

流水线的 FPGA 低功耗设计^①

李宏钧 胡小龙 (中南大学 信息科学与工程学院 湖南 长沙 410075)

摘要: 在组合逻辑中加入寄存器级形成流水线, 减少了信号毛刺的产生和传播, 从而降低 FPGA 动态功耗, 通过 XPower 功耗分析工具总结出了流水线设计和非流水线设计的功耗, 为了做出更完整的对比, 使用了低翻转率信号, 随机翻转率信号和高翻转率信号作为输入, 最后得出结论, 对于高翻转率的信号, 使用流水线可以一定程度的降低 FPGA 的功耗, 对于低翻转率的信号, 使用的流水线可能会使用比非流水线更多的功耗, 并分析了其原因。

关键词: 现场可编程门阵列; 低功耗; 流水线; 毛刺; 乘累加单元

FPGA Low Power Design Using Pipeline

LI Hong-Jun, HU Xiao-Long

(Institute of Information Science and Engineering, Central South University, Changsha 410075, China)

Abstract: By adding new registers in combinational logic, pipeline can reduce the generation and propagation of glitch. This paper compares the power dissipation of pipeline design and non-pipeline design, using low transition probability signals, random transition probability signals and high transition probability signals. It comes to the conclusion that pipeline can reduce power dissipation in high transition probability signals design, but for low transition probability signals design, pipeline may consume more power, and its reason is analyzed.

Keywords: FPGA; low power; pipeline; glitch; MAC;

1 引言

FPGA(Field Programmable Gate Array)以其可重构与现场可编程的灵活性, 相比于 ASIC 而言, 有着更短的开发周期和更小的前期研发资金投入, 在原型板的开发中占据着重要的地位, 获得了越来越多的人的青睐。随着问题规模的复杂和对速度等性能的要求不断提高, FPGA 设计的功耗问题日益成为开发设计的瓶颈, 对于以电池供电的便携式的设计而言, 降低功耗意味着更长的电池寿命和续航时间, 也能扩大 FPGA 在诸如小型手持设备终端上的应用; 对于使用交流电网的设计而言, 降低功耗则意味着降低设备的工作温度, 减少对冷却设备的要求, 同时更低的功耗也减少器件的过多的发热可以提高设计的稳定性和延长芯片的寿命, 有研究表明, 温度降低 20°C, 可以使芯片的总体寿命延长 10 倍^[1]。

FPGA 的功耗包含静态功耗和动态功耗两部分。

$$P_{total} = V_{DD} I_{Leak} + V_{DD} I_{dp} + C_L V_{DD}^2 F + \frac{\beta}{12} (V_{DD} - 2V_T)^3 \tau F,$$

其中 V_{DD} 是供电电压, I_{Leak} 是泄露电流, I_{dp} 是输入电压的直接电流通路电流, C_L 是负载电容, V_T 是门限电压, τ 是平均的上升沿时间, F 是时钟频率。前两项构成了 FPGA 的静态功耗, 后两项构成 FPGA 的动态功耗。静态功耗就是指逻辑门在没有开关活动时候的功耗, 包括泄露电流和导电电流两部分, 随温度和工艺的不同而不同。静态功耗主要取决于所选的 FPGA 产品; FPGA 的动态功耗是所有电容性节点充放电产生的组合功耗和电路转换过程中的短路电流形成的功耗^[2]。

如上所述, 根据 FPGA 功耗的组成部分, 降低 FPGA 功耗的技术主要也就包含两大类: 第一类就是降低 FPGA 的静态功耗, 这可以使用更好的架构如三层氧化物防泄露, 更低的或根据不同性能要求对电

① 收稿时间:2009-12-13;收到修改稿时间:2010-01-07

路不同模块提供不同的供电电压、合适的 CMOS 门限电压等一些电路基本组成结构上的调整来减少泄露电流；第二类是降低 FPGA 的动态功耗，主要是在保证设计满足应用要求的情况下，通过降低设计中的有效信号翻转率，减少毛刺无效信号的产生和传播来降低功耗。这些方法包括使用流水线，门控电路，内嵌的定制功能块，能减少平均布线长度和平均节点负载电容的更好的自适应探索布局布线算法，更好的适合于 FPGA 逻辑实现的设计算法等^[3,4]。降低 FPGA 功耗的各种方法可以结合起来获得更好的降低功耗的效果和获得更满意的性能。同时为了达到整个系统的功耗尽量低，降低功耗并不仅仅是降低每个时钟周期的功耗，而应该考虑完成完整的计算需要消耗的功耗。

在第二部分将着重分析流水线可以降低 FPGA 动态功耗的原理，第三部分考虑了一个在实际中经常使用的 MAC(Multiply-Accumulate Unit)例子作为验证实例，最后对比了流水线设计和非流水线设计降低功耗的效果并提出了将来的研究。

