



[12] 发明专利说明书

专利号 ZL 200410064975.7

[45] 授权公告日 2008 年 5 月 14 日

[11] 授权公告号 CN 100388255C

[22] 申请日 2004.10.10

[21] 申请号 200410064975.7

[73] 专利权人 中兴通讯股份有限公司

地址 518057 深圳市南山区高新技术产业园科技南路中兴通讯大厦法律部

[72] 发明人 杨思远 吴 边 邵贵阳 孟春才
段晓雪

[56] 参考文献

US20030093607A1 2003.5.15

US6119189A 2000.9.12

US6204687B1 2001.3.20

WO03034199A2 2003.4.24

CN1414488A 2003.4.30

审查员 郑 宁

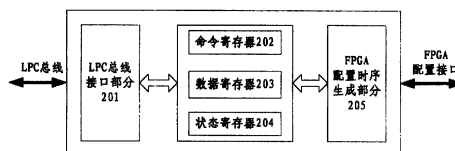
权利要求书 1 页 说明书 4 页 附图 3 页

[54] 发明名称

一种接口转换模块和对 FPGA 进行配置的方法

[57] 摘要

一种接口转换模块和对 FPGA 进行配置的方法，接口转换模块包括 LPC 总线接口部分、命令寄存器、数据寄存器、状态寄存器和 FPGA 配置时序生成部分；对 FPGA 进行配置的方法包括：处理器经过 LPC 总线发启动配置命令给接口转换模块生成 FPGA 下载的初始化时序并进行初始化；通过 LPC 总线送配置数据给接口转换模块生成 FPGA 下载时序，将数据下载到 FPGA 中；通过 LPC 总线发配置结束命令给接口转换模块生成相应配置结束时序；或者接口转换模块在数据配置完成以后，自动完成相应配置结束时序。本发明可以提高 FPGA 配置速度和可靠性；使用标准的 LPC 总线做为接口，模块兼容性很好，还可以提高硬件系统设计的灵活性。



1、一种接口转换模块，其特征在于，包括低针数总线接口部分（201）、命令寄存器（202）、数据寄存器（203）、状态寄存器（204）和FPGA配置时序生成部分（205）；所述低针数总线接口部分完成低针数总线读写寄存器组的时序；所述数据寄存器（203）暂存一个字节的FPGA配置数据；所述命令寄存器（202）控制由处理器通过低针数总线写入的操作命令；所述状态寄存器（204）用来查询该接口转换模块工作状态；所述FPGA配置时序生成部分（205）完成串行配置时序，将数据寄存器中的配置数据转换成串行配置数据流下载到FPGA中。

2、一种如权利要求1所述的接口转换模块对FPGA进行配置的方法，包括：

2.1 处理器经过低针数总线发启动配置命令给接口转换模块，接口转换模块接收到启动命令，开始生成FPGA下载的初始化时序并对所述FPGA进行初始化；

2.2 处理器通过低针数总线送配置数据给接口转换模块，接口转换模块接收到所述配置数据后，生成FPGA下载时序，将所述配置数据下载到FPGA中；

2.3 处理器通过低针数总线发配置结束命令给接口转换模块，接口转换模块接收到配置结束命令后，开始生成FPGA配置过程所需要的配置结束时序；或者接口转换模块在数据配置完成以后，自动完成FPGA配置过程所需要的配置结束时序。

3、权利要求2所述的对FPGA进行配置的方法，其特征在于，在初始化、配置数据和配置结束的每一步骤后，都有检测程序，检测对应的操作是否成功；前一操作成功才进行下一操作。

4、权利要求3所述的对FPGA进行配置的方法，其特征在于，检测程序检测对应的操作是否成功是通过读接口转换模块中的状态寄存器来实现的。

5、权利要求2所述的对FPGA进行配置的方法，其特征在于，配置数据的过程中，处理器每次发送一个字节的配置数据数据给所述接口转换模块。

一种接口转换模块和对 FPGA 进行配置的方法

技术领域

本发明涉及一种对 FPGA 进行配置的方法,更具体的说,是在电路板上通过 LPC(Low pin count 低针数)总线对 FPGA (field programmable gates array 现场可编程门阵列)进行配置的方法。

背景技术

为了方便产品的现场升级和减少用于配置 FPGA 的专用 PROM (Programmable Read-Only Memory 可编程序的只读存储器)芯片成本,很多产品都要求电路板上 FPGA 能够通过 CPU (处理器)进行在线配置。

