

一种实现高速异步 FIFO 的 FPGA 方法

黄忠朝, 赵于前

HUANG Zhong-chao, ZHAO Yu-qian

中南大学 信息物理工程学院 生物医学工程系, 长沙 410083

Department of Biomedical Engineering, School of Info-Physics and Geomatics Engineering, Central South University, Changsha 410083, China

E-mail: lipse_huang@163.com

HUANG Zhong-chao, ZHAO Yu-qian. Implementation method of high-speed asynchronous FIFO using FPGA. Computer Engineering and Applications, 2010, 46(3): 13-15.

Abstract: To overcome the metastability and ensure the validation of data transfer, the asynchronous First In First Out (FIFO) modules are often used to buffer data in systems with data transfers crossing clock domains. Because of the existence of the addressing pointers, which often adopt Gray-code counters, and the "full & empty" generation logic in a usual asynchronous FIFO module, the signals passing two modules may suffer large delay. As a result, the working frequency of the whole module is limited. Based on a premise that the "full" state will never occur in a high-frequency system, a method of implementing high speed asynchronous FIFO in FPGA is proposed. The focus on this way is that the "full" flag generation logic and redundant RAM depth are omitted, i.e., only the "empty" flag is generated. So, the design of FIFO is simplified. The results from simulation and synthesis design show that the working speed of the whole module is greatly increased.

Key words: Field-Programmable Gate Array (FPGA); metastability; gray-code; high-speed FIFO

摘要:在跨时钟域传递数据的系统中,常采用异步 FIFO (First In First Out, 先进先出队列) 口来缓冲传输的数据,以克服亚稳态产生的错误,保证数据的正确传输。但由于常规异步 FIFO 模块中的 RAM 存储器读写寻址指针常采用格雷码计数器以及“空满”控制逻辑的存在,将使通过这两个模块的信号通路延时对整个模块的工作频率造成制约。提出了一种在 FPGA 内实现高速异步 FIFO 的方法,该方法针对不可能产生满信号的高频系统,通过省略“满”信号产生模块和多余的存储器位深来简化常规的 FIFO 模块,而只保留“空”信号产生模块。仿真和综合设计结果表明,整个模块的工作频率得到一定提高。

关键词:现场可编程门阵列 (FPGA); 亚稳态; 格雷码; 高速 FIFO

DOI: 10.3778/j.issn.1002-8331.2010.03.004 **文章编号:** 1002-8331(2010)03-0013-03 **文献标识码:** A **中图分类号:** TP752.1

1 引言

设计同步化是当今集成电路设计的重要思想之一,即对所有时钟控制器件都采用同一个时钟来控制。但在实际应用系统中,实现完全同步化的设计是很困难的,一般情形下都需完成数据在多个时钟域间的传递(如高速模块和低速模块之间的数据交换)。此时,如何保持系统的稳定,或者克服跨时钟域间数据传递时的亚稳态,以顺利完成数据的传输就成为一个关键的问题。

解决亚稳态问题的通常做法是对每位信号加同步器或增加握手信号,但这样会增加系统的复杂度且影响传输速度。较好的解决办法是在两个时钟域的交界处设计一个异步 FIFO (First In First Out 先进先出队列)来实现数据流的传输,由发送时钟域将数据写入,接收时钟域将数据读出,在数据传输的同时实现了数据的缓存,是一种经常使用的方^[1-3]。常规 FIFO

口主要由双口 RAM 存储器、“空满”控制逻辑、存储器读写寻址指针组成,但 RAM 存储器读写寻址指针由于常采用格雷码计数器以及“空满”控制逻辑的存在,将使通过这两个模块的信号通路延时对整个模块的工作频率造成制约。

针对这种局限性,先简单地介绍了亚稳态的工作原理,并详细地分析了常规 FIFO 口对模块工作频率制约的主要原因,然后结合 FPGA 优秀的逻辑控制功能,及低功耗、高可靠性、可重构性、开发周期短和开发费用低廉等特性,提出了一种在 FPGA 内实现高速异步 FIFO 的方法。该方法主要针对不可能产生满信号的高频系统,通过省略“满”信号产生模块和多余的存储器位深来简化常规的 FIFO 模块,即只产生“空”信号,从而使整个模块的工作频率大大提高,适合于高频数据采集系统,如视频图像采集预处理系统。

基金项目:高等学校博士学科点专项科研基金(the Research Fund for the Doctoral Program of Higher Education under Grant No.200805530185)。

作者简介:黄忠朝(1976-),男,博士学位,讲师,主要研究领域:医学信号与图像处理,视频采集与处理系统设计;赵于前(1973-),男,博士,副教授,主要研究方向:计算机应用技术、图像与信号处理。

