

一种用刷新技术实现 SRAM 抗 SEU 错误累积的方法

陈庆宇, 吴龙胜, 郝 奎, 艾 刁

(西安微电子技术研究所, 陕西 西安 710054)

摘 要: 为了防止空间应用 SRAM 出现 SEU 错误累积, 提出了一种优化的读→校验→回写刷新机制. 该机制实时监测处理器状态, 当处理器对外部主存进行读操作时, 由存储器控制器自主地(即不需处理器干预)对读操作的存储单元进行刷新操作; 当处理器进行访问外部主存以外的其他操作时, 由存储器控制器自主的对所有的存储单元进行遍历式刷新操作, 该机制可以避免长时间未被读的存储单元发生 SEU 错误的累积, 保证 SRAM 单元中发生错误的比特位数小于校验码的纠错能力. 最后, 通过向 SRAM 随机注错的方法对本机制的存储器控制器进行验证, 结果表明存储器控制器满足设计要求.

关键词: 空间应用; SRAM; 存储器控制器; 刷新; 单粒子翻转

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)07-0113-05

A Method for Resisting SRAM SEU Accumulation with Scrubbing

CHEN Qing-yu, WU Long-sheng, HAO Kui, AI Diao

(Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: To prevent the accumulation of SEU errors in the SRAM of space application, this paper puts forward an optimized scrubbing mechanism featuring with reading, checking and writing back. The mechanism makes a real-time monitor for the processor state. When the processor requests the external main memory to read, the memory controller will scrub the storage unit which was reading without CPU intervention. While the processor has no operation on main memory, the memory controller is of independence to refresh all SRAM cells, which is traverse type. The mechanism can avoid SEU errors accumulation in SRAM cells which has not been read for a long time, and guarantee that the error bits in an SRAM cell is less than the check code error detection ability. Finally, this paper makes the memory controller which is implemented the scrubbing mechanism a verification, the results show that the memory controller can archive the design requirements.

Key words: space application; SRAM; memory controller; scrubbing; SEU

1 引言

为了确保辐射环境中电子系统的可靠性, 面向空间应用的电路必须具备自我恢复能力^[1]. 随着集成电路工艺尺寸的减小, SRAM 存储单元中截止管漏源区反偏 PN 结空间电荷区能承载的临界电荷有所降低, 导致 SRAM 中 SEU 事件愈加不容忽视. 据研究, 如果芯片尺寸按照摩尔定律每三年降低 40% 的话, 单粒子效应引起的 SEU 错误每年增加一倍^[2]. SRAM 作为空间电子系统中数据和指令的存

储单元, 提高其抗 SEU 能力具有重要的现实意义.

目前, 主要从工艺制造、电路设计和系统级容错等方面提高 SRAM 抗 SEU 的能力. (1) 工艺方面^[2], 采用 SOI(Silicon on Insulator)、SOS(Silicon on Sapphire)等特殊工艺, 可以减弱 SEU 的发生. (2) 电路设计方面^[3-4], 对 N 管而言, 可以增大栅面积、提高栅氧化层厚度及沟道长宽比; 对 P 管而言, 可以减小沟道长宽比及降低栅氧化层厚度等. 另外可以增加反馈电阻或者采用具有冗余功能的双向互锁单元. (3) 系统级容错方面^[5-7], 大多采用三模冗

余、定期刷新(deterministic scrubbing)、概率刷新(probabilistic scrubbing)等措施. 电路加固和特殊工艺制造的 SRAM, 通常会造成集成度降低、读写周期延长、功耗增加等问题, 限制了这类 SRAM 的应用范围. 三模冗余通过备份存储信息实现容错能力, 但是会增加系统的面积、功耗、成本等. 系统设计师采用刷新技术提高 SRAM 抗 SEU 的能力. 所谓定时刷新指每经过一定的刷新间隔, 处理器通过存储器控制器对 SRAM 的所有存储单元进行一次读→校验→回写的刷新操作, 该技术会占用处理器正常执行的时间, 不能满足当前电子系统的实时性要求. 概率刷新指处理器对 SRAM 某存储单元进行读操作时, 会进行校验, 如果发生可纠正的错误, 则进行回写操作, 概率刷新技术对某些未进行读操作的存储单元无法进行保护.

计算机科学中最重要的程序特征是局部性原理^[8]: 程序会经常重复使用它最近使用过的指令和数据, 说明大部分程序并不是均衡地访问所有的代码和数据, 因此设计中必须考虑长时间不被访问的存储单元的 SEU 问题. 本文提出一种面向 SRAM 抗 SEU 错误累积的方法, 该方法通过对概率刷新技术的优化, 从而避免长时间未进行读操作的存储单元中 SEU 错误的累积, 有效保证两次刷新之间 SEU 引起的翻转位数不超过错误校验码 ECC(error correction coding)的纠正能力.

