

# 基于FPGA的异步FIFO设计

张维旭, 贺占庄

(西安微电子技术研究所, 陕西 西安 710065)

**摘要:**在现代集成电路设计中, 一个系统往往包含了多个时钟, 如何在异步时钟间传递数据成为一个很重要的问题。而使用异步FIFO可以有效地解决这个问题。异步FIFO是一种在电子系统中得到广泛应用的器件, 文中介绍了一种基于FPGA的异步FIFO设计方法。使用这种方法可以设计出高速、高可靠的异步FIFO。

**关键词:**FIFO; 亚稳态; 格雷码; FPGA

**中图分类号:**TP311.1

**文献标识码:**A

**文章编号:**1673-629X(2006)07-0168-03

## Asynchronous FIFO Design Based on FPGA

ZHANG Wei-xu, HE Zhan-zhuang

(Xi'an Microelectronic Technology Research Institute, Xi'an 710065, China)

**Abstract:** In modern IC designs, a system always contains several clocks. Transmitting data among asynchronous clocks become an important problem. But it can solve this problem by using asynchronous FIFO. Asynchronous FIFO is a device that was widely used in electronic system. This paper introduces a method of FIFO design based on FPGA. Can design high speed and reliability asynchronous FIFO.

**Key words:** FIFO; metastability; Gray Code; FPGA

### 1 异步FIFO介绍

随着现代芯片设计规模的不断扩大, 集成电路越来越复杂, 一个系统中往往包含多个时钟。如何在异步时钟间传送数据就成为电路设计中一个重要的问题。异步FIFO (First In First Out) 是用来解决这个问题。FIFO 常用于数据的缓存和容纳异步信号的频率或相位的差异, 使用异步FIFO可以在两个不同的时钟系统之间快速准确地传输实时数据。异步FIFO在网络接口、图像处理等方面都得到了广泛的应用。

异步FIFO是一种先进先出电路, 用来存储、缓冲在两个异步时钟之间的数据传输。在异步电路中, 由于时钟之间的周期和相位完全独立, 因此数据的丢失概率不为零。文中将介绍使用FPGA设计高可靠性、高速度的异步FIFO电路的一种方法。

图1给出了FIFO的接口信号和内部模块图。由图1可以看出: 整个系统分为两个独立的时钟域——读时钟域和写时钟域; FIFO的存储介质为一块双口RAM, 可以同时读写操作。在写时钟域部分, 由写地址生成逻辑产生写控制信号和写地址; 读时钟部分由读地址生成逻辑产生

读控制信号和读地址。由写地址和读地址相互比较生成空满标志。

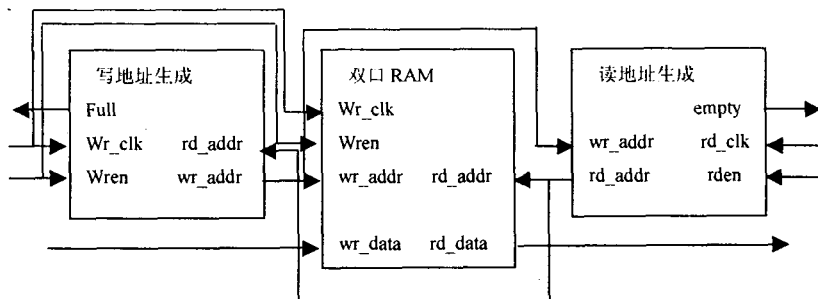


图1 FIFO接口信号和内部模块图

### 2 异步FIFO设计的难点

异步FIFO设计有两个难点: 一是如何根据FIFO的指针信号正确地判断FIFO的空满状态; 二是如何同步异步信号, 避免触发器亚稳态的产生。

#### 2.1 亚稳态问题的解决

在数字电路中, 触发器要满足 setup/hold 的时间要求。当一个信号被寄存器锁存时, 如果信号和时钟之间不能满足这个要求, Q端的值是不确定的, 并且在一个未知的时刻会固定到高电平或低电平。这个过程称为亚稳态。图2所示为异步时钟和亚稳态。

亚稳态必然会发生在异步FIFO中, 因为在异步FIFO中, 电路的外部输入和内部时钟没有任何时间关系, 因此存在 setup/hold 冲突是必然的, 同时在电路内部的两

收稿日期: 2005-10-20

作者简介: 张维旭(1978-), 男, 山东潍坊人, 硕士研究生, 研究方向为SOC; 贺占庄, 研究员, 硕士生导师, 研究方向为计算机控制技术。

个没有关系的时钟域之间传递信号,也会出现 setup/hold 冲突。虽然亚稳态是不可避免的,但是,可以通过下面的方法将其降低到一个能够接受的范围之内<sup>[1]</sup>。

