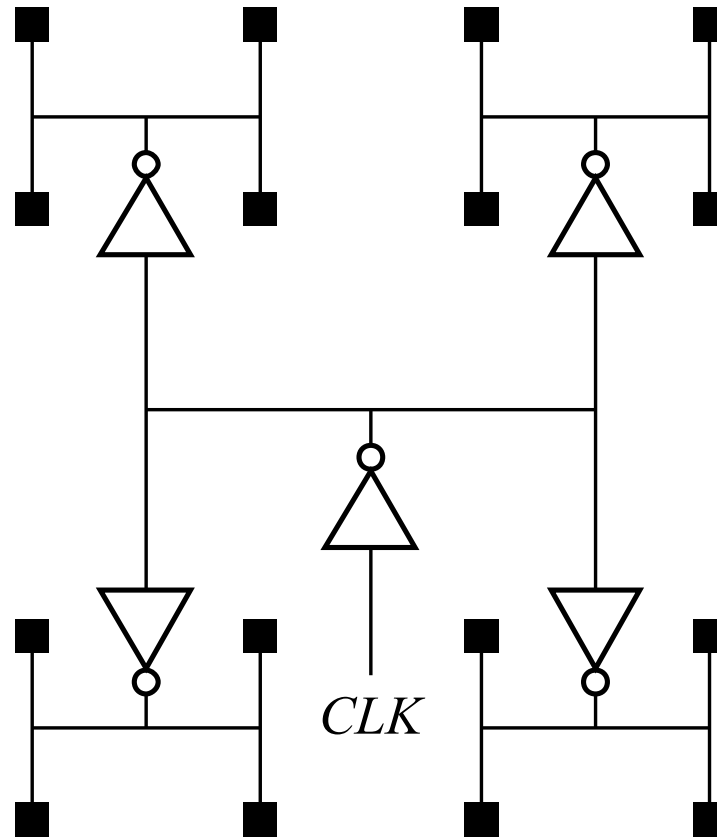
$$T_{max} = \text{Period} - \text{Input Delay} - \text{Setup_Time} \text{ (assuming 0 clock skew)}$$
$$= 2 - 0.6 - 0.2 = 1.2 \text{ ns}$$

