文章编号:1000 - 582X(2002)12 - 0069 - 05

消除 CPLD/FPGA 器件设计中的毛刺*

何　伟，张　玲
(重庆大学 通信工程学院，重庆 400044)

摘　要:复杂可编程逻辑器件 CPLD 和现场可编程门列阵 FPGA 在现代数字系统设计中起着越来越重要的作用，但系统设计中出现的毛刺往往成为系统设计成功与否的障碍。本文介绍了 CPLD/FPGA 设计中毛刺产生的原因，详细分析了在 EDA 环境下对毛刺进行判断与定位的原理，论述了利用 EDA 工具消除毛刺的 3 种方法与具体实现。通过实例分析表明，借助于功能强大的 EDA 工具，不仅可以在设计中十分方便地发现毛刺，而且可以精确定位，进而寻找消除毛刺的最佳方法和进行结果的验证。

关键词:毛刺；竞争与冒险；复杂可编程逻辑器件；现场可编程门列阵；电子设计自动化
中图分类号:TN47；TN79　　**文献标识码**:A

数字系统设计中的毛刺(竞争 - 冒险)现象是长期困绕电子设计工程师的设计问题之一，是影响设计工程师设计效率和数字系统设计有效性和可靠性的主要因素。由于毛刺的存在，使得系统存在诸多潜在的不稳定因素，尤其是对尖峰脉冲或脉冲边沿敏感的电路就更是如此。毛刺通常主要对电路的触发清零端 CLR、触发复位端 RESET、CP 端、锁存器的门控端和专用芯片的控制端等产生严重的影响，会使电路发生误动作，从而造成数字系统的逻辑混乱[1]。

随着 VLSI 和计算机技术的飞速发展，数字系统的设计发生了革命性的变化，今天的数字系统设计已经进入广泛应用大规模高密度可编程逻辑器件(HDPLD)的电子设计自动化(EDA)的新时代，一台电脑、一套 EDA 软件，在家中也能高效地实现大规模集成电路和数字系统的设计[2]。尽管如此，毛刺仍然是设计中所必须克服和解决的主要问题之一，现代数字系统设计中对毛刺的判断、定位和分析完全可以在远离硬件的 EDA 环境下通过精确的时序仿真和时间分析来完成，毛刺的消除方法同传统方法相比有新的特点和本质不同。遗憾的是关于在 EDA 环境下判断和消除毛刺的系统方法在资料中很少报导，笔者就采用高密度可编程逻辑器件的现代数字系统设计中关于毛刺判断、定位和消除的问题进行论述。

1　毛刺的产生

要消除毛刺，首先要了解毛刺产生的原理。在高密度可编程器件(HDPLD)内部，延时处处存在，且延时主要由门延时和路径延时两部分组成，其中路径延时是主要的。图 1 是一个最简单的组合电路，图 2 是该电路的时间分析的结果，图 3 是其时序仿真的波形图[3-6]。

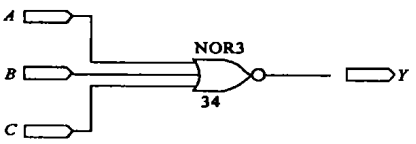


图 1　简单组合电路

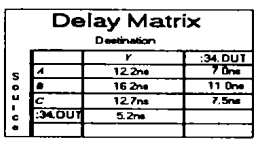


图 2　“图 1”的工作波形

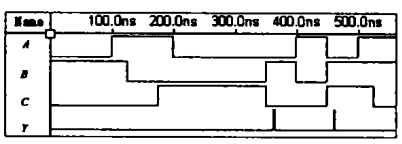


图 3　“图 1”的时间分析

从图 2 可知，输入脚 A、B、C 至输出脚 Y 的延时分别是 12.2 ns、16.2 ns 和 12.7 ns，这其中包含了 3 部分：其一是输入脚 A、B 和 C 至或非门输入端的路径延时；其二是门延时，图 2 表明这两部分之和分别为 7.0 ns、

* 收稿日期:2002 - 09 - 23
作者简介:何伟(1964 -)，男，四川南充人，重庆大学副教授，硕士，主要从事电子系统设计的研究。

11.0 ns 和 7.5 ns,该延时主要由第一部分构成;最后是门的输出端至输出脚 Y 的路径延时 5.2 ns。由于门延时对 3 个输入脚都是相同的,因此它们在延时上的差别完全来源于第一部分。通过计算知 B 比 C 和 A 的延时要大 4.0 ns 和 3.5 ns。