在通常情况下,一般是把处理器的 I/O (输入输出)管脚和 FPGA 的串行配置管脚直接相连,通过软件模拟 FPGA 的配置时序对 FPGA 进行配置。

但是在以 Intel 的 X86 架构为处理器平台的产品中,处理器并没有通用的 I/O 管脚可以和 FPGA 相连,虽然可以通过桥片的 I/O 和 FPGA 相连,并通过软件控制桥片的 I/O 模拟配置时序对 FPGA 进行下载,但是这样不仅占用桥片的 I/O 资源,灵活性差(因为需要桥片的固定 I/O 管脚),通用性不好(因为不同桥片的 I/O 定义不一定一样),而且配置速度慢,软件的复杂度增加,可靠性降低。

针对目前 FPGA 在线加载方法的不足,可以采用通用的总线接口来代替桥片的固定 I/O 管脚来对 FPGA 进行配置。现在大多数的桥片都有 LPC 总线接口,用来代替 ISA (Industry Standard Architecture 工业标准结构)接口,LPC 总线需要的信号线比 ISA 大大减少,只需要 6 根信号线。

发明内容:

本发明要解决的问题,是提供一种通过 LPC 总线对 FPGA 进行配置的接口转换模块,并提供用该接口转换模块通过 LPC 总线对 FPGA 进行配置的方法。

本发明中的接口转换模块,包括 LPC 总线接口部分、命令寄存器、数据寄存器、状态寄存器和 FPGA 配置时序生成部分;所述 LPC 总线接口部分完成 LPC 总线读写寄存器组的时序;所述数据寄存器暂存一个字节的 FPGA 配置数据;所述命令寄存器控制由处理器通过 LPC 总线写入的操作命令;所述状态寄存器用来查询该接口转换模块工作状态;所述 FPGA 配置时序生成部分完成串行配置时序,将数据寄存器中的配置数据转

换成串行配置数据流下载到 FPGA 中。

本发明中的对 FPGA 进行配置的方法，包括：

处理器经过 LPC 总线发启动配置命令给接口转换模块，接口转换模块接收到启动命令，开始生成 FPGA 下载的初始化时序并对 FPGA 进行初始化；若初始化时序已经完成，可以跳过这一步；

处理器通过 LPC 总线送配置数据给接口转换模块，接口转换模块接收到配置数据后，生成 FPGA 下载时序，将配置数据下载到 FPGA 中；配置数据的过程中，每次配置一个字节的配置数据；

处理器通过 LPC 总线发配置结束命令给接口转换模块，接口转换模块接收到配置结束命令后，开始生成 FPGA 配置过程所需要的配置结束时序；或者接口转换模块在数据配置完成以后，自动完成 FPGA 配置过程所需要的配置结束时序。

在本方法中，在初始化、配置数据和配置结束的每一步骤后，都有检测程序，检测对应的操作是否成功；前一操作成功才进行下一操作。检测程序是否成功是通过读接口转换模块中的状态寄存器来实现的。

与现有技术相比：1、由于软件不再需要模拟 FPGA 的配置时序，大大提高了 FPGA 配置速度和可靠性；2、使用标准的 LPC 总线做为接口，模块兼容性很好，采用这种方法设计的软件和硬件可以使用在带 LPC 总线的各种型号的处理系统，提高了硬件系统设计的灵活性。

附图说明：

图 1 为本发明应用环境的系统架构框图；

图 2 为本发明中的接口转换模块内部结构框图；

图 3 为 LPC 接口部分状态机转换图；

图 4 为 FPGA 配置时序生成部分状态机转换图；

图 5 为本发明中的方法流程图。

具体实施方式：

下面结合附图对技术方案的实施作进一步的详细描述：

如图 1 所示，本发明的应用环境系统结构框图，该系统包括：

处理器小系统 101，可以包括处理器、内存、桥片和 BIOS 等，是一个可以运行程序的独立的处理器系统，可以运行操作系统和本发明中的配置程序。

接口转换模块 102，本模块作为一个 LPC 总线装置挂在 LPC 总线上，完成 LPC 总

线到 FPGA 串行配置的时序转换功能，处理器通过本模块来完成对 FPGA 的在线配置。该硬件模块内部结构框图如图 2 所示，主要包括 LPC 总线接口部分 201、寄存器组（202-204）和 FPGA 配置时序生成部分 205。LPC 总线接口部分主要完成 LPC 总线读写寄存器组的时序；数据寄存器 203 暂存一个字节的 FPGA 配置数据；FPGA 配置时序生成部分完成串行配置时序，将数据寄存器中的配置数据转换成串行配置数据流下载到 FPGA103 中；命令寄存器 202 和状态寄存器 204 主要提供给软件一个控制和检测此模块状态的接口。