2 存储器控制器的控制算法

本文实现的一种用刷新技术实现 SRAM 抗 SEU 错误累积的方法, 其特征在于: 当处理器对主存进行读操作时, 由微处理器的存储器控制器自主的(即不需处理器干预)对读操作的存储单元进行读→校验→回写的刷新操作. 当处理器进行访问主存以外的其他操作时, 存储器控制器利用自身的空闲时间, 从 SRAM 的零地址开始, 对 SRAM 所有的存储单元进行遍历式刷新操作, 遍历式刷新操作以最基本的校验单元为刷新步长, 每当完成一个存储单元的刷新操作, 刷新地址递增, 控制状态机回到空闲态, 重新检测处理器是否存在访存操作. 主要控制算法如下所述:

Step1: 当存储控制器在 idle 态时, 检测是否存在处理器读操作, 若存在, 转向 Step9, 否则, 执行 Step2;

Step2: 检测是否存在处理器写操作, 若存在, 转向 Step14, 否则, 执行 Step3;

Step3: 存储控制器进行遍历式刷新的读操作, 读取刷新地址对应的数据 D 和校验位 B ;

Step4: 根据 Step3 读出的数据 D , 按校验码生成方程组计算其校验位 B_1 , 将 B 和 B_1 进行异或运算, 得到校验结果 $U(U = U_0 \cup U_1 \cup U_2)$, U_0 表示无错, U_1 指数据 D 中存在纠正的错误, U_2 指数据 D 中存在不可纠正的多位错误;

Step5: 判断 Step4 得出的校验结果 U 是否属于 U_0 , 若 $U \in U_0$, 则刷新地址自加刷新字长, 执行 Step6, 否则执行 Step7;

Step6: 存储控制器返回 idle 态, 结束;

Step7: 判断 Step4 得出的校验结果 U 是否属于 U_1 , 若 $U \in U_1$, 刷新地址不变, 执行 Step8, 否则刷新地址加 $0x4$, 执行 Step6;

Step8: 存储器控制器进行遍历式刷新的写操作, 将校验后的数据和校验位写入刷新地址对应存储单元, 写入完成后, 刷新地址加 $0x4$, 转向执行 Step6;

Step9: 存储器控制器进行处理器的读操作, 读取处理器访存地址对应的数据 D 和校验位 B ;

Step10: 根据 Step9 读出的数据 D , 按校验码生成方程组计算其校验位 B_1 , 将 B 和 B_1 进行异或运算, 得到校验结果 $U(U = U_0 \cup U_1 \cup U_2)$, U_0 表示无错, U_1 指数据 D 中存在纠正的错误, U_2 指数据 D 中存在不可纠正的多位错误;

Step11: 判断 Step9 得出的校验结果 U 是否属于 U_0 , 若 $U \in U_0$, 则执行 Step6, 否则, 顺序执行 Step12;

Step12: 判断 Step9 得出的校验结果 U 是否属于 U_1 , 若 $U \in U_1$, 则存储器控制器进行概率刷新的写入操作, 将校验后的数据和校验码写入处理器访问地址对应的 SRAM 存储单元, 否则, 顺序执行 Step13;

Step13: 存储器控制器进行错误处理, 向处理器返回错误信息, 之后执行 Step6;

Step14: 存储器控制器进行处理器的写操作, 将数据和校验码写入到处理器访问地址对应的 SRAM 存储单元, 写操作完成后, 执行 Step6.

上述控制算法中, 校验码生成方程组指根据校验码生成矩阵得到, 遍历式刷新中要求每次刷新完成后, 刷新地址自加刷新字长, 刷新字长指 ECC 校验算法的数据位宽对应的地址长度(如校验算法针对 32 位的数据, 刷新字长为 $0x4$, 针对 8 位的数据, 刷新字长为 $0x1$). 特定的刷新字长意味着遍历式刷

新操作以最基本的 ECC 校验单元为基本单位,每当完成一个基本单位的刷新操作,刷新地址递增,控制状态机回到空闲态,重新检测处理器是否存在访存操作,即当遍历式刷新操作被处理器的访存操作打断时,由硬件保证刷新地址不变,再次进行遍历式刷新操作时,需从上次被打断的刷新地址开始。