现在来分析图 3 的波形图。显然该图出现了 2 个尖峰脉冲,分别出现在 362.7 ns、462.2 ns 处,脉宽分别为 3.5 ns、0.5 ns。进一步分析可知,在 350.0 ns 时 B、C 信号同时变化,B 由 0 变为 1,C 由 1 变为 0。如果没有延时或 B、C 的延时相同,显然毛刺不会出现,因为输入信号是同时变化的,ABC 的变化过程是 001→010,输出将恒为 0。但是如上的假设并不成立,C 将在 12.7 ns 后先起作用,B 要到 16.2 ns 后才起作用,于是 ABC 的变化过程是 001→000→010,在 $ABC=000$ 时 $Y=1$,此时间正好是 3.5 ns。所以在 350.0 ns 处的 B、C 变化造成在 362.7 ns 处产生 3.5 ns 宽的毛刺并不奇怪。

同理可以分析出为什么在 450.0 ns 处的 ABC 变化会造成在 462.2 ns 处产生 0.5 ns 宽的毛刺,该期间输入信号的变化过程为 100→000→001→011。在数字电路中,把门电路两个或两个以上输入信号同时发生状态变换,且至少有两个信号是向相反的逻辑电平跳变的现象叫做竞争[1]。显然毛刺是由于竞争而产生的,但并不是所有的竞争都要产生毛刺,图 3 中 400.0 ns 处的竞争就没有产生毛刺,其原因是 0 向 1 的跳变早于 1 向 0 的跳变。

以上分析可见,竞争是产生毛刺的原因,信号间延时不同又是引起门电路竞争的原因。与传统设计不同的是延时主要由路径延时决定,而非门延时。

2 毛刺的判断、定位与分析

传统设计中的毛刺是在硬件测试中才能被发现,这需要做很多实验室的工作。相比之下,现代数字系统设计中对毛刺的判断、定位和处理要容易得多,这些工作完全可以在远离硬件的环境中进行,即在 EDA 软件下通过精确的时序仿真和时间分析来完成。而且,设计过程的任意一次精确的局部或全局软件时序仿真中毛刺都是难于逃脱,换句话说毛刺不是在整个设计完成后才被发现的,而是在设计过程中被发现的,这更有利于问题的解决。下面通过一个实例进行讨论。

图 4 是一个四象限对称的函数信号发生器数据产生控制电路。该电路设计工作原理如下:初始条件下 32 进制双向计数器 counter32 在 clk 信号的作用下首先进行加法计数,其输出信号送 0 检测模块 detect0 和 30 检测模块 detect30,当 detect30 检测到计数值为“11110”时输出一个正脉冲,该脉冲作为 RS 触发器的 R 输入端将 RS 触发器置 0 端,从而将 counter32 置成减法计数;当 detect0 检测到计数值为“00000”时输出一个正脉冲,该脉冲作为 RS 触发器的 S 输入端将 RS 触发器置 1 端,从而又将 counter32 置成加法计数。在这样的控制下 counter32 实际完成的是 60 进制的计数,计数值的变化规律是 0→1→2→...→29→30→29→...→2→1→0→1,此循环周而复始,RS 触发器的输出信号 RS-Q 通过二分频器 divider2 后输出的信号 Q 用于对 D/A 输出后的放大器的反向控制。

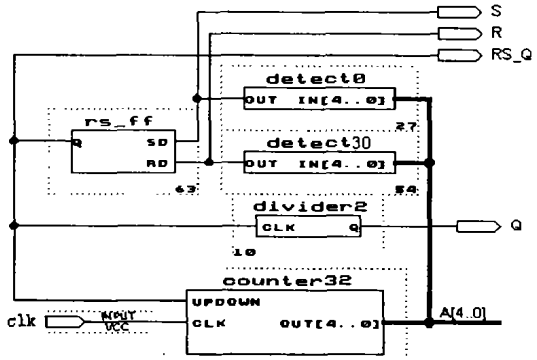
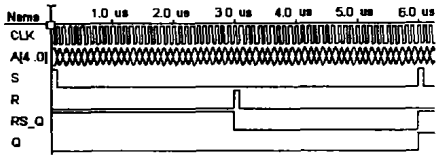
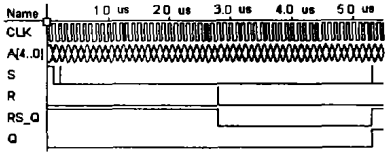