2 流水线降低功耗的原理

通过流水线设计降低 FPGA 的功耗，主要包括两大类研究，一种是在保证性能的情况下使用更低的供电电压，或对于不同性能要求的流水线使用不同的电压^[5]，第二种是在不改变电压的情况下，使电路的时序余量最大，性能最好^[6]。本文主要针对第二种类型，从逻辑设计的角度分析了流水线对功耗的影响。

流水线属于同步时序电路，它使用组合逻辑和触发器实现电路功能，主要信号和输出信号都由时钟驱动触发器产生，以固定的时间间隔采样信号，可以以统一的方式处理长路径和短路径，能够避免毛刺，使信号稳定。同时同步时序逻辑可移植性更好，模块间的接口也比较稳定可靠。目前商用的 FPGA 都是面向同步的电路设计而优化的，提倡在设计中全部使用同步逻辑电路。

如图 1 所示，对 n 输入的 LUT 而言，输出和输入的逻辑关系可以表示为： $z = f(x_1, x_2, x_3, \dots, x_n)$ ，其中的 Z 为输出： $x_1, x_2, x_3, \dots, x_n$ 为输入数据，并且， $x_i \in \{0, 1, -\}$ 其中的 $-$ 表示无关输入，并非不定态。假设任何两个输入彼此不相关，任意输入的变化概率为 $P(x_i)$ ，由此输入导致的输出的变化概率为 $P(x_i, z)$ ，则输入 x_i 导致输出 z 发生变化的概率可以定义为： $P_{x_i, z} = \frac{\partial z}{\partial x_i}$ ，输出 z 的

翻转密度和输入的转换密度可以表示为： $D_z = \sum_{i=1}^n P_{x_i, z} \cdot D_{x_i}$ ，其中 D_{x_i} 是输入 x_i 的翻转密度，翻转密度是指单个信号在一个时钟周期内的平均翻转次数，而对于组成流水线的非计算单元 FD 而言，其信号变化率，当输入的信号是稳定的不含有毛刺信号的时候，可以认为输入和输出的变化率相等^[7]，即为 $p_i = p_o$ ，但当输入不稳定，由于寄存器的时钟沿触发和数据暂存作用，输出的变化率要小于输入的变化率，而流水线设计的引入正是要让不稳定的输入变成比较稳定的输出，减少无效信号翻转，从而降低设计的功耗。

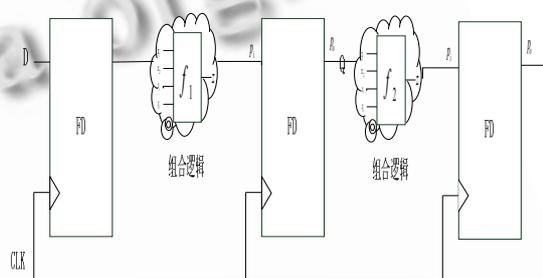


图 1 流水线基本单元

流水线分割了组合逻辑功能，把每个流水块的逻辑功能限制在相邻的逻辑单元中，有更短的布线长度和更小的负载电容，它还可以滤除毛刺波的传输，对于组合逻辑而言，当产生的毛刺比逻辑门的惯性延迟短的时候毛刺会被逻辑门所吸收，而更长的毛刺则会继续向后面的逻辑传输，导致多米诺骨牌效应。毛刺的传播在使系统变得不稳定并增加调试难度的同时，也会消耗了大量的能量，而加入了寄存器的流水线设计则可以很大程度上滤除这种毛刺效应。

流水线中要对时序进行合理安排，每个操作步骤的划分要合理，各个流水段的处理时间应尽量相等，如果简单操作和复杂操作都使用同样长度的流水线，将会造成时序周期和逻辑资源的浪费^[8]，但确定流水线的长度时候就必须以最长的流水操作为基准，所以为了控制不同复杂长度的流水线，就需要使用有限状态机来控制操作或者对复杂的逻辑进行划分，增加流水线段个数或使用多个并行的模块来提高复杂逻辑的处理速度，对操作进行时序分割了以后，简单的操作可能只需要一个时钟周期，而复杂的操作可能需要很多个周期。

3 流水线与非流水线设计的比较研究

一般的多路选择, 加法, 乘法等需要较多组合逻辑运算的往往在运算过程中产生大量的毛刺, 因此为了达到对比的效果突出流水线设计降低功耗作用, 考虑使用如下的乘累加单元 MAC 功能电路作为对比测试: 四个 18 位的输入数据被分为 a, b, 和 c, d 两组, 在第一组 a, b 中选择一个数据, 在另外一组 c, d 中也选择一个数据, 然后二者相乘, 然后剩余的每组一个数相乘, 然后把获得的乘积值相加, 输出得到结果。

