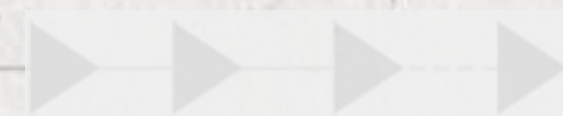


芯动力——硬件加速设计方法

第三章 同步电路与跨时钟域电路设计(2)

邸志雄@西南交通大学
zxdi@home.swjtu.edu.cn



亚稳态—单bit信号的跨时钟域传输

亚稳态

双锁存器法

优缺点、适用条件？

使用几级DFF进行异步信号同步最合适？

寄存器同步法是否可以消除亚稳态？

单bit信号跨时钟域如何传输？

慢到快如何实现

双锁存器法

边沿检测同步电路适用条件？

脉冲同步器适用条件？

快到慢如何实现

结绳法对脉冲宽度和间隔的需求？

结绳法分类？

采样中“快到慢”与“慢到快”在考虑问题时有什么区别？

为什么不能用慢时钟直接采用快时钟域的信号？

多bit指示信号跨时钟域如何传输？

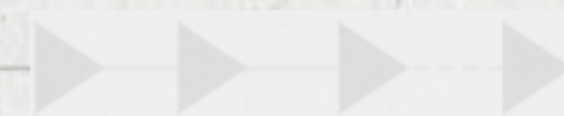
多bit数据流跨时钟域如何传输？

亚稳态—单bit信号的跨时钟域传输

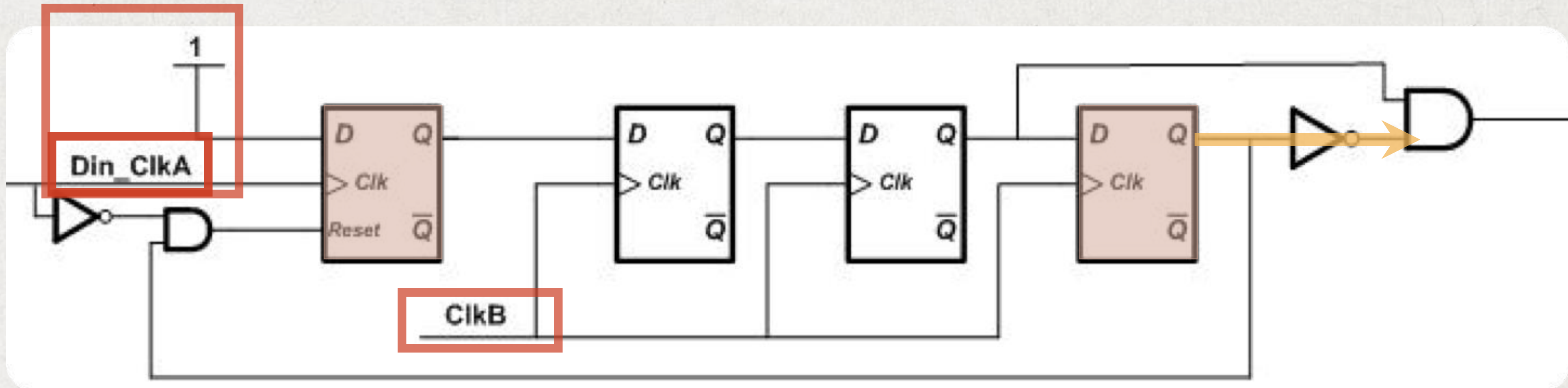
- 信号从快时钟域到慢时钟域过渡时，慢时钟将可能无法对变化太快的信号实现正确采样；
- 上述同步器法对两个时钟之间的关系要求很严格，而“结绳法”适合任何时钟域的过渡。

“结绳法” 原理：

- 将快时钟信号的脉冲周期延长，等到慢时钟同步采样后再“解绳”，还原为原来的脉冲周期宽度。



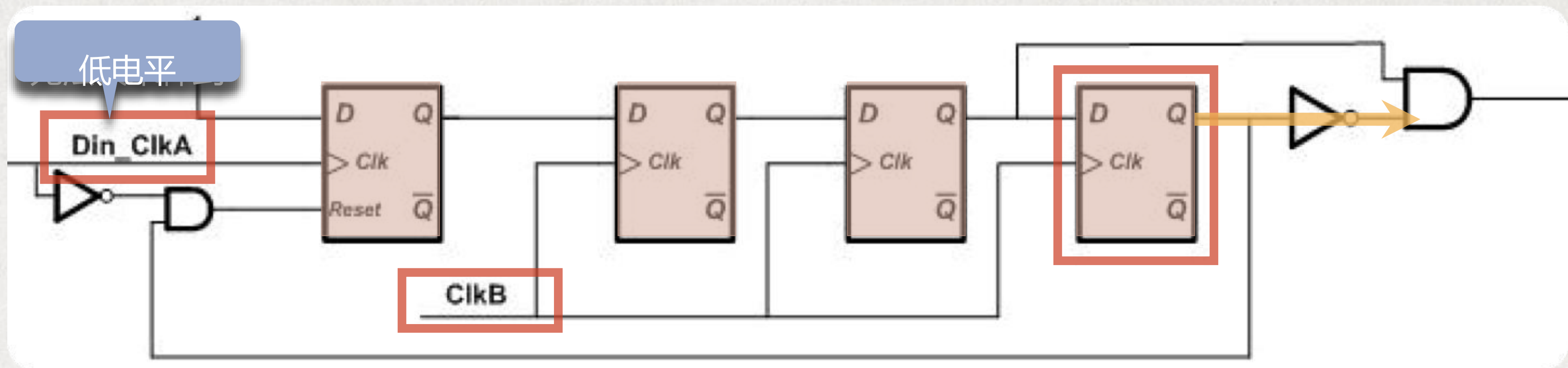
- 利用数据的边沿作时钟（图中上升沿）。（可以将脉冲无限延长，直到可以采集到数据，然后复位，要考虑产生数据的频率）。



说明：

- 这种结绳法的原理是，数据作为Din clkA，即当数据有上升沿(0->1)时，寄存器1的输出将会稳定在高电平，此时等待ClkB采样；当ClkB完成采样后，寄存器4会输出高电平，若此时Din_clkA为低电平，那么即可完成复位，开始下一次采样等待。

- 这里需要注意的是当数据来临(即上升沿)时, ClkB域需要等待3个ClkB才会在寄存器4输出并完成输入端的复位, 所以Din_ClkA如果变化较快, 即持续时间短于3个ClkB, 也就是Din_ClkA频率大于ClkB的1/3, 那么这时Din_ClkA的变化将无法被采样到, 因为ClkB域需要3个ClkB才能完成采样, 并且此时Din_ClkA必须是低电平才能复位, 采用异步复位。

