

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.



[12] 发明专利说明书

G06F 17/50 (2006.01)

G06F 11/14 (2006.01)

G11C 29/00 (2006.01)

专利号 ZL 200510030176.2

[45] 授权公告日 2009年4月22日

[11] 授权公告号 CN 100481091C

[22] 申请日 2005.9.29

[21] 申请号 200510030176.2

[73] 专利权人 上海奇码数字信息有限公司

地址 201203 上海市张江高科技园区碧波路328号B栋4楼

[72] 发明人 刘彦 周振亚

[56] 参考文献

US5086502A 1992.2.4

US2002/0093691A1 2002.7.18

CN1627267A 2005.6.15

CN1530841A 2004.9.22

审查员 庄锦军

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 张政权

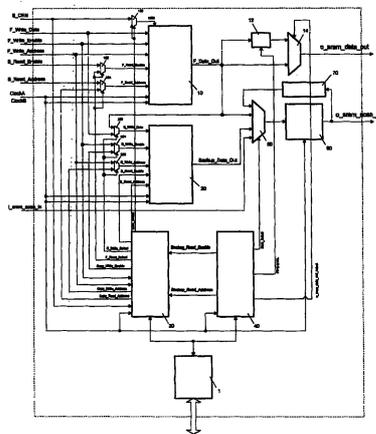
权利要求书2页 说明书11页 附图3页

[54] 发明名称

快照装置中备份存储器的动态更新方法、模块以及系统

[57] 摘要

提供了一种快照装置中备份静态随机存取存储器的动态更新方法，包括把对主存储器的写操作同时复制到备份存储器的步骤；以及利用主存储器的空闲期间，把主存储器的内容复制到备份存储器的步骤。还提供一种静态随机存取存储器快照模块和包含该模块的仿真系统，所述模块包括主存储器、备份存储器以及输出模块，所述静态随机存取存储器快照模块还包括缓存及控制模块，所述的缓存分别与主存储器的输出端及输出模块的一个输入端连接，所述控制模块控制所述缓存的数据锁定以及控制所述主存储器和备份存储器执行以下操作：对主存储器的写操作同时被复制到备份存储器；在主存储器的空闲期间，主存储器的内容复制到备份存储器。



1. 一种快照装置中备份静态随机存取存储器的动态更新方法，所述方法包括以下步骤：

把对主存储器的写操作同时复制到备份静态随机存取存储器的步骤；

利用主存储器的空闲期间，把主存储器的内容复制到备份静态随机存取存储器的步骤。

2. 如权利要求1所述方法，其特征在于，利用主存储器的空闲期间把主存储器的内容复制到备份静态随机存取存储器的步骤包括以下步骤：

a) 对主存储器进行读操作；

b) 判断由所述读操作读取的主存储器数据是否来得及写入备份静态随机存取存储器；

c) 如果步骤b)的判断结果为是，则把所述读操作读取的主存储器数据写入备份静态随机存取存储器，然后回到步骤a)，准备复制下一个数据；

d) 如果步骤b)的判断结果若为否，则把该数据暂时存储于附加的缓存中；

e) 在步骤d)暂时存储数据的期间，判断对主存储器存储该数据的地址是否进行了写操作，如果是，则转到步骤a)，准备复制下一个数据，如果否，则把该数据写入备份静态随机存取存储器，然后回到步骤a)，准备复制下一个数据。

3. 如权利要求1或2所述的方法，其特征在于，所述主存储器的空闲期间是系统不对主存储器进行任何操作的期间。

4. 一种静态随机存取存储器快照模块，包括主存储器、备份存储器以及输出模块，备份存储器的输出端与输出模块的一个输入端连接，其特征在于，所述静态随机存取存储器快照模块还包括缓存以及控制模块，所述的缓存的输入端与主存储器的输出端连接，缓存的输出端与输出模块的一个输入端连接，所述控制模块控制所述缓存的数据锁定以及控制所述主存储器和备份存储器执行以下操作：

对主存储器的写操作同时被复制到备份存储器；

在主存储器的空闲期间，主存储器的内容复制到备份存储器。

5. 如权利要求4所述的静态随机存取存储器快照模块，其特征在于，所述

控制模块包括更新控制模块，在所述主存储器进行写操作时，所述更新控制模块将所述写操作复制到备份存储器，

在所述主存储器的空闲期间，所述更新控制模块对主存储器进行读操作，并将读出的数据写入备份存储器，

如果在所述主存储器的下一次写操作之前，所述更新控制模块来不及把由所述读操作得到的数据写入备份存储器，所述更新控制模块就把该数据暂时保存在更新控制模块中，

在该数据的暂时保存期间，若所述数据在所述主存储器内的地址进行了写操作，则所述更新控制模块取消所述数据的复制操作。

6. 一种包含如权利要求 4 或 5 所述的静态随机存取存储器快照模块的硬件仿真系统。

快照装置中备份存储器的动态更新方法、模块以及系统

技术领域

本发明涉及一种快照装置中备份静态随机存取存储器的动态更新方法，是结合复制系统对主存储器的写操作与复制主存储器的内容两个措施实现动态更新，最终能够使快照与硬件仿真互不干扰同时进行。

