

如何用 FPGA 进行时序分析设计

FPGA，即现场可编程门阵列，它是作为专用集成电路（ASIC）领域中的一种半定制电路而出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。对于时序如何用 FPGA 来分析与设计，本文将详细介绍。

基本的电子系统如图 1 所示，一般自己的设计都需要时序分析，如图 1 所示的 Design，上部分为时序组合逻辑，下部分只有组合逻辑。而对其进行时序分析时，一般都以时钟为参考的，因此一般主要分析上半部分。在进行时序分析之前，需要了解时序分析的一些基本概念，如时钟抖动、时钟偏斜（ T_{skew} ）、建立时间（ T_{su} ）、保持时间（ T_h ）等。时序分析也就是分析每一个触发器（寄存器）是否满足建立时间/保持时间，而时序的设计的实质就是满足每一个触发器的建立时间/保持时间的要求。

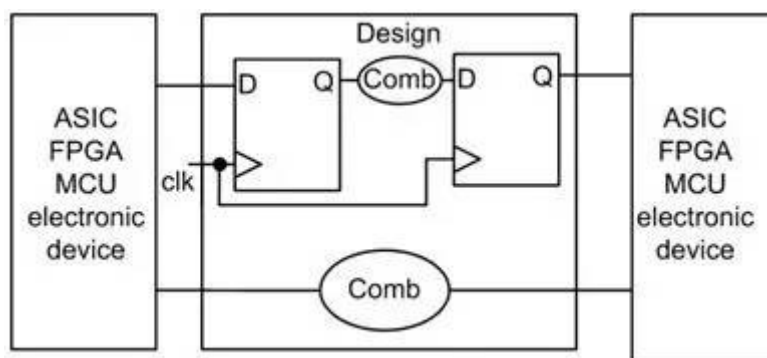


图 1 基本的电子系统

一、时钟抖动和时钟偏斜

1、时钟抖动

时钟信号边沿变化的不确定时间称为时钟抖动，如图 2 所示。一般情况下的时序分析是不考虑时钟抖动，如果考虑时钟抖动，则建立时间应该是 $T_{su} + T_1$ ，保持时间应该是 $T_h + T_2$ 。

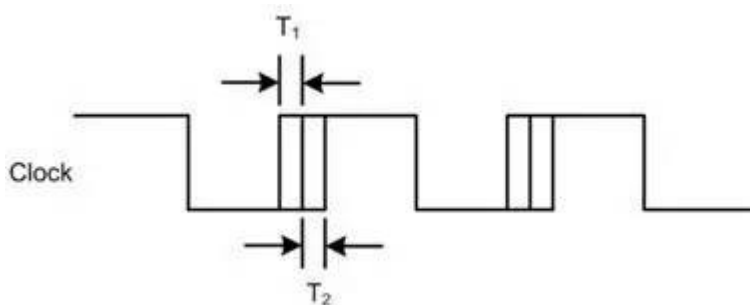


图 2 时钟抖动时序图

2、时钟偏斜

时序偏斜分析图如图 3 所示。时钟的分析起点是源寄存器 (Reg1)，终点是目标寄存器 (Reg2)。时钟在图中的结构中传输也会有延迟，时钟信号从时钟源传输到源寄存器的延时为 T_{c2s} ，传输到目标寄存器的延时为 T_{c2d} 。时钟网络的延时为 T_{c2s} 与 T_{c2d} 之差，即 $T_{skew} = T_{c2d} - T_{c2s}$ 。