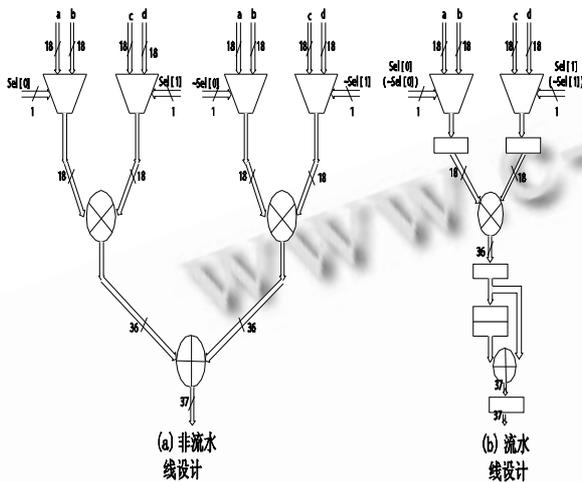


图 2 非流水线和流水线设计结构图

在图 2 中图(a), 是没有加入流水结构的实现方法, 该电路将会完全用组合逻辑来实现, 因为非流水线结构对输入数据没有缓存作用, 所以为了实现上述的功能需要四个二路选择器, 两个乘法器单元和一个加法单元; 在图 2 中图(b)是流水线结构设计, 并对非流水线设计中的四个选择器与两个乘法器和一个加法器进行了相同功能的功能单元合并, 形成了只有两个选择器与一个乘法器和一个加法器。相同的功能单元合并复用以后会导致计算出来的部分结果有先有后, 后进入乘法器的一对乘法运算结果要比开始的一对晚两个时钟周期, 这样为了不改变时钟同时也时序对齐, 先运算得到的部分结果要被暂存两周期, 然后与后选择的部分乘积结果同时到达加法器单元, 这个两级流水线在具体的设计使用的时候可以只使用一个 FD 来实现, 在该实验所用的 Xilinx 芯片元件库中, 可以使用一个 FD 来延迟一个信号一到十六个时钟周期, 这样同时也降低了设计中该使用的资源数, 对功耗有好

的促进作用, 同时为了减少加法单元的结果中的毛刺波的传播, 在加法单元之后也加入了一级流水缓存, 在最后输出的结果如果有比较多的毛刺而没有进行流水滤除, 会导致在 PCB 板上的传播, 而内部布线与 PCB 走线会有不同数量级的功耗差别, 而对每个模块的输出结果进行同步寄存器缓冲处理一下, 也是一种好的设计实现习惯, 这样可以使不稳定的信号限制在本模块内, 使下一个输入获得比较稳定的输入信号, 避免了不稳定毛刺波信号的跨模块传输。

为了测试尽可能多的情况, 使用了三种不同类型的输入向量进行测试: 第一种类型为: 低翻转率的信号, 这里采用所有位都为全 1 的数据模拟; 第二种类型为: 任意的随机输入信号, 为获得 18 位的随机数, 使用系统提供的 random() 函数产生测试向量, 同时为了在测试多次的时候有不同的结果, 使用系统的当前时间作为种子, 然后计算随机数据, 这样可以保证在多次不同的测试中有不同的随机结果; 第三种类型为: 高翻转率的信号, 使用相邻两位都不同的输入。并且乘积的两个数对应位互补。为了简化测试程序的设计, 系统的时钟都在整个测试过程中一直有效, 如果仅在有效数据采集运算的时候才给出时钟, 这样可以减少时钟网络的功耗, 但使用门控电路禁止时钟或使用选择器复用时钟本身会导致时钟质量的下降, 增加时钟的毛刺与不稳定。一个比较好的折中方法是对时钟信号不进行门控, 而对于输入数据和输出数据, 则加入使能信号, 只在信号的被采集或正确结果输出的时候才使能该信号, 这样可以过滤掉没有必要的内部信号多余翻转。

4 实验结果

设计验证使用 Xilinx 的 XC4VLX100 的芯片, 该芯片的基本逻辑单元为四输入的 LUT, 配套的综合和实现工具为 ISE 9.2i 开发套件, 仿真使用的是 Modelsim 6.1f-SE verilog 版本, 综合工具的选项配置为: 都勾选功耗优化, 其余的皆为默认选项。

功耗的分析需要得到信号的翻转频率, 使用 modelsim 可以获得两种记录波形的格式数据, 一种是 WLF(wave log file)格式文件, 该文件记录了信号的波形, 可以从中得到信号翻转, 但该文件时 modelsim 私有格式, 只能由他自己生成和打开, 同