背景技术

SoC (system on a chip: 片上系统) 技术已经成为当今超大规模集成电路的发展趋势, 它将原来由许多芯片完成的功能集中到一块芯片中完成。但 SoC 不是各个芯片功能的简单叠加, 而是从整个系统的功能和性能出发, 用软硬结合的设计和验证方法, 利用 IP 复用及深亚微米技术, 在一个芯片上实现复杂的功能。虽然 SoC 技术可为设计更加复杂的产品提供高效的解决方案, 缩短微电子产品的上市时间, 但由于 SoC 芯片的规模较大、功能复杂, 增加了设计过程中的仿真验证工作的难度。仿真一般有软件仿真和硬件仿真, 软件仿真具有电路状态全可见的特点, 而硬件仿真具有速度快的优势。因此, 目前的验证一般采用硬件仿真与软件仿真结合的方式进行。首先, 利用硬件仿真系统导出发生错误前某一时刻的电路状态并将其导入软件仿真系统; 然后利用软件仿真系统利用导入的该时刻的电路状态重现该错误并对其进行具体分析。这种硬件仿真与软件仿真结合的验证方案中, 测试项目的大部分是在硬件仿真系统上运行的, 只有发生错误的时刻的附近的一小部分在软件仿真系统上运行, 所以它兼具硬件仿真的高效性与软件仿真的彻底性的优点。

从硬件仿真系统中导出某一时刻电路状态的过程称为 snapshot (以下简称“快照”)。目前有两种实现快照的方式: 一种是硬件仿真系统在运行一个测试项目的过程中只进行一次快照 (以下称单次快照 (single-snapshot)), 另一种是硬件仿真系统在运行一个测试项目的过程中每间隔一定的时间就进行一次快照 (以下称多次快照 (multi-snapshot))。单次快照的前提是得知

发生错误的时间点，在发生错误前的某一恰当的时刻进行快照。在单次快照方案中，若硬件仿真系统在运行某一测试项目的过程中发生多次错误，为了在软件仿真系统上逐一重现这些错误，就要在每一错误发生前的某一时刻进行快照，这样就要求硬件仿真系统运行该测试项目 $n+1$ 次（ n 为发生错误的次数）。若采用多次快照方案，硬件仿真系统只需要运行测试项目一次就能获得多个间隔一定时间的快照。当要在软件仿真系统上重现某一错误时，只要选择该错误发生前某一时刻的快照即可。往往大规模集成电路在一个测试项目的仿真过程中很可能发生多个错误，因此，多次快照方案更适合大规模集成电路的仿真验证。

然而，现有的多次快照方案是在预定的时间点暂停硬件仿真，在该暂停期间进行快照，快照结束后继续硬件仿真。由于暂停，因此这种多次快照方案中仿真与快照不能同时进行，所以效率比较低。

电路状态包括寄存器的值与静态随机存取存储器（SRAM）的内容。实现多次快照最理想的方式是采用备份寄存器以及备份静态随机存取存储器存储电路状态。在不进行快照时，这些备份寄存器以及备份静态随机存取存储器和与被测设计对应的寄存器以及静态随机存取存储器保持一致，快照前冻结所述的备份寄存器以及备份静态随机存取存储器，然后通过扫描电路把备份寄存器以及备份静态随机存取存储器中的数据导出，这样就可实现快照与硬件仿真同步进行，互不影响。由于电路状态随着硬件仿真的进行不停地变化，在进行一次快照后，所述备份寄存器以及备份静态随机存取存储器的值可能已经和与被测设计对应的寄存器以及静态随机存取存储器的值不同了，因此我们可以在一个时钟单位内使备份寄存器和与被测设计对应的寄存器的值再次保持一致（因为利用一个时钟单位就能够对寄存器的值进行改写），但却无法使备份静态随机存取存储器和与被测设计对应的静态随机存取存储器的值重新保持一致，这是进行快照的同时不影响硬件仿真效率的最大的技术难点。

台湾大学电机工程研究所研究生吕东荣的论文——《A Snapshot Method to Provide Full Visibility for Functional Debugging Using FPGA》中提出了多次快照的想法，但却并没有提出恢复备份静态随机存取存储器的解决方案，因此，仍然无法在设有静态随机存取存储器的设计上实现硬件仿真与多次

快照的同步进行。

发明内容

因此，在本领域中亟需一种能够在短时间内使备份静态随机存取存储器和与被测设计对应的静态随机存取存储器重新保持一致的方案，以实现硬件仿真与多次快照同步进行，从而进一步提高大规模集成电路的仿真验证效率。