接口转换模块 102 内部各部分工作原理和流程说明如下：

接口转换模块 102 内部 LPC 总线接口部分 201 设计的状态机流程图如图 3 所示。状态机开始处于起始状态 301，检测到帧同步信号以后进入下一个状态获得交易类型 302，接下来的几个周期获取地址 303，如果地址正确，并且是写交易，那么就从 LPC 总线上接收数据 307，然后经过总线同步 308 和驱动转换 309 以后，交易完成。如果是读交易，那么经过总线驱动转换 304 和总线同步 305 以后，送状态寄存器数据到 LPC 总线上 306，然后经过总线驱动转换 309 后，总线交易完成。

接口转换模块 102 内部命令寄存器 202 详细说明举例，如下表所示：

表一

| 状态 | 意义 |
|------|--|
| 状态 1 | 启动 FPGA 配置（告诉模块进行 FPGA 配置初始化） |
| 状态 2 | 配置数据结束（告诉模块所有的配置数据都已经传输完毕，模块开始监测 FPGA 侧配置完成信号） |
| 其他 | 未定义 |

接口转换模块 102 内部状态寄存器 204 详细说明举例，如下表所示：

表二

| 状态 | 意义 |
|------|-------------------------------|
| 状态 1 | 初始化错误（没有监测到 nstatus） |
| 状态 2 | 初始化成功（表示初始化成功，告诉软件可以开始送数据了） |
| 状态 3 | 字节配置过程中出现位错误 |
| 状态 4 | 一个字节数据传输完成（告诉软件可以送下一个字节的数据了）。 |
| 状态 5 | 配置成功 |
| 状态 6 | 配置失败 |
| 状态 7 | 模块正在监测配置完成信号（供查询用，非必要） |
| 状态 8 | 模块正在进行字节数据配置过程（供查询用，非必要） |
| 其他 | 未定义 |

接口转换模块 102 内部 FPGA 配置时序生成部分 205 状态机图如图 4 所示。FPGA 配置接口以从串配置(Passive Serial Configuration)接口为例。状态机开始处于起始状态 401, 接收到启动命令后, 进入初始化配置 402, 完成初始化时序, 检测 FPGA 初始化状态是否成功 403 并写相应的状态寄存器 404; 然后开始数据配置, 包括接收配置数据 406、生成数据配置时序 407 和检测配置完成状态 408; 如果所有的数据配置完成 405, 延时相应的时间 409, 检测 FPGA 配置完成信号 410, 如果检测到配置完成信号, 配置过程成功。

本发明通过接口转换模块 102 配置 FPGA 的方法流程图如图 5 所示(命令寄存器 202 和状态寄存器 204 的具体编码以及对应含义见表一和表二):

