

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 12/00 (2006.01)

G06F 13/14 (2006.01)



[12] 发明专利说明书

专利号 ZL 03137543.X

[45] 授权公告日 2007 年 1 月 24 日

[11] 授权公告号 CN 1296830C

[22] 申请日 2003.6.8 [21] 申请号 03137543.X

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 李庆东

[56] 参考文献

US 5497497 A 1996.3.5 G06F 9/40

CA 2069972 A1 1993.3.1 G06F 1/24

审查员 李 科

[74] 专利代理机构 中原信达知识产权代理有限责任公司

代理人 王永安

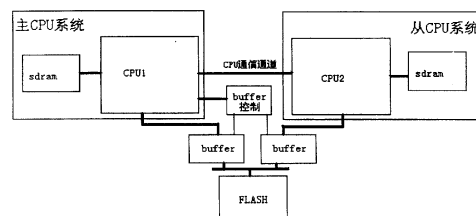
权利要求书 3 页 说明书 8 页 附图 3 页

[54] 发明名称

中央处理单元启动的方法及系统

[57] 摘要

一种中央处理单元启动的方法及系统，在多中央处理单元系统中，利用同一个存储器存放各个中央处理单元的引导程序，各个中央处理单元与该存储器之间的数据/地址总线分别通过缓冲器进行隔离；上电或整个系统复位后，主中央处理单元使得各从中央处理单元处于复位状态而无法启动，主中央处理单元给出片选信号选通对应的缓冲器及存储器，正常启动；主中央处理单元启动完毕或启动到某个步骤后，解除从中央处理单元的复位；从中央处理单元给出片选信号选通对应的缓冲器及存储器，正常启动。本发明相对于现有的从不同 BOOTROM 启动而言节省了成本，由于各个 CPU 都可以访问 flash，可以很方便地实现对所有 CPU 的软件进行升级。



1、一种中央处理单元启动的方法，其特征在于：在多中央处理单元系统中，利用同一个存储器存放各个中央处理单元的引导程序，各个中央处理单元与该存储器之间的数据/地址总线分别通过缓冲器进行隔离；中央处理单元系统启动流程步骤如下：

1) 上电或整个系统复位后，主中央处理单元使得各从中央处理单元处于复位状态而无法启动，主中央处理单元给出片选信号选通与主中央处理单元对应的缓冲器及存储器，正常启动；

2) 主中央处理单元启动完毕或启动到某个步骤后，解除从中央处理单元的复位，并调整主中央处理单元输出的控制信号，使得从中央处理单元可以访问存储器；

3) 从中央处理单元给出片选信号选通该从中央处理单元对应的缓冲器及存储器，正常启动。

2、根据权利要求1所述的中央处理单元启动的方法，其特征在于：所述的存储器采用闪存，各个中央处理单元的引导程序和应用软件在同一个闪存中根据地址划分存放在不同的空间。

3、根据权利要求1所述的中央处理单元启动的方法，其特征在于：从中央处理单元已正常启动或启动到不需要存储器后，主中央处理单元通过调整其输出的控制信号，取消从中央处理单元选通该从中央处理单元对应的缓冲器的权限，防止从中央处理单元误访问存储器。

4、根据权利要求3所述的中央处理单元启动的方法，其特征在于：所述的从中央处理单元已正常启动的信息通过中央处理单元之间的通信通道获得，也可以估算一定的时间，确保该从中央处理单元已正常启

动或已启动到不需要存储器的阶段。

5、根据权利要求1所述的中央处理单元启动的方法，其特征在于：
在各中央处理单元正常运行期间，如果从中央处理单元需要访问存储器，必须通过中央处理单元之间的通信通道向主中央处理单元提出申请，得到允许后，主中央处理单元调整其输出的控制信号，使该从中央
5 处理单元可以选通读取存储器；读取完毕后通知主中央处理单元，主中央处理单元调整其输出的控制信号，取消该从中央处理单元选通读取存储器的权限。