时钟偏差和抖动的来源



同步时钟分布中的偏差和抖动的来源

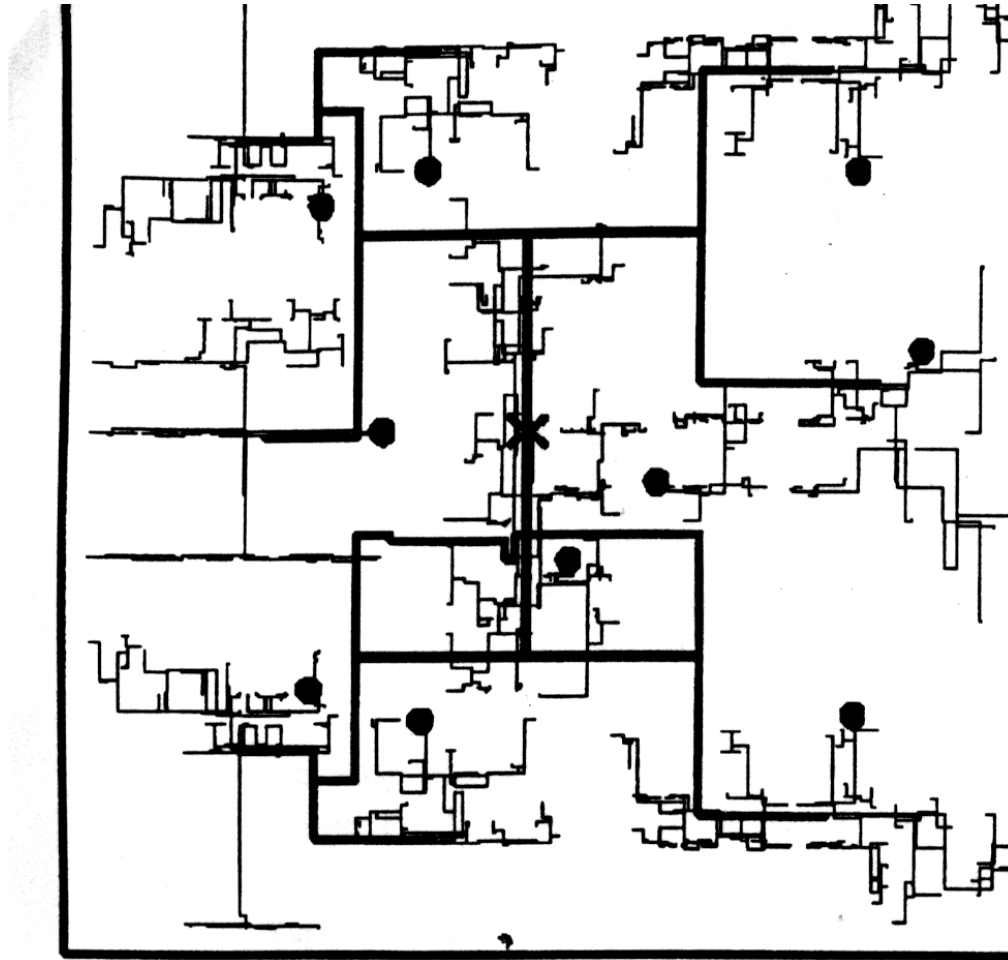
三、时钟分布技术



16个叶子节点的H树时钟分布网络

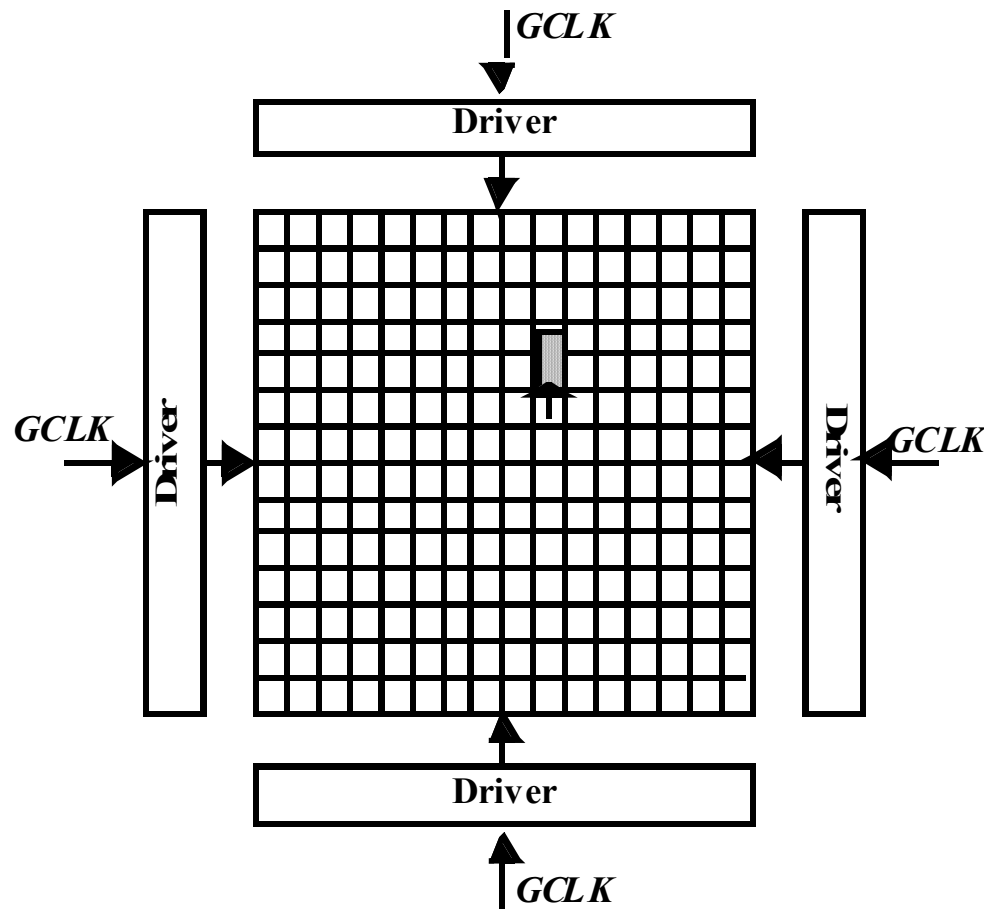
Clock is distributed in a tree-like fashion

实际的H树



[Restle98]

网络结构的时钟分布



- *No rc-matching*
- *Large power*

Example: DEC Alpha 21164

Clock Frequency: 300 MHz - 9.3 Million Transistors

Total Clock Load: 3.75 nF

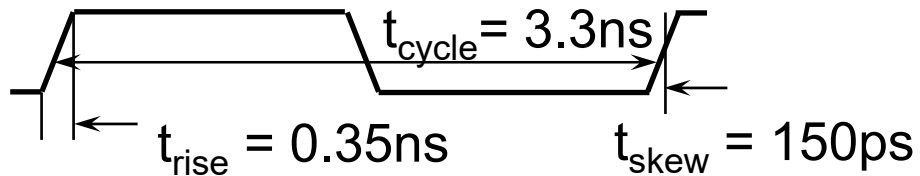
Power in Clock Distribution network : 20 W (out of 50)

Uses Two Level Clock Distribution:

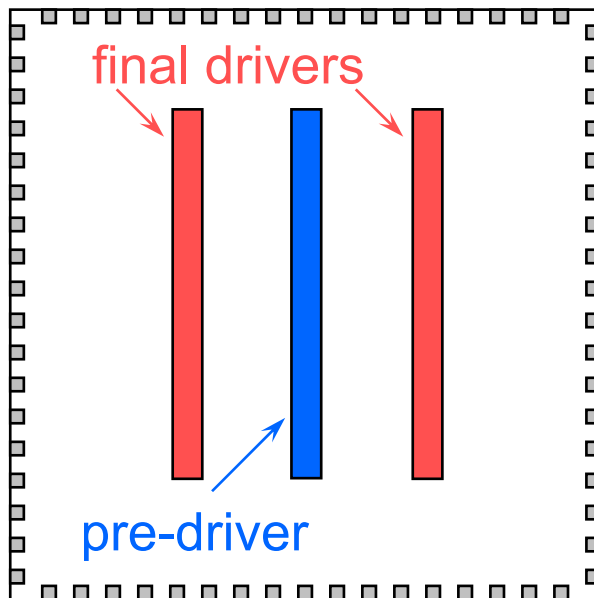
- **Single 6-stage driver at center of chip**
- **Secondary buffers drive left and right side clock grid in Metal3 and Metal4**

Total driver size: 58 cm!