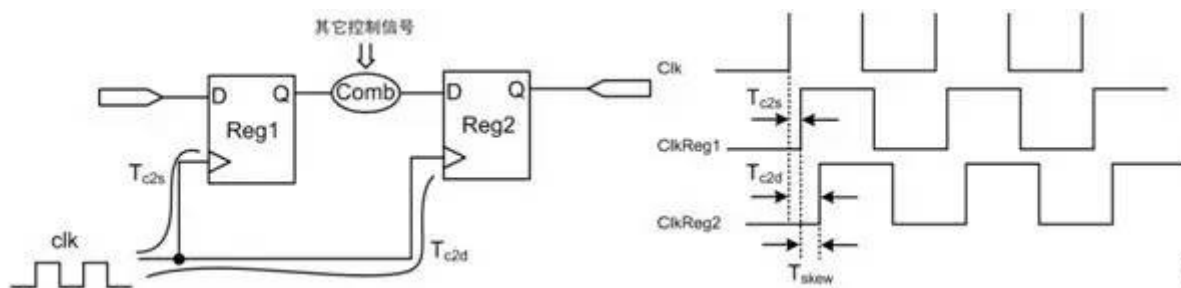


图 3 时钟偏斜时序图

二、建立时间和保持时间

建立时间 (Setup Time) 常用 T_{su} 表示，指的是在触发器的时钟信号上升沿到来以前，数据和使能信号稳定不变的时间，如果建立时间不够，数据将不能在这个时钟上升沿被打入触发器，使能信号无效，也就是说在这个时钟周期对数据的操作时无效的；保持时间 (Hold Time) 常用 T_h 表示，指的是在触发器的时钟信号上升沿到来以后，数据和使能信号稳定不变的时间，如果保持时间不够，数据同样不能被打入触发器，对数据的操作同样是无效的，使能信号无效。数据要能稳定传输，就必须满足建立时间和保持时间的关系，图 4 标识了它们间的关系。

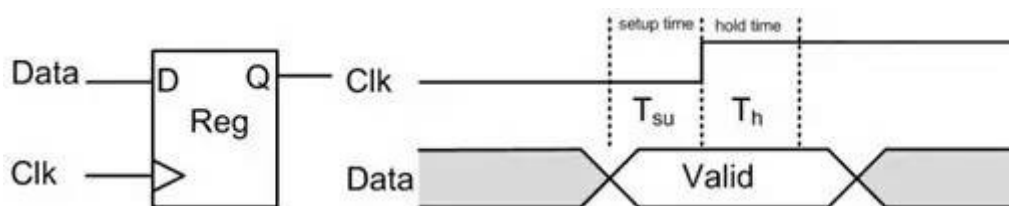


图 4 建立时间/操持时间的概念

三、发送沿和捕获沿

(1) 发送沿 (Launch Edge)：前级寄存器发送数据对应的时钟沿，是时序分析的起点；

(2) 捕获沿 (Latch Edge)：后记寄存器捕获数据对应的时钟沿，是时序分析的终点。相对于 launch Edge 通常为一个时钟周期，但不绝对，如多周期。

“信号跳变抵达窗口”：对 latch 寄存器来说，从 previous 时钟对应的 Hold Time 开始，到 current 时钟对应的 Setup Time 结束。

“信号电平采样窗口”：对 latch 寄存器来说，从 current 时钟对应的 Setup Time 开始，到 current 时钟对应的 Hold Time 结束。

