ASIC 中的异步时序设计

王夏泉

(华中科技大学 电子与信息工程系，武汉 430074)

摘要：绝大部分的 ASIC 设计工程师在实际工作中都会遇到异步设计的问题，本文针对异步时序产生的问题，介绍了几种同步的策略，特别是结绳法和异步 FIFO 的异步比较法都是比较新颖的方法。

关键词：异步时序，MTBF，双锁存器法，结绳法，异步 FIFO，异步比较。

Asynchronous design in ASIC

Abstract: Most of the ASICs that are ever designed are driven by multiple asynchronous clocks. Aiming at the issue of asynchronous design, this paper introduced several solutions. Especially, the methods of toggle and asynchronous compare in asynchronous FIFO design are good ideas.

Keywords: asynchronous timing, MTBF，two registers, toggle, asynchronous FIFO, asynchronous compare.

1. 前言

在一般的 ASIC 教程中，大家接触的大都是同步时序的设计，即单时钟的设计。但是在实际的工程中，纯粹单时钟设计的情况很少，特别是在设计模块与外围芯片的通讯中，跨时钟域的情况经常不可避免。作者在实际工作中就遇到了一些异步时序设计的问题，由于最初对异步时序产生的问题估计不足，导致在设计的后期不得不对设计进行返工，本文介绍的几种同步策略也正是在实践中学习摸索的结果。本文旨在向读者介绍几种实用的同步方法，不可能对异步时序设计涉及的问题覆盖完全。由于篇幅限制，本文主要描述同步策略的核心思想，而不涉及到具体的实现。

2. 问题的产生－亚稳态

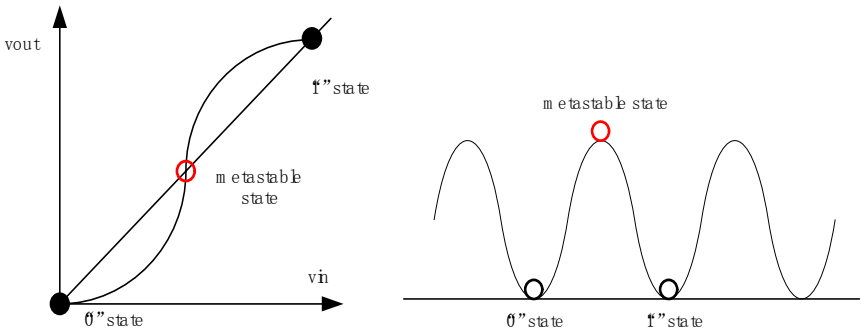
2.1 异步时序的定义

异步时序设计指的是在设计中有两个或以上的时钟，且时钟之间是同频不同相或不同频率的关系。而异步时序设计的关键就是把数据或控制信号正确地进行跨时钟域传输。

2.2 亚稳态

每一个触发器都有其规定的建立(setup)和保持(hold)时间参数，在这个时间参数内，输入信号在时钟的上升沿是不允许发生变化的。如果在信号的建立时间中对其进行采样，得到的结果将是不可预知的，即亚稳态。

下面从触发器的物理特性方面对亚稳态进行描述：[1]



2-1 亚稳态问题

触发器进入亚稳态的时间可以用参数 MTBF(mean time between failures)来描述，MTBF 即触发器采样失败的时间间隔，其公式描述如下：

$$MTBF = e^{(tr/\tau)} / T_0 fa$$

其中：

tr ： 分辨时间(从时钟沿开始)