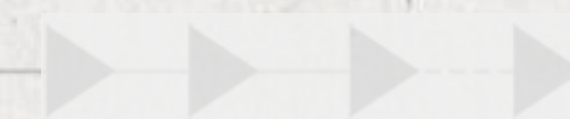
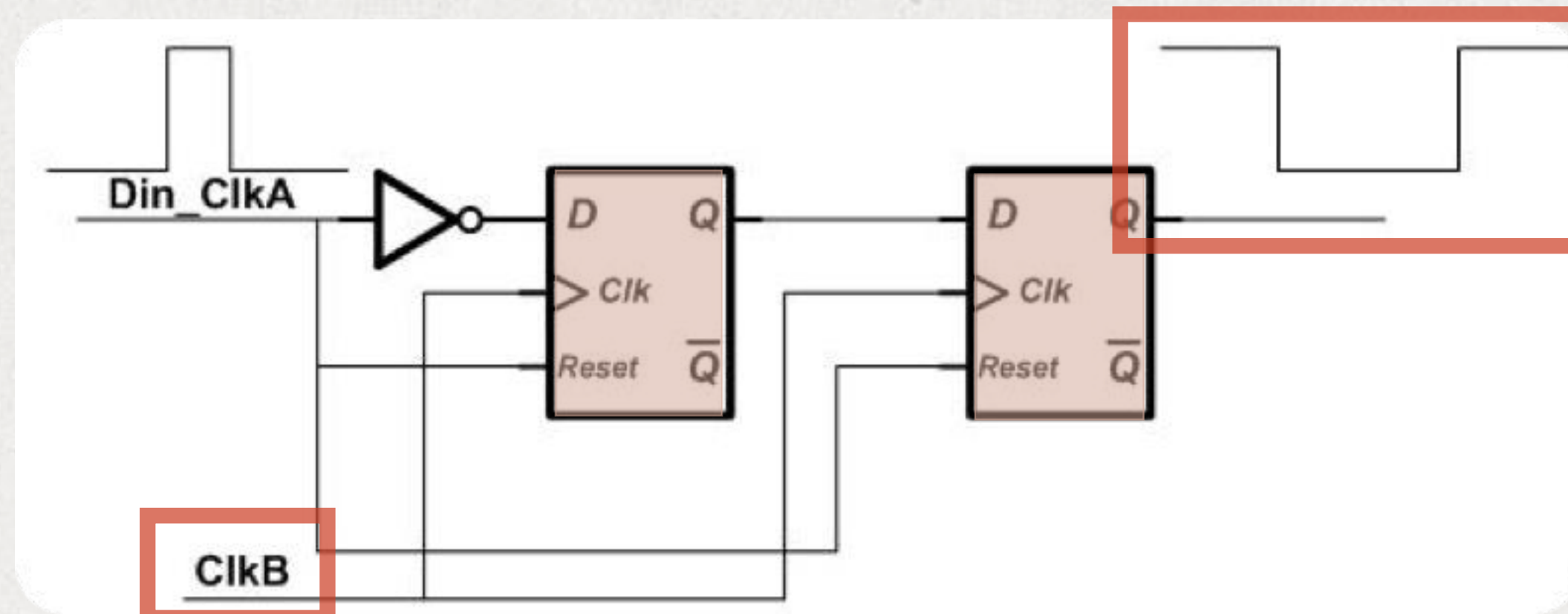
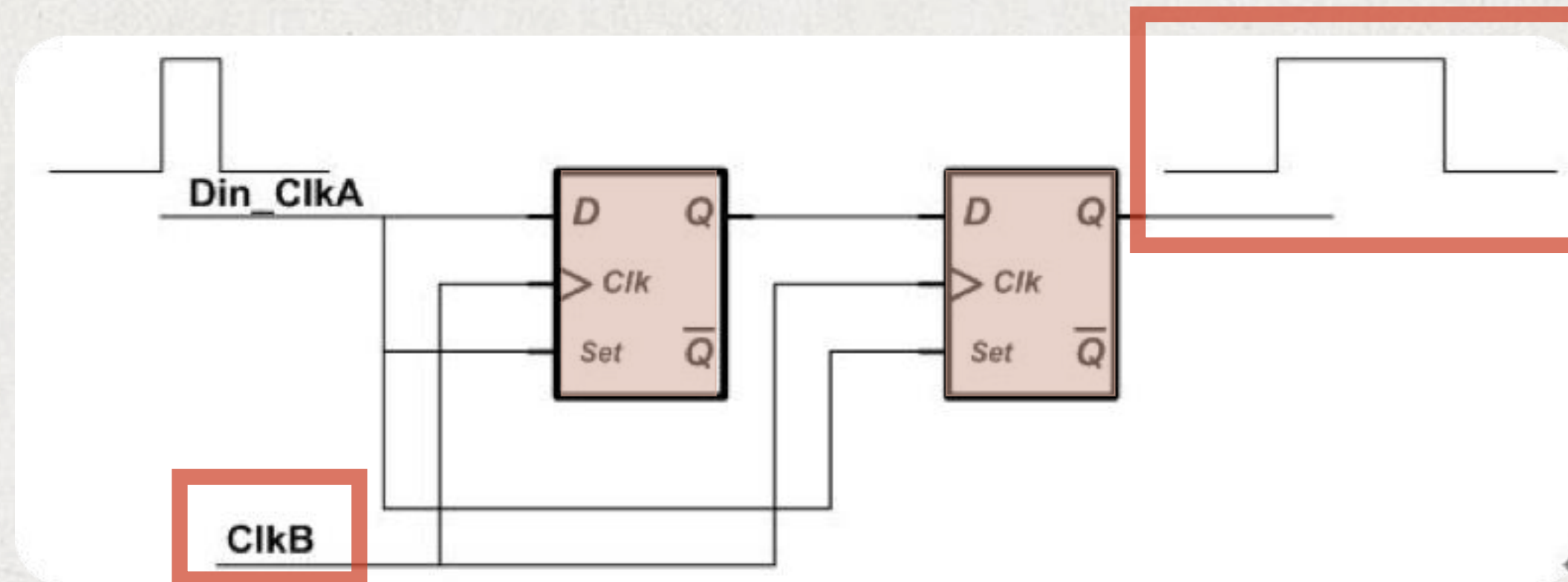


在慢时钟采样快时钟的时候, 结绳法适合采样数据较少(即脉冲间隔较大)的控制信号。即脉冲间隔 $T_a > 3T_b$; 即等待3个clkB时钟后, 完成复位, 才允许下一个输入脉冲。