本发明提供一种快照装置中备份静态随机存取存储器的动态更新方法，所述方法包括以下步骤：

把对主存储器的写操作同时复制到备份存储器的步骤；

利用主存储器的空闲期间，把主存储器的内容复制到备份存储器的步骤。

如上所述方法中，利用主存储器的空闲期间把主存储器的内容复制到备份存储器的步骤包括以下步骤：

a) 对主存储器进行读操作；

b) 判断是否可将由所述读操作读取的主存储器数据写入备份存储器；

c) 如果步骤 b) 的判断结果为是，则把所述读操作读取的主存储器数据写入备份存储器，然后回到步骤 a)，准备复制下一个数据；

d) 如果步骤 b) 的判断结果若为否，则把该数据暂时存储于附加的缓存中；

e) 在步骤 d) 暂时存储数据的期间，判断主存储器存储该数据的地址是否被改写，如果是，则转到步骤 a)，准备复制下一个数据，如果否，则转到步骤 c)。

在如上所述的方法中，所述主存储器的空闲期间是系统不对主存储器进行操作的期间。

本发明还提供一种静态随机存取存储器快照模块，包括主存储器、备份存储器以及输出模块，备份存储器的输出端与输出模块的一个输入端连接，其特征在于，所述静态随机存取存储器快照模块还包括缓存以及控制模块，所述的缓存的输入端与主存储器的输出端连接，缓存的输出端与输出模块的一个输入端连接，所述控制模块控制所述缓存的数据锁定以及控制所述主存储器和备份存储器执行以下操作：

对主存储器的写操作同时被复制到备份存储器；

在主存储器的空闲期间，主存储器的内容复制到备份存储器。

如上所述的静态随机存取存储器快照模块，所述控制模块包括更新控制模块，在所述主存储器进行写操作时，所述更新控制模块将所述写操作复制到备份存储器，

在所述主存储器的空闲期间，所述更新控制模块对主存储器进行读操作，并将读出的数据写入备份存储器，

如果在所述主存储器的下一次写操作之前，所述更新控制模块来不及把由所述读操作得到的数据写入备份存储器，所述更新控制模块就把该数据暂时保存在更新控制模块中，

在该数据的暂时保存期间，若所述数据在所述主存储器内的地址进行了写操作，则所述更新控制模块取消所述数据的复制操作。

本发明还提供一种包含如上所述的静态随机存取存储器快照模块的硬件仿真系统。

附图说明

图 1 为本发明静态随机存取存储器快照模块的硬件框图；

图 2 根据本发明的一方面的主存储器或者备份存储器输出数据与扫描寄存器内的存储位的对应关系；

图 3 为根据本发明的一方面的主存储器或者备份存储器输出数据与扫描寄存器内的存储位的对应关系，其中前一静态随机存取存储器快照模块的扫描输出与扫描寄存器的第八位连接；

图 4 例示出本发明的静态随机存取存储器快照模块的工作时序。

具体实施方式

请参阅图 1，本发明的静态随机存取存储器快照模块包括存储器快照状态机 1、主存储器 10、备份存储器 20、更新控制模块 30、快照控制模块 40、第一复用模块 50、第二复用模块 14、扫描寄存器 60 以及扫描逻辑模块 70。其中，所述的主存储器 10 与备份存储器 20 均为两端口（Two Port）的静态随机存取存储器。主存储器 10 用于被测设计，备份存储器 20 则用于快照。

为减少引脚占用，通过一条扫描链完成所有存储器的快照，所以当前静态

随机存取存储器快照模块需要引入前一静态随机存取存储器快照模块的快照输出 `i_sram_scan_in` 以作为一路快照输出选择。另外，由于主存储器 10 输出端口的值也属于电路状态的一部分，因此需要设置一个缓存 12 保存开始快照的那一刻主存储器 10 输出端口的值，以防止这个值随着系统对主存储器 10 进行的读操作而丢失。缓存 12 的另外一个作用是在更新备份存储器 20 的过程中，读取主存储器 10 的内容复制到备份存储器 20 时，保存读取前主存储器 10 输出端口的值，供系统选择读取。缓存 12 对主存储器 10 输出端口的值的保存由快照控制模块 40 的控制信号 `BUF_CTRL` 控制。主存储器 10 的输出 `F_Data_Out` 和缓存 12 的输出都输入到第二复用模块 14，第二复用模块 14 在快照控制模块 40 输出的控制信号 `o_sram_data_out_select` 的控制下选择两者之一作为当前主存储器的数据输出 `o_sram_data_out`。同时，缓存 12 的输出还输入到第一复用模块 50，作为其一路输出选择，用以输出快照前主存储器 10 的输出端口的值。