τ ,T_0 ： 触发器参数

f ： 采样时钟频率

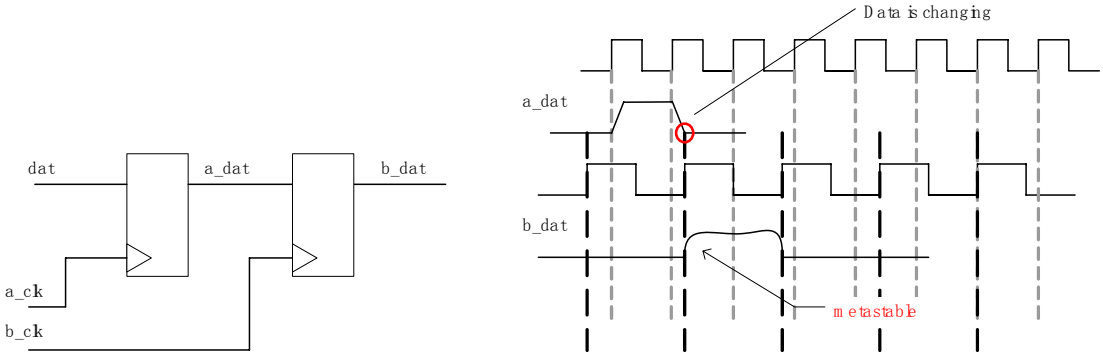
a ： 异步事件触发的频率

对于一个典型的 0.25μm 工艺的 ASIC 库中的一个触发器，我们取如下的参数：

tr = 2.3ns, τ = 0.31ns, T_0 = 9.6as, f=100MHZ, a = 10MHZ, MTBF = 2.01 days

即触发器每两天便可能出现一次亚稳态。

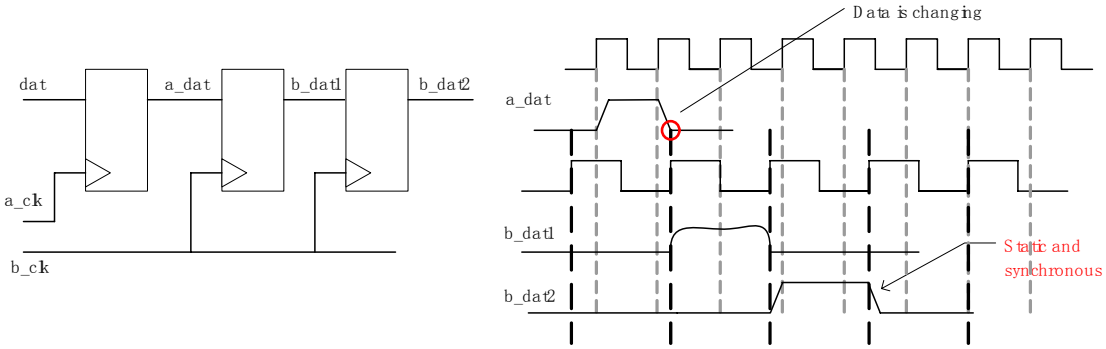
如下图所示，一个信号在过渡到另一个时钟域时，如果仅仅用一个触发器将其锁存，那么用 b_clk 进行采样的结果将可能是亚稳态。这也是信号在跨时钟域时应该注意的问题。



2-2 单锁存器法产生的问题

3. 同步策略一 — 双锁存器法

为了避免上节所述的亚稳态问题，就应当使参数 MTBF 尽可能的大，通常采用的方法是双锁存器法，即在一个信号进入另一个时钟域之前，将该信号用两个锁存器连续锁存两次，最后得到的采样结果就可以消除亚稳态问题。



3-1 双锁存器法解决亚稳态问题

当使用了双锁存器以后，b_dat2 的MTBF由以下公式可以得出：[1]

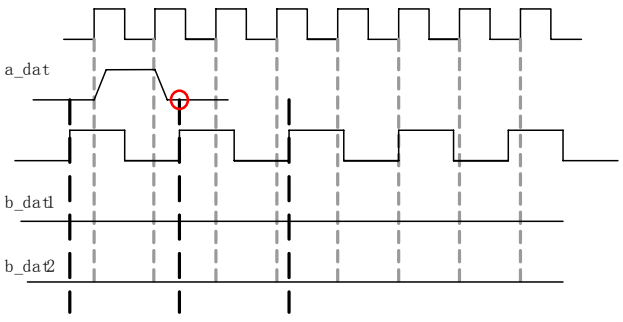
$$MTBF = e^{(tr/\tau)} / T_0 fa \times e^{(tr/\tau)} / T_0 f$$

如果我们仍然使用上一节所提供的参数，则b_dat2 的MTBF为 $9.57*10^9$(years)。由上述结果可以看出，双锁存器法可以消除亚稳态问题。

4. 同步策略二 — 结绳法

细心的读者也许会发现，在上面的例子中，如果a_clk的频率比b_clk频率高，将可能会出现因为dat变化太快而使b_clk无法采到的问题。即在信号从快时钟域向慢时钟域过渡的时候，如果信号变化太快，慢时钟将可能无法对该信号进行正确采样，如下图所示。所以在使用双锁存器法的时候，应该使原始信号保持足够长的时间，以便另一个时钟域的锁存器可以正确的对其进行采样。[3]

