

FPGA 复位设计的关键问题分析

徐 菲¹ 李卓群¹ 陈亚都¹ 徐梓雲¹ 张 瑶¹
XU Fei LI Zhuoqun CHEN Yadu XU Ziyun ZHANG Yao

摘 要

在 FPGA（field programmable gate array）设计中，复位是确保系统正确初始化和稳定运行的关键环节。因此，文章聚焦于 FPGA 中常用复位方法的时序性能展开深入分析，并结合实例分析了不同复位设计方法对 AMD 和 Altera 的 FPGA 资源使用率的影响。其次，对 FPGA 复位设计中的复位信号来源进行分析，剖析了跨时钟域时使用异步复位同步释放方式的实现原理。针对复位信号高扇出问题，提出基于复位信号分层驱动和优化策略。并对子模块的复位设计进行了分析并提出解决方法，确保了复位信号的同步性和可靠性。

关键词

复位；FPGA；异步复位同步释放；复位来源；高扇出

doi: 10.3969/j.issn.1672-9528.2025.06.038

0 引言

FPGA（field programmable gate array）以其灵活性、定制性、可重配性、可扩展性和并行处理能力^[1-2]得到了越来越广泛的应用。随着系统功能越来越复杂，对复位设计的要求也越来越高。由于不同的 FPGA 厂商对其触发器、Block RAM 的上电默认状态不统一，FPGA 需要通过复位使电路处于一个已知的状态，确保逻辑电路同步工作。此外，复位可以使系统发生问题后，能够恢复到正常状态。

不同的复位方式对 FPGA 的时序性能和资源使用率影响不同，进而影响到整个系统的功能和性能。因此对 FPGA 复位设计中的关键问题进行分析，对保证系统的稳定性和可靠性具有重要的意义。

1 常用复位方法时序分析

1.1 同步复位

同步复位时，只有当时钟沿采集到同步复位信号的有效电平时，才会在时钟沿到达时刻进行复位操作^[3]。

在时序方面，同步复位是同步时序电路，有利于时序约束和时序分析。同时，由于同步复位只在有效的时钟边沿到来时进行复位，因此可以滤除高于时钟频率的复位毛刺。

但是，同步复位信号依赖于时钟，在下列场景中复位可能失效：（1）如图 1 的场景①所示，复位信号持续时间小于一个时钟周期，导致复位信号丢失；（2）如图 1 的场景

②所示，如果使用了门控时钟，在复位有效的时候时钟可能还处在静止状态，导致电路不能复位。

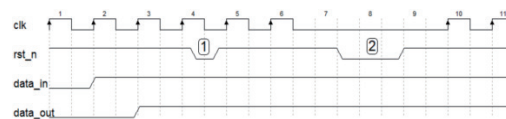


图 1 同步复位失效场景

1.2 异步复位

异步复位信号不受时钟信号的控制，当复位信号到来时立即进行复位操作。

异步复位响应速度快，但是易受毛刺干扰，而且若复位信号的释放时刻和时钟边沿接近，很容易使寄存器的输出产生亚稳态。此外，由于存在时钟偏移，异步复位信号经过不同路径到达各个寄存器的时间不同，可能导致亚稳态或复位的实际释放时刻不同^[4]。如图 2 所示，复位信号 rst_n 经过不同的路径到达触发器的时间不同，rst_n_path1 在时钟沿 2 之前释放，满足恢复时间；rst_n_path2 刚好在时钟沿 2 上释放，可能导致亚稳态；rst_n_path3 在时钟沿 2 之后释放，此时比 rst_n_path1 和 rst_n_path2 晚释放一个时钟周期，导致一部分寄存器在时钟沿 2 之前完成复位，而另一部分寄存器在该时钟 2 之后才完成复位，进而导致后续的逻辑功能出现混乱。