步骤 1: 首先向命令寄存器 202 写入启动命令。

步骤 2: 延时一段时间后读模块状态寄存器 204, 判断初始化状态。如果初始化错误, 进入错误处理程序, 配置失败。

步骤 3: 如果初始化成功, 则往数据寄存器 203 中写入配置数据。

步骤 4: 读模块状态寄存器 204, 判断在这个字节数据配置过程中有没有位错误。如果有位错误, 进入错误处理程序, 配置失败。

步骤 5: 如果没有位错误, 继续往数据寄存器 203 中写入下一个数据; 如此反复, 直到所有数据配置完毕。

步骤 6: 向命令寄存器 202 写入配置数据结束命令。

步骤 7: 延时一段时间后读模块状态寄存器 204, 判断配置完成状态。如果配置失败, 进入配置错误处理程序, 配置失败。

步骤 8: 如果状态寄存器 204 的状态为配置成功, 则表示 FPGA 配置成功, 整个配置过程结束。

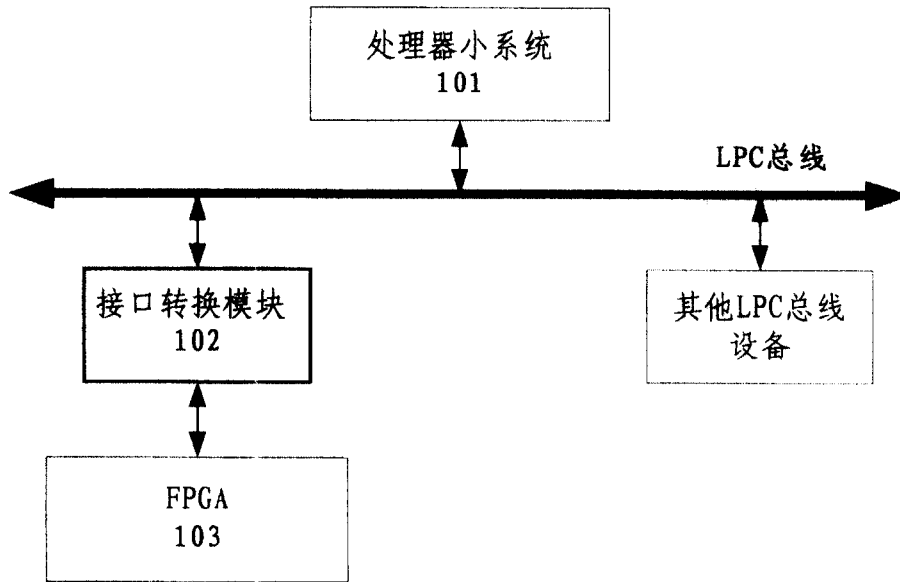