图 4 函数发生器数据产生控制电路

图 5(a)是图 4 的功能仿真波形图,图中可以看出在周期为 0.1us 的 clk 的作用下,RS-Q 的高低电平均为 3 us,而 Q 信号是 RS-Q 的二分频输出。R、S、RS-Q 和 Q 输出的波形完全符合设计要求,因此作为功能设计,该电路是成功的。但是要使该电路下载后的集成芯片能正常工作,还必须适配到具体的 CPLD/FPGA 后进行精确的时序仿真。图 5(b)就是图 4 适配到 FLEX10K10 芯片后的时序仿真波形图。比较(a)、(b)两图发现,(b)图的逻辑功能发生了改变。进一步分析可知,detect0 的输出信号 S 出现了 2 个毛刺,定量测量知第 1 个毛刺宽度为 0.4 ns。毛刺出现的位置在 A[4..0]从 00001 向 00010 的跳变过程中,竞争出现在 A_0 和 A_1 上。从图 6 的延时矩阵可知 A[4..0]从 counter32 的 q[4..0] 到 detect0 的延时为 $A_0=6.6$ ns、$A_1=7.0$ ns,简单分析后可知跳变过程为 00001→00000(0.4 ns)→00010,所以在 detect0 的输出理应出现一个 0.4 ns 的毛刺。第 2 个毛刺宽度为 3.7 ns,位置在 A[4..0]从 00100 向 00011 的跳变过程中,同理 detect30 的输出信号 R 也出现了一个宽度为 0.5 ns 的毛刺,位置在 A[4..0]从 11011 向 11100 的跳变过程中。此时 counter32 已经不是 60 进制的计数,显然系统的逻辑功能完全被毛刺给修改了,这正说明了毛刺的危害。



(a) 功能仿真图



(b)时序仿真图

图 5　"图 4"的工作波形

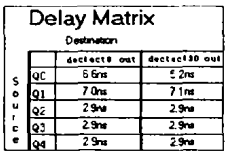


图 6　时间分析

需要说明的是以上在 EDA 系统下对毛刺的判断、定位和分析是非常精确和接近实际的，同时又清楚地看到了毛刺的危害。而且整个过程只需敲敲键盘、击击鼠标、看看屏幕便能轻松完成，而不需要泡在实验室里满头大汗地埋头苦干。

3　消除毛刺的方法

3.1　调整路径延时

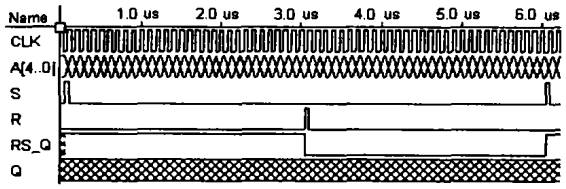
毛刺归根到底是由于延时不同而引起的，因此只要让门电路的所有输入信号具有相同的延时，毛刺就不会产生。所以消除毛刺最原始、最直接的方法就是调整延时。因为 CPLD/FPGA 的内部延时主要由路径延时组成，所以调整延时实际上就是调整门电路在芯片内部的位置以改变路径的长度从而达到改变延时的目的。调整工作需要有十分的耐心仔细地去做，需要反复调节和时间分析，好在 EDA 软件下进行这样的调节和时序、时间分析操作都非常的方便、简单和快捷，有的 EDA 软件还提供有专门处理这类问题的应用。

但是，这种方法有着极大的局限性。首先，在有些设计中要求的延时调节范围非常大，如果系统本身较复杂，内部利用率已很高，调节就非常困难。其次，即便是调整好的电路，在外部使用条件如温度、电压等发生变化时，其延时特性也会随着发生变化从而使毛刺再次出现。再次，这种方法设计的电路必须针对特定的芯片，电路没有再利用的价值。