时该文件也是默认生成的文件，另外一种格式的是 VCD(Value Change Dump)格式文件，该文件为 IEEE1364 标准定义的 ASCII 格式文件，其中格式化的记录了头信息，变量的预定义和变量值的变化信息，但该文件不是默认生成的，所以为了得到该文件，需要在 testbench 中使用 dumpvars 函数调用和使用 do 文件描述需要记录的信号名称，为了完整，do 文件需要描述设计中所有的使用到的信号，该验证设计中为了各个工具之间的兼容性，使用 VCD 格式来获得信号翻转率。

功耗分析的流程为：代码设计完成后，编写对应的用户约束文件 UCF 和布局约束文件 PCF，综合得到经过实现的设计文件 NCD 和资源使用情况 html 格式报告，再根据上节中的不同类型输入数据进行仿真生成 VCD 文件，然后将以上三个文件导入 XPower 中得到功耗总结报告 PWR。实验分析结果如图 3 所示。(单位：毫瓦)

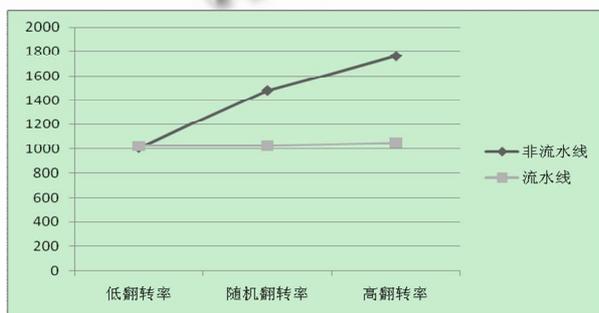


图 3 非流水线与流水线的功耗对比

从对比可以看到，流水线的设计对于信号的翻转率比较高的时候，设计的功耗可以得到较好的降低，但当信号的翻转率很低的时候，流水线设计并不能带来功耗的降低，反而会增加设计的功耗。当信号的翻转率很低的时候，在非流水线的设计中，尽管信号结果输出会有毛刺，但频率低，对功耗影响不是很大，相反的在流水线设计中因为引入了大量的寄存器，所以会比较组合逻辑消耗更多的功耗，而在信号的翻转率很高的时候，流水线的寄存器已经不是系统总体功耗的主要部分，过多的信号翻转和毛刺的传播成了影响功耗的主要因素，实验过程中也发现，如果在设计中采用使能技术，也就是说，外部来的输入信号，只在需要的时候才采集，当不需要的时候禁止寄存器使能端可以更多的降低设计的功耗。

5 小结

本文基于实际开发项目，结合 FPGA 的功耗的两部分组成，进行了流水线与非流水线设计的功耗比较与实验，通过对比功耗总结结果可以发现，对于低翻转率的信号流水线达不到降低功耗的效果，但可以降低高信号翻转率的电路设计的功耗，流水线设计在很长的组合逻辑路径中插入寄存器，虽然增加了运算周期数，却能大大减少组合逻辑链路延时，减少设计的逻辑深度，在提高整个系统的工作频率的同时，也降低了设计的功耗。本文对从事 FPGA 低功耗设计的人士会有参考借鉴价值。

参考文献

- 1 Tuan T, Steve T. The power consumptions of FPGA architectures. 今日电子, 2007, 15(9):31-33.
- 2 Andr G, Wayne B, Jean L. Low Power Digital Design in FPGAs: A study of Pipeline Architectures implemented in a FPGA using a low supply voltage to reduce power consumption. Sophie D. 2000 IEEE International Symposium on Circuits and Systems, Geneva: International conference center Geneva, 2000: 220-224.
- 3 许文曜, 徐新民. 基于新型 FPGA 的自可重构系统设计研究[硕士学位论文].杭州:浙江大学, 2008.
- 4 Anurag T. Low Power FPGA Design Techniques for Embedded System. Computer Science and Engineering, 2005, 7(1):132-150.
- 5 Sutter G, Boemo E. Experiments in low power FPGA design. Latin American Applied Research, 2007, 37(26):35-37.
- 6 Anand R, Sujit D. Glitch Analysis and Reduction in Register Transfer Level Power Optimization. Jha T. Proceedings of ACM/IEEE Design Automation Conference. San Francisco: Proceedings of the 46th Design Automation Conference. 1996:331-336.
- 7 Lamoureux J. Activity Estimation for Field-Programmable Gate Arrays. Madrid. Field Programmable Logic and Applications of International Conference, Milan: FPL06. 2006:35-37.
- 8 倪继利, 陈曦, 李挥. CPU 源代码分析与芯片设计及 linux 移植. 北京: 电子工业出版社, 2007.36-45.