包括备份存储器 20 的输出 `Backup_Data_Out`、扫描逻辑模块 70 的输出、缓存 12 的输出以及前述的 `i_sram_scan_in` 在内，当前的静态随机存取存储器快照模块共有 4 路数据供输出接口选择作为扫描输出，该选择是由第一复用模块 50 在快照控制模块 40 的控制信号 `Shift_Select` 的控制下实现的。扫描寄存器 60 和扫描逻辑模块 70 把输出数据转化成串行的格式后作为当前的静态随机存取存储器快照模块的扫描输出 `o_sram_scan_out`。扫描寄存器 60 类似于一个先进先出缓存。

假设系统存储器的位宽是 8 位，扫描寄存器 60 的深度也是 8。主存储器 10 和备份存储器 20 通过第一复用模块 50 与扫描寄存器 60 的连接关系是一位对应一位的连接方式。如图 2 所示，当主存储器 10 或者备份存储器 20 输出一个 8 位的数据 (`sram_in_0~sram_in_7`) 后，这个 8 位的数据便如图示的关系一位对应一位地存储于扫描寄存器 60 中，扫描逻辑模块 70 再通过循环移位以先进先出的方式把这个数据转化成串行的格式输出，这个 8 位的数据串行输出完成后扫描寄存器 60 才能接收下一个 8 位的数据。静态随机存取存储器快照模块的扫描输出是串行的，也就是说 `i_sram_scan_in` 输入的（或 `o_sram_scan_out` 输出的）是串行的数据。为简化时序控制，`i_sram_scan_in`

通过第一复用模块 50 直接与扫描寄存器 60 的第八位连接，如图 3 所示。

本发明对存储器进行快照扫描的过程如下：首先，输出缓存 12 的数据；其次，输出备份存储器 20 的内容；最后，更新备份存储器 20 使之与主存储器 10 再次保持一致。本发明的重点在于如何更新备份存储器 20 使之与主存储器 10 再次保持一致以便进行下一次快照。简而言之，本发明是结合复制系统对主存储器 10 的写操作与复制主存储器 10 的内容两项措施来完成对备份存储器 20 的更新，详述如后。

在完成一次存储器的快照后，存储器快照状态机 1 通知快照控制模块 40 停止对备份存储器 20 进行读操作，快照控制模块 40 通过 Shift_Select 信号使第一复用模块 50 选择 i_sram_scan_in 为输出。此时，需要更新备份存储器 20 使之与主存储器 10 再次保持一致，以便进行下一次快照。更新备份存储器 20 的措施有两个，而且必须结合这两个措施才能保证备份存储器 20 最终与主存储器 10 完全一致：一是把系统对主存储器 10 的写操作复制到备份存储器 20，二是利用空闲期间对主存储器 10 从头至尾进行一次读操作，并把相应的数据复制到备份存储器 20。为复制主存储器 10 的内容而对主存储器 10 进行的读操作需要在系统不对主存储器 10 进行读操作时才能进行；为复制主存储器 10 的内容而对备份存储器 20 进行的写操作，只要系统不对主存储器 10 进行写操作时就可进行。采用不同的静态随机存取存储器（如单端口（one port）以及双端口（dual port）的静态随机存取存储器），就可能需要在不同的时机下才能对其进行相应的操作，以下用空闲期间来概括可对其进行操作的时机。

主存储器 10 的输入信号有激活存储器的信号 CEN、主存储器的写入数据 F_Write_Data、对主存储器 10 进行写操作的使能信号 F_Write_Enable、对主存储器 10 进行写操作的地址 F_Write_Address、对主存储器 10 进行读操作的使能信号 F_Read_Enable、对主存储器 10 进行读操作的地址 F_Read_Address 以及时钟 ClockA/ClockB。为了使备份存储器 20 与主存储器 10 时刻保持一致，需要把系统对主存储器 10 的写操作复制到备份存储器 20，也就是把所述的 F_Write_Data、F_Write_Enable、F_Write_Address 三个信号引入备份存储器对应的输入。

可能对主存储器 10 进行读操作的对象有两个，分别是系统和更新控制模

块 30。因此，需要设置第三复用模块 100 以选择系统激活存储器的信号 S_CEN 或者更新控制模块 30 激活存储器的信号 CEN_Select 作为激活主存储器 10 的信号 CEN，CEN_Select 同时控制第三复用模块 100 的输出选择，如此实现对主存储器 10 的激活控制。若要对存储器进行读操作还有两个信号是必需的：F_Read_Enable 与 F_Read_Address，因此，还需要设置第四复用模块 102 及第五复用模块 104。第四复用模块 102 的两个输入分别为系统对主存储器 10 进行读操作的使能信号 S_Read_Enable 与更新控制模块 30 对主存储器 10 进行读操作的使能信号 F_Read_Select；第五复用模块 104 的两个输入分别为系统对主存储器 10 进行读操作的地址 S_Read_Address 与更新控制模块 30 为复制主存储器 10 的内容而对主存储器 10 进行读操作的地址 Copy_Read_Address；F_Read_Select 同时控制第四复用模块 102 及第五复用模块 104 的输出选择。