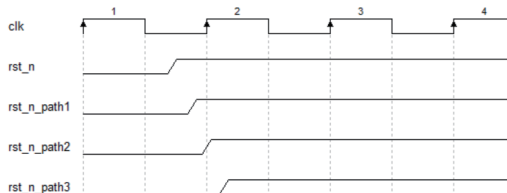


图 2 异步复位信号的释放

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710076

1.3 异步复位，同步释放

异步复位，同步释放是在复位信号到来时立刻进行复位，在复位信号撤离时，等到下一个时钟沿到达时才撤离^[5]。综合了异步复位响应快和同步复位释放时序良好的优点。

因此，进行 FPGA 复位设计时，对外部输入的复位信号或者跨时钟域的复位信号进行异步复位、同步释放，然后用同步后的局部复位信号对该时钟域内的所有模块进行复位。确保同一复位路径上复位方式的一致性，避免复位时相关寄存器的跳变时间不一致^[6]，降低亚稳态。

2 复位对 FPGA 资源使用率的影响

在不同厂家的 FPGA 器件中，不同的复位方式对 FPGA 的资源利用率也不同。

2.1 AMD 公司的 FPGA

AMD 公司的 FPGA，相比低电平复位，使用高电平复位可以节省一个 LUT (look-up table)。图 3~6 分别是在 Vivado 软件中实现的 AMD FPGA 低电平同步复位、低电平异步复位、高电平同步复位和高电平异步复位电路图，通过对比可以看出，复位信号经过图 3 和图 4 时，多消耗了一个 LUT。原因是 AMD FPGA 触发器的 SR 控制端是高电平有效的，如果 RTL 代码采用低电平有效的复位方式，综合器将在复位信号驱动寄存器 SR 端之前插入一个查找表 LUT 来实现反相器。如果设计中所有的复位都如此，则会增加较多资源。

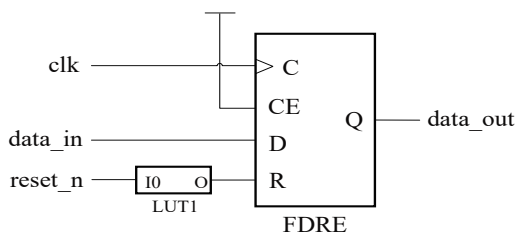


图 3 AMD FPGA 低电平同步复位电路图

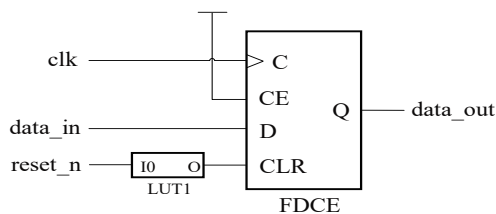


图 4 AMD FPGA 低电平异步复位电路图

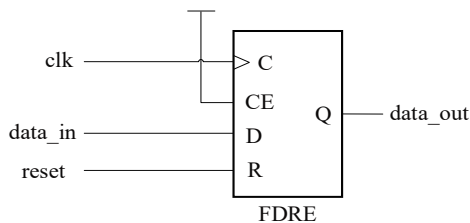


图 5 AMD FPGA 高电平同步复位电路图

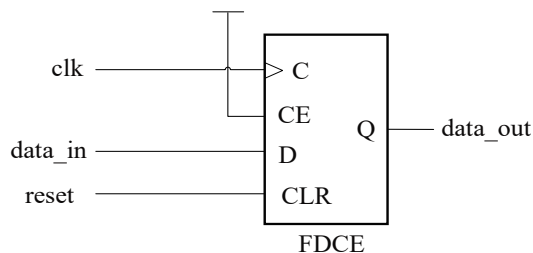


图 6 AMD FPGA 高电平异步复位电路图

对 AMD 公司的 FPGA，在使用 DSP48E1 或 BRAM 等资源时，推荐同步复位。原因是相比异步复位，使用同步复位时，可以利用 DSP48E1 或 BRAM 等资源内部的寄存器，提高 FPGA 器件的利用率，不需额外消耗 FPGA 寄存器资源，进而节省 FPGA 的总体资源。

例如，使用 AMD 公司的 FPGA 实现两个 16 bit 数据的乘法，并在 Vivado 软件中实现。表 1 和表 2 分别是 Vivado 软件中实现的高电平同步复位和高电平异步复位乘法的资源消耗情况。通过对比可以看出，在使用 DSP48E1 实现两个 16 bit 数据的乘法时，且复位信号为高电平的条件下，同步复位比异步复位节省了 16 个 LUT 和 1 个 FF (flip-flop)。