上述控制算法利用处理器进行其他操作的时间

(即存储器控制器的空闲时间),由硬件直接插入针对所有 SRAM 单元遍历式刷新操作,为长时间不被访问的存储单元的 SEU 累积提供了一种有效的解决措施.该方法在不占用处理器执行时间的基础上,提供了一种有效的抗 SEU 错误累积方法,提高了存储系统的可靠性,其控制流程图如图 1 所示。

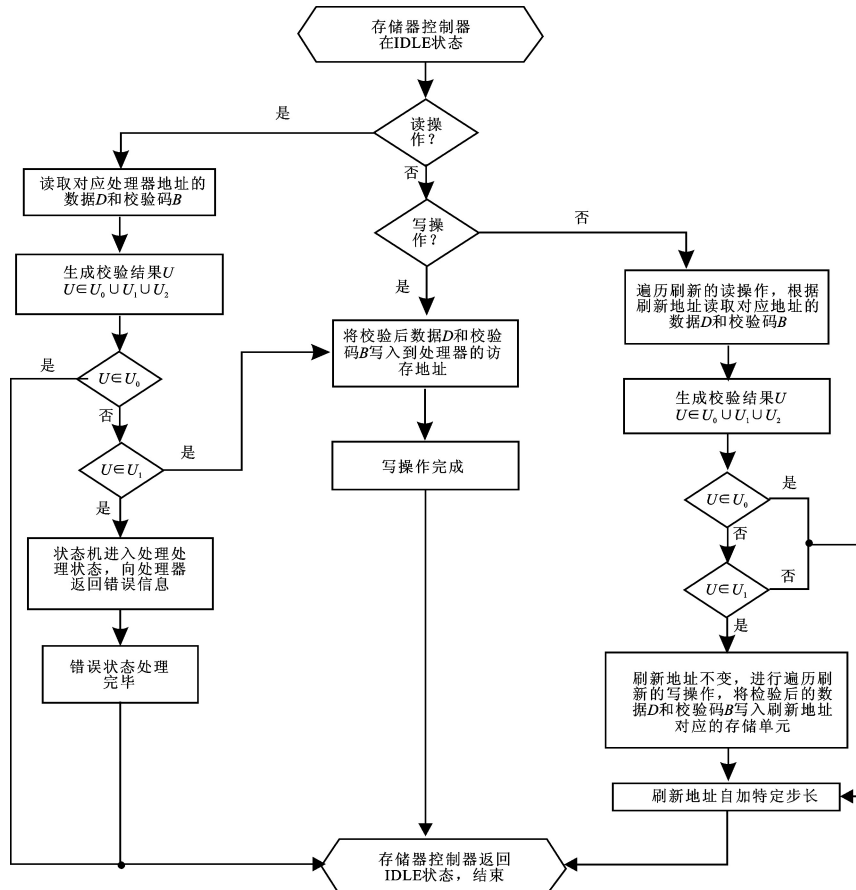


图 1 抗 SEU 错误累积的控制算法流程图

3 存储器控制器的实现

综合考虑存储器控制器的可复用性、SRAM 的时序和容错要求,采用自顶向下的实现方式,对存储器控制器进行功能模块的划分,最后使用 VHDL 语言完功能设计.存储器控制器电路由寄存器组、控制信息生成电路、状态转移控制电路、纠检错模块、多路选择器、校验码编码模块、读写数据寄存器及若干控制信号等组成,具体结构如图 2 所示。

寄存器组包含存储器控制器中所有处理器可见的控制和状态寄存器,处理器通过配置寄存器组规定存储器控制器工作状态(EDAC 是否使能等).出于可复用性的考虑^[9],寄存器组基于 AHB 总线接口,采用基址+偏移地址的地址规划方式。

控制信息生成电路在检测 AHB 访问信息的同时读取寄存器组中的控制信息,生成处理器的读写标志、EDAC 使能等信号,这些控制信号连接至状态转移控制电路。

状态转移控制电路通过状态机实现本文前述的控制算法,在相应的读写状态时生成访问 SRAM 所需的读写、片选信号,同时控制纠检错模块的 EDAC 使能,接收校验结果,如果校验结果出现单错,那么状态机进入写状态,通过多路选择器选择校验后的数据进行回写,进行 SRAM 的刷新操作,状态机的状态转移图如图 3 所示。

校验码编码模块将写入到 SRAM 存储单元中的 32 位数据生成 7 位校验码,纠检错模块对读出数据和校验码的正确性进行判断并给出校验结果.上述两个

模块均采用基于萧码(Hsiao code)的编码规则,具备纠

一检二的能力,且不会发生类似于汉明码的误纠情况.