本发明对备份存储器 20 的更新结合了复制系统对主存储器 10 的写操作与复制主存储器 10 的内容两个措施，因此，可能对备份存储器 20 进行写操作的对象也可能有两个：一个是系统，另一个是更新控制模块 30。因此，就可能需要设置第六复用模块 202、第七复用模块 204 及第八复用模块 206。第六复用模块 202 有两个输入：一个是 F_Write_Data，另一个是主存储器 10 的输出 F_Data_Out（用于复制主存储器 10 的内容到备份存储器）。第七复用模块 204 有两个输入：一个是 F_Write_Enable，另一个是更新控制模块 30 给出的 Copy_Write_Enable（复制主存储器 10 的内容到备份存储器 20 的写使能信号）。第八复用模块 206 有两个输入：一个是系统的 F_Write_Address，另一个是更新控制模块 30 给出的 Copy_Write_Address（复制主存储器 10 的内容到备份存储器 20 的地址）。更新控制模块 30 给出的 B_Write_Select 信号控制第六复用模块 202、第七复用模块 204 及第八复用模块 206 的输出选择。

以上说明了各组件之间的信号走向和连接关系，下面对快照和备份存储器 20 的更新流程进行说明。当系统通知存储器快照状态机 1 开始进行快照时，快照更新控制模块 30 把 Copy_Write_Enable 置为无效，并通过 B_Write_Select 选择 Copy_Write_Enable 为第七复用模块 204 的输出，屏蔽对备份存储器 20 的写操作，使备份存储器 20 的内容与系统开始进行快照的那一时刻的主存储器 10 的内容保持相同，即锁定了开始快照时主存储器的内容。缓存 12 也在这一

刻锁定其值，保持这一刻主存储器 10 输出端口的值。

当系统通知存储器快照状态机 1 对存储器进行快照时，快照控制模块 40 通过 Buffer CTRL 锁定缓存 12 的值，接着通过 Shift_Select 控制选择缓存 12 的输出为第一复用模块 50 的输出，输出主存储器 10 输出端口的值；输出完毕后，快照控制模块 40 再通过 Shift_Select 控制选择备份存储器 20 的输出 Backup_Data_Out 作为第一复用模块 50 的输出，并产生 Backup_Read_Enable 信号（对备份存储器 20 进行读操作的使能信号）与 Backup_Read_Address 信号（对备份存储器 20 进行读操作的地址）送到更新控制模块 30 对备份存储器 20 进行读操作，把备份存储器 20 的内容输出，当备份存储器 20 的内容输出完毕后当前存储器的快照结束。

对当前存储器的快照完毕后就要更新备份存储器 20，使之与主存储器 10 再次保持一致，以便下一次快照，这是本发明的重点所在。快照控制模块 40 通过地址比较得知快照结束时通知存储器快照状态机 1，存储器快照状态机 1 通知更新控制模块 30 开始对备份存储器 20 进行更新。S_CEN 同时给第三复用模块 100 与更新控制模块 30，这样更新控制模块 30 就知道系统何时对主存储器 10 进行操作，何时不对主存储器 10 进行操作。当系统对主存储器 10 进行写操作时，更新控制模块 30 通过 B_Write_Select 复制该写操作到备份存储器 20，换言之，备份存储器 20 随着主存储器 10 的变化而变化，因此，只要在这个变化的过程中把主存储器 10 从起始地址到末尾地址的内容复制一遍到备份存储器 20 就可使两者内容保持一致。对静态随机存取存储器一个地址的读取只需要一个时钟周期，更新控制模块 30 在主存储器 10 的空闲期间至少可完成一个读操作，在系统对主存储器 10 进行下一次写操作之前，若更新控制模块 30 来不及把由所述读操作得到的数据写入备份存储器 20，就先把这个数据保存在更新控制模块 30。在这个数据的保存期间内，若系统对主存储器 10 的该地址进行了写操作，由于备份存储器 20 复制了系统对主存储器 10 的写操作，其该地址的值已经与主存储器 10 的相同，因此就取消这个数据的复制操作。更新控制模块 30 把从 F_Write_Address 得到的地址与所述被保存的数据原先在主存储器 10 中的地址进行比较即可判断出系统在数据保存期间是否对这个地址进行了写操作。