6、根据权利要求1所述的中央处理单元启动的方法，其特征在于：
10 如果存储器的空间较大，某些中央处理单元的地址范围不够时，根据该中央处理单元程序在存储器中存放的地址，在缓冲器之前给高位的存储器地址线做上、下拉处理，将其从外部强制成所需的固定电平，使得该中央处理单元可以访问到分配的存储器空间。

7、一种中央处理单元启动系统，其特征在于：
15 主、从中央处理单元的数据/地址总线分别通过缓冲器隔离后连接到同一存储器；

主中央处理单元的片选信号直接输入到与主中央处理单元对应的缓冲器的使能信号管脚，同时该片选信号在与各从中央处理单元的片选信号相与后输入存储器，控制存储器的选通；

20 从中央处理单元的片选信号在与所述主中央处理单元的片选信号相与之前，由上拉电阻拉高，并被与该从中央处理单元对应的缓冲器隔离；

主中央处理单元复位信号经下拉电阻拉低后输出到各从中央处理单元的复位管脚，同时，该复位信号在进行非运算后，与经上拉电阻拉
25 高的主中央处理单元输出的控制信号及从中央处理单元的片选信号进

行或运算，所产生的信号输入该从中央处理单元对应的缓冲器的使能信号管脚。

8、根据权利要求7所述的中央处理单元启动系统，其特征在于：所述的存储器为具有块保护功能的闪存器件。

- 5 9、根据权利要求7或8所述的中央处理单元启动系统，其特征在于：主、从中央处理单元之间可通过通信通道相连进行通讯。

10、根据权利要求7或8所述的中央处理单元启动系统，其特征在于：各中央处理单元的数据/地址总线还分别连接同步动态存储器或其它存储器件。

中央处理单元启动的方法及系统

技术领域

本发明涉及电数字数据处理，尤其涉及一种中央处理单元（CPU）

5 启动的方法及系统。

背景技术

现在中央处理单元（CPU）启动都有单独的启动系统，特别是在多CPU系统中，每个CPU由各自的启动芯片（BOOTROM）启动，这样的
10 设计方式使用的器件数量多，而且不易实现BOOTROM软件升级，造成系统维护和增强功能的困难。

如图1所示，在现有技术方案中，各个CPU系统（如图中CPU1系统和CPU2系统）都有独立的BOOTROM，系统只能从该CPU对应的BOOTROM启动。由于整个系统中，通常只以某个CPU为主，其他CPU
15 协助处理业务，各CPU之间只进行简单的通信，系统设计中为了降低成本，一般只实现主CPU的BOOTROM升级，而各个从CPU的BOOTROM不升级。如果实现所有BOOTROM的升级，则需要为每个BOOTROM设计加载电路，系统设计复杂度增加，成本上升。

现有技术方案的缺点为：

20 （1）在多CPU系统中，每个CPU通道由各自独立的BOOTROM引导启动，系统中需要使用多个BOOTROM存储器，增加了系统成本。

（2）系统需要升级所有CPU的软件时，为降低成本，只实现主CPU的升级，从CPU不升级。

(3) 实现从CPU升级需要增加系统复杂度, 增加系统成本。

发明内容

本发明所要解决的技术问题是: 克服现有技术各个 CPU 通过独立的
5 BOOTROM 启动所带来的成本高、升级复杂等缺点, 提供一种 CPU 启动的方法和系统, 从而降低系统成本, 并方便实现 CPU 的升级。

本发明为解决上述技术问题所采用的技术方案为:

这种中央处理单元启动的方法, 在多中央处理单元系统中, 利用同一个存储器存放各个中央处理单元的引导程序, 各个中央处理单元与该
10 存储器之间的数据/地址总线分别通过缓冲器进行隔离; 中央处理单元系统启动流程步骤如下:

1) 上电或整个系统复位后, 主中央处理单元使得各从中央处理单元处于复位状态而无法启动, 主中央处理单元给出片选信号选通与主中央处理单元对应的缓冲器及存储器, 正常启动;

15 2) 主中央处理单元启动完毕或启动到某个步骤后, 解除从中央处理单元的复位, 并调整主中央处理单元输出的控制信号, 使得从中央处理单元可以访问存储器;