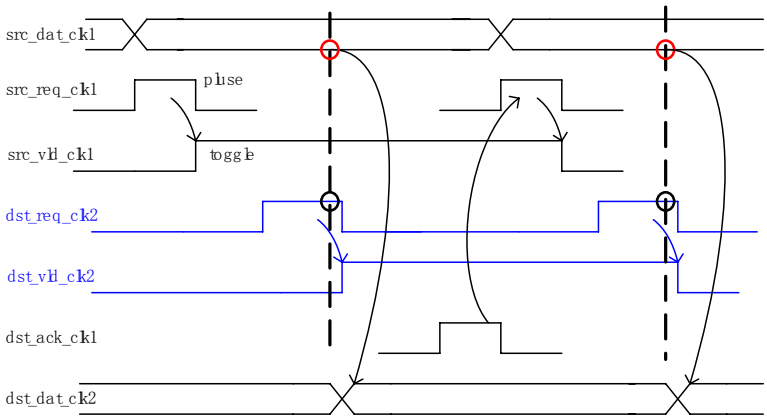
Problem：



4-1 采样失败的例子

针对上述问题，本节将介绍一种安全的跨时钟域的方法：“结绳法”。如下图所示，因为“结绳法”适合任何时钟域的过渡（clk1 和clk2 的频率和相位关系可以任意选定），所以没有表明两个时钟坐标。[2]

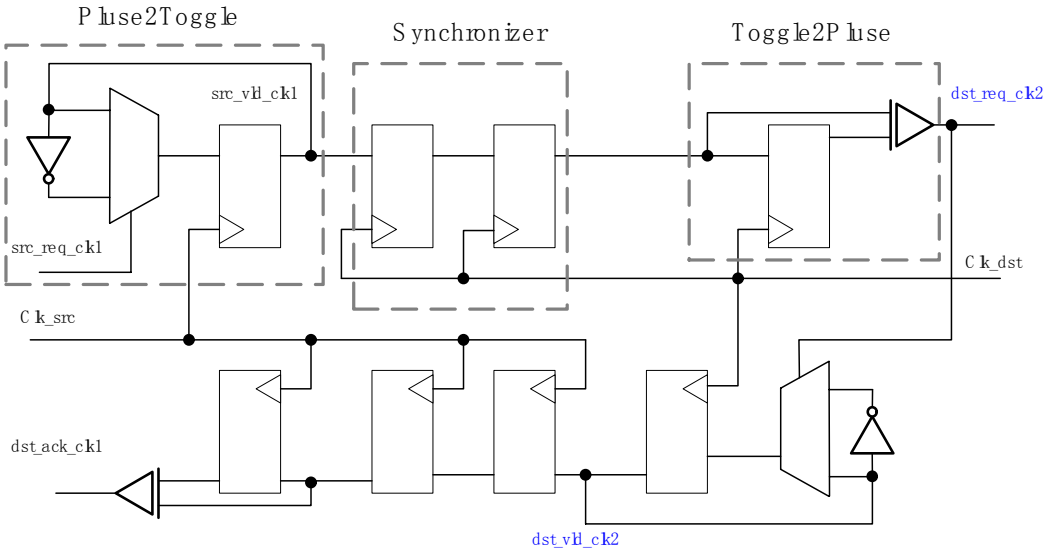
Pluse-toggle:



4-2 “结绳法”波形示意图

其中标明_clk1 的信号表示该信号属于 clk1 时钟域，同理标明_clk2 的信号表示该信号属于 clk2 时钟域。在两次 src_req_clk1 之间被 src_vld_clk1“结绳”(pluse2toggle)，在将 src_vld_clk1 用双锁存器同步以后，将该信号转换为 dst_req_clk2(toggle2pluse)。同理，用 dst_vld_clk2 将 dst_req_clk2“结绳”，dst_vld_clk2 表明在 clk2 时钟域中，src_dat_clk1 已经可以进行正确采样了。最后将 dst_vld_clk2 转换为 dst_ack_clk1(synchronizer and toggle2pluse), dst_ack_clk1 表明 src_dat_clk1 已经被 clk2 正确采样了， 此后 clk1 时钟域就可以安全地传输下一个数据了。可以看出，“结绳法”关键是将信号结绳以后，使其保持了足够长的时间，以便另一个时钟可以正确地采样。

