

通用异步串行接口的 VHDL 实用化设计

作者：杨晓斌 赵花荣 赵明生 转贴自：微计算机信息

摘 要：通用异步串行接口 (Universal Asynchronous Receiver Transmitter, UART) 在通信、控制等领域得到了广泛应用。根据 UART 接口特点和应用需求，以提高 VHDL 设计的稳定性和降低功耗为目标，本文讨论了 UART 接口中时钟域划分、时钟分频、亚稳态、同步 FIFO 设计等问题和解决方案。

关键词：通用异步串行接口 VHDL 亚稳态 现场可编程逻辑阵列

1 引言

FPGA 从实现粘合逻辑逐步发展成为设计平台的核心，在电子、通信以及航空航天等领域得到了广泛应用。本人最近实现的中频软件无线电硬件平台，就以 FPGA 为核心，实现上变频、下变频等中频数字信号处理，并且构成 A/D/A、DSP 和 ARM 模块之间的通信中心。这种以 FPGA 为核心的架构使得硬件平台结构灵活，具有可重构性，为软件无线电的各种算法分配方案提供了有力支撑。

除了和 TMS320C6416 之间的数据流采用 EMIF 接口外，FPGA 的其它接口均采用 UART。为了软件开发和移植的便利，UART 设计要做到兼容 ST16C550 的功能。稳定可靠则是作为软件无线电硬件平台关键接口的基本要求。考虑到嵌入式系统的特点，在设计中应尽量降低功耗。本文围绕这些目标，介绍了在 UART 实用化设计中所遇到的一些重要问题、解决方案以及最终结果。

2 UART 及 ST16C550 概述

UART 是广泛使用的串行数据传输协议，它在收发分离的串行链路上进行全双工异步通信。发送过程接收来自数据总线上的并行数据，按照低位序方式并串转换，然后根据控制寄存器的设置生成串行数据流；相应的，接收过程把串行数据流转换成并行数据，产生中断以及状态信息，并对数据传输过程中的异常进行处理。ST16C550 是广泛使用的一款 UART 接口芯片，是 NS16C550 的改进版本。它收发均带有 16 字节的 FIFO，可以通过设定波特率设置寄存器来进行收发时钟的分

频控制，传输速率从 50bps 到 1.5Mbps。具体内容可参见数据手册[1]。

3 实用化设计的主要问题和解决方案

3.1 框架设计

根据 UART 的功能和数据流特点，系统划分为 5 个模块：时钟生成模块，完成时钟分频和时钟分配；界面模块，完成 UART 其它模块和数据总线的交互；发送模块，缓冲接收到的数据并按照设置生成串行信号；接收模块，按照设置将接收信号串并变换并将数据送到 FIFO 中；Modem 模块完成与 Modem 信息交互和控制，功能相对简单独立。图 1 为 UART 顶层模块示意图。

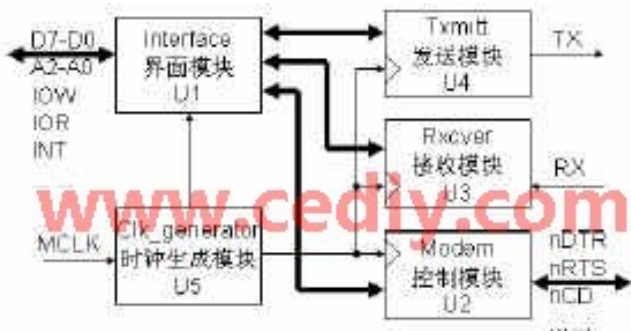


图 1：UART 顶层模块图

3.2 时钟域的划分

在同步电路设计中，减小时钟数量可简化设计，提高系统的稳定性。不相关的时钟域之间的数据传递不可避免的存在亚稳态问题，带来稳定性能的下降。时钟速率与功耗呈线性关系，当工艺一定时，低功耗设计需要我们降低时钟频率和信号翻转次数。下面从这些设计策略和通信效率来分析不同时钟域划分方案。

方案一：低速时钟方案。首先根据控制寄存器的设置对外部提供的时钟进行分频，生成全局唯一的时钟。这种方案的优点是系统实现简单、面积最小、功耗最低。缺点也很明显，CPU 时钟远远高于芯片的工作时钟，与 UART 传递数据时将占用 CPU 过多的时间。

方案二：高速时钟方案。系统直接采用外部提供的时钟为唯一时钟，根据控制寄存器的设置生成收发模块的同步时钟使能信号，来达到分频的目的。这种方案全局只有一个时钟，设计简单。唯一缺点是功耗较大。