以下结合图4对本发明的静态随机存取存储器快照模块的工作时序进行说明,其中,S_Write_Enable表示系统发出的写使能信号,而F_Write_Enable为主存储器10的写使能信号,本实施例中只有系统可对主存储器10进行写操作,故S_Write_Enable和F_Write_Enable相同。图4中所示的第一阶段的快照状态(快照状态为深色的一段期间)中,静态随机存取存储器快照模块不进行快照与更新。更新控制模块30通过F_Read_Select选择S_Read_Enable与S_Read_Address为第四复用模块102与第五复用模块104的输入,即只允许系统对主存储器10进行读操作。更新控制模块30通过B_Write_Select选择F_Write_Data、F_Write_Enable、F_Write_Address为第六复用模块202、第七复用模块204及第八复用模块206的输出,即将系统对主存储器10的写操作复制到备份存储器20。当S_Read_Enable(系统给出的读操作使能信号)有效时,F_Read_Enable有效,系统对主存储器10进行读操作。当F_Write_Enable有效时,B_Write_Enable有效,系统对主存储器10和备份存储器20进行写操作。

第二阶段是快照阶段,在图4的快照状态中标为shift-state。更新控制模块30通过F_Read_Select选择S_Read_Enable和S_Read_Address为第四复用模块102及第五复用模块104的输出,只有系统可对主存储器10进行读操作。更新控制模块30把Copy_Write_Enable置为无效,并通过B_Write_Select选择F_Data_Out(主存储器10的输出)、Copy_Write_Enable、Copy_Write_Address为第六复用模块202、第七复用模块204及第八复用模块206的输出,由于Copy_Write_Enable被置为无效,因此,此时备份存储器20被冻结。系统对主存储器10的操作与快照互不影响,快照控制模块40可在任何时刻通过更新控制模块30对备份存储器20进行读操作。

第三阶段是对备份存储器20进行更新的阶段,图4中的快照状态标为update-back-up-sram-state。在更新过程中,对备份存储器20,既要复制系统对主存储器10的写操作,还要复制主存储器10的内容,后者的进行以不影响前者为前提(如前所述)。系统需要对主存储器10进行写操作时,更新控制模块30通过B_Write_Select选择F_Write_Data、F_Write_Enable及F_Write_Address为第六复用模块202、第七复用模块204及第八复用模块206

的输出，在备份存储器 20 上复制系统对主存储器 10 的写操作。图 4 中该阶段 F_Write_Enable 第一个有效电平就是系统对主存储器 10 和备份存储器 20 同时进行的写操作。当主存储器 10 处于空闲期间，更新控制模块 30 将对备份存储器 20 复制主存储器 10 的内容。更新控制模块 30 通过 F_Read_Select 选择 F_Read_Select 和 Copy_Read_Address 作为第四复用模块 102 及第五复用模块 104 的输出，对主存储器 10 进行读操作，图 4 中该阶段 F_Read_Enable 的第二个有效电平就是更新控制模块 30 对主存储器 10 进行的读操作。若空闲期间足够长，紧接着这个读操作，更新控制模块 30 把 B_Write_Enable 置为有效电平，并通过 B_Write_Select 选择 F_Data_Out、Copy_Write_Enable、Copy_Write_Address 为第六复用模块 202、第七复用模块 204 及第八复用模块 206 的输出，把由所述读操作得到的数据写入备份存储器 20，图 4 中该阶段 B_Write_Enable 的第二个有效电平就是更新控制模块 30 为复制主存储器 10 的内容对备份存储器 20 进行的写操作。图 4 中该阶段 F_Read_Enable 出现的第三个有效电平是更新控制模块 30 对主存储器 10 进行的读操作，但紧随着这个 F_Read_Enable 有效电平的结束，出现了 S_Write_Enable 的有效电平，这说明系统要对主存储器 10 进行写操作，因为这个写操作同样要被复制到备份存储器 20，所以由更新控制模块 30 从主存储器 10 读出的数据来不及写入备份存储器 20，这个数据暂时被更新控制模块 30 保存（例如存储于附加的存储模块中，未示出），在系统对主存储器 10 和备份存储器 20 进行写操作的过程中，更新控制模块 30 把 F_Write_Address 给出的地址与所述被保存数据原先在主存储器 10 中的地址进行比较，若 F_Write_Address 给出的地址中有一个与该地址相同，就表示主存储器 10 该地址的数据已经被系统改写，而备份存储器 20 因为复制了系统对主存储器 10 的写操作，其该地址的数据已经与主存储器 10 该地址的数据保持一致，为复制这个数据的写操作取消。反之，则应该在系统对主存储器 10 的写操作完毕后把这个数据写入备份存储器 20，图 4 中该阶段 B_Write_Enable 出现的第四个有效电平就是把这个数据写入备份存储器 20 的操作。

其中，值得说明的是，可以限制为复制主存储器 10 的内容对备份存储器 20 的写操在 S_CEN 为无效的时候进行，也可以只限制其在 F_Write_Enable 无

效的时候（其范围比前者大）进行，本发明实施例采用后者进行说明。但是，本发明的思想同样能够简单地适用于前者。

在本实施例中，虽然以分立模块的方式描述了各种模块之间的关系和信号走向，但是，如本领域的技术人员所熟知的那样，例如可以把更新控制模块 30、快照控制模块 40、以及存储器快照状态机结合于一个功能模块中，例如存储器控制模块，对存储器的读写以及复制和快照进行控制。也可以把缓存 12、第一复用模块 50、第二复用模块 14、扫描寄存器 60 以及扫描逻辑模块 70 结合于一个功能模块中实现，例如存储器输出模块。这些等效替换是本领域的常用技术手段，都应视为本发明的范围之内。