launch 寄存器必须保证驱动的信号跳变到达 latch 寄存器的时刻处于“信号跳变抵达窗口”内，才能保证不破坏 latch 寄存器的“信号电平采样窗口”。

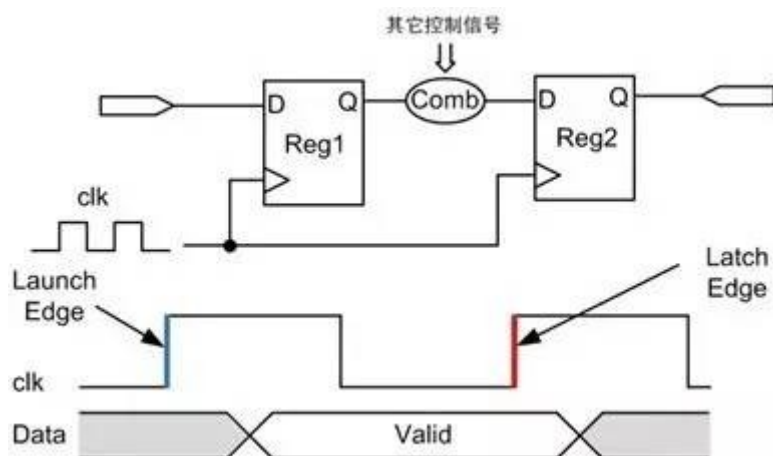


图 5 Launch Edge 和 Latch Edge

四、数据和时钟的时序分析

如图 6 所示，为分析建立时间/保持时间的基本电路图。 T_{clk1} 为 Reg1 的时钟延时， T_{clk2} 为 Reg2 的时钟延时， T_{co} 为 Reg1 固有延时， T_{data} 为 Reg1 的到 Reg2 之间组合逻辑的延时， T_{su} 为 Reg2 的建立时间， T_h 为 Reg2 的保持时间，设时钟 clk 周期为 T ，这里分析数据的建立时间和保持时间。

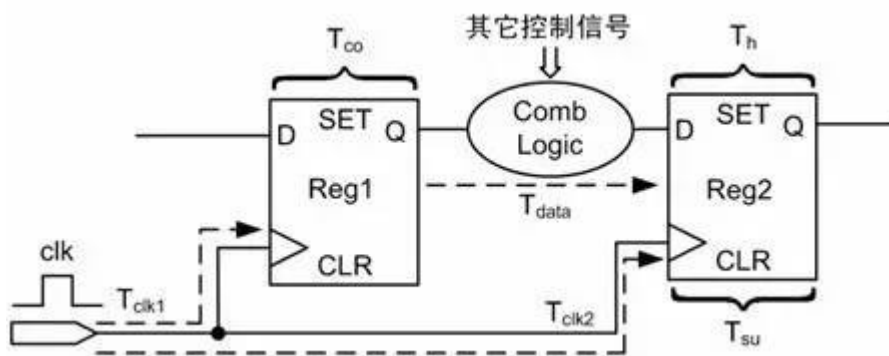


图 6 基本电路图

1、建立时间的分析

如图 7 所示，建立时间的分析是以第一个 launch Edge 为基准，在 Latch Edge 查看结果。建立时

嵌入式资源免费下载: <http://www.kontronn.com>

间的裕量 (T 为时钟周期) :

$$\text{Setup Stack} = (T + T_{\text{clk2}}) - T_{\text{su}} - (T_{\text{clk1}} + T_{\text{co}} + T_{\text{data}})$$

假设 $\Delta T = T_{\text{clk2}} - T_{\text{clk1}}$, 则:

$$\text{Setup Stack} = (T + \Delta T) - T_{\text{su}} - (T_{\text{co}} + T_{\text{data}})$$

可见 $\Delta T < 0$ 影响建立时间, 使建立时间的要求更加苛刻。因此对于 ΔT 尽量避免, 采用同步单时钟, 并且尽量采用全局的时钟信号, 这样 ΔT 几乎为 0, ΔT 的影响几乎不存在, 可以忽略不计。