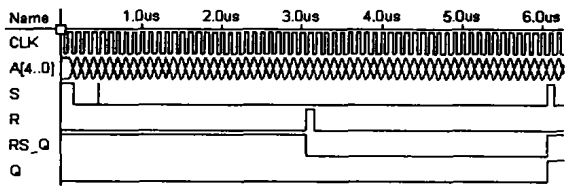
3.2　引入选通信号

由于竞争仅仅发生在输入信号变化转换的瞬间，在稳定状态是没有竞争的，所以在输入信号稳定后进行选通就可以彻底消除尖峰脉冲。传统设计中也可通过引入选通信号的方法消除毛刺，但受 MSI 功能的影响，其应用的要么受到极大的限制，要么将增加电路的复杂性。在 EDA 的应用中引入选通脉冲具有非常大的灵活性，因为所有功能模块电路都可以自己重构，选通脉冲的引入或许只是修改两句 VHDL 语言就可完成，引入的位置也具有极大的随意性。

仍以图 4 为例，图 7(a)是在图 4 的 detect0 和 detect30 的输出端引入选通脉冲的，由于 A[4..0]是在 clk 的上升沿发生跳变的，则选通信号采用 clk 的低电平。由图可见引入选通后 R、S 信号上的毛刺被彻底消除了，但是却带来了新的问题，即 R、S 的输出信号将不再是电平输出信号了，而是变成了脉冲信号，这种变化有时会影响电路的正常工作，图 7(a)表明它使 RS-FF 的初始状态不确定，导致二分频的输出 Q 的状态将始终不确定，这是该法的缺点。



(a)对 R、S-选通的时序图



(b)对 A[4..0]选通的时序图

图 7　"图 4"引入选通后的工作波形

图 7(b)是在图 4 的 counter32 的输出端引入选通脉冲的，选通信号也是 clk 的低电平，由于 counter32 的输出端至 detect0 和 detect30 的路径延时较大而且不同，所以毛刺仍可能存在，但是引入选通后至少 counter32 的输出信号的跳变是同时完成的，因此改变了信号间的延时的关系从而改变了毛刺出现的位置，只要新出现的毛刺的位置对电路没有不良后果，这样的设计就是成功的。图 7(b)正是这种情况，因为 RS-FF 中出现在 S 脉冲与 R 脉冲之间的 S 信号上的毛刺，以及出现在 R 脉冲与 S 脉冲之间的 R 信号上的毛刺对触发器的状态没有任何影响。

3.3　修改逻辑设计

应用可编程逻辑器件进行设计的最大特点是它的所有功能模块都是自己够建的，或者是可修改的，这种修改是不受任何限制的。图 4 中的计数器 counter32

如果修改成单信号变化的计数器，就从根本上堵绝了产生毛刺的源泉，因为任何时候计数器的输出信号只有一个信号的状态发生变化。表 1 就是单信号变化计数器的真值表。这样的计数器用 VHDL 语言非常容易实现。

表 1 单信号变化计数器的真值表

计数值	输出状态	计数值	输出状态	计数值	输出状态	计数值	输出状态
0	00000	8	01111	16	11000	24	10100
1	00001	9	01110	17	11001	25	10110
2	00011	10	01100	18	11011	26	10111
3	00010	11	01101	19	11010	27	10101
4	00110	12	01001	20	10010	28	11101
5	00100	13	01011	21	10011	29	11111
6	00101	14	01010	22	10001	30	11110
7	00111	15	01000	23	10000	31	11100

这种方法的缺点是计数器的输出状态没有什么规律，不便于其它电路利用。如果希望计数器的输出状态还要有规律变化以便其它电路使用，可以考虑采用双输出电路，一路是单信号变化输出专门提供给 detect0 之类的检测电路，另一路是正常计数输出供其它功能电路使用。