3) 从中央处理单元给出片选信号选通该从中央处理单元对应的缓冲器及存储器, 正常启动。

20 其中, 所述的存储器可采用闪存, 各个中央处理单元的引导程序和应用软件在同一个闪存中根据地址划分存放在不同的空间。

其中, 从中央处理单元已正常启动或启动到不需要存储器后, 主中央处理单元可以通过调整其输出的控制信号, 取消从中央处理单元选通该从中央处理单元对应的缓冲器的权限, 防止从中央处理单元误访问存
25 储器。

其中，所述的从中央处理单元已正常启动的信息通过中央处理单元之间的通信通道获得，也可以估算一定的时间，确保该从中央处理单元已正常启动或已启动到不需要存储器的阶段。

其中，在各中央处理单元正常运行期间，如果从中央处理单元需要
5 访问存储器，必须通过中央处理单元之间的通信通道向主中央处理单元提出申请，得到允许后，主中央处理单元调整其输出的控制信号，使该从中央处理单元可以选通读取存储器；读取完毕后通知主中央处理单元，主中央处理单元调整其输出的控制信号，取消该从中央处理单元选通读取存储器的权限。

10 其中，如果存储器的空间较大，某些中央处理单元的地址范围不够时，根据该中央处理单元程序在存储器中存放的地址，在缓冲器之前给高位的存储器地址线做上、下拉处理，将其从外部强制成所需的固定电平，使得该中央处理单元可以访问到分配的存储器空间。

相应的一种中央处理单元启动系统，其特征在于：

15 主、从中央处理单元的数据/地址总线分别通过缓冲器隔离后连接到同一存储器；

主中央处理单元的片选信号直接输入到与主中央处理单元对应的缓冲器的使能信号管脚，同时该片选信号在与各从中央处理单元的片选信号相与后输入存储器，控制存储器的选通；

20 从中央处理单元的片选信号在与所述主中央处理单元的片选信号相与之前，由上拉电阻拉高，并被与该从中央处理单元对应的缓冲器隔离；

主中央处理单元复位信号经下拉电阻拉低后输出到各从中央处理单元的复位管脚，同时，该复位信号在进行非运算后，与经上拉电阻拉
25 高的主中央处理单元输出的控制信号及从中央处理单元的片选信号进行或运算，所产生的信号输入该从中央处理单元对应的缓冲器的使能信

号管脚。

其中，所述的存储器可以是具有块保护功能的闪存器件。

其中，主、从中央处理单元之间可通过通信通道相连进行通讯。

其中，各中央处理单元的数据/地址总线还分别连接同步动态存储器
5 或其它存储器件。

本发明的有益效果为：本发明在多 CPU 系统中使用 flash 来放置各 CPU 的引导程序和应用软件，所有 CPU 共用一个 flash，从同一个 flash 启动，相对于现有技术从不同的 BOOTROM 启动而言节省了成本，由于各个 CPU 都可以访问 flash，可以很方便地实现对所有 CPU 的软件进
10 行升级。

附图说明

图 1 为现有的 CPU 启动原理示意图；

图 2 为本发明 CPU 启动原理示意图；

15 图 3 为本发明多个 CPU 从同一个 flash 启动的控制电路图；

图 4 为本发明 CPU 启动流程图；

图 5 为本发明 CPU 地址线范围不足时的处理示意图。

具体实施方式

20 下面根据附图和实施例对本发明作进一步详细说明：

如图2所示为本发明CPU启动原理示意图，本发明使用可读写的存储器（如flash闪存器件）作为载体，取代现有的启动芯片（bootrom），使用flash器件存放CPU软件，所有CPU共用同一个flash，从同一个flash启动。由于CPU启动时读取bootrom的时间很短，程序搬运到同步动态存储
25 器（sdram）后不再访问bootrom，所以可以通过主CPU来控制各个CPU

系统对bootrom按顺序先后访问，达到共享的目的。为防止各CPU随意选取通flash以及隔离各个总线防止干扰，各个CPU系统与flash之间的数据/地址总线、及部分或所有片选信号通过缓冲器（buffer）进行隔离，由主CPU（图中CPU1）对各buffer进行控制，实现各CPU对flash的先后访问。