表 1 AMD FPGA 高电平同步复位乘法的资源消耗

资源	资源消耗个数	资源利用率
DSP	1	0.63
IO	66	19.53
BUFG	1	3.13

表 2 AMD FPGA 高电平异步复位乘法的资源消耗

资源	资源消耗个数	资源利用率
LUT	16	0.03
FF	1	0.01
DSP	1	0.63
IO	66	19.53
BUFG	1	3.13

2.2 Altera 公司的 FPGA

Altera 公司的 FPGA，使用异步低电平复位可以提高 FPGA 器件的利用率^[7]。

在 Quartus Prime 软件中实现复位电路，图 7 和图 8 分别是 Altera FPGA 同步低电平和同步高电平复位电路图，图 9 和图 10 分别为异步低电平和异步高电平复位电路图，对比图 7~10 可以看出，使用异步低电平复位可以节省逻辑单元。这是因为 Altera 公司的 FPGA 的触发器仅支持异步复位端口，异步低电平复位方式与其硬件架构完全匹配，可充分利用器件特性，如果采用同步复位和异步高电平复位方式，需要额外消耗逻辑资源。

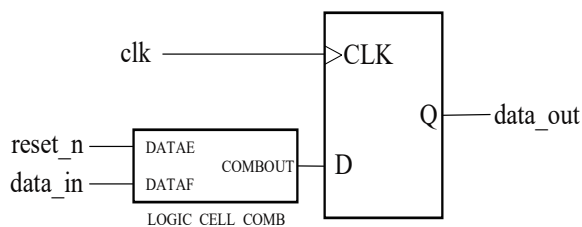


图 7 Altera FPGA 同步低电平复位电路图

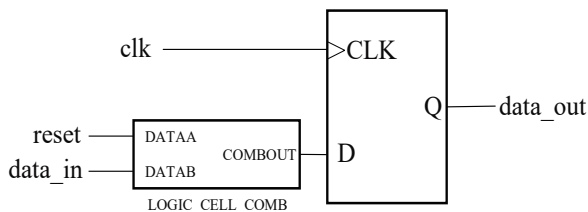


图 8 Altera FPGA 同步高电平复位电路图

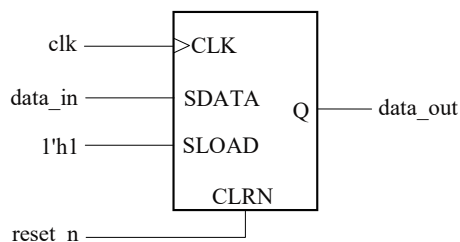


图 9 Altera FPGA 异步低电平复位电路图

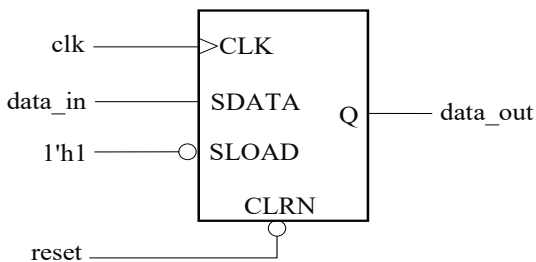


图 10 Altera FPGA 异步高电平复位电路图

3 FPGA 复位关键问题分析

3.1 复位信号来源

根据复位信号的来源不同，FPGA 的复位分为外部复位引脚复位和内部上电自复位。

外部复位：外部复位通常采用按键开关或者专用的复位芯片产生复位信号^[8]，对 FPGA 进行复位。外部复位的复位信号与 FPGA 是异步的^[9]，且需要额外的硬件电路。

内部复位：FPGA 不使用外部的复位信号，采用自复位的方式产生复位信号。可采用计数器的方式产生复位信号，FPGA 上电后处于复位状态，通过计数器控制上电自复位信号的转换，当计数器达到预设的值，复位信号释放，从而完成自复位。此外，PLL (phase-locked loop) 的 LOCK 信号也可以作为内部复位信号触发源使用。