如果建立时间的裕量 Setup Stack 小于 0, 则不满足建立时间, 也就会产生不稳定态, 并通过寄存器传输下去。

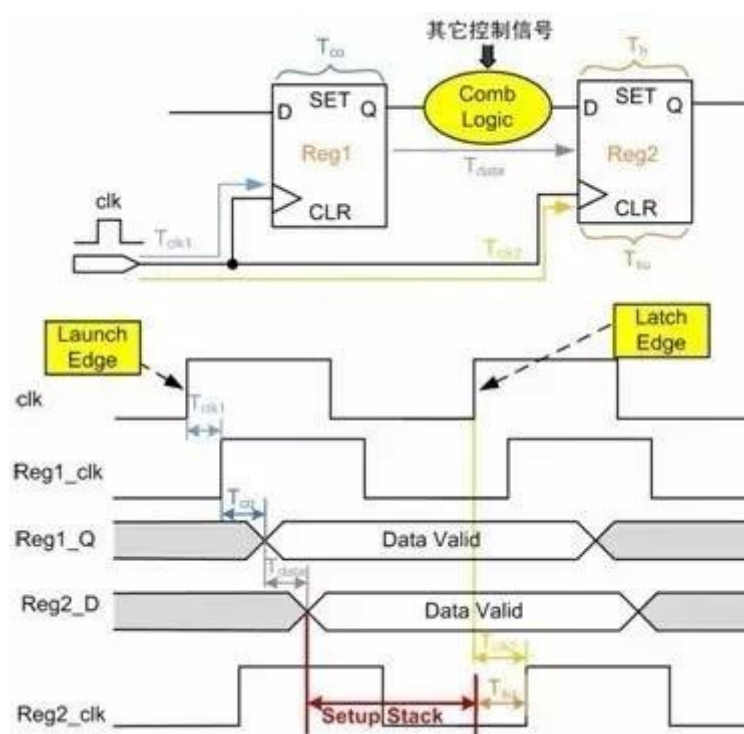


图 7 建立时间时序分析图

2、保持时间的分析

如图 8 所示, 保持时间的分析是以第二个 launch Edge 为基准, 在 Latch Edge 查看结果。保持时间的裕量:

$$\text{Hold Stack} = (T_{\text{clk1}} + T_{\text{co}} + T_{\text{data}}) - T_{\text{clk2}} - T_{\text{h}}$$

假设 $\Delta T = T_{clk2} - T_{clk1}$, 则:

$$\text{Hold Stack} = (T_{co} + T_{data}) - \Delta T - T_h$$

可见 $\Delta T > 0$ 影响保持时间, 使保持时间的要求更加苛刻。因此对于 ΔT 尽量避免, 采用同步单时钟, 并且尽量采用全局的时钟信号, 这样 ΔT 几乎为 0, ΔT 的影响几乎不存在, 可以忽略不计。