5 同时主CPU可以在正常运行时选通flash，进行各个CPU软件的升级加载。

如图3所示，以两个CPU系统为例说明本发明的技术方案，图中CPU1为主CPU，CPU2为从CPU。CPU1、CPU2的数字地址总线接口（date_addr_bus）分别经buffer1、buffer2的隔离后与同一flash的数字地址总线接口（date_addr_bus）相接。在buffer和flash分别选通时，相应的
10 CPU可以访问flash，从flash的相应区域读取数据和进行数据搬运到相应的SDRAM，从而启动。

图3中，缓冲器buffer1和buffer2的/G管脚为开关选通信号，低有效。该管脚为高时，相应的CPU和flash的数据/地址总线隔离；该管脚为低时，相应的CPU和flash的数据/地址总线连通。

15 各CPU之间的通信通道在不同的系统中可以不同，可以为RS232、RS485、I2C等串口；也可以为其他接口，如通用测试与操作接口（UTOPIA）等；也可以为自定义的通信接口。

CPU1的片选信号/CS1直接输入buffer1的/G管脚，同时/CS1在与CPU2的片选信号/CS2相与（and）后形成flash的选通信号/CS3。

20 CPU2的片选信号/CS2在做与运算送给flash之前，通过buffer2隔离，防止CPU2不取得flash控制权时选通flash。/cs2_buf的上拉电阻保证上电期间和buffer2隔离时，该信号为高状态，不会误选通flash。

CPU1的复位输出信号/reset_out连到CPU2的复位管脚/reset，控制CPU2的复位状态。如果有多个从CPU，每个从CPU各需要一个与主CPU
25 相连的复位信号，不能共用，复位信号低有效。

CPU1的control信号先和/reset_out的非（not）进行或运算（or）后产

生信号CPU1_ctr，再将CPU1_ctr与/cs2进行或运算后输出CPU2_OE信号到buffer2的/G管脚，CPU2_OE作为使能信号控制buffer2的通断，防止CPU2随意选通buffer2，访问flash。平时由于上拉电阻的上拉，control信号为高，不允许CPU2选通buffer2。允许CPU2选通buffer2时，control
5 信号为低。上拉电阻保证上电期间control信号为高状态。

这样，当系统启动时，各CPU的启动过程如下（假设以下器件的控制信号都为低有效）：

上电后或整个系统复位后，CPU1先正常启动，CPU1的/reset_out管脚输出低电平，使得CPU2的复位输入为低（同时该管脚被拉低，保证
10 CPU1启动初期也能为低电平），CPU2处于复位状态，无法启动；同时/reset_out经过非和或运算后使得CPU2_OE为高，关闭buffer2。CPU1正常启动，给出/CS1低信号，/CS1输入buffer1的/G管脚选通buffer1，同时/CS1与/CS2_buf相与产生低信号/CS3送入flash，选取通flash，取得flash的读写权限，从flash的相应区域读取数据和进行数据搬运到SDRAM。

CPU1启动完毕或者启动到某个步骤后（根据实际情况决定），
15 /reset_out管脚拉高，使得CPU2走出复位状态，正常启动，同时control信号保持为低电平，使得CPU2可以控制buffer2的选通。CPU2正常启动，给出/CS2低信号，输入/G管脚的信号为低，buffer2选通，输出/CS2_buf为低，从而选通flash，取得flash的读写权限，从flash的相应区域读取数据和进行数据搬运到SDRAM。
20

在CPU2启动期间，CPU1必须保证不访问flash，以免发生冲突。CPU2启动完毕的信息可以通过CPU之间的通信通道获得，也可以估算一定的时间，确保CPU2已正常启动或已启动到不需要flash的阶段（CPU从flash启动的时间通常很短，几秒钟之内可以完成基本启动，不再需要从flash
25 读取数据）。确保CPU2不再需要flash后，CPU1的control管脚拉高，取消CPU2选通buffer2的权限，防止两个CPU同时访问flash，造成冲突，损坏