图 1

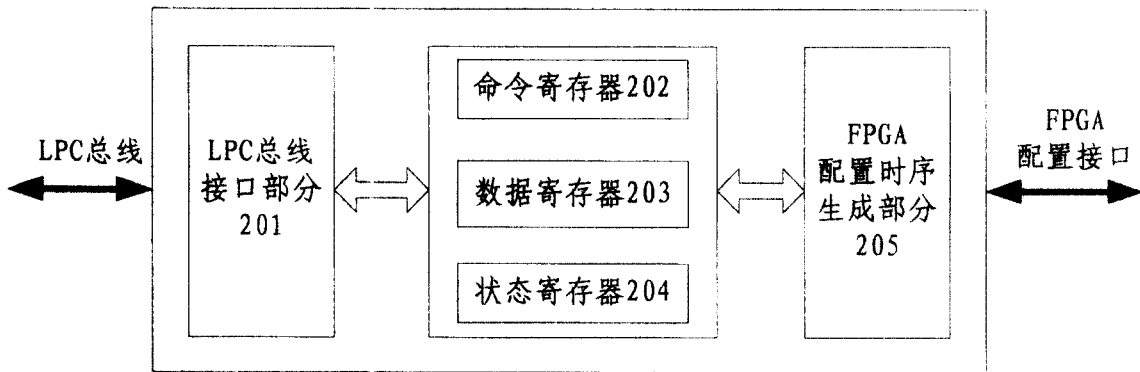


图 2

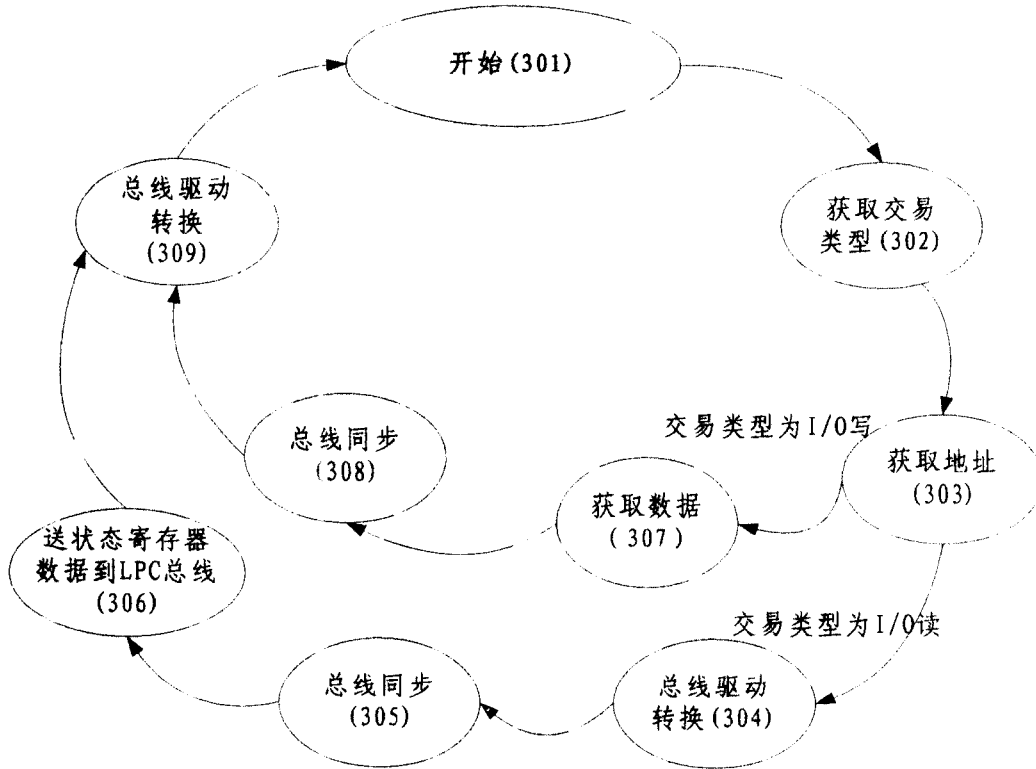


图 3

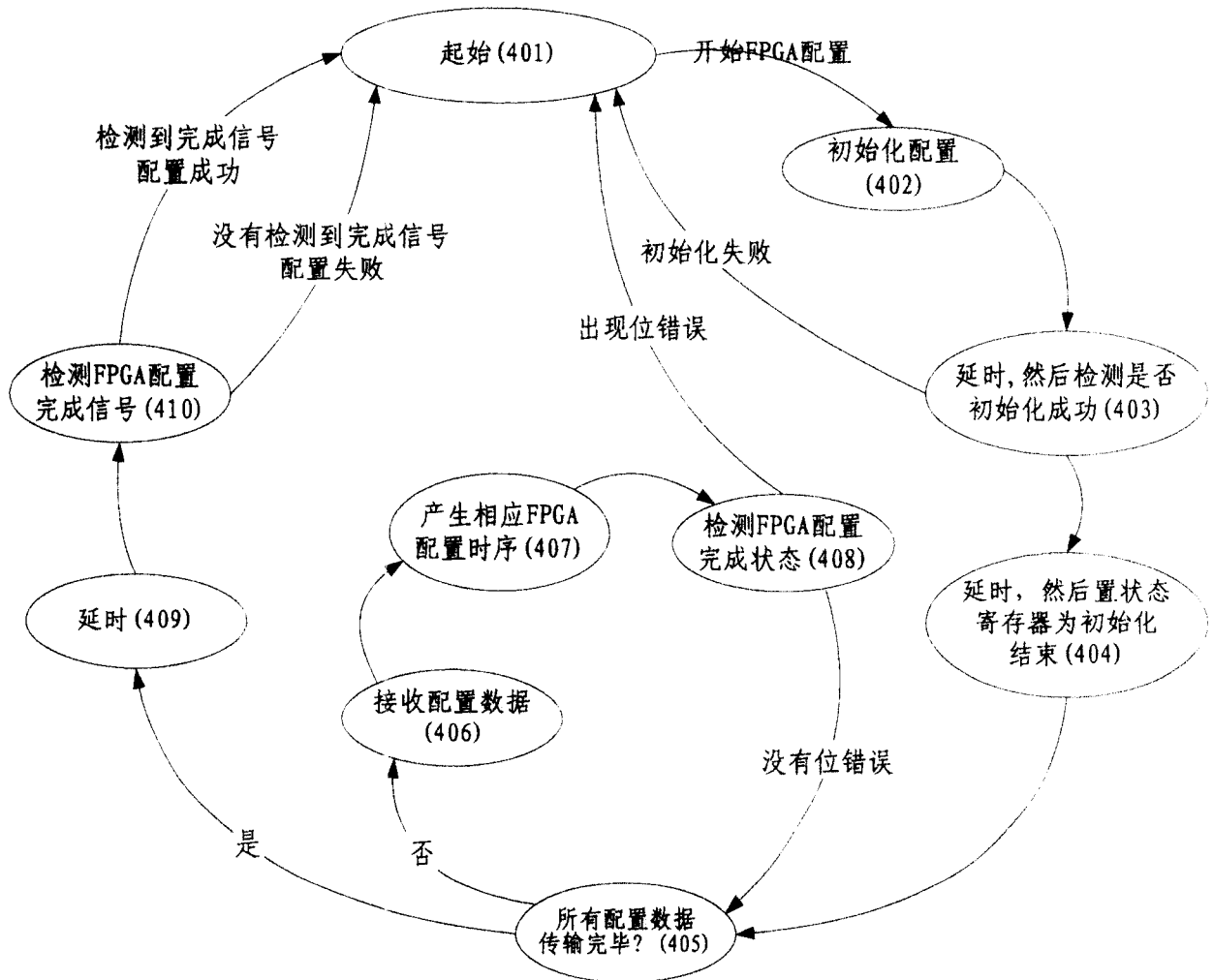


图 4

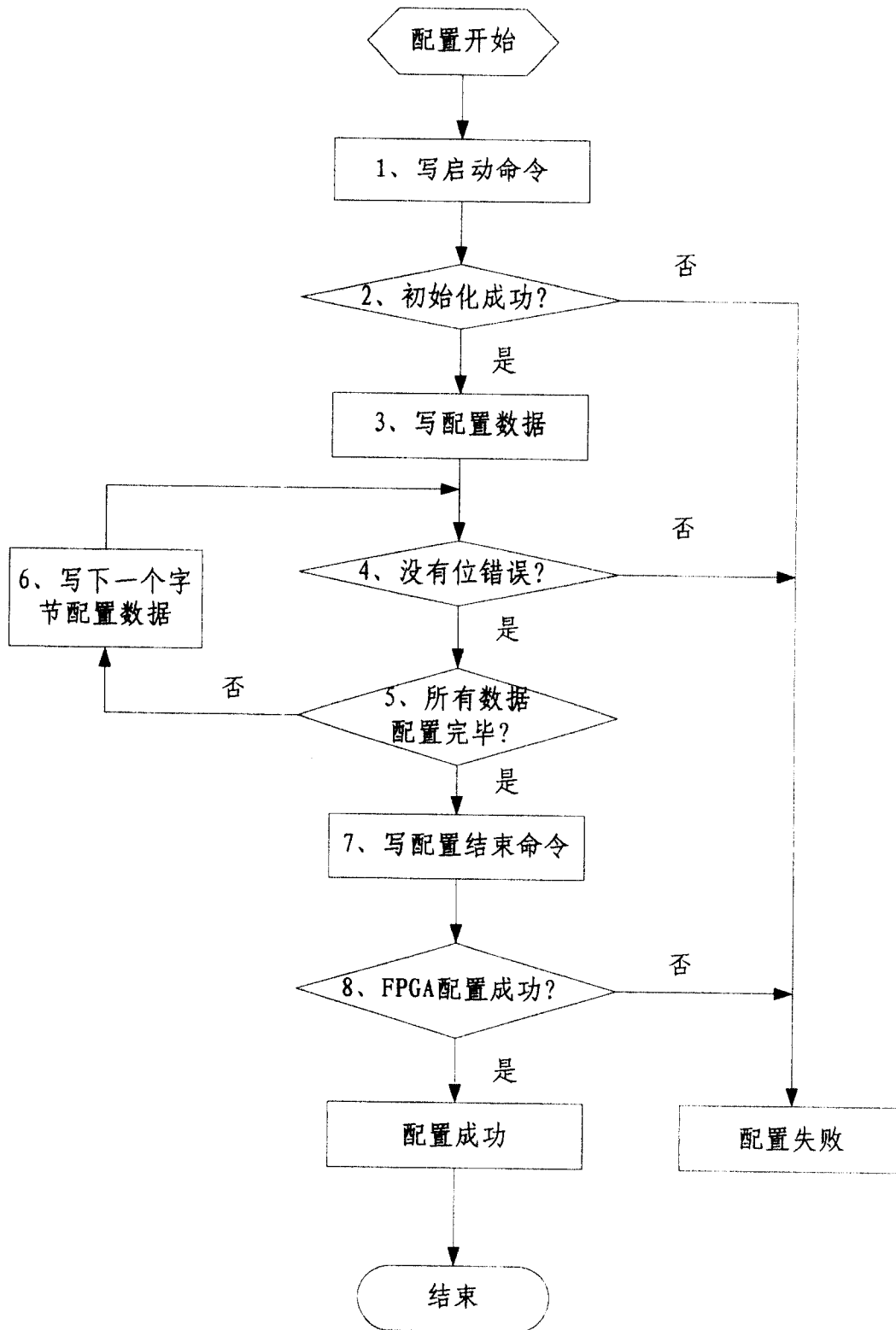


图 5