快时钟→慢时钟:

结绳法2

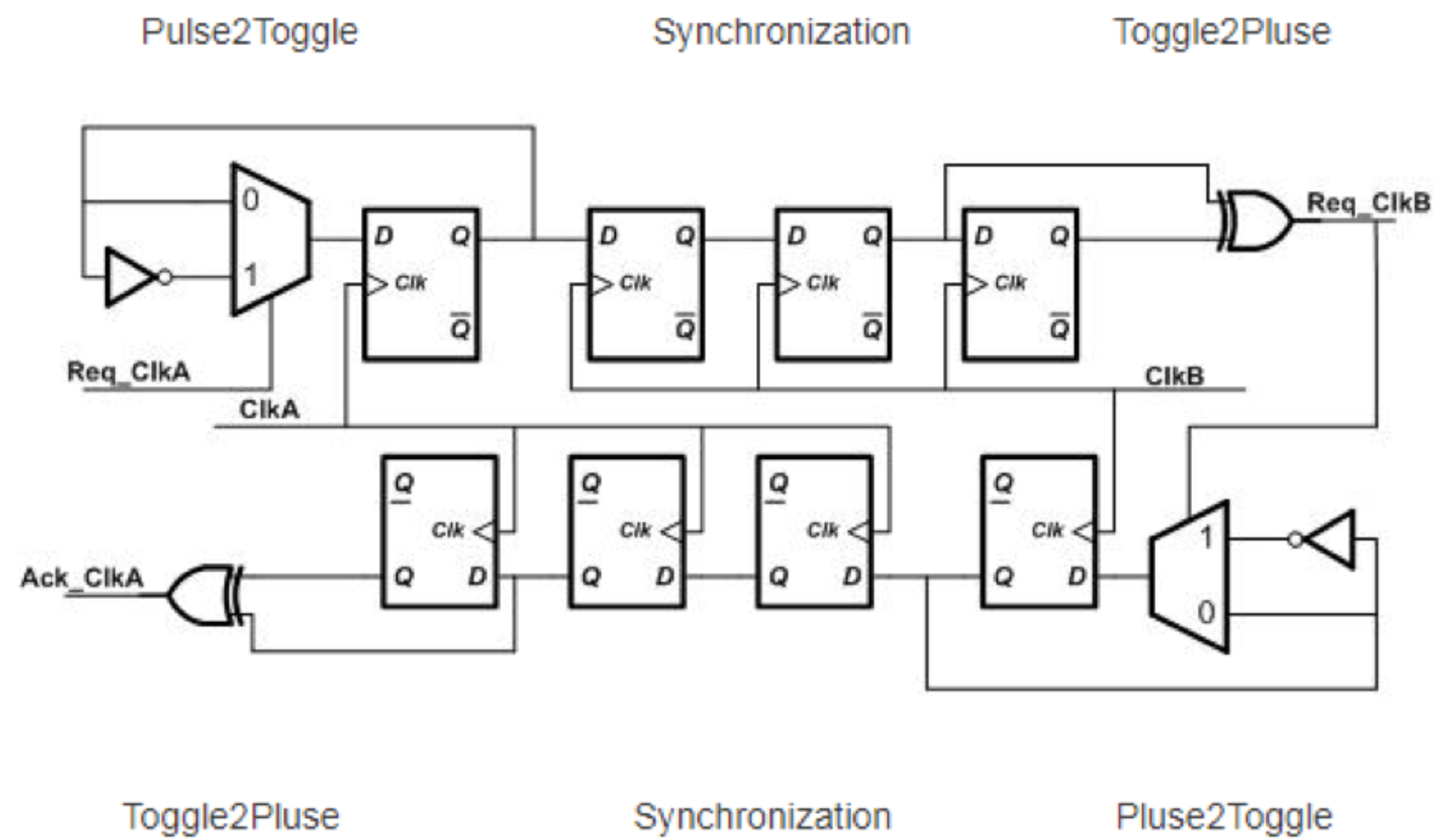
- 利用数据作为异步复位, 置位信号



快时钟→慢时钟:

结绳法3

• 利用握手协议



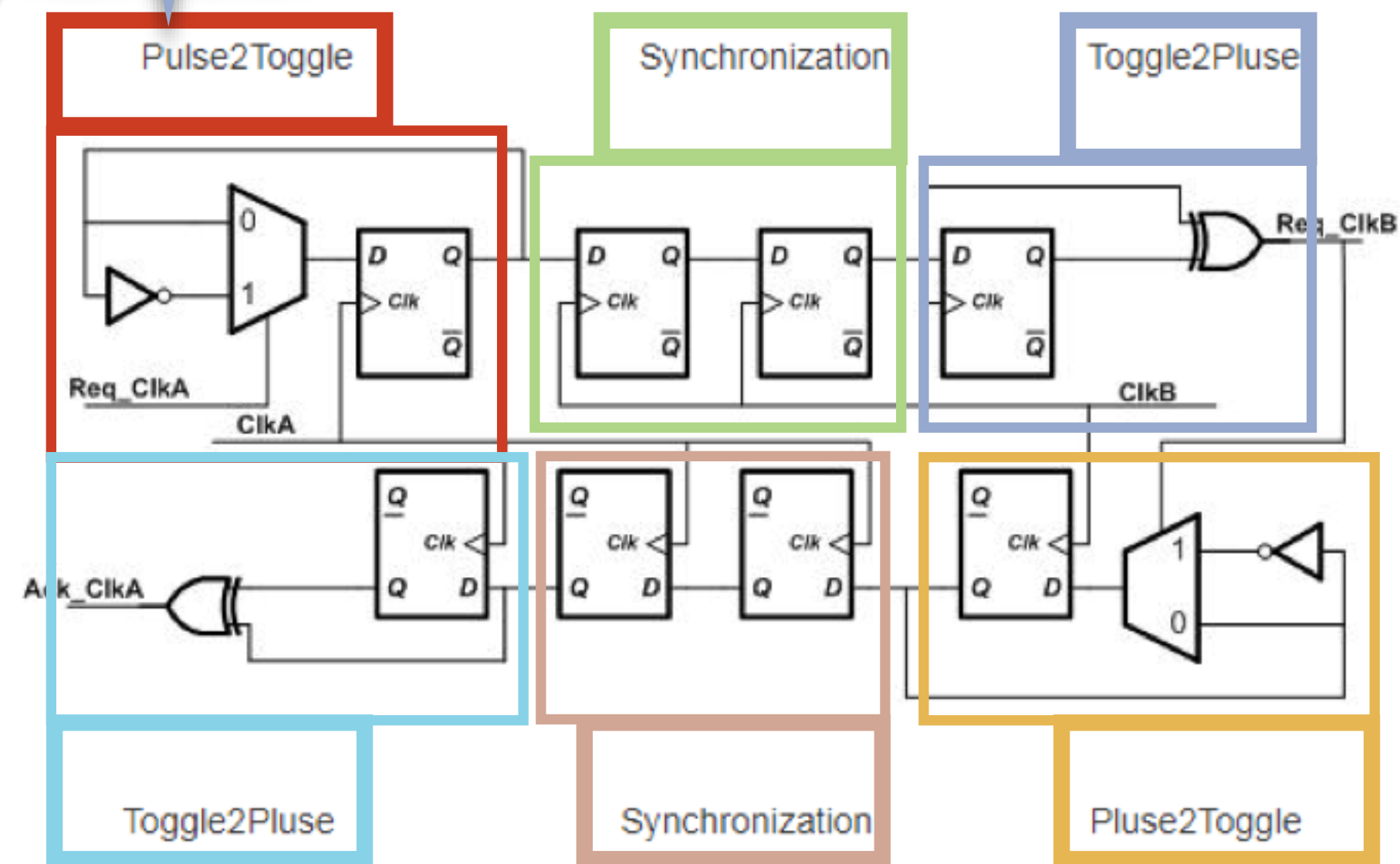
快时钟→慢时钟:

结绳法3

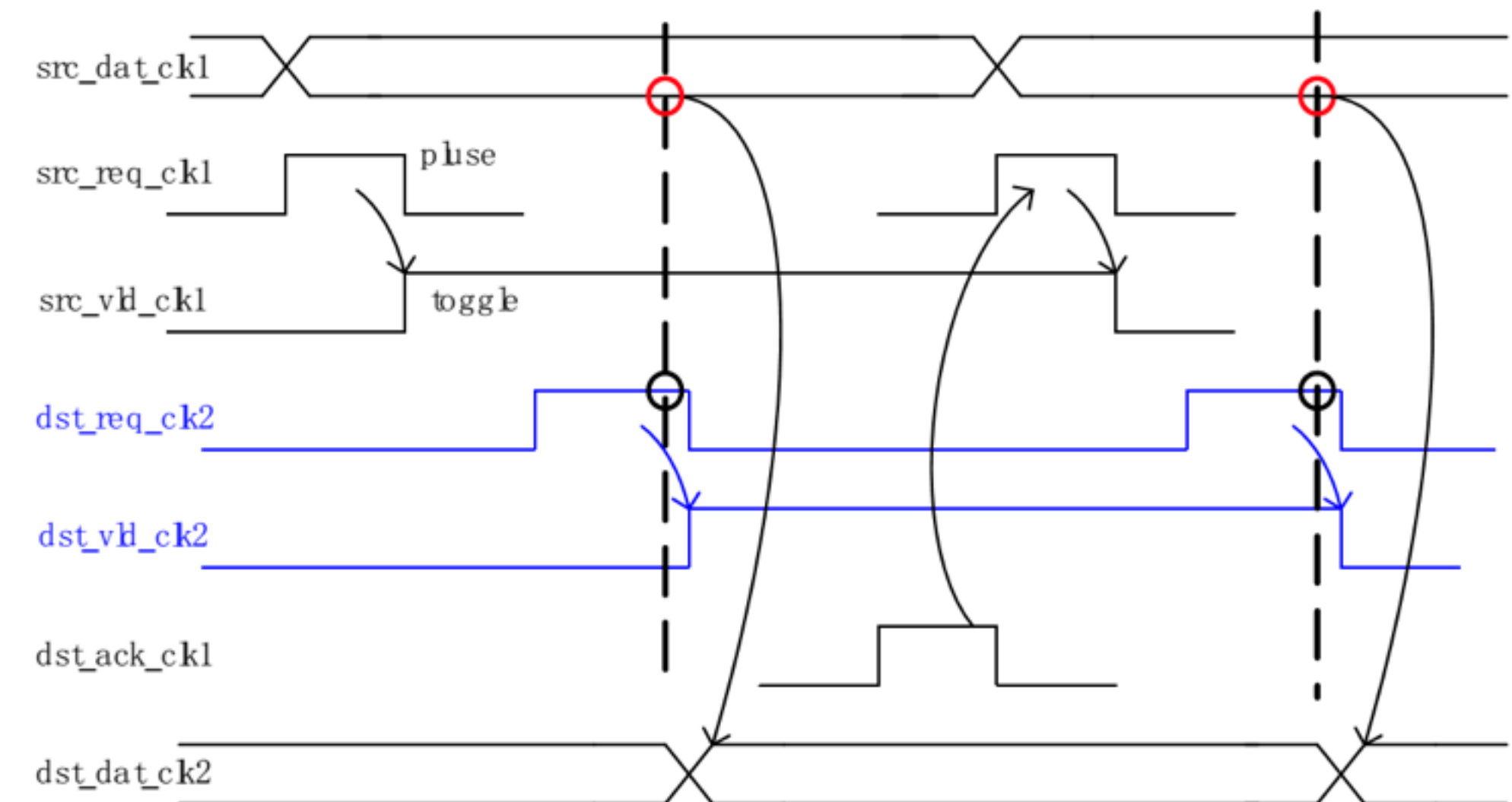
• 利用握手协议

翻转电路

将快时钟域的信号进行结绳



Pulse-to-toggle:



快时钟→慢时钟

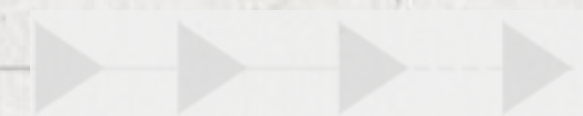
- 结绳就是将单脉冲延长，以方便采集到数据。


结绳的方法归结为2类:

- 利用脉冲的边沿做时钟;
 - 利用脉冲的电平(部分场合要求最小脉冲宽度)做选择器或者异步复位, 置位。
- 另外的关键点就是什么时候结绳结束(采集到了数据就要让对方回到初始状态)

这里的操作也有2种方法:

- 利用采集到的脉冲做异步复位, 置位。
- 利用采集到的脉冲再次结绳采集做握手响应信号。

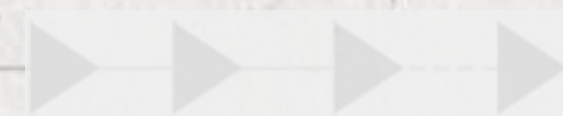




快时钟→慢时钟

总结:

- 结绳法可以解决快时钟域向慢时钟域过渡的问题，且其适用的范围很广。
- 结绳法实现较为复杂，特别是其效率不高，在对设计性能要求较高的场合应该慎用。



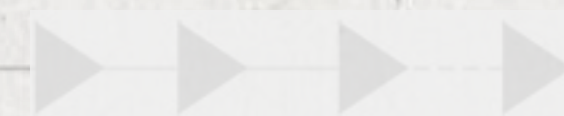
- 采样中“快到慢”与“慢到快”在考虑问题时有什么区别?
- 为什么不能用慢时钟直接采用快时钟域的信号?