Alpha 21164 Clocking

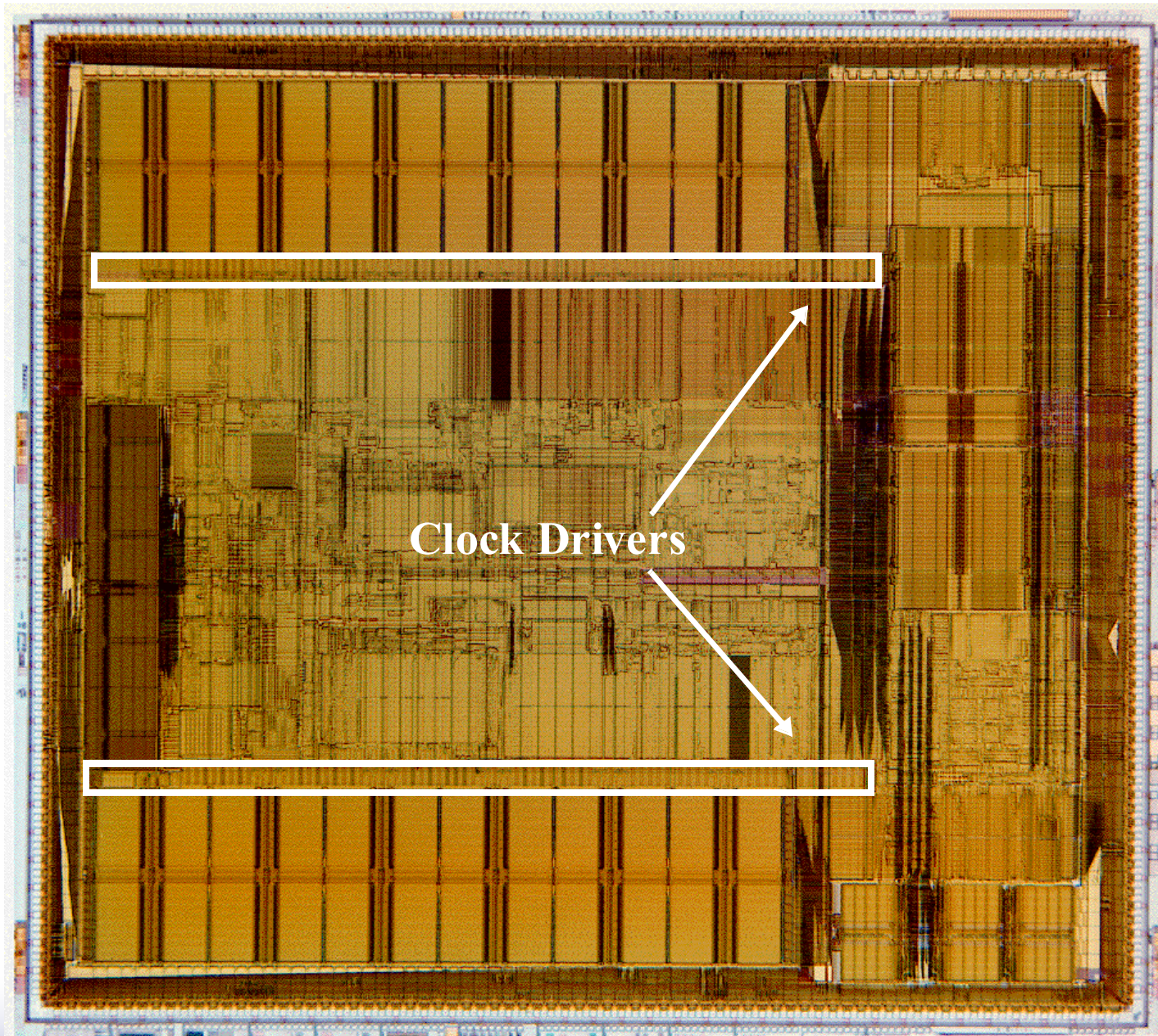


Clock waveform



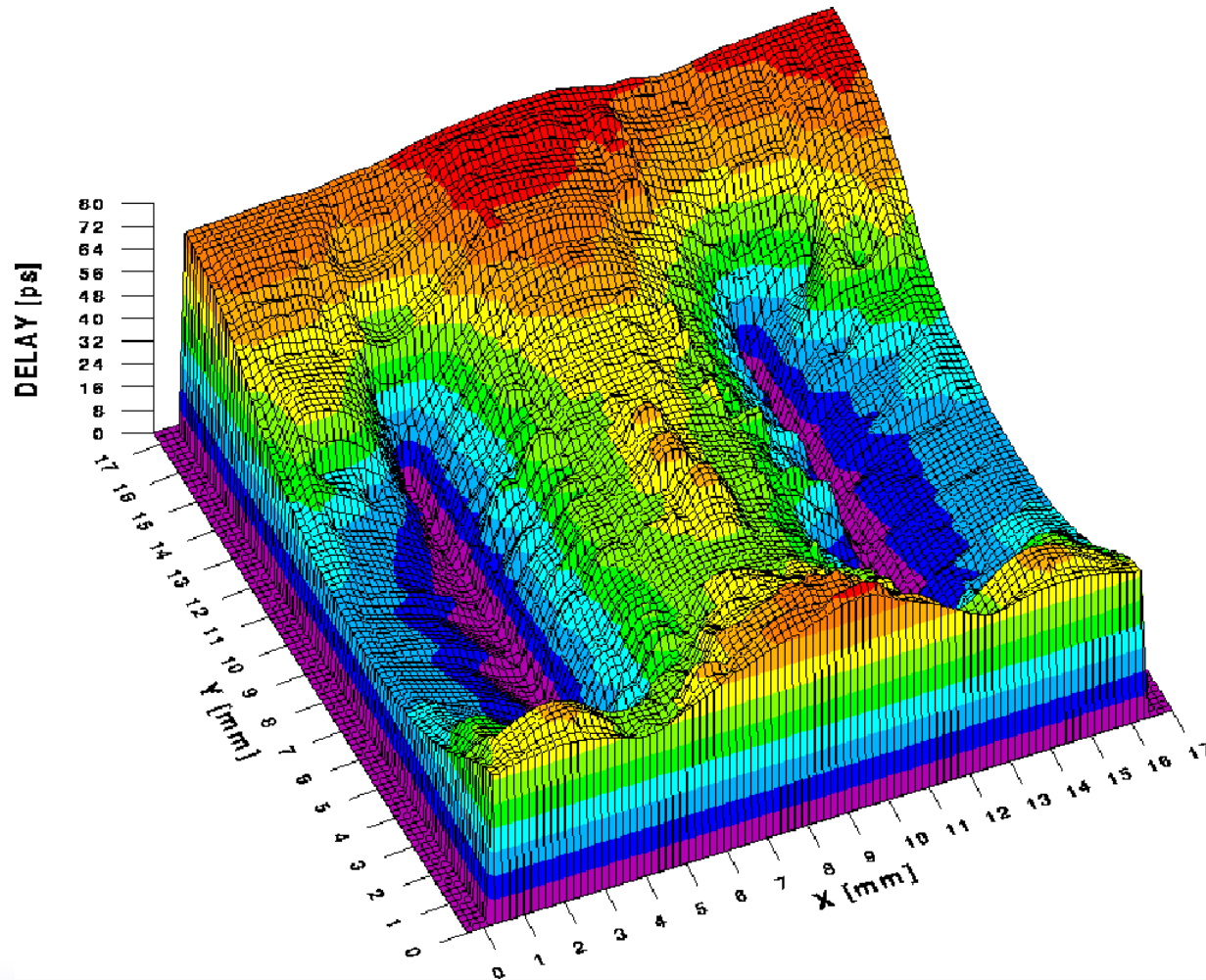
Location of clock driver on die

- ❑ 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
- ❑ More complex race checking
- ❑ Device variation