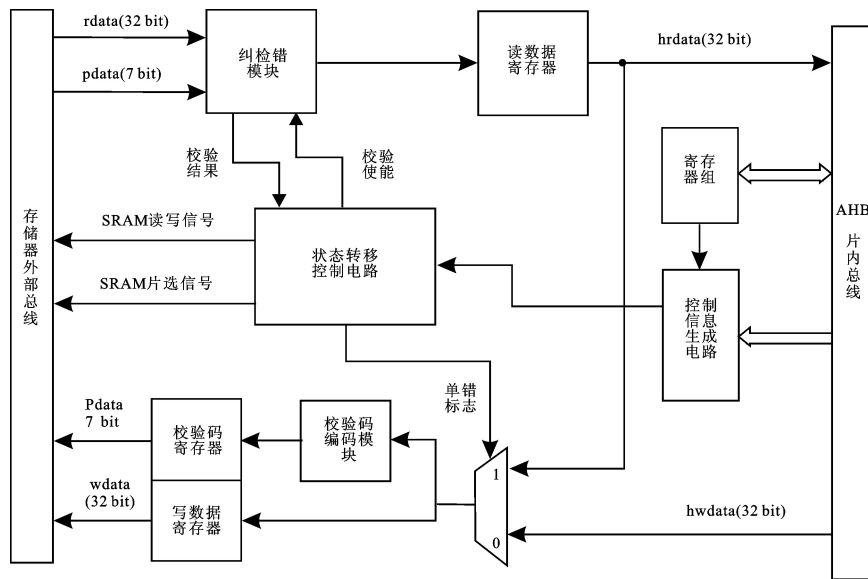


图2 存储器控制器的结构框图

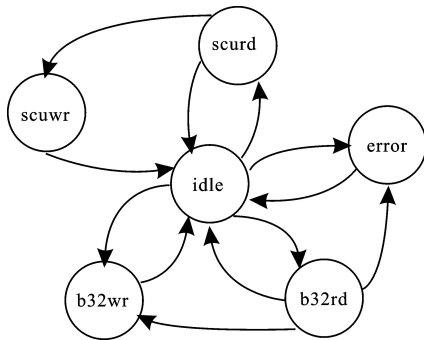


图3 状态转移控制电路的状态转移图

下面结合图2和图3对本文设计的存储器控制器的工作过程进行说明. 存储器控制器在idle状态如果检测到写操作,那么进入b32wr状态,同时对写数据wdata生成7位校验码,在状态转移控制电路的控制下,将数据和校验码写入对应的存储器单元,然后状态机返回idle状态;若在idle状态检测到读操作,则进入b32rd状态,将读出的数据和校验码送到纠检错模块,纠检错模块将校验结果送到状态转移控制电路,根据校验结果,状态可能跳转至idle、b32wr或者error. 以校验结果发生单错为例,状态转移控制电路控制状态机进入b32wr状态,同时输出单错标志到多路选择器的控制端,选择将经过校验纠错的hrdata输入到校验编码模块,生成校验码,完成回写操作,返回idle状态;若idle状态下无法检测到处理器读写操作,那么进入scurd状态,读出刷新地址对应的数据和校验码,然后根据校验结果控制状态机的转移,状态机的转移完全符合本文前述的控制算法,最后返回idle状态.

4 存储器控制器的仿真验证

在验证过程中,通过编写验证测试程序,用Cygwin Bash Shell编译器将测试程序生成二进制机器码,存放在romv.dat,随后将romv.dat文件加载进ROM模型中,最后利用仿真器Questasim 10.0c对存储器控制器进行仿真,其仿真环境如图4所示. 需要说明的是,系统设计中,ROM的起始地址为0x0,SRAM的起始地址为0x40000000,处理器复位后从0x0地址开始取指令、执行.

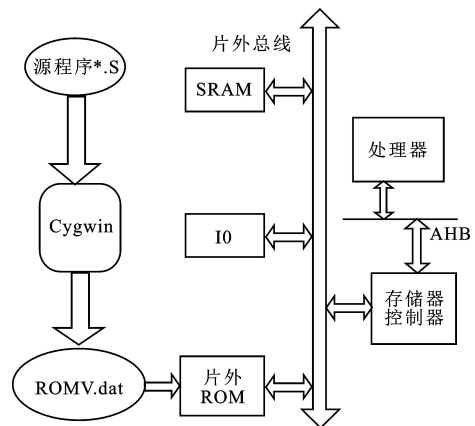


图4 存储器控制器的系统级仿真环境

测试程序中主要的组成部分依次为:(1)处理器初始化程序,包括使能SRAM的EDAC功能;(2)对SRAM的0x40000000~0x40000200的1k字写入固定值;(3)执行一段循环程序,使处理器Cache命中,这段循环程序避免对步骤(2)的1k字进行任意操作,循环程序所需的处理器时间 $T > 2T_j$ (T_j 为