慢到快

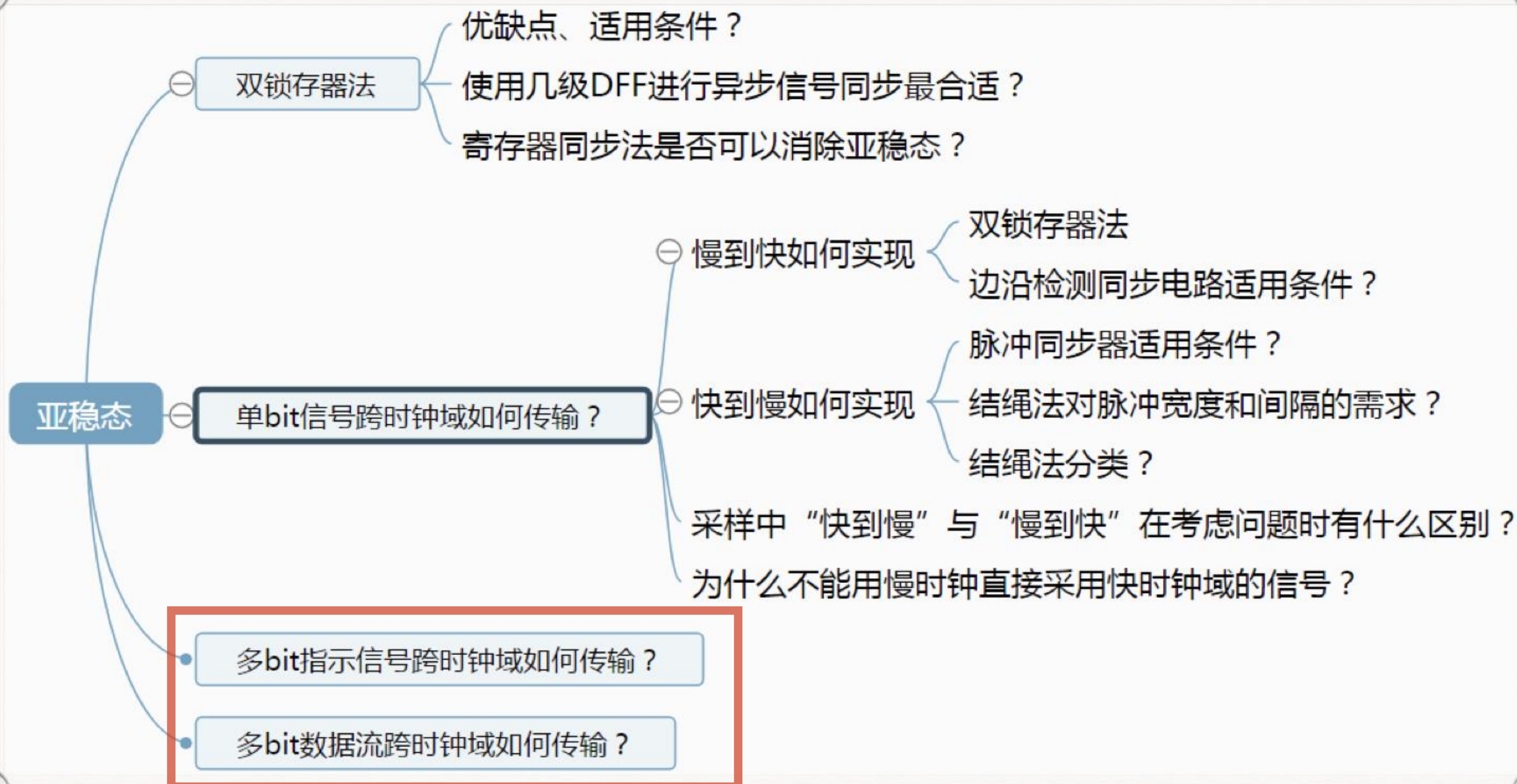
- 只需要考虑亚稳态问题

快到慢

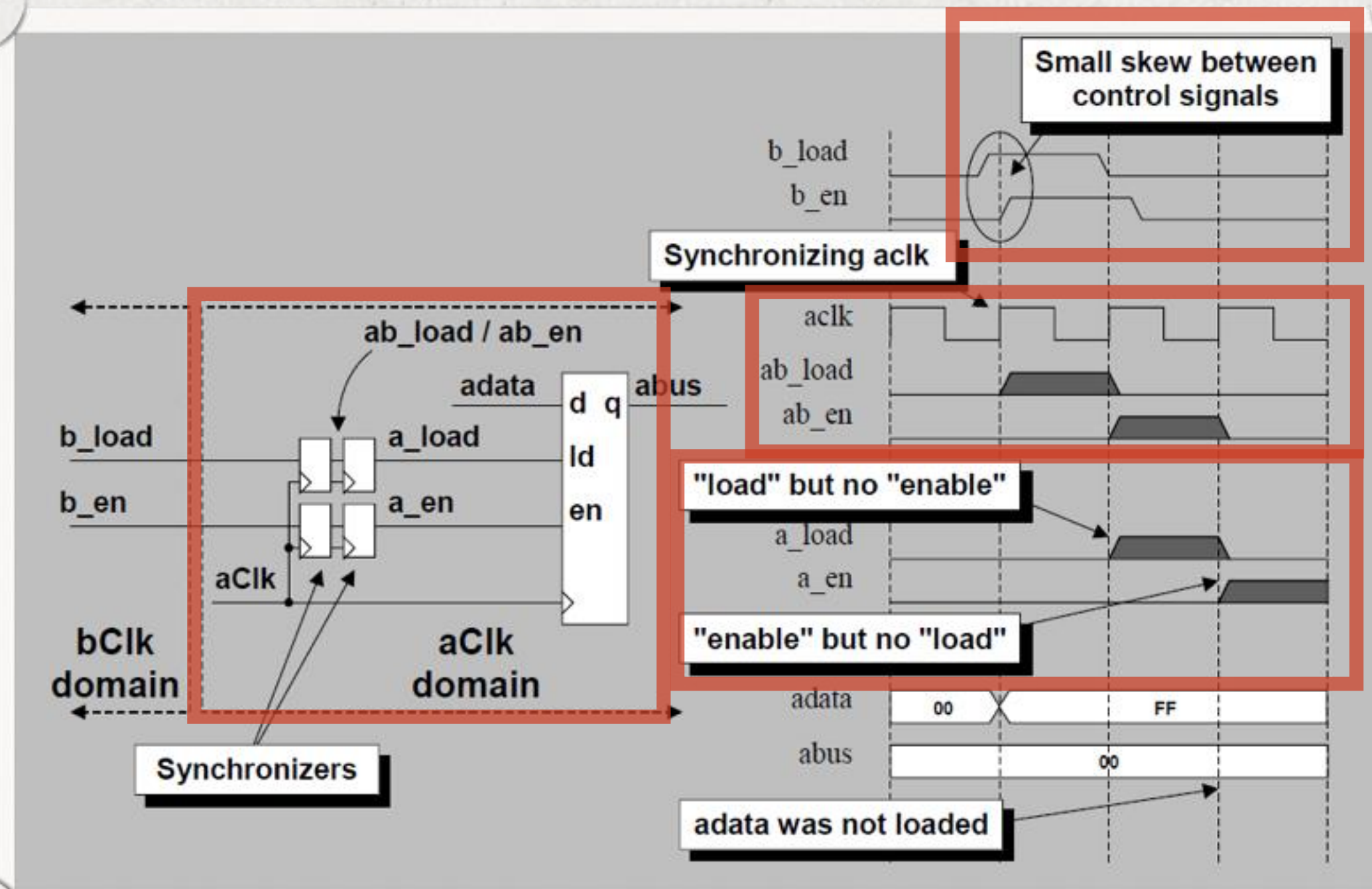
- 除亚稳态问题外，还需考虑慢时钟的采样速率问题。因为根据采样定理，采样频率低于信号最高频率2倍的时候，是无法完整采样的。