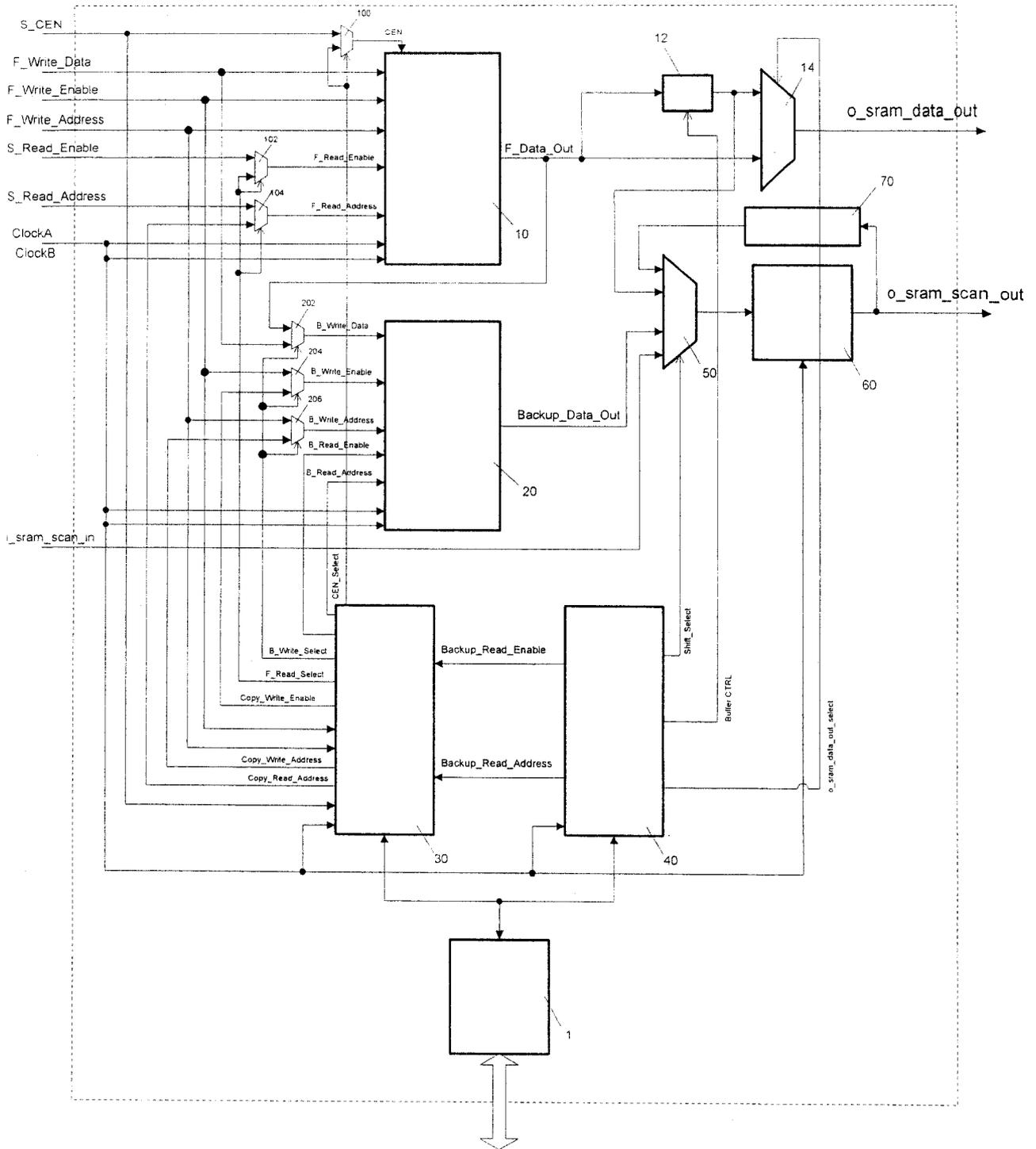


图 1

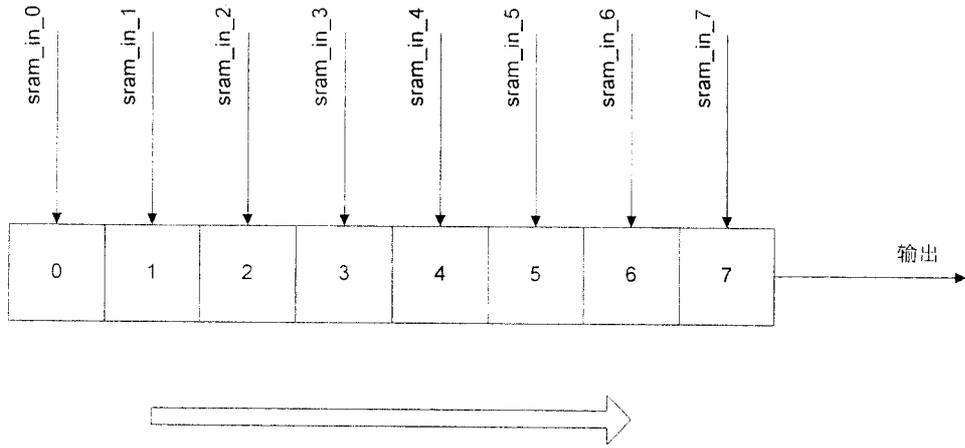


图 2