器件。

在正常运行期间，如果CPU2需要访问flash，必须通过CPU之间的通信通道提出申请，得到允许后同时CPU1的control管脚拉低，CPU2可以选通读取flash。读取完毕后通知CPU1，CPU1的control管脚拉高，取消CPU2选通buffer2的权限。如果正常运行期间CPU2不需要访问flash，没有本操作。

图3中只给出了三个片选信号：/cs1为主CPU给flash的片选信号（输出），/cs2为从CPU给flash的片选信号（输出），/cs3为flash的选通信号（输入）。

当从CPU不只CPU2一个时，同样根据control、/reset_out和从CPU给出的片选信号作为使能信号输入相应buffer的/G管脚来控制该buffer的通断，并通过该从CPU给出的片选信号与主CPU给出的片选信号相与来控制flash的通断，从而实现该从CPU对flash的访问。为了安全起见，也可以增加主CPU的control信号来控制相应的从CPU，达到主CPU可以单独控制各个从CPU的buffer通断。

如图4所示为本发明CPU启动流程图，启动流程概括如下：

1) 上电或整个系统复位后，主CPU使得各从CPU处于复位状态而无法启动，主CPU给出片选信号选通主CPU与flash之间的buffer及flash，正常启动；

2) 主CPU启动完毕或启动到某个步骤后，解除从CPU的复位，并把control信号拉低，使得从CPU可以访问flash；

3) 从CPU给出片选信号选通该从CPU与flash之间的buffer及flash，正常启动，在此期间主CPU不能访问flash；

4) 确保从CPU不再访问flash后，主CPU把control信号拉高，防止从CPU误访问flash；

5) 主CPU和从CPU完成其它启动操作；

6) 主CPU和从CPU正常运行。

本系统中的flash空间要进行划分,给各个CPU划分足够的地址空间,放置软件和其他信息。由于各个CPU都可以访问flash,可以很方便的实现在线加载各个软件,升级各个CPU的软件。

- 5 如果flash的空间较大,某些CPU的地址范围不够时,可以采用在buffer之前给高位的flash地址线做上、下拉处理,通过上、下拉处理后,从外部将高位的地址线强制成所需的固定电平,使得该CPU可以访问到分配的flash空间。如图5所示,当CPU地址线为A0—A18,共19根,而flash地址线为A0—A20,共21根时,图4圈中部分采用上、下拉处理,根据该
- 10 CPU在flash中存储的地址,在buffer之前将A19强制为低电平,A20强制为高电平,从而在buffer选通时,该CPU可以访问到分配的flash空间(0x100000-0x17FFFF的512k空间分配给该从CPU使用)。

如果某些CPU的数据线宽不够,比如flash为16bit的数据线,而从CPU1为8位的数据线时,则该CPU只能访问8位对应的flash空间。

- 15 本发明节省了成本,降低了系统复杂度,同时可以很方便的实现对所有CPU的软件进行升级,只要能访问flash器件的CPU都能升级软件。通常主CPU能够访问flash器件的所有空间,同时它能够决定flash器件的使用权限,由它来升级软件比较简单方便。需要注意的是,由于基本输入输出系统(BIOS)软件部分软件比较重要,可选用具有块保护功能的
- 20 flash器件,从而可在平时使得各个CPU的BIOS软件空间处于保护状态,防止误操作。