下面这段程序就是双输出计数器计数控制的片段。

```
process(clk)
begin
  if clk'event and clk = '1' then
    if updown = '1' then
      state <= state + 1;
    else
      state <= state - 1;
    end if;
  end if;
end process;
process(state)
begin
  case state is
    when 0 => A <= "00001";B <= "00001";
    when 1 => A <= "00010";B <= "00011";
    when 2 => A <= "00011";B <= "00010";
    when 3 => A <= "00100";B <= "00110";
    when 4 => A <= "00101";B <= "00100";
    when 5 => A <= "00110";B <= "00101";
    when 6 => A <= "00111";B <= "00111";
    when 7 => A <= "01000";B <= "01111";
    when 8 => A <= "01001";B <= "01110";
    when 9 => A <= "01010";B <= "01100";
    when 10 => A <= "01011";B <= "01101";
    when 11 => A <= "01100";B <= "01001";
    when 12 => A <= "01101";B <= "01011";
    when 13 => A <= "01110";B <= "01010";
    when 14 => A <= "01111";B <= "01000";
    when 15 => A <= "10000";B <= "11000";
    when 16 => A <= "10001";B <= "11001";
    when 17 => A <= "10010";B <= "11011";
    when 18 => A <= "10011";B <= "11010";
    when 19 => A <= "10100";B <= "10010";
    when 20 => A <= "10101";B <= "10011";
    when 21 => A <= "10110";B <= "10001";
    when 22 => A <= "10111";B <= "10000";
    when 23 => A <= "11000";B <= "10100";
    when 24 => A <= "11001";B <= "10110";
    when 25 => A <= "11010";B <= "10111";
    when 26 => A <= "11011";B <= "10101";
    when 27 => A <= "11100";B <= "11101";
    when 28 => A <= "11101";B <= "11111";
    when 29 => A <= "11110";B <= "11110";
    when 30 => A <= "11111";B <= "11100";
    when 31 => A <= "00000";B <= "00000";
    when others => null;
  end case;
end process;
```

5 结 论

EDA 环境中的数字系统设计，毛刺依然是系统成功设计的障碍，或者可以说对毛刺的处理仍然是现代数字系统设计所必须解决的问题。通过 EDA 软件的精确分析和仿真可以方便地查找毛刺出现的位置、原因，并制定相应的解决方法。讨论的 3 种方法中调整延时法是最费时、费事的方法，设计的电路有较大的局限性；选通法常常能取得较好的效果，但要求电路必须能正常工作在脉冲信号下，这也限制了它的使用；修改逻辑设计的方法由于从根本上消除了毛刺产生的根源，因此具有最普遍的意义，但同时由于该法必须全面分析、掌握和设计电路的工作状态及其转变，有时还需要多路输出，因此又带来了它的复杂性。

参考文献：

[1] 王毓银.数字电路逻辑设计[M].北京：高等教育出版社，

1999.
[2] 潘松.VHDL 实用教程[M].成都:电子科技大学出版社,2000.
[3] 宋万杰.CPLD 技术及其应用[M].西安:西安电子科技大学出版社,1999.
[4] 曾繁泰.可编程器件应用导论[M].北京:清华大学出版社,2001.
[5] 曾繁泰.EDA 工程概论[M].北京:清华大学出版社,2002.
[6] Altera 公司:MAX + PLUSⅡ[Z].1997.

Eliminate the glitch in design of CPLD/FPGA

HE Wei, ZHANG Ling
(College of Communications Engineering, Chongqing University, Chongqing 400044, China)

Abstract: Complex Programmable Logic Device (CPLD) and Field Programmable Gate Array (FPGA) play more and more important roles in design of modern digital system. But sometimes, the glitch in design of the system become handicap of succeedable design. This paper analyses the cause, judge and locate of glitch in design of modern digital system apply the component of CPLD/FPGA. Three methods about eliminate the glitch make use of EDA tools are discussed and some examples are given.

Key words: glitch; race and hazard; CPLD; FPGA; EDA

(责任编辑　吕赛英)

(上接第 65 页)

A Research on the Characteristics of Mexican-hat Mother Wavelet Used in Optic Realization

TIAN Feng-chun
(College of Communications Engineering, Chongqing University, Chongqing 400044, China)

Abstract: The Mexican-hat mother wavelet used in optic realization is analysed. Its characteristics on time-frequency domain localization, accurate reconstruction condition, regularity order and orthonormality are discussed. The error concept for the Mexican-hat wavelet that the non-tensor product two dimensional form of mother wavelet is obtained from the turning of its one dimensional form is corrected. The reason is explained for that the Mexican-hat mother wavelet can't be used in conventional discrete wavelet transform for image data compression, but when it is used in 2-D image wavelet transform realized by optic way, it features excellent energy localization.

Key words: wavelet transform; optic realization; Mexican-hat; mother wavelet

(责任编辑　张　苹)