如果保持时间的裕量 Hold Stack 小于 0, 则不满足保持时间, 也就会产生不稳定态, 并通过寄存器传输下去。

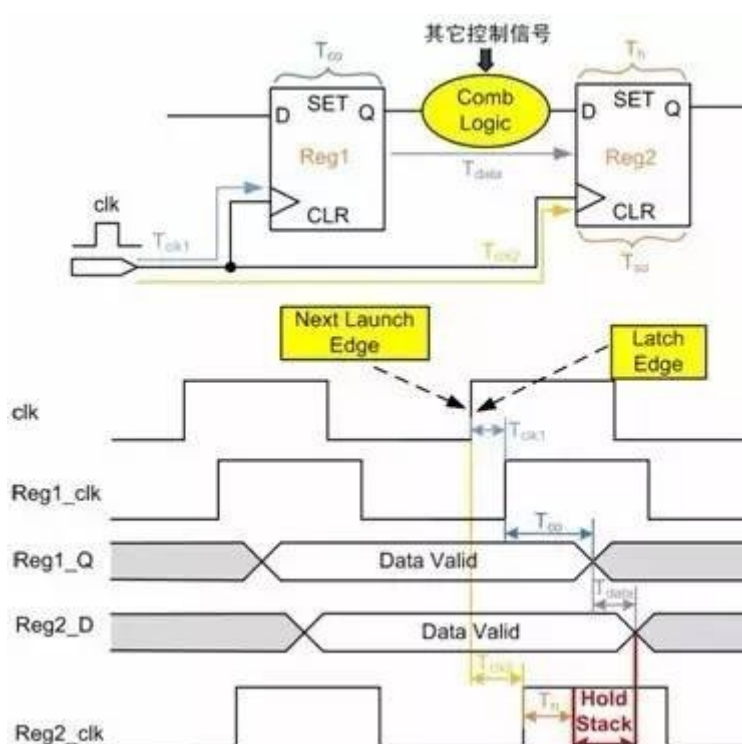


图 8 保持时间时序分析图

五、DT6000S 项目实例

DT6000S 项目上有 4 路光以太网接口连接到 FPGA, 由 FPGA 进行实现 MAC 层和解码 IEC61850 的 SV 和 GOOSE。以太网 PHY 通过 MII 接口和 FPGA, 因此 FPGA 与外部的接口有 4 路 MII 接口。项目初期是实现 1 路光以太网接口, 并且验证功能正确之后, 但是后期变成的 4 路光以太网时, 总会存在 1 路光以太网通信不正常。经过分析得到是 FPGA 通 MII 接口和 PHY 的时序不满足。如图 9 所示为 MII 接口的时序图, 时序不满足分为 TX_CLK 和 RX_CLK。

其一是 PHY 输出的 TX_CLK 和 FPGA 依据 TX_CLK 产生的 TXD[3:0]&TX_EN 延时大, 主要延时为内部逻辑的延时, PCB 延时小并且一致, 导致 PHY 的 TX_CLK 的建立时间不满足, 从而导致发送数据错误。

其二是 PHY 输出的 RX_CLK 和 RXD[3:0]&RX_DV&RX_ER 到 FPGA 内部同步触发器的延时之差太大, 导致 FPGA 内部同步触发器的 RX_CLK 的建立时间不满足, 从而导致接收数据错误。

因此 FPGA 在综合时需要添加约束, 使之时序满足要求, 约束的条件为 TXD[3:0]和 TX_EN 的输出延时要少。RX_CLK 和 RXD[3:0]&RX_DV&RX_ER 路径延时之差要小。添加约束之后, 4 路 MII 接口的光以太网数据通信就正常了。

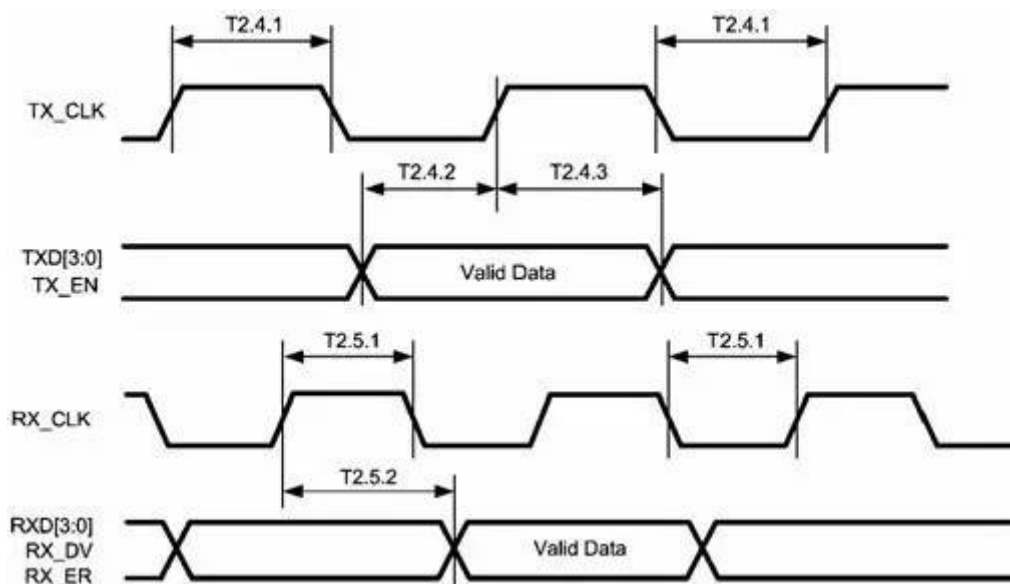


图 9 MII 时序图

本文阐述了时序分析基础, 说明概念的同时进行了时序分析, 通过时序分析理解建立时间和保持时间。