图 5-2 描述了“结绳法”具体的实现方法，可以看出结绳法的实现主要包括三个基本单元：Pluse2Toggle, Synchronizer 和 Toggle2Pluse。



4-3 “结绳法”实现图

其中，Pluse2Toggle 模块负责将两个脉冲信号“结绳”，即将单脉冲信号延长；Synchronizer 模块即用双锁存器法将得到的信号过渡到另一个时钟域；Toggle2Pluse 模块是 Pluse2Toggle 功能相对，即将延长的脉冲信号还原为单脉冲，这里用到了异或门。整体的设计思想即用 Pluse2Toggle 将信号延长，用 Synchronizer 过渡，用 Toggle2Pluse 还原，以保证另一个时钟域可以正确采到，而接收方用相反的流程送回响应信号。

“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。但是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。

5. 同步策略三 — 异步 FIFO

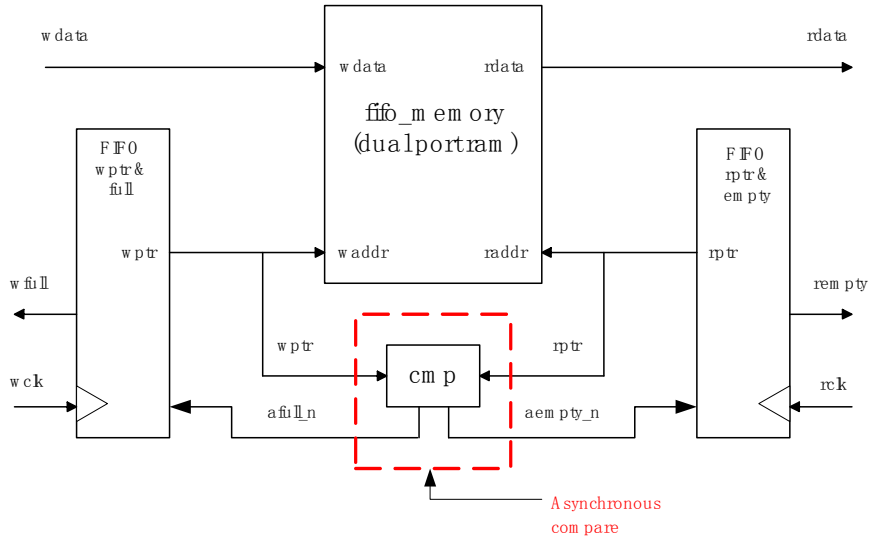
在有大量的数据需要进行跨时钟域传输，并且对数据传输速度要求比较高的场合，一般采用异步 FIFO。异步 FIFO 即用一种时钟写入数据，而用另外一种时钟读出数据，其中这两个读写时钟是异步的。

不管是什么类型的 FIFO，其关键点是产生读、写地址和空、满的标志。好的 FIFO 设计的基本要求是：写满而不溢出，能读空而不多读。

一个异步 FIFO 一般由如下部分组成：

1. Memory, 作为数据的存储器；
2. 写逻辑部分，主要负责产生写信号和地址；
3. 读逻辑部分，主要负责产生读信号和地址；
4. 地址比较部分，主要负责产生 FIFO 空、满的标志。

而其中如何正确的产生 FIFO 空、满的标志，是异步 FIFO 设计成败的关键。本文着重介绍一种新颖的地址比较方法，而如异步 FIFO 经常使用的“格雷码”以及读写逻辑的实现，在本节中不再赘述。

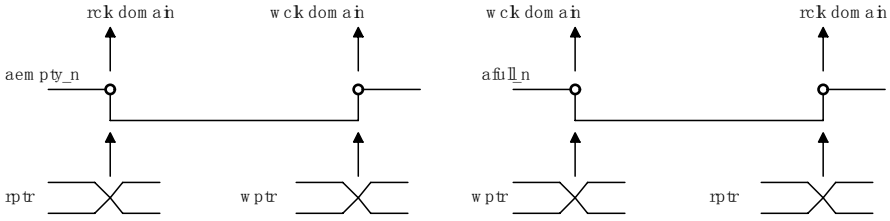


5-1 异步 FIFO 结构图

一般的异步FIFO都是采用先将某一个地址与另一个地址同步以后，再进行比较，这种

方法一般效率不高。本节介绍一种异步比较的方法，如下图所示：[4]