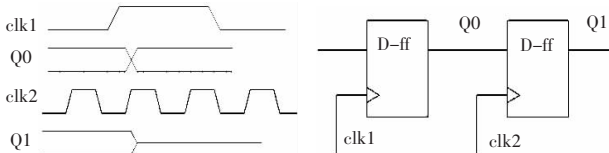
收稿日期:2009-10-12 **修回日期:**2009-12-11

2 亚稳态发生原理

亚稳态是指触发器无法在某个规定时间段内达到一个可确认的状态。当一个触发器进入亚稳态时,既无法预测该单元的输出电平,也无法预测何时输出才能稳定在某个正确的电平上。在这个稳定期间,触发器输出一些中间级电平,或者可能处于振荡状态,并且这种无用的输出电平可以沿信号通道上的各个触发器级联式传播下去^[4-5]。

简单来讲,在数字系统中,每一时刻系统的每一个寄存器都会有一个稳定的状态 1 或者 0。此时,寄存器的输出电压在下级门电路的噪声容限范围内。而当系统的某一个寄存器当前输出的电压值既可能被系统认为是 1 也可能被认为是 0 的时候,即输出的电压处在下级门电路的噪声容限范围外的时候,这种状态即为亚稳态。

亚稳态通常发生在跨时钟域传递数据的系统中。图 1(a) 是一个跨时钟域传递数据的情况,前级寄存器工作于时钟 CLK1,后级寄存器工作于 CLK2。当前一级的数据正在改变的时候,下一级电路进行读数据即可产生亚稳态。图 1(b)即为(a)中寄存器直接传递数据时亚稳态发生的时序原理。由图 1(b)可以看出,当 CLK1 上升沿来临后不久便是 CLK2 上升沿时,此时前级寄存器输出信号 Q0 正在变化,下级寄存器对其读出的数据 Q1 既可能是 1,也可能是 0,即亚稳态。



(a)分别工作于两个不同时钟的寄存器直接传递数据图 (b)寄存器工作时亚稳态发生的时序原理

图 1 亚稳态发生原理图

3 常规 FIFO 模块分析

常规的 FIFO 口包括:双口 RAM 存储器,“空满”控制逻辑,存储器读写寻址指针等。图 2 是一个位宽为 8、位深为 4 的常规 FIFO,图中的 chk_full 和 chk_empty 模块分别为“满”控制逻辑和“空”控制逻辑,其输入分别为被同步的“准当前”读写寻址指针值,输出为“空满”标志位。W_ptr 和 R_ptr 模块分别为写和读寻址指针,即循环格雷码计数器,在此采用格雷码计数是为了减少由于亚稳态产生的错误位(最多一位),以方便后续的处理。G2B 模块将格雷码转化为二进制码,并输出低两位进行寻址。Sync0/ Sync 1 是同步模块,将异步的地址信号同步化,以方便“空满”控制模块进行处理,其具体的寻址原理见参考文献^[4-5],在此不再赘述。

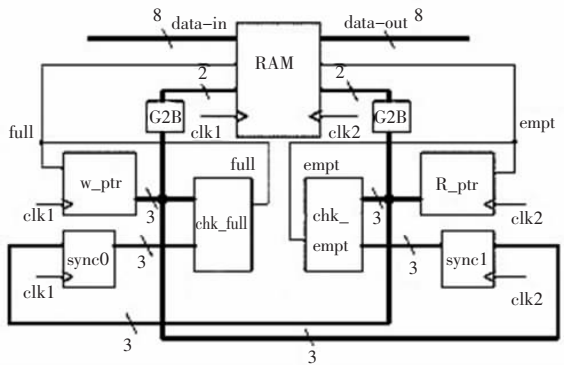


图 2 常规 FIFO 模块图

图 3 是图 2 中常规 FIFO 模块在综合软件 Synplify Pro 中综合的关键路径。综合选用的器件是 Altera Cyclone II, part: EP2C5, package: QC208, speed: -6。由图可以看出该关键路径即为寻址指针寻址的路径,若综合时对特定的路径加上约束,可以改变关键路径。但是由于有“空满”控制逻辑,格雷码计数器内部逻辑和寻址逻辑的存在,关键路径总会在这几个模块之间产生,因而其工作频率在一定的 FPGA 芯片内不可能达到很高。