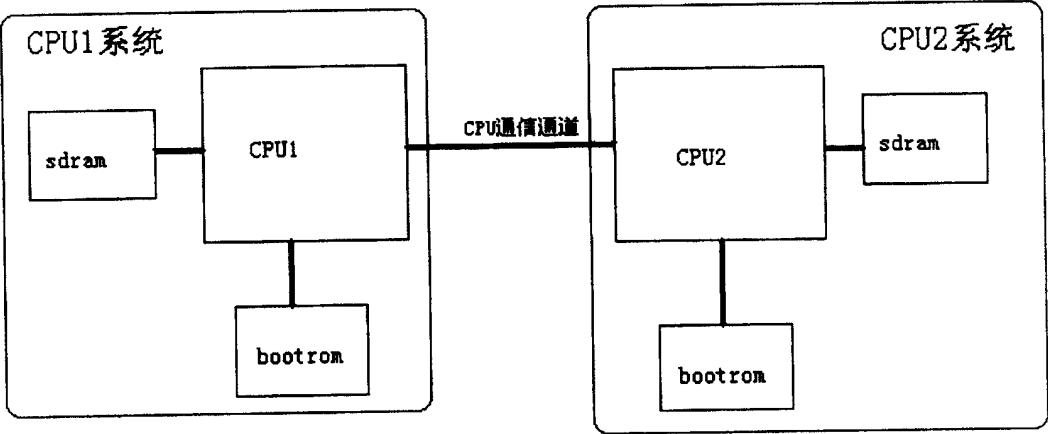


图1

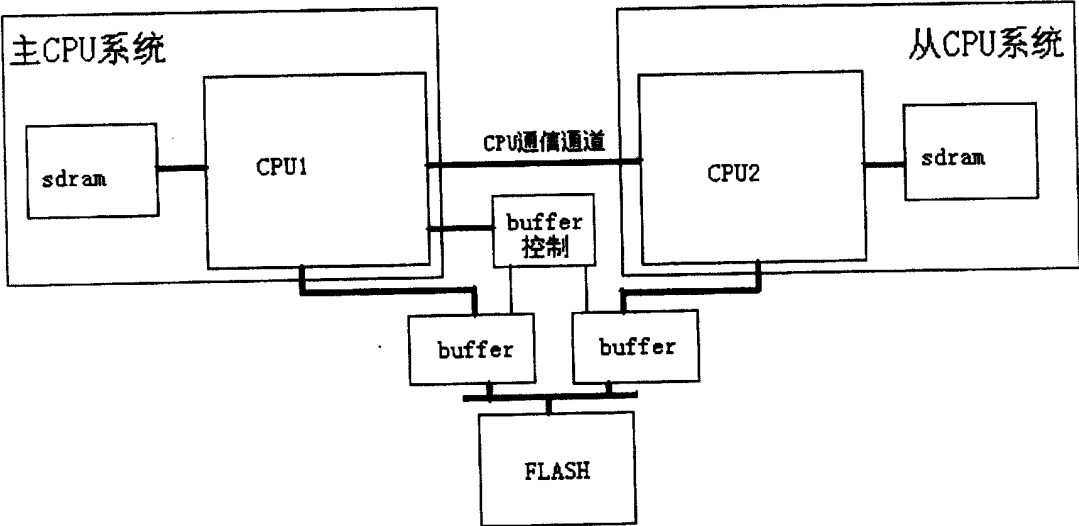


图2

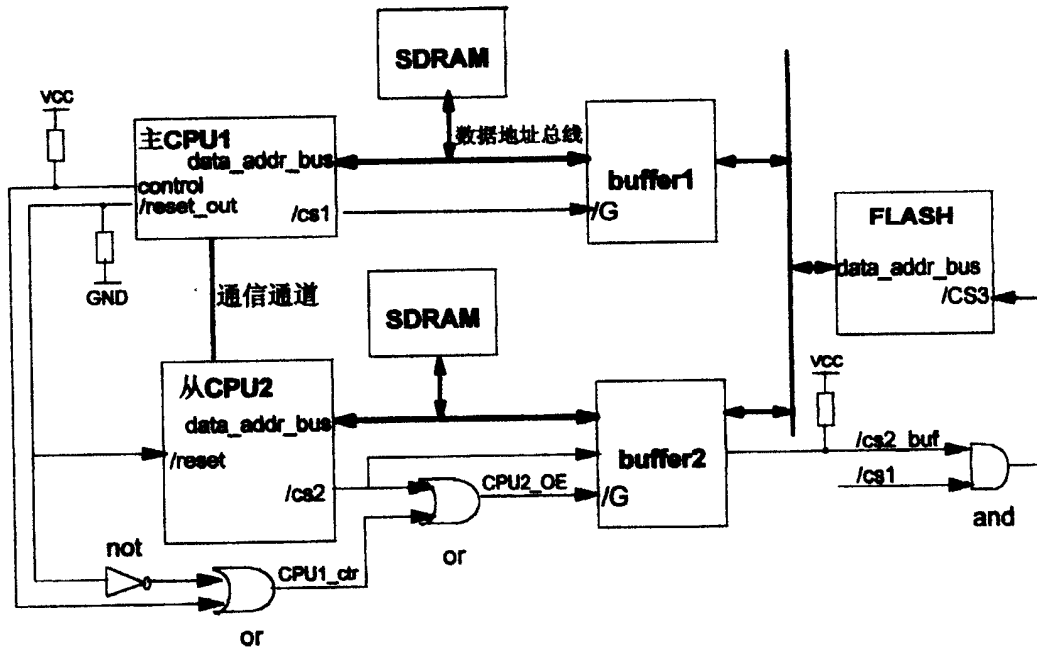