(1)对写地址/读地址用格雷码。同步多个异步输入信号出现亚稳态的概率远远大于同步一个异步信号的概率。对多个触发器的输出所组成的写地址/读地址可能采用格雷码。由于格雷码每次只有一个数据位变化,采用格雷码可以有效地减少亚稳态的产生。

(2)采用触发器来同步异步输入信号,如图3中的两级触发器可以将出现亚稳态的几率降低到一个很小的程度。不过,使用这个方法会增加一级延时。

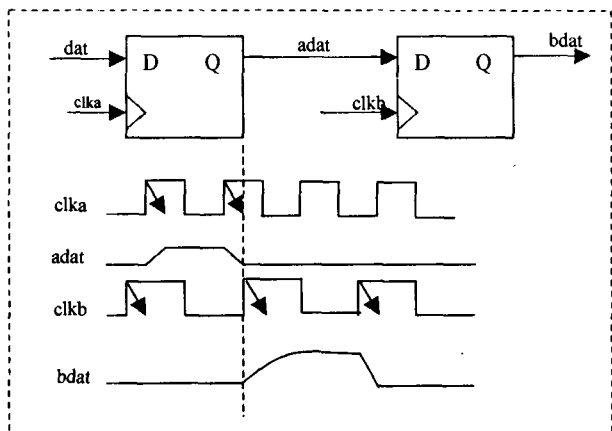


图2 异步时钟和亚稳态

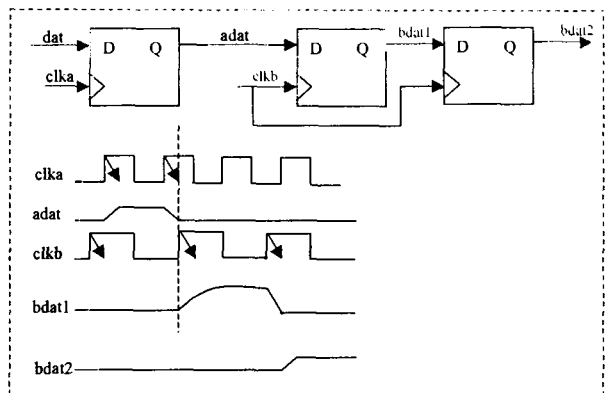


图3 采用触发器后的异步时钟和亚稳态

## 2.2 指针及空满信号的生成

空满标志的产生是整个FIFO的核心部分。这部分设计的好坏,将直接影响FIFO的性能。空满标志产生的原则:写满不溢出,读空不多读。即无论在什么情况下,都不能出现读写指针对同一个存储地址操作的情况<sup>[2]</sup>。

首先来看一下同步FIFO指针移动及空满信号的生成过程。对于同步FIFO读写指针都指向一个内存的起始位置,每进行一次读写操作,相应的指针就增加一次,指向下一个位置。当指针移动到最后的位置时,它又重新回到初始位置。在FIFO非空或非满的状态下,这个过程将随着读写控制信号的变化一直变化下去。如果FIFO处于空状态,再进行一个读出动作会产生向下溢出(under-flow),一个无效的数据被读出;同样地,对于一个写满的

FIFO如果进行一个写入动作,会产生向上溢出(overflow),一个有用的数据会被新写入的数据覆盖。为了避免这种错误的产生,应该对FIFO设置满和空两个信号。对满信号置位表示FIFO已经处于满状态,不能再执行写入动作;对满信号复位表示FIFO非满,还可以执行写入动作;对空信号置位表示FIFO处于空状态,这时没有有效数据可以读出;对空信号复位说明FIFO非空,这时可以进行读操作。

当读指针和写指针相等时,FIFO可能处于满状态或空状态。可以用不同的方法判断FIFO究竟处于什么样的状态,也就是说究竟是写指针从后面追上了读指针还是读指针从后面追上了写指针。文中采用设置一个额外状态位的方法来区分两种状态。读写指针由它的地址位和状态位组成。地址位随着相应的操作递增,指针由最后位置返回初始位置时状态位取反。当读写指针的状态位和地址位全部吻合的时候,读写指针经历了相同次数的循环移动,这时FIFO处于空状态;如果读写指针的地址位相同而状态位相反,说明写指针多经过了一次循环,这时FIFO处于满状态。使用下面的公式来判断空满状态:

empty = (读指针 == 写指针)

full = (读地址 == 写地址 && 读指针状态位 != 写指针状态位)

如上所述,标志是通过对比读写指针的比较产生的,如果在读写时钟完全异步时对读写地址进行比较,可能得到错误的结果。例如,在读指针变化过程中对读写指针进行比较,这是由于读指针各位并不是同步变化,就有出现错误的判断,生成错误的空满信号。若FIFO未满时满信号已经置位,就降低了写数据的速率;而若FIFO已满却没有置位,就产生溢出错误。同理,空信号也会出现类似的错误,这对于实际应用来说是绝对不允许的<sup>[3]</sup>。