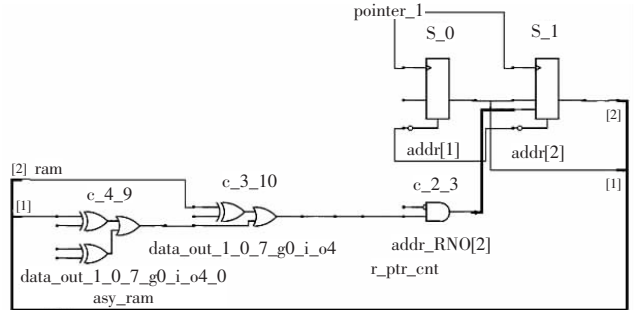


图 3 常规 FIFO 的综合结果的关键路径

为说明这一点,对图 3 中关键路径进行仿真,得出如下时序分析报告:

```
##### START OF TIMING REPORT #####
# Timing Report written on Wed Jun 17 13:54:05 2009
Top view: fifo_s
.....
Performance Summary
*****
Worst slack in design: 1.590
```

Starting Clock	Requested frequency/MHz	Estimated Frequency/MHz	Requested Period/ns	Estimated Period/ns	Slack/ns
clk1	27.0	339.8	37.037	2.943	34.094
clk2	200.0	293.3	5.000	3.410	1.590

该时序分析结果是将以上综合结果中 CLK1 约束为 27 MHz, 将 CLK2 约束在 200 MHz 的工作频率后的分析结果。由结果可以看出,最坏情况的时钟裕度为 1.590 ns,由此可以推测,该模块的最高工作频率不可能高于 300 MHz。

4 高速 FIFO 设计与仿真分析

为解决常规对整个模块工作频率的制约问题,针对不可能产生满信号的高频系统,利用 FPGA 功耗低、可在线编程、可重构性和灵活的逻辑编程控制功能特征,提出了一种在 FPGA 内实现高速异步 FIFO 的方法,其主要思想是通过省略“满”信号产生模块和多余的存储位来简化常规的 FIFO 模块,而只保留“空”信号产生模块,从而提高整个模块的工作频率。

图 4 是所设计的 FPGA 高速 FIFO 模块图。图中的 RAM 即为 8 个(或 n 个)并联工作于 CLK2 的寄存器,其输入数据总线端传输的数据工作于 CLK1。其下的两个 T 触发器,即一个异或门和一个 D 触发器组成了空信号产生逻辑。

当 rst 复位信号端出现高电平时,T'触发器和 T 触发器的输出全部复位 0,此时异步异或逻辑输出端 async_empty_1 输出为 0(空信号有效,不可读状态),T 触发器的控制输入为 0,因而在以后的 CLK2 上升沿上(内部高速系统对 FIFO 口的无效读操作)T 触发器不会翻转。直到 CLK1 的上升沿(外部低速系统对 FIFO 口的写操作)到来时,T'触发器翻转,此时异步异或

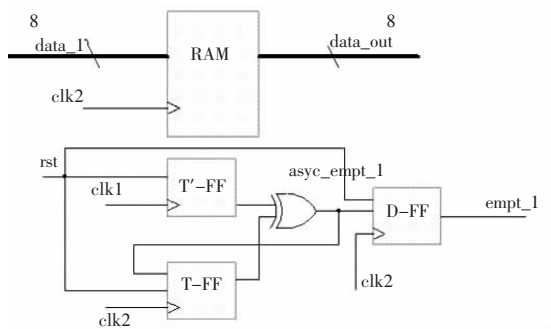


图4 高速 FIFO 模块图

逻辑门输出变为 1 (空信号无效,即 FIFO 非空,可读状态),在下一个 CLK2 上升沿(内部高速系统对 FIFO 口的有效读操作)到来时,T 触发器翻转,asyc_empt_1 恢复低电平(空信号有效),等待下一个 CLK1 (外部低速系统对 FIFO 的写操作)的到来。表 1 是“空”信号产生的真值表。

表 1 高速 FIFO 空信号产生真值表

T'(clk1)	T'(clk2)	asyc_empt_1	状态
0	0	0	空
0	1	1	满
1	1	0	空
1	0	1	满

图 4 中存储器的数据端工作在外部时钟频率 CLK1, 而寄存器工作在内部高速时钟频率 CLK2, 因此, 8 位并行寄存器的输出端很有可能会输出错误数据, 但是当空信号被拉高时(内部高速系统对 FIFO 的有效读操作), 即为外部系统写操作刚结束时, 此时寄存器输入数据总线上的数据一定是稳定的, 因而寄存器输出端的数据一定是正确的。

图 5 是高速 FIFO 模块的仿真结果。由图可以看出每出现一次 CLK1 上升沿(每对 FIFO 进行一次写操作), 异步空信号 asy_empt_1 便出现一次正脉冲(空信号无效, 内部高速系统读有效信号)。而且, 由仿真结果中可以看到, 产生的异步空信号 asy_empt_1 时宽时窄, 这是因为该信号由异步逻辑产生, 其信号宽度取决于两个时钟(CLK1 和 CLK2)上升沿相距的距离: 距离最宽为一个 CLK2 周期, 最窄为图 4 中 T 触发器到异或门输出的延时。因此, 当信号宽度小于后级输入门的建立时间时便很有可能会引起后级的读写错误。为此在异步空信号后端加一级 D 触发器, 将异步的空信号输入到 D 触发器锁存起来, 保存一个 CLK2 时钟周期以后再拉低, 便使得空信号的信号宽度始终为一个 CLK2 时钟周期, 从而保证后级的读写正确, 如图 5 中 empt_1 信号的仿真结果所示。