图3

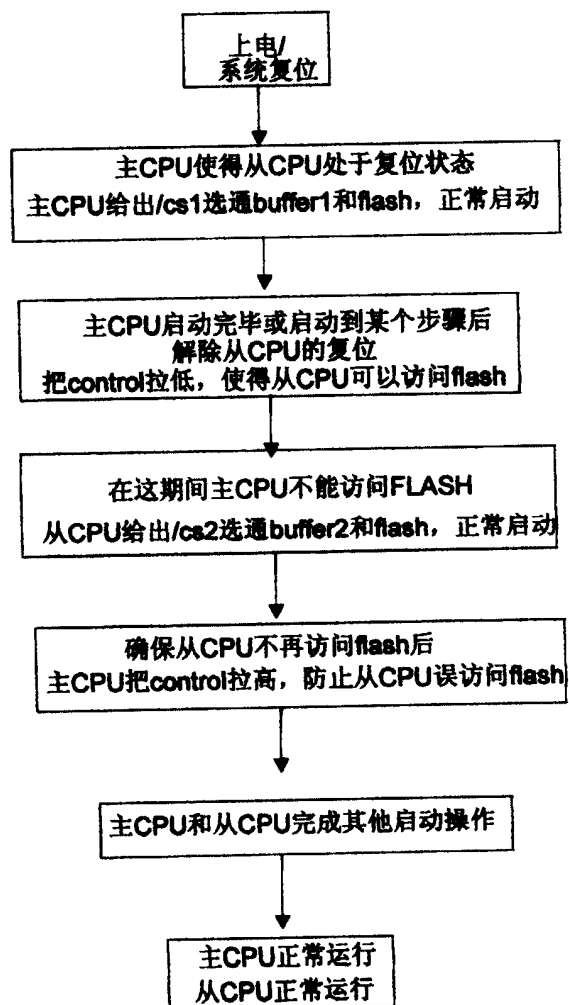


图4

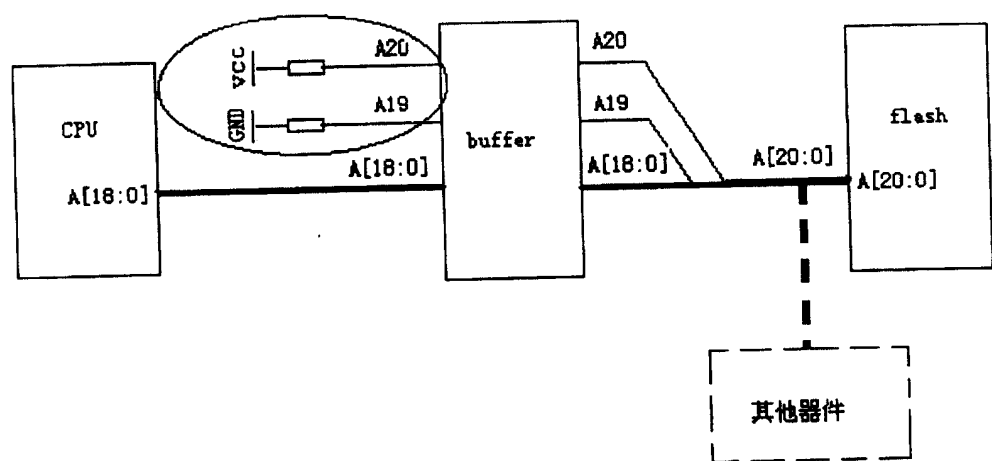


图5