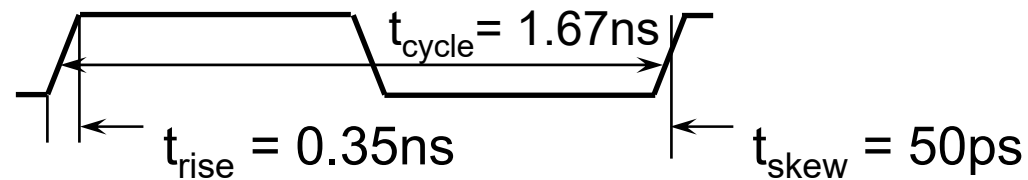
Clock Drivers

Clock Skew in Alpha Processor

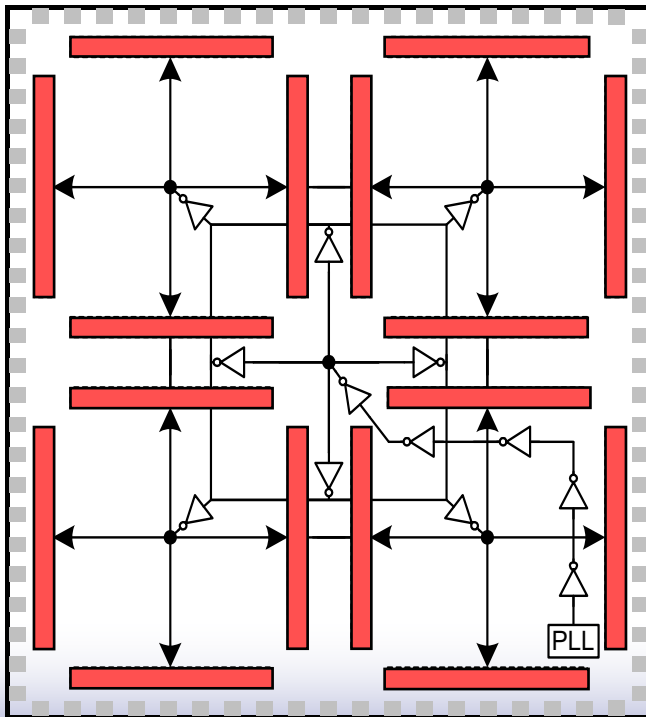


EV6 (Alpha 21264) Clocking

600 MHz – 0.35 micron CMOS

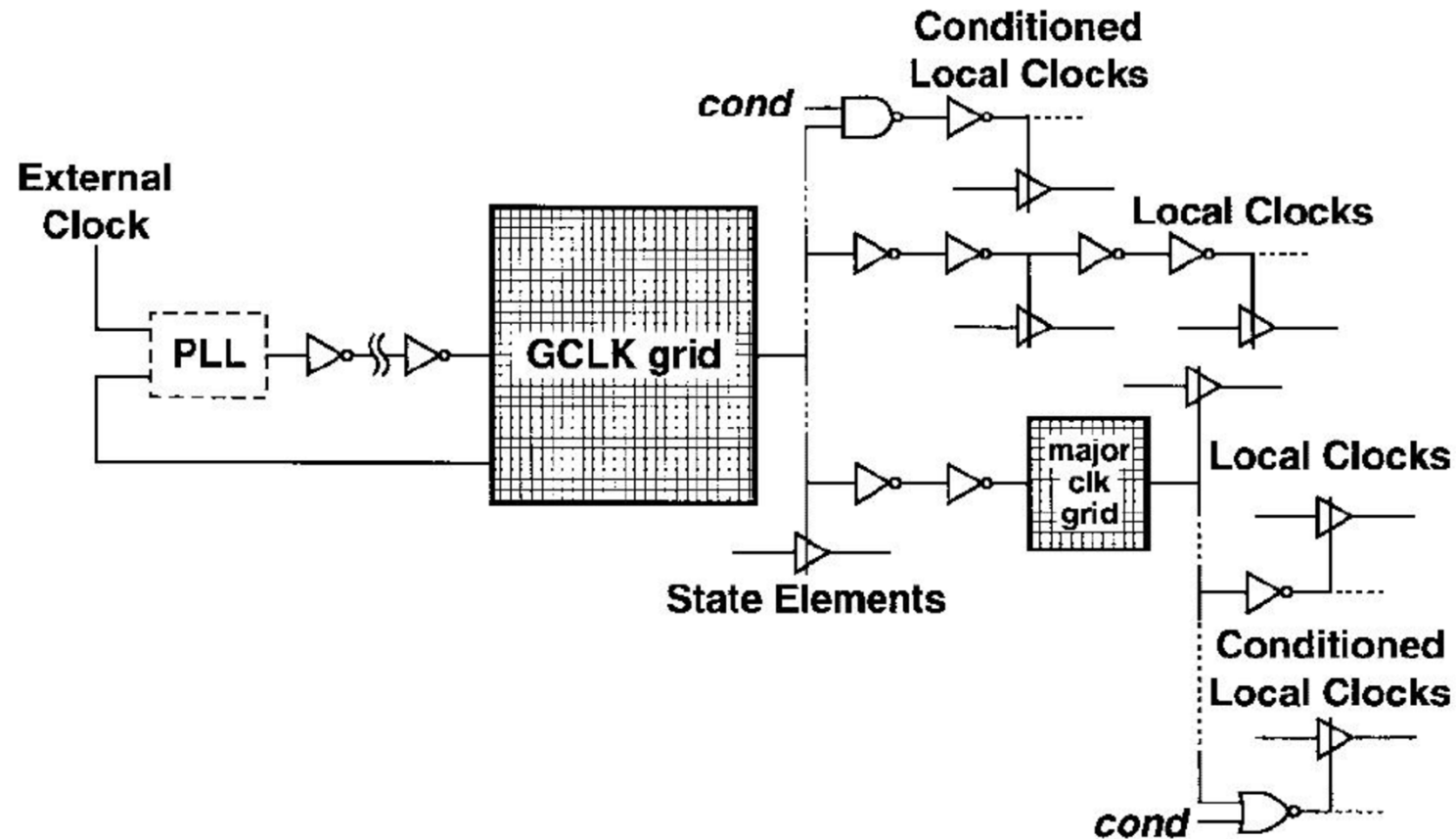


Global clock waveform

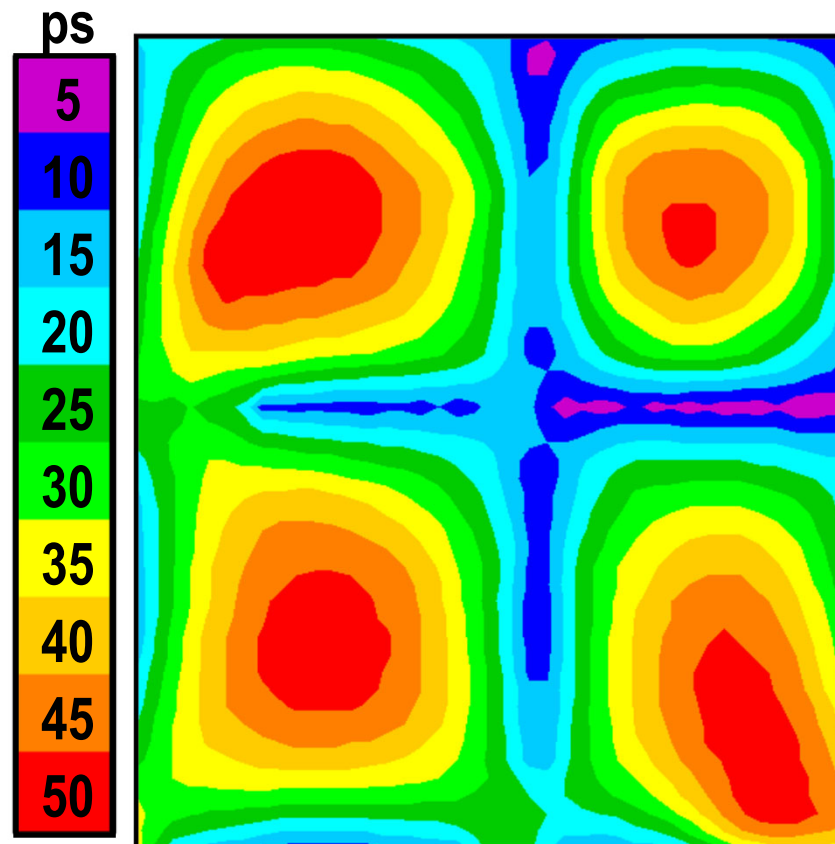


- ❑ 2 Phase, with multiple conditional buffered clocks
 - 2.8 nF clock load
 - 40 cm final driver width