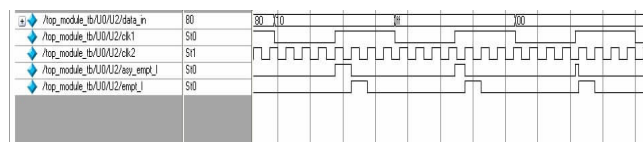


图5 高速 FIFO 仿真波形图

但是这样做的代价是提高了对高速 FIFO 中“高速”含义的要求。由以上分析可以知道, 不带同步 D 触发器的高速 FIFO 能正常工作的要求是内部高速系统的工作频率是外部的 2 倍以上, 这在一般的高速数字系统(如视频处理系统)很容易满足。在加上同步 D 触发器以后, 为使该高速 FIFO 模块正常工作, 其内部高速系统的时钟频率必须是外部的 3 倍以上, 这个

要求在高速处理的系统中也是不难达到的。

当后级系统与高速 FIFO 集成于同一 FPGA 内的情况时, 则必须在 FIFO 输出端用输出有效使能信号(即空信号)来标示输出信号的有效。若后级系统在 FPGA 外部时, 则可以将空信号连接到 FPGA 的输出三态口的控制端, 将输出数据总线连接到外部输出端口, 从而使得当信号输出无效时, FIFO 在 FPGA 的输出引脚状态为三态, 当 FIFO 输出引脚出现数据时, 即为有效的输出信号。

图 6 是该高速 FIFO 模块综合结果的关键路径图, 从图中可以看出, 该关键路径由高速 FIFO 内部产生异步空信号的异步异或逻辑产生, 即整个关键路径只经历一个异或逻辑门。因此, 该关键路径延时不会很长, 系统的工作频率可以达到很高。

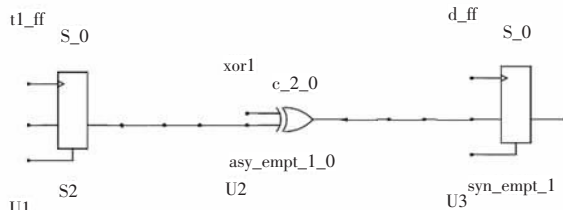


图6 高速 FIFO 的时序综合关键路径图

以下是将 CLK1 限制在 27 MHz, 将 CLK2 限制在 900 MHz 时的 FIFO 时序仿真报告。从报告可以看出, 此时的时钟裕度为 0.076 ns, 大于零, 因而系统可以正常工作。实际上, 只要 CLK1 满足小于 CLK2 的 1/3, 整个系统便可以正常工作。此时系统的时钟是常规 FIFO 的三倍, 故称之为高速 FIFO 口。

```

##### START OF TIMING REPORT #####
# Timing Report written on Sun Jun 07 01:55:54 2009
Top view:                fifo_s
Requested Frequency:     27.0 MHz
Wire load mode:         top
Paths requested:        5
.....
Worst slack in design:  0.076
    
```

Starting Clock	Requested frequency/MHz	Estimated Frequency/MHz	Requested Period/ns	Estimated Period/ns	Slack/ns
clk1	27.0	827.8	37.037	1.208	35.829
clk2	900.0	966.3	1.111	1.035	0.076

5 结论

提出了一种高速异步 FIFO 的 FPGA 设计方法, 并用于高速数据处理系统中。通过利用高速系统(内部时钟频率是外部系统 3 倍以上)中 FIFO 不可能出现“写满”状态的特点, 省略了常规 FIFO 口中的“满”信号控制位。并用一排并联的寄存器代替双口 RAM 存储器, 省略了存储器寻址指针。同时分别采用一个 T 触发器、一个 T' 触发器和一个异或门产生“空”信号, 通过一个 D 触发器将异步的空信号同步, 从而使 FIFO 内寄存器间的逻辑延迟大大减少, 在 FPGA 内成功实现了高速 FIFO。目前, 该方法已用于而所设计的“基于 FPGA 图像采集预处理系统”中, 大大提高了整个系统的工作速度, 提高了效率。

参考文献:

[1] 熊红兵, 陈琦. 基于 FPGA 的异步 FIFO 设计与实现[J]. 微计算机信息, 2006, 22(6-2): 216-218.