FPGA 启动过程通常分为 8 个阶段：设备上电、清除配

置内存、采样模式引脚、同步、检测设备 ID、加载位流数据、CRC 校验和启动阶段^[10]。启动阶段包括等待时钟稳定、准备 I/O 状态等。启动阶段结束后，进入用户模式，FPGA 开始执行用户设计的逻辑功能。内部复位就是在用户模式阶段执行的，此时 FPGA 时钟已经稳定，通过内部复位将电路从一个已知状态开始工作。

3.2 异步复位，同步释放

对于外部输入的复位信号以及跨时钟域的复位信号，需要进行同步化处理，避免亚稳态问题。即采用异步复位，同步释放的方式，在复位信号到来时不受时钟信号的同步，而复位信号释放时受时钟信号的同步。既保证了能快速响应复位信号，又使复位信号的释放与时钟同步。

低电平异步复位同步释放的电路如图 11 所示，该电路使用 AMD 的 FPGA 实现。从图 11 中可以看出，复位信号 reset_n 经过 LUT1 连接到 FDCE 的异步清零端 CLR。由于采用异步低电平复位，在 AMD 的 FPGA 中需要使用一个 LUT 来实现取反操作。两个 FDCE 的时钟使能端 CE 均为 1，当复位信号 reset_n 为 0 时，经过取反操作，连接到 FDCE 的异步清零端 CLR 为 1，此时两个 FDCE 的输出 Q 端值为 0；当复位信号 reset_n 为 1 时，经过取反操作，连接到 FDCE 的异步清零端 CLR 为 0，此时第一级 D 触发器将数据输入 D 端的值 1 传递到输出 Q 端，第二级 D 触发器将数据输入 D 端的值，即第一级触发器的输出值 1，传递到输出 Q 端，最终经过两级触发器，输出值为 1。

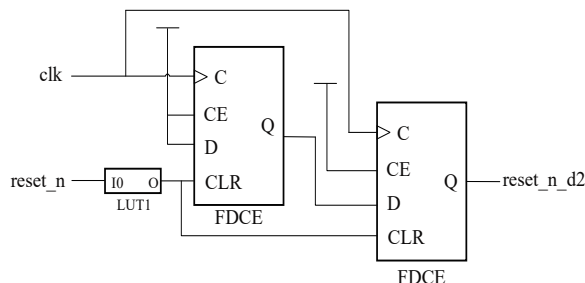


图 11 异步复位同步释放电路

3.3 复位信号高扇出优化

高扇出是指一个信号驱动多个下级寄存器的情况。如所有复位电路均使用同一个复位信号，会造成复位信号扇出高，在时钟频率比较高且驱动的寄存器比较多时，会降低驱动能力，容易导致时序路径过长、布线拥塞及功耗增加等问题。

为解决复位信号的高扇出问题，提出复位信号分层驱动的优化策略。将全局复位信号分解为多级局部复位信号，以优化负载分布并降低扇出。复位信号的两层驱动优化策略如下：

全局层：通过专用全局缓冲器（如 AMD FPGA 中的 BUFG^[11]）将全局复位信号分发到 FPGA 芯片的各区域。BUFG 是 FPGA 中用于增强信号驱动能力和减少信号在传输过程中的偏斜的重要资源。通过 BUFG 分发复位信号，可以有效解决复位信号的高扇出问题，并确保复位信号在整个芯片内的稳定传输。

局部层：采用寄存器复制的方式，即将复位信号进行复制，下级模块使用复制后的复位信号进行复位，从而防止复位信号扇出过大。原理如下：假设一个复位信号 reset 驱动 2 000 个寄存器，那么此时复位信号的扇出为 2 000。如果有多个寄存器信号都跟复位信号 reset 保持一致，那么这些寄存器信号也可以用来驱动这 2 000 个寄存器实现复位。如果有 5 个类似的寄存器信号，将需要驱动的 2 000 个寄存器均匀分配给这 5 个寄存器信号，即每个寄存器信号分配驱动 400 个寄存器实现复位，这样每个复位信号的扇出就由 2 000 降到了 400，极大降低了信号扇出。