- ❑ Local clocks can be gated “off” to save power
- ❑ Reduced load/skew
- ❑ Reduced thermal issues
- ❑ Multiple clocks complicate race checking

Alpha 21264 Clocking

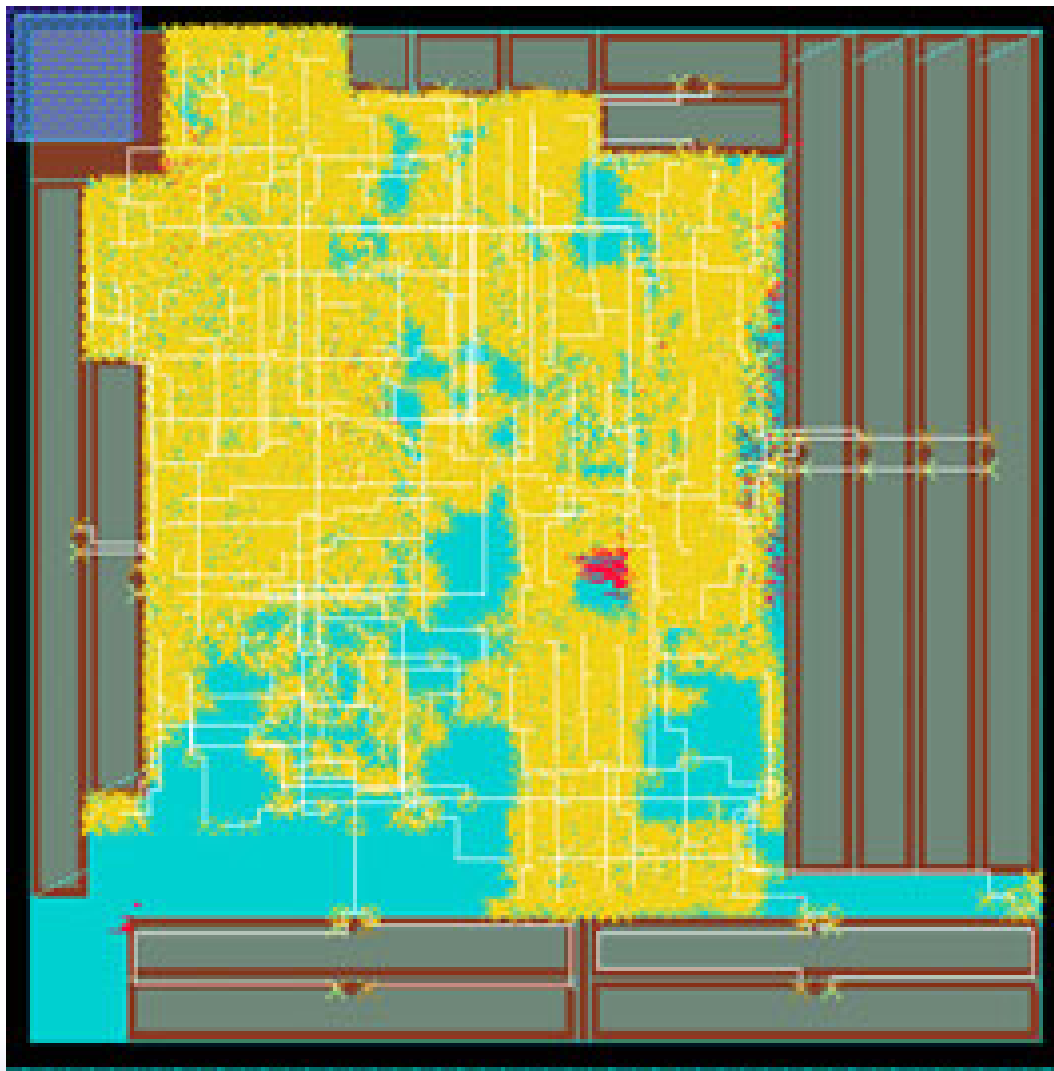


EV6 Clock Results



GCLK Skew
(at Vdd/2 Crossings)

龙腾R1的时钟树分布



时钟偏差和抖动的处理方法

- 为使偏差最小，可以采用H树或者更为一般的布线匹配的树结构，使从中央时钟分配源到单个钟控元件的时钟路径均衡。在采用时钟树布线时，必须使包括导线和晶体管负载在内的每条路径时钟负载相等。
- 采用局部时钟网络（而不是树型布线）可以减小时钟偏斜，但代价是增加了电容负载和功耗。