从上面的分析中可以看出,异步FIFO之所以会发生上述错误,是因为在地址变化时,多位地址的各位变化的时间不一样,异步时钟对其采样时数值可能不是地址最后的数值,这样就产生了错误的空满信号。格雷码是一种在相邻计数数值之间只有一位发生变化的编码方式,读写指针采用格雷码的编码方式,就可以解决上述错误。

## 3 异步FIFO的FPGA实现

FPGA(Field-Programmable Gate Array,现场可编程门阵列)在现代数字电路设计中发挥着越来越重要的作用。从设计简单的接口电路到设计复杂的状态机,FPGA所扮演的角色已经不容忽视。FPGA所具有的静态可重复编程和动态在系统重构的特性,使得硬件的功能可以像软件一样通过编程来修改,使用FPGA可以方便地对FIFO进行配置<sup>[4]</sup>。下面给出FIFO的Verilog编码。

程序1 自然码到格雷码的转换:

```
module norm_to_gray(din,dout);
parameter width=8;
```

```
input[width-1:0] din;
output[width-1:0] dout;
assign dout = din ^ {1'b0, din[3:1]};
endmodule
```

#### 程序 2 格雷码到自然码的转换:

```
module norm_to_gray(din,dout);
parameter width=8;
input[width-1:0] din;
output[width-1:0] dout;
integer i,j;
reg temp;
reg[width-1:0] buff;
always @(din)
for (i=width-1;i>=0;i=i-1)
begin
temp=0;
for(j=width-1;j>=i;j=j-1)
begin
temp=temp ^ din[j];
end
buff[i]=temp;
end
assign dout=buff;
endmodule
```

#### 程序 3 写地址产生(只给出地址产生部分):

```
always @(posedge wr_clk or negedge rst)
//rst 是复位信号
begin
if(!rst)
wadd<=0; //初始自然码写指针
else
begin
wadd<=wadd+1;
norm_to_gray(wadd,wadd_gray);
//wadd_gray 是格雷码写指针
end
end
```

end

#### 程序 4 满标志产生:

```
always @(radd_gray or wadd_gray)
//radd_gray 是读指针格雷码
begin
gray_to_norm(wadd_gray,wadd);
gray_to_norm(radd_gray,radd);
if(radd[width-1]! = wadd[width-1]
&& radd[width-2:0] = wadd[width-2:0])
full<=1;
else
full<=0;
end
```

类似的,可以写出读地址产生模块和空标志产生模块<sup>[5]</sup>。

## 4 结 论

文中针对异步 FIFO 设计中的难点和要点,提出了具体的解决方案,并用 verilog 语言给出了电路的设计,利用 Altera 的 Cyclone II 系列 FPGA 实现,并被应用于多种电路中,在实际应用中取得了较好的效果。

#### 参考文献:

- [1] KANOPOULOS N, HALLENBECK J J. A First - In, First - out Memory for Signal Processing Applications[J]. IEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, 1986, CAS-33(5):556-558.
- [2] 夏宇闻. Verilog 数字系统设计教程[M]. 北京:北京航空航天大学出版社,2004.
- [3] 罗 昊. 一种异步 FIFO 的设计方法[EB/OL]. <http://www.dianzi.net/show.asp?id=1733>, 2004-12-19.
- [4] 任爱锋, 初秀琴, 常 存, 等. 基于 FPGA 的嵌入式系统设计[M]. 西安:西安电子科技大学出版社,2004.
- [5] 杜建国. Verilog HDL 硬件描述语言[M]. 北京:国防工业出版社,2004.

(上接第 16 页)

案材料,然后进行删除或修改。

## 4 结束语

DIQA 知识库,是用 XML 表示知识,各知识库拥有统一的标准化文档格式,并与平台、应用软件无关,通过 DOM 处理文档信息,便于维护。实验表明,用该方法构建起来的知识库,极大地方便了在系统平台下实现资源共享和知识库维护。DIQA 知识库作为原型系统平台下存储知识资源的容器,在“信息高速公路”中会有广阔的应用前景<sup>[5]</sup>。

#### 参考文献:

- [1] 刘亚军,徐 易. 一种基于加权语义相似度模型的自动答疑系统[J]. 东南大学学报(自然科学版),2004.34(5):609-612.
- [2] 吕俊峰,潘会友. 基于 XML 的格式化网络课件[J]. 太原理工大学学报,2002,33(2):204-207.
- [3] 徐 易. 智能答疑系统的研究与实现[D]. 南京:东南大学,2002.
- [4] 侯晓强,徐春荣,勾海波. Java 服务器编程实例[M]. 北京:清华大学出版社,2003. 131-142.
- [5] 张丽晖. Mobile Agent 原型系统的研究与开发[D]. 南京:东南大学,2004.