在实际设计中，综合考虑效率和功耗的要求，采用了两个关联时钟的方案。与 CPU 的接口界面直接采用外部提供的最高时钟信号，而其它模块采用由波特率设置寄存器控制的分频时钟。这样在系统中存在两个关联的时钟域，设计时需要两个时钟域边界的逻辑进行分析和处理。

3.3 时钟分频

同步数字电路设计中,时钟是整个电路中最重要信号。时钟信号上的毛刺会引起系统的逻辑混乱，大规模的数字芯片还对时钟歪斜(Clock skew)和负载提出了要求。为了适应这些需求，FPGA 内部一般设有数量不等的全局时钟网络。

使用同步计数器或状态机进行时钟分频是一种较好的方案。在设计中计数器或状态机应直接产生分频时钟信号，而不应该对计数器或状态机进行译码来产生时钟信号，因为译码等组合逻辑可能给时钟带来毛刺，引起系统不稳定。UART 当波特率设置寄存器为 0 或 1 时，时钟信号不需要分频，故分频电路中使用了一个多路选择器。时钟分频电路如图 2 所示。

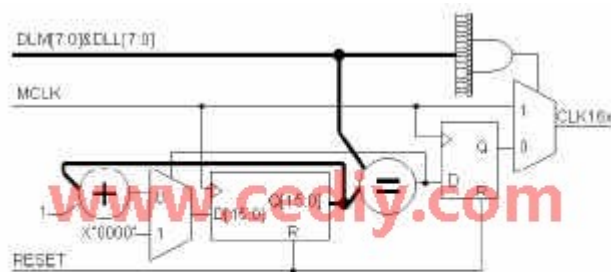


图 2: 时钟分频电路的设计

3.4 异步时钟与亚稳态

UART 使用独立的时钟信号，使得 CPU 与 UART 以及 UART 之间的信号都处在不同的时钟域。为了减少时序上的冲突，跨时钟域的数据传递首先需要同步处理。但由于时钟频率和相位的差异，就不可避免存在亚稳态问题[2]。所谓亚稳态，是指触发器/锁存器的输入信号时序不能满足设置时间和保持时间的要求，将有可能使得触发器/锁存器的输出没有正确的锁定到逻辑 0 或逻辑 1，处在一个未知的状态，如滞留在中间状态，或者震荡。这里以 SN74ABT7819 的参数为例来分

析亚稳态、说明提高系统稳定性的方法[3]。

的负指数分布。 T_0 表示器件进入亚稳态可能性的孔径时间。SN74ABT7819 在室温、5V 电压时, τ 亚稳态滞留时间是随机的, 服从参量为 $\tau=0.30\text{ns}$, $T_0=7\text{ps}$, 输入信号建立时间 $TSU1=5\text{ns}$, 传播延时 $TPD1=9\text{ns}$, 芯片内部触发器建立时间以及传播延时约 $TPD2+TSU2=1.3\text{ns}$ 。设定触发器异步输入信号边沿频率 $FD=10\text{MHz}$, 芯片工作频率即触发器时钟 $FC=50\text{MHz}$ 。则每个接收外界输入信号的触发器平均失效时间 MTBF (the Mean Time Between Failures) 为:

$$MTBF_1 = \frac{e^{\frac{T_{sample}-T_{PD1}-T_{SU1}}{\tau}}}{T_0 F_C F_i} = \frac{e^{\frac{20\text{ns}-9\text{ns}-5\text{ns}}{0.30\text{ns}}}}{7\text{ps} \cdot 10\text{MHz} \cdot 50\text{MHz}} = 1.4 \times 10^5 \text{s} = 1.6\text{days}$$

系统中有多路并行信号跨越不同的时钟域, 这样系统总的 MTBF 会很小, 将以小时或分钟计。可见单触发器同步电路不能满足稳定性的要求。为了减小亚稳态的影响, 将亚稳态控制在时钟域边界, 可以采用传统的握手通讯方式[4]或者双触发器同步电路, 前一方案在通信速率较低时有效, 后者则在实践中广泛使用, 这里仅对后者进一步说明。三级触发器同步电路一般是不必要的。

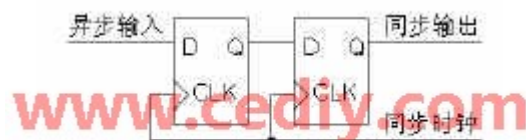


图 3: 双触发器同步电路

在图 3 中, 异步输入经过两级触发器同步生成同步输出。即使第一个触发器输出存在亚稳态, 经过一个时钟周期后, 第二个触发器输出仍处于亚稳态的概率非常小, 此电路的平均失效时间 MTBF 已经是一个无限长的时间:

$$MTBF_2 = MTBF_1 \times e^{\frac{T_{sample}-(T_{PD2}+T_{SU2})}{\tau}} = 1.39e+5 \times e^{\frac{20\text{ns}-1.3\text{ns}}{0.30\text{ns}}} = 1.6e+32\text{s}$$