时钟偏差和抖动的处理方法（续）

- 如果与数据相关的时钟负载变化引起了显著的抖动，就应当使用时钟负载不受数据影响的差分寄存器。
- 数据如果沿一个方向流动，可以使数据和时钟按相反方向布线。这样可以消除竞争，代价是性能有所降低。

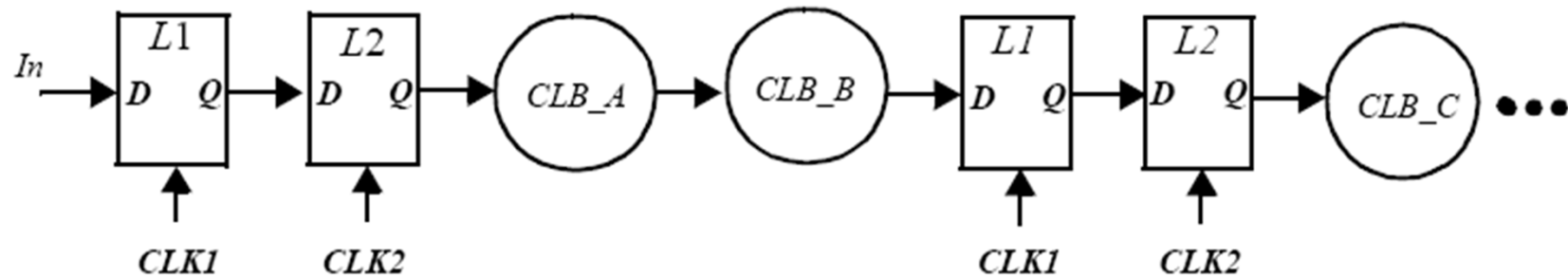
时钟偏差和抖动的处理方法（续）

- 通过将时钟线与相邻信号线屏蔽，可以避免与数据相关的噪声。把电源线（Vdd或GND）放在时钟线的旁边可以减少或避免与相邻信号网络的耦合。
- 由于层间绝缘电介质厚度不均造成互连电容的变化可以通过采用虚设填料来大大减少。虚设填料的是使用非常普遍，它通过提高绝缘层的一致性来减少时钟偏差。系统性的偏差应通过模拟并予以弥补。

时钟偏差和抖动的处理方法（续）

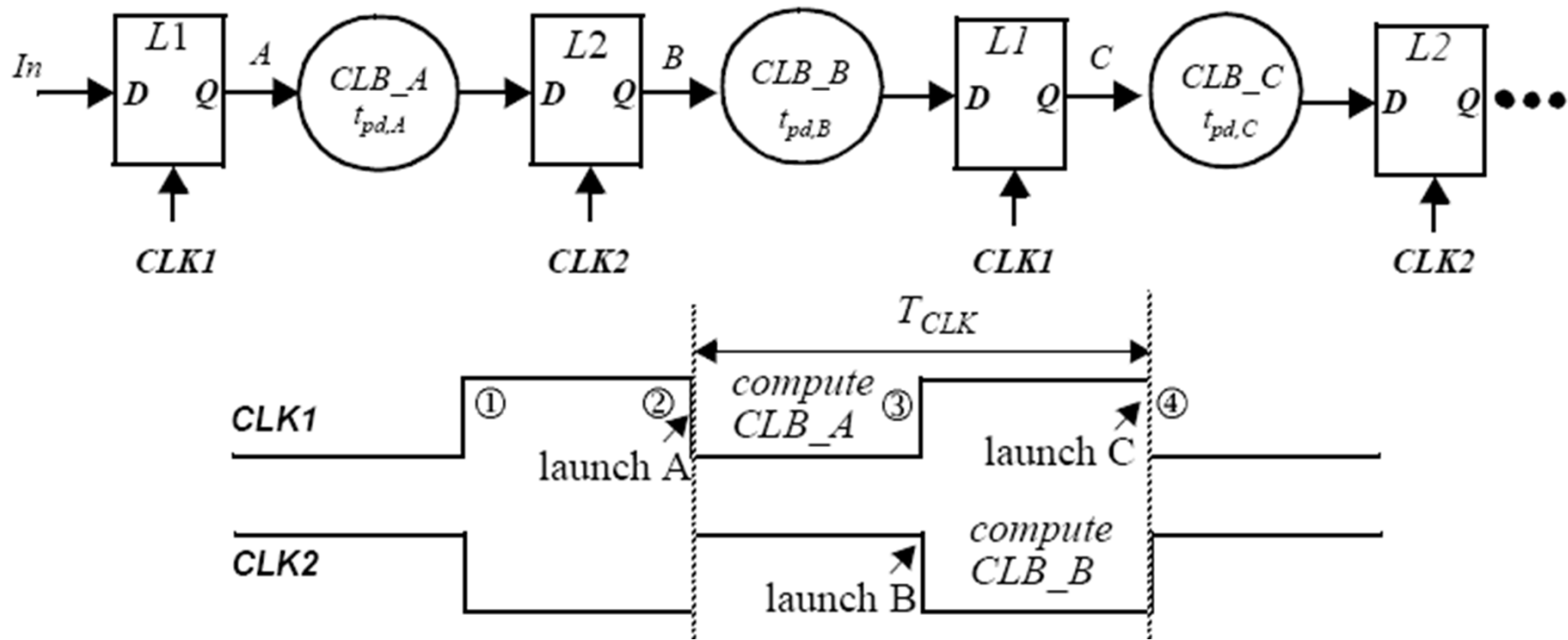
- ❑ 芯片上各处温度不均匀会引起时钟缓冲延时的变化，采用基于延时锁定环（delay-locked loop）的反馈电路可以弥补温度偏差问题。
- ❑ 电源供电不稳是引起抖动的重要原因，因为这会使经过时钟缓冲器的延时在周期与周期之间不同。通过加入片上去耦电容可以减少高频的电源电压变化。遗憾的是，去耦电容需要很大的面积，因此必须采取有效的封装方法以缩小芯片面积。