亚稳态—多bit信号的跨时钟域传输



- 如何传递两个同时需要的信号(b_load和b_en)?

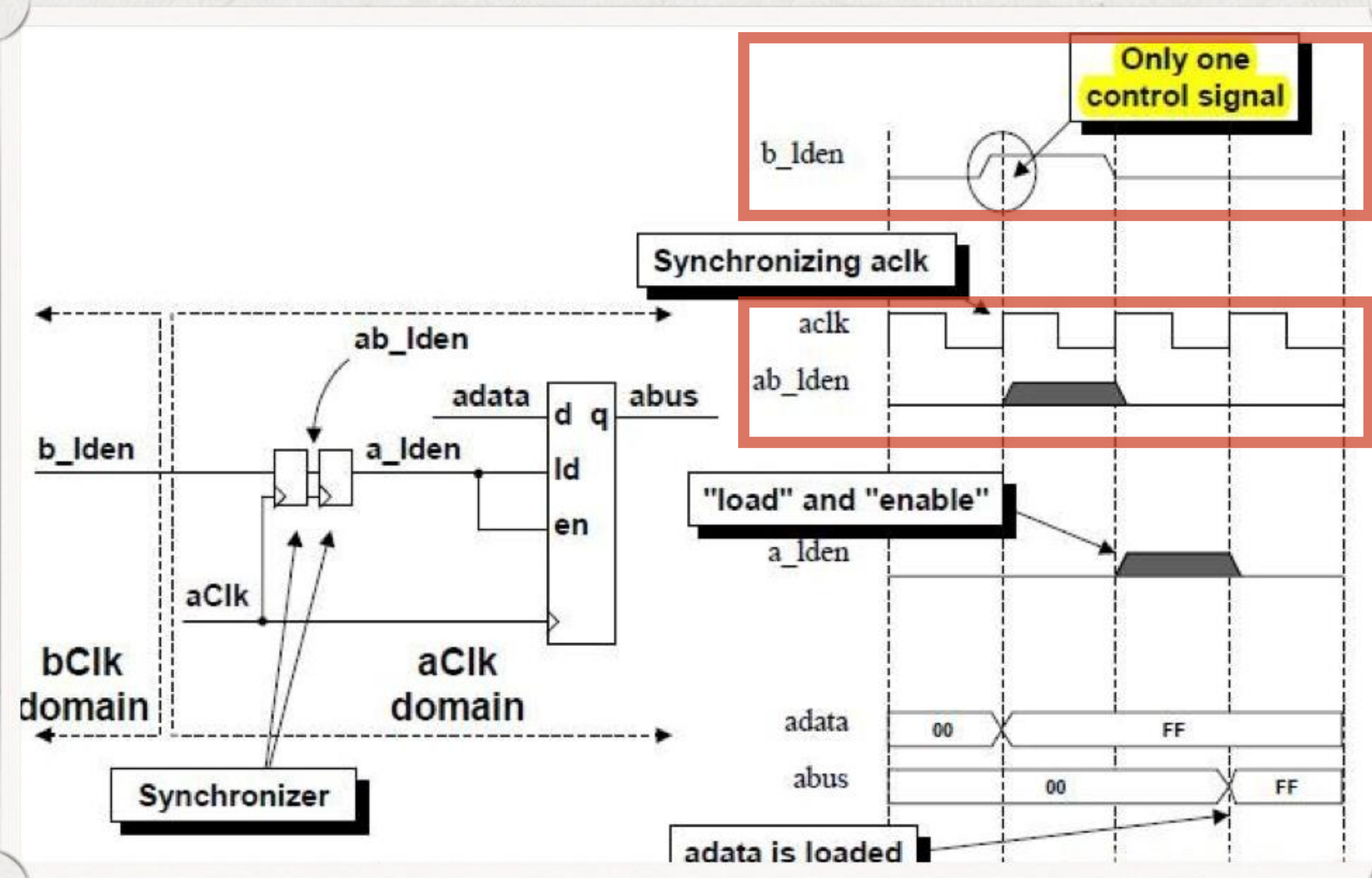


多个控制信号跨时钟域
仅仅通过简单的同步器
同步有可能是 unsafe 的

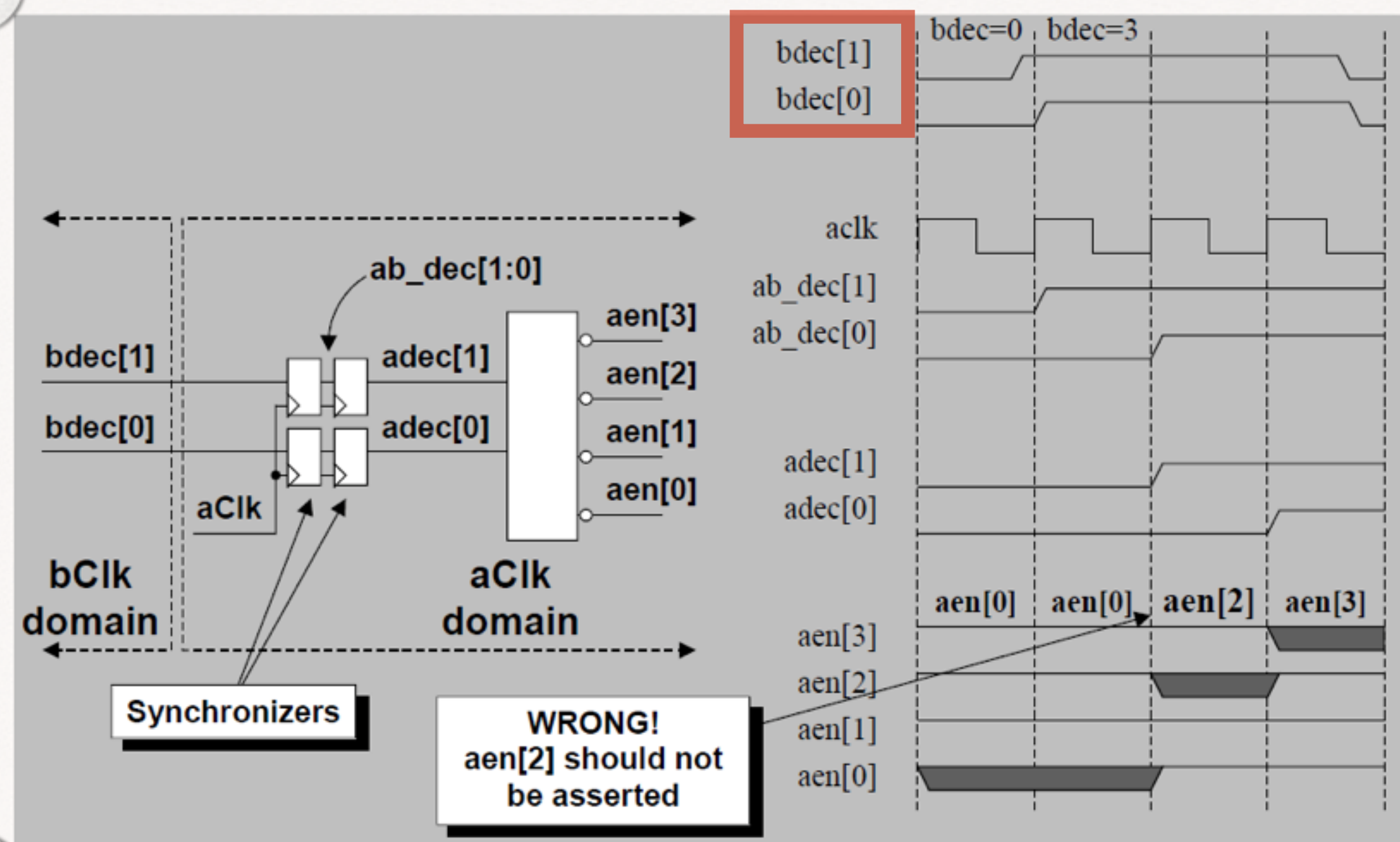
b_load和b_en同步至a_clk时钟域，如果这两个信号有一个小的skew，将导致在a_clk时钟域中两个信号并不是在同一时刻起作用，与在b_clk中的逻辑关系不同。

解决方法:

- 将b_load和b_en信号在b_clk时钟域中合并成一个信号b_lden, 然后同步至a_clk中。



- 如果遇到不能合并的情况，如译码信号。



如果Bdec[0]、bdec[1]间存在 skew将导致同步至a_clk中后译码错误，出现误码。