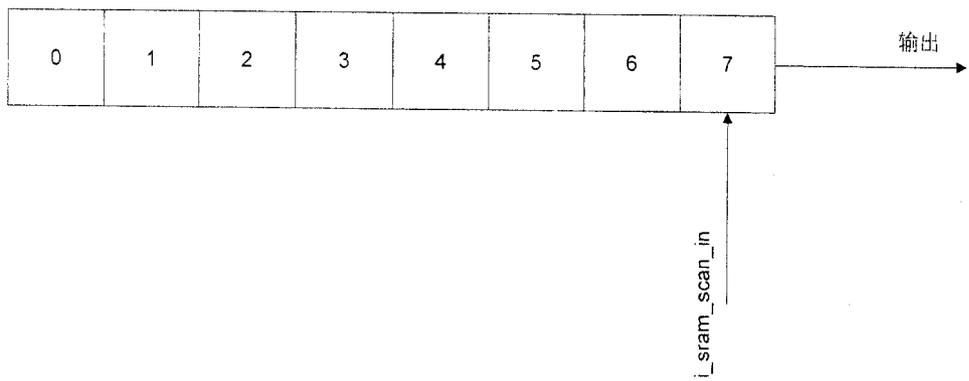


图 3

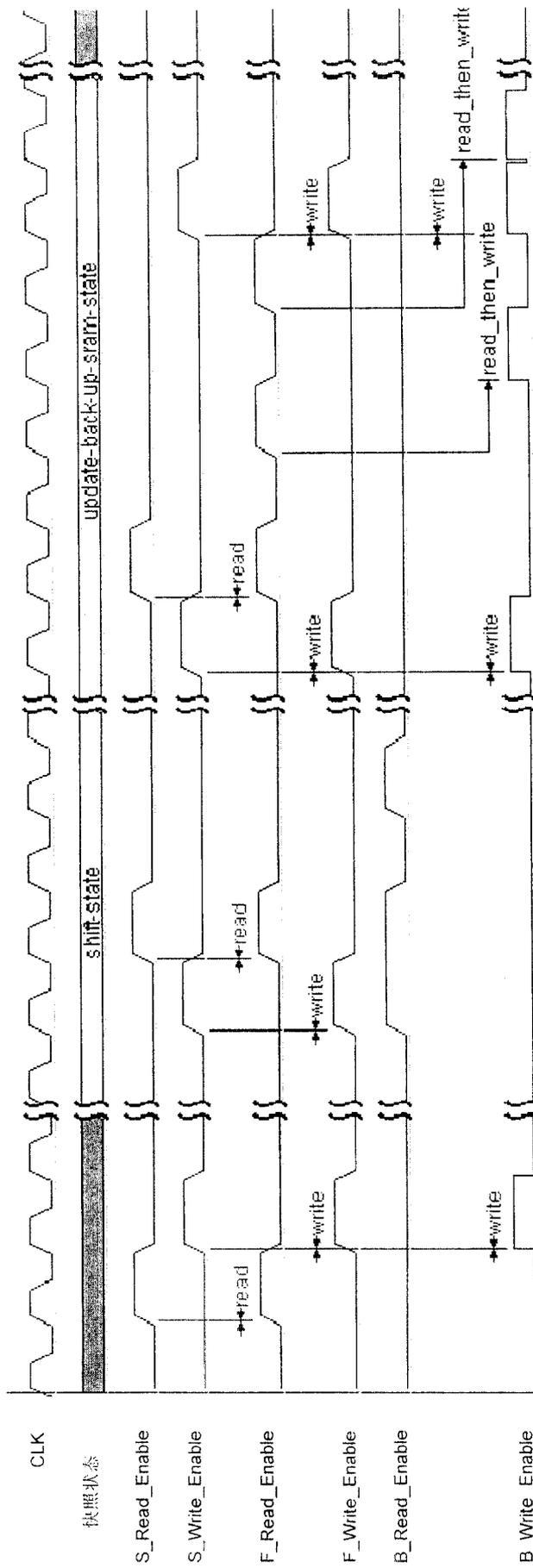


图 4