四、锁存式时钟控制



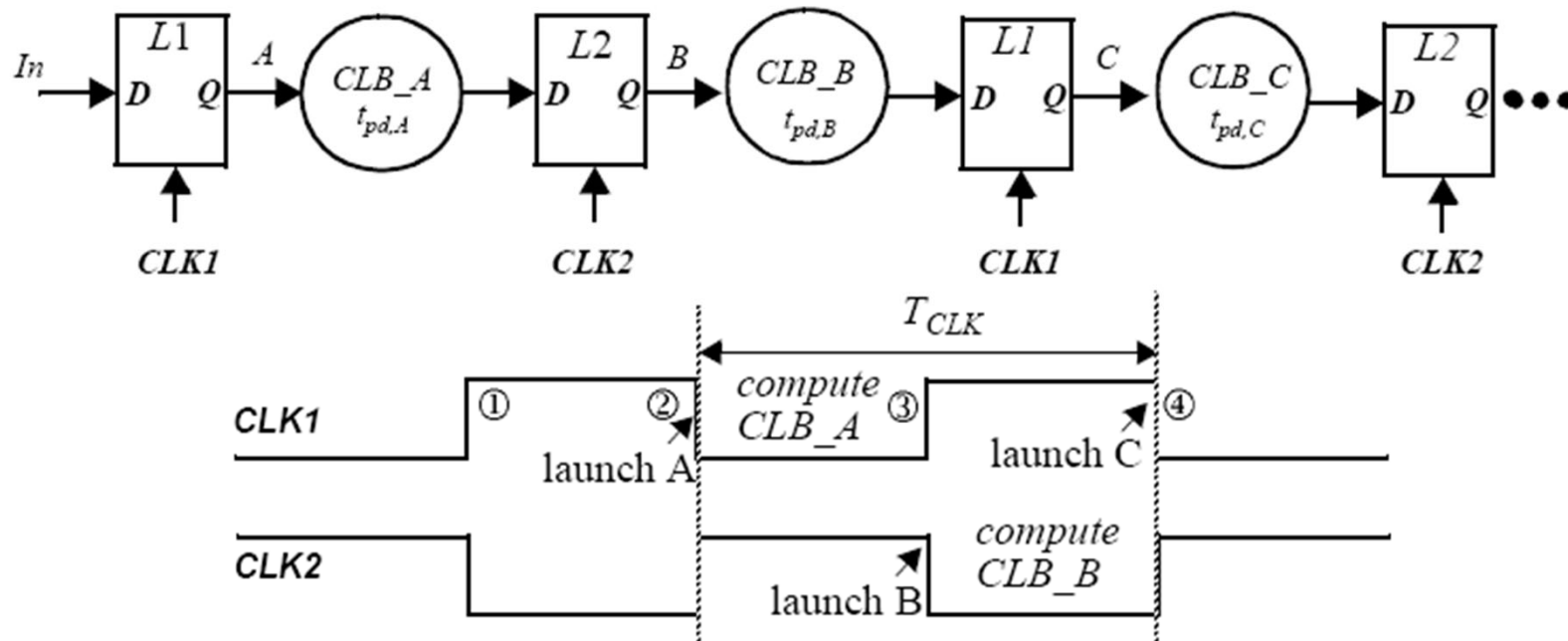
- ❑ 在边沿触发的系统中，两个寄存器之间最坏情况下的逻辑路径决定了整个系统的最小时钟周期。
- ❑ 如果一个逻辑块在时钟周期结束之前已完成操作，它必须等待下一个时钟边沿。
- ❑ 采用锁存式方法使时序控制更加灵活，它允许这一级可以让出等待时间或从其它级借用。
- ❑ 把锁存式方法看做是在主从触发器的两个锁存器之间加入了一个逻辑。

四、锁存式时钟控制



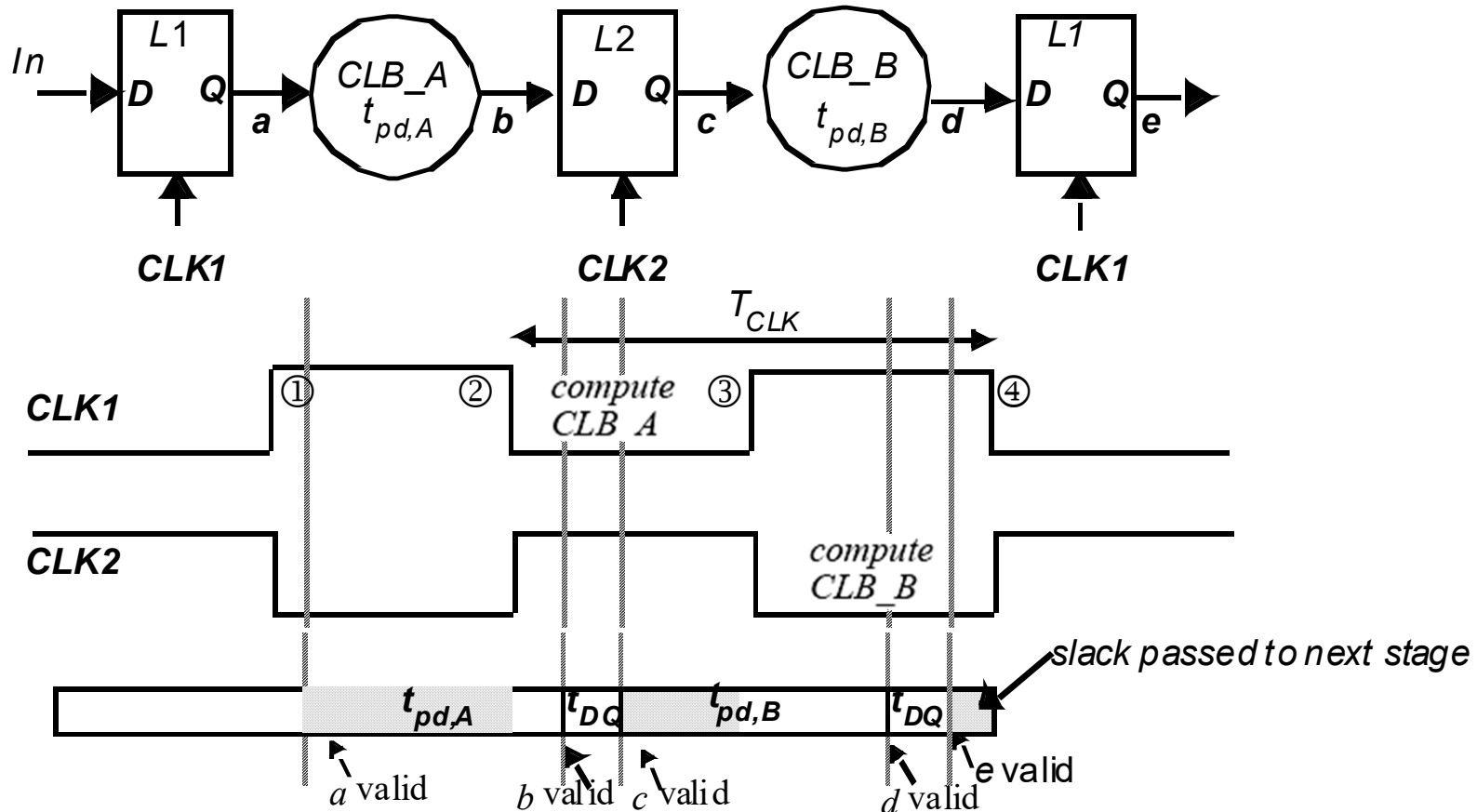
- CLK1和CLK2是互为反向的两项时钟。
- 在CLK1的下降沿(边沿②)处组合逻辑CLB_A有一个稳定的输入。它的最大求值时间等于 $T_{CLK}/2$ ，即CLK1全部为低电平相位的时间。
- 在CLK2的下降沿(边沿③)，锁存输出CLB_A，并开始计算CLK_B。
- CLK2的低电平计算CLK_B，并在CLK1的下降沿(边沿④)得到输出。
- 类似于边沿触发系统。用来计算CLB_A和CLB_B组合的时间都是 T_{CLK}