Xilinx 未给出 Virtex II 系列的亚稳态描述参数, 它能工作在更高的时钟频率上, 亚稳态的参数会比 SN74ABT7819 优秀。但以上分析和设计规则依然适用。

3.5 同步 FIFO 的设计

设计中根据收发模块对状态标志要求的不同，分别进行同步 FIFO 的设计以节约逻辑资源。其中，发送模块的 FIFO 只需要全空/全满标志，采用地址相等不相等的比较逻辑和地址绕回指示位来产生。具体过程为：地址随着相应的操作递增，写指针由内存的最后位置返回到初始位置时将地址绕回指示位置 1，读指针返回时则置 0。因此，当读写指针地址相同时，若地址绕回指示位为 0，则读写指针经历了相同次数的循环移动，FIFO 处于空状态(图 4a)；若地址绕回指示位为 1，则写指针比读指针多循环一次，FIFO 处于满状态(图 4b)。



图 4：发送模块 FIFO 空满标志的生成

接收模块需要在 FIFO 中数据量达到一个设定的数值时产生一个中断，由于设定的数值是任意的，这样空/满标记的产生必须使用减法器，消耗的逻辑资源稍大。综合后的逻辑资源使用情况也说明了这点。

3.6 锁存器的使用

使电路复杂化的常见原因之一是设计中存在许多不必要的锁存器，使得电路复杂，工作速度降低，系统可靠性变差。综合时应该仔细检查是否合理使用了锁存器。由于 UART 接口的功能特点，设计中共使用了 4 个锁存器，用来锁存 A(2:0) 和 CSn 信号。当 UART 和 CPU 总线处在同一个芯片中时，这些锁存器可以用寄存器取代。

4 综合结果

整个设计以 VHDL 语言来实现。在 SYNOPSYS Design Compiler 中使用 LSI_10K

库，设定系统工作频率为 25MHz，其他使用预定的选项，综合后最大路径延时为 10.66ns，预期工作频率大于 90MHz。资源使用情况如表 1 所示。

Cell	Reference	Aera
U1	Intface	1020.00
U2	Modem	131.00
U3	Rxcver	2694.00
U4	Txmirt	2412.00
U5	Clk_generator	333.00
Total 5 cells		6590.00

表 1: VHDL 综合的资源使用情况

在 XC2V1000-6 芯片中，以 Synplify 为综合工具，则使用了 188 个寄存器，占用了 1%的逻辑资源。最大路径延迟预期 9.043ns，预期工作频率 110MHz。

5 仿真与验证

设计中对 UART 各模块分别撰写了相应的测试程序，验证了各模块的正确性。然后把这些模块装配在一起，在系统级上再进行了接收、发送和中断功能的门级验证。这种模块化分层次的验证过程在调试中有效地缩小了查找错误的范围，提高了调试效率并保证了代码的健壮性。

6 结束语

稳定性和低功耗是嵌入式通信系统的重要设计目标。实现需要的功能有时并不困难，难的是提高系统稳定性和有效降低功耗。时钟的规划和亚稳态的处理与这两个目标有着非常密切的关系，是实用化设计关注的重点之一。通过稳定性、功耗与资源等方面的综合考虑，该设计在所实现的软件无线电硬件平台上得到了成功应用，达到了实用化设计的目标。

参考文献：

[] EXAR Corp., ST16C550 Datasheet[Z], www.exar.com, 2005

[2] L.Kleeman, A.Cantoni, Metastable behavior in digital systems, IEEE Design and Test of Computers[J], Volume 4, No. 6, 1987

[3] Chris Wellheuser, Metastability Performance of Clocked FIFOs[Z], www.ti.com,

1996

[4] Peter Alfke, 跨越异步时钟边界传输数据的解决方案[Z], www.eetchina.com,

2001

作者简介:

杨晓斌, 男, 硕士研究生, 清华大学电子工程系; 研究方向: 数字电路设计, 软件无线电;

联系方式: 北京海淀区清华大学东主楼 9 区 318 室 (100084);

Email:yangxb02@mails.tsinghua.edu.cn;

赵花荣, 女, 硕士研究生, 清华大学电子工程系; 研究方向: 数字信号处理, 软件无线电; Email:zhaohr03@mails.tsinghua.edu.cn;

赵明生, 男, 博士, 清华大学电子工程系副教授; 研究方向: 软件无线电、神经网络及其在信号处理中的应用; Email:zhaoms@tsinghua.edu.cn。