在这种情况下，建议加入另一个控制信号，确保bdec[0]、bec[1]稳定时采。例如在bdec[0]、bec[1]稳定输出后一到两个周期b_clk域输出一个en信号，通知a clk域此时可以采bdec[0]、bec[1]信号。这样可确保正确采样。



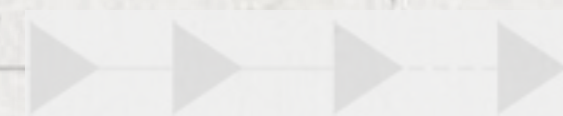
多bit数据流跨时钟域如何传输?

数据流和指示信号不同:

- 数据流大多具有连续性，即背靠背传输；
- 数据流要求信号具有较快的传输速度；

解决方法:

FIFO



同步策略 — FIFO

FIFO

first in first out

- 是一种先进先出的储存结构

与普通存储器的区别:

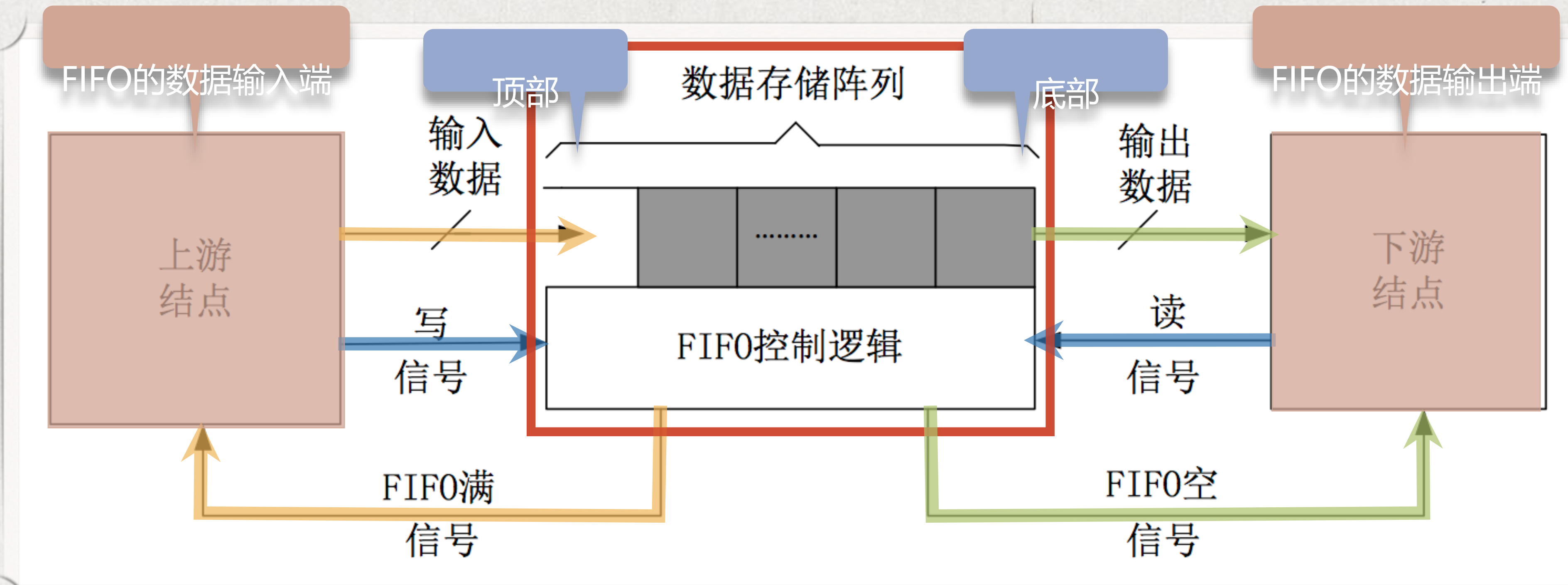
- 没有外部读写地址线 使用简单

缺点:

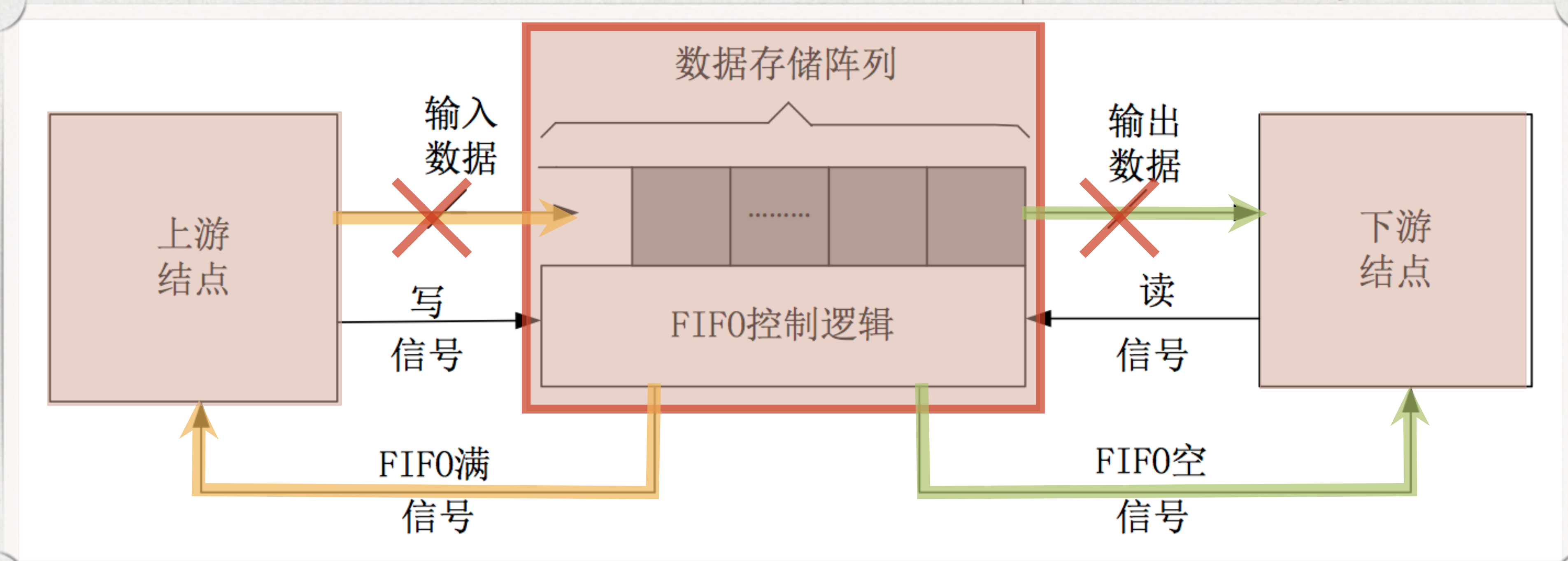
- 只能顺序写入数据，顺序的读出数据，其数据地址由内部读写指针自动加1完成，不能像普通存储器那样可以由地址线决定读取或写入某个指定的地址。

IC

- 模块与模块之间的通信设计中，多时钟的情况已经不可避免；数据在不同时钟域之间的传输很容易引起亚稳态；异步FIFO就是一种简单、快捷的解决方案。



- FIFO的上游结点是FIFO的数据输入端，在写信号有效时，数据将被写入FIFO的顶部（由FIFO内部的写指针控制），并且在FIFO内部，写指针后移一个单元，同时FIFO的满信号（FIFO full Signal）将控制上游结点是否发送数据；FIFO的下游节点是FIFO的数据输出端，当读信号有效时，FIFO中位于FIFO底部单元的数据将被读出（由FIFO内部的读指针控制），并且在FIFO内部读指针将后移一个单元，同时FIFO空信号（FIFO empty Signal）将控制下游节点是否读出数据。



- FIFO的上游结点是FIFO的数据输入端，在写信号有效时，数据将被写入FIFO的顶部（由FIFO内部的写指针控制），并且在FIFO内部，写指针后移一个单元，同时FIFO的满信号（FIFO full Signal）将控制上游结点是否发送数据；FIFO的下游节点是FIFO的数据输出端，当读信号有效时，FIFO中位于FIFO底部单元的数据将被读出（由FIFO内部的读指针控制），并且在FIFO内部读指针将后移一个单元，同时FIFO空信号（FIFO empty Signal）将控制下游节点是否读出数据。

同步策略 — FIFO

用途1:

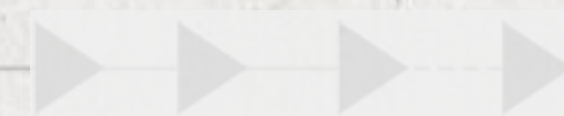
异步FIFO读写分别采用相互异步的不同时钟

- 在现代集成电路芯片中，随着设计规模的不断扩大，一个系统中往往含有数个时钟，多时钟域带来的一个问题就是，如何设计异步时钟之间的接口电路。异步FIFO是这个问题的一种简便、快捷的解决方案，使用异步FIFO可以在两个不同时钟系统之间快速而方便地传输实时数据。

用途2:

对于不同宽度的数据接口也可以用FIFO

例如单片机位8位数据输出，而DSP可能是16位数据输入，在单片机与DSP连接时就可以使用FIFO来达到数据匹配的目的。





同步策略 — FIFO

FIFO结构分类:

同步FIFO

- 指读时钟和写时钟为同一个时钟，在时钟沿来临时同时发生读写操作；

异步FIFO

- 指读写时钟不一致，读写时钟是互相独立的。