四、锁存式时钟控制



- ❑ 在锁存式系统中，由于锁存器在其导通期间是透明的，一个逻辑块可以利用上一个逻辑块没有用完的时间，称之为用剩时间借用。
- ❑ 优点：允许处于周期边界之间的逻辑使用比一个周期更长的时间，同时满足对周期时间的总限制。
- ❑ CLB_A 的输入信号必须在 $CLK1$ 的下降沿(边沿②)有效，如果前一级结束的较早， CLB_A 可以使用的最长时间是它的相位时间加上前面计算的剩余时间。

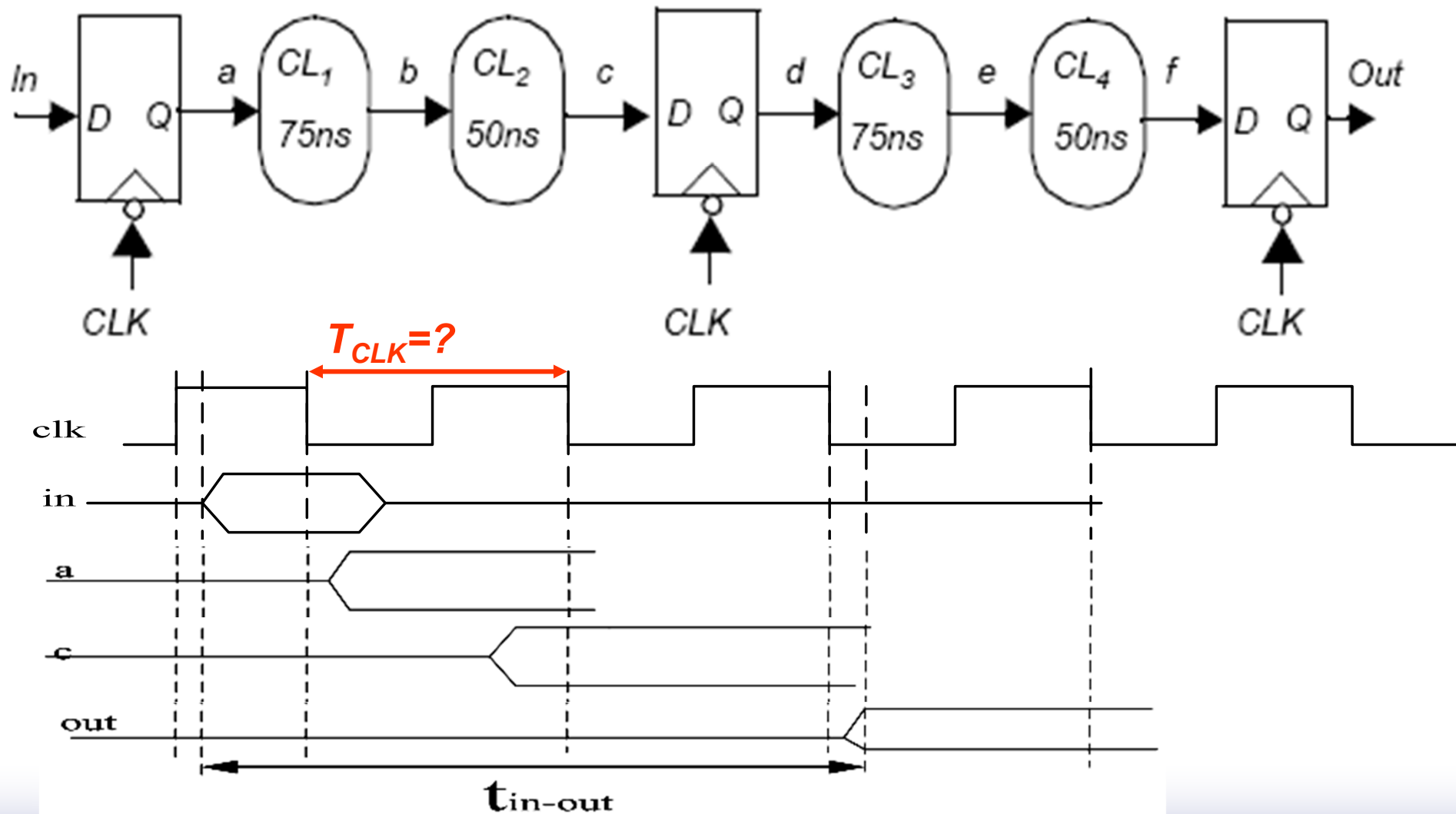
用剩时间借用



- 信号*a*在边沿②之前有效，*CLB_A*开始计算，要求在③之前完成计算。
- 信号*c*有效，*CLB_B*开始计算，要求在④之前完成计算。
- 整个工作周期的延时*CLB_B*和*CLB_A*延时之和大于一个时钟周期。
- 上一级中可借用到的最长时间是半个时钟周期。最大的逻辑延时为1.5个时钟周期。

Register —Based Pipeline

P379 例10.3



P42

Latch —Based Pipeline