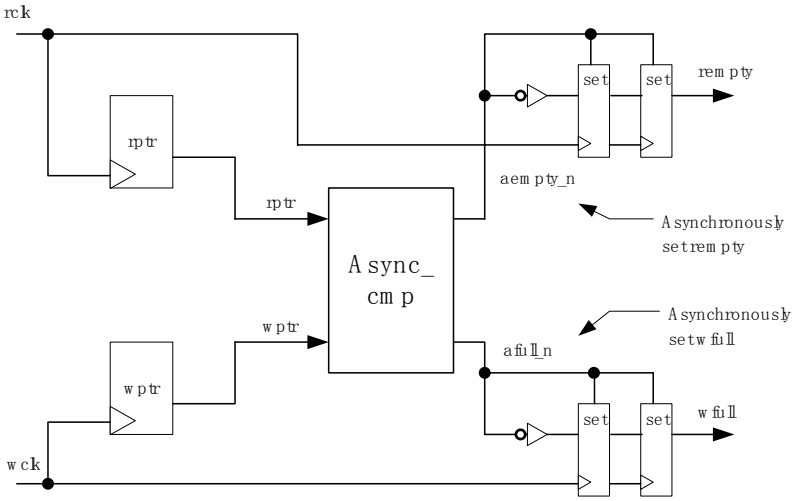
我们注意到，当 rptr 与 wptr 进行异步比较的时候，由于 rptr 的变化(assert)，才产生 aempty_n(FIFO 空标志),即 aempty_n 的下降沿是与 rptr 同属于一个时钟域的，同理，由于 wptr 的变化(assert)，使 aempty_n 无效(de-assert),即 aempty_n 的上升沿是与 wptr 同属于一个时钟域的。



5-2 异步比较的时钟域

现在我们就可以利用上述理论基础，来实现从 aempty_n 到 rempty 的过渡。其中，rempty 是属于 rclk 时钟域的。由于 aempty_n 的下降沿是属于 rclk 时钟域的，所以我们可以用它来作为 rempty 的复位信号，而 aempty_n 的上升沿是属于 wclk 时钟域的，所以我们用双锁存器法将其过渡到 rclk 时钟域，最后得到的 rempty 信号就属于 rclk 时钟域。同理可以得到 wfull 信号。

下图是异步比较器的具体实现方法：



5-3 异步比较实现方案

如上图所示，异步比较法的关键是用异步比较的结果信号的下降沿作为最终比较结果的复位信号，而其上升沿则用传统的双锁存器法进行同步。最终得到的信号的上升沿与下降沿都是属于同一个时钟域。与传统的先将地址信号同步然后进行同步比较相比，异步比较法效率更高，实现也更简单。

6. 总结

本文着重介绍了异步时序产生的问题及原因和基本的解决方法，同时还介绍了两种同步的策略。现将异步时序设计的基本思想总结如下：

- 由于寄存器存在亚稳态问题，所以在进行异步传输的时候，应注意对传输数据以及控制信号进行同步处理。
- 在模块划分时，尽量将各个模块划分为同步时序模块，而用单一的同步模块(synchronizer)来完成跨时钟域的同步，以便使设计尽量简单。
- 在同步模块(synchronizer)中应注意信号的命名，一般如该信号属于 xclk 时钟域，则在信号名后标明_xclk，以便有效区分不同时钟域的信号。
- 根据设计的特点选择同步的策略。本文介绍了几种同步的策略，各有特点且实现难易不同，读者在实际的应用中，应注意根据自身设计的特点选择适当的同步策略。如只有少数信号跨时钟域，一般用双锁存器法即可；而如果有大量数据需要高速跨时钟域传输，一般采用异步 FIFO；而对于快时钟域向慢时钟域过渡的情况，可以考虑使用结绳法。

7. 参考文献

1. Ryan Donohue, Synchronization_pres, Rdonohue@yahoo.com
2. Michael Crews and Yong Yuenyongsgool, Practical design for transferring signals between clock domains, www.edn.com
3. Clifford E. Cummings，Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs, SNUG San Jose 2001.
4. Clifford E. Cummings，Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons, SNUG San Jose 2002.

本文基于与美国 combrio 公司合作的“同步光纤网络(SONET)数据报文监控/分类系统”的国际合作项目。

王夏泉，男，硕士研究生，主要研究方向：宽带网络通信技术、数字集成电路设计与验证技术。武汉,华中科技大学电子与信息工程系(430074).

E-mail: net_xqwang@263.net