3.4 子模块内的复位方式

进行复位设计时，应由专门的复位生成模块来集中对复位信号进行生成和处理，经过专门的复位生成模块产生的复位信号已经与各个子模块的时钟同步。复位生成模块负责对外部输入的复位引脚进行同步处理，或者产生内部的上电自复位信号。对于有多个时钟域的设计，需要对跨时钟域的复位信号进行异步复位，同步释放处理。此外，当复位信号驱动寄存器比较多时，建议将复位信号进行复制，减少复位信号的扇出。最后，局部复位信号进入到各个子模块。

各子模块进行复位设计时，尽量减少复位信号的使用^[12]。对于流水线的数路路径和没有反馈的电路，可不用复位，因为经过几个周期，错误的数据会被移除系统。对于控制路径，需要进行复位操作将电路置于已知的状态。

对于确实需要使用复位的路径，各个子模块具体是使用异步复位还是同步复位，低电平复位还是高电平复位，可结合系统的复位需求、工程代码复用以及器件特性，选择合适的复位方式。

4 结语

本文对 FPGA 的常用复位方法进行了时序分析，分析了同步复位、异步复位方法的优点以及存在的时序风险。然后，通过对 AMD 和 Altera 公司的 FPGA 进行复位电路和乘法电路的实现，分析了不同复位方式对 FPGA 器件的资源利用率的影响。最后，对复位信号的来源、复位信号的同步化处理、复位信号高扇出优化以及子模块内的复位方法等关键问题进行了分析，为 FPGA 复位设计提供了参考。

每种复位方式都有各自的优缺点，在进行系统复位设计时，应该根据具体的应用场景、目标器件的特性选择复位方式。

参考文献：

- [1] 禹芳. 一种基于 FPGA 的时序控制优化方法 [J]. 信息技术与信息化, 2024(5):132-135.
- [2] 李倩. 基于 FPGA 的嵌入式系统概述 [J]. 硅谷, 2014, 7(9): 147-152.
- [3] 郝建, 原茵茵. FPGA 复位的可靠性设计方法 [J]. 电子科技, 2013, 26(10): 125-127.
- [4] 樊继明, 陆锦宏. FPGA 深度解析 [M]. 北京: 北京航空航天大学出版社, 2015:117-130.
- [5] 李超, 赵启林, 戴兆麟, 等. 一种 SoC 时钟复位管理电路设计与验证 [J]. 单片机与嵌入式系统应用, 2023,23(6):8-11.
- [6] 陈云, 陈恩耀, 刘国斌, 等. FPGA 中关于复位的设计和研究 [J]. 数字技术与应用, 2018,36(3):182-184.
- [7] 姚建波, 沈亮, 程国华. 复位对 FPGA 亚稳态和资源利用率的影响 [J]. 微型机与应用, 2014,33(12):15-17.
- [8] 任磊, 王长宇, 黄克英. 基于 DSP 与 FPGA 的工业计算机设计 [J]. 山西电子技术, 2024(6):41-42.
- [9] 祝宇, 董冠涛, 张硕. FPGA 软件亚稳态处理技术研究 [J]. 中国检验检测, 2020, 28(3):14-17.
- [10] AMD. 7 Series FPGAs configuration (UG470 v1.17)[EB/OL]. (2023-12-05)[2025-03-02].https://docs.amd.com/v/u/en-US/ug470_7Series_Config.
- [11] AMD. 7 Series FPGAs clocking resources user guide (UG472 v1.14) [EB/OL]. (2018-07-30)[2025-03-06]. https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking.
- [12] CHAPMAN K. Get smart about reset: think local, not global[EB/OL].(2008-03-07)[2025-04-02].<https://docs.amd.com/v/u/en-US/wp272>.

【作者简介】

徐菲（1993—），女，山东日照人，硕士，助理工程师，研究方向：航电系统硬件开发，email:meredithxa@126.com。

（收稿日期：2025-03-02 修回日期：2025-06-04）