1 k 字的错误注入周期), (4) 经过循环程序的 T 时间之后, 对(2)的 1 k 字进行读操作. 上述程序执行过程中, 通过调用工具可识别的脚本(Questasim 中为 mem load), 在步骤(2)之后对 0x40000000 ~ 0x40000200 的每个字中任意 1 bit 位进行翻转, 循环周期为 T_j . 根据上述说明, 如果存储器控制器未对 SRAM 进行刷新, 那么执行步骤(4)会发生不可纠正的数据错误, 使处理器进行陷阱服务程序. 仿真证明, 处理器正常结束测试程序, 未进入陷阱服务程序, 即本文的遍历刷新算法在可以一定条件下避免 SEU 错误的累积.

存储器控制器的概率刷新时序图如 5 所示, 处理器从 0x400 地址取数据, 得到数据 0x87654320 和校验位 0x2A(0x2A 对应着 0x87654321), 这种情况下相当于发生单错, 存储器控制器将 0x87654320 纠正后传递给处理器, 并将纠正后的数据 0x87654321 写入 0x400. 概率刷新保护发生读操作的 SRAM 存储单元, 避免其 SEU 错误累积. 图 6 所示为存储器空闲状态下的遍历式刷新, ahbsi.htrans 为 0, 说明处理器未进行访存操作, 遍历刷新对 0x724 进行读操作, 得到数据 0x000003C9, 校验过程中发现单错, 之后将校验后的数据 0x000001C9 写回到 0x724, 遍历刷新可以有效避免长时间未进行读操作的存储单元的 SEU 错误累积.

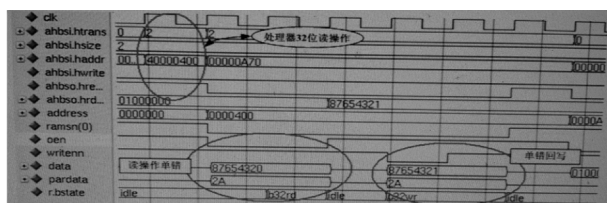


图 5 概率刷新技术时序图

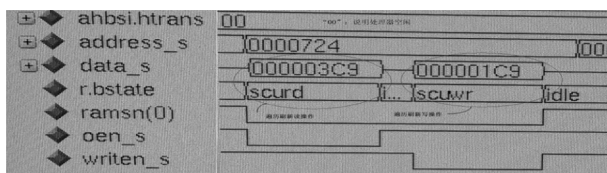


图 6 遍历刷新技术时序图

5 结束语

为了防止空间应用 SRAM 出现 SEU 错误累积, 本文提出了一种基于存储器控制器空闲时间的刷新控制算法, 之后搭建验证环境、编写测试用例对存储器控制器的抗 SEU 错误累积的能力进行了验证, 仿真结果表明, 控制算法和存储器控制器符合设计预期, 在不占用处理器执行的基础上, 提供了一种

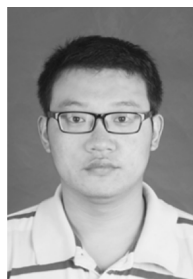
有效的抗 SEU 错误累积的方法, 提高了存储系统的可靠性, 为面向空间应用的处理器设计提供了一种加固思路.

将来的工作是改进 Cache 结构, 提高 Cache 命中率, 为存储器控制器提供更多的空闲时间, 从而进一步提高其抗 SEU 错误累积的能力.

参考文献:

- [1] 李玉红, 赵元富, 岳素格, 等. 0.18 μm 工艺下单粒子加固锁存器的设计与仿真[J]. 微电子学与计算机, 2007, 24(12): 66-69.
- [2] 张英武, 郭天雷, 袁国顺. 高可靠微处理器的设计[J]. 微电子学与计算机, 2009, 26(1): 59-62.
- [3] 贺朝会, 李国政, 罗晋生, 等. CMOS SRAM 单粒子翻转效应的解析分析[J]. 半导体学报, 2000, 21(2): 174-178.
- [4] 王步冉, 夏克强. SRAM 的 SEU 效应及加固技术三维数值模拟[J]. 质量与可靠性, 2012(3): 43-46.
- [5] Juan Antonio Maestro, Pedro Reviriego. Reliability of single-error correction protected memories[J]. IEEE Transactions on Reliability, 2009, 58(1): 193-201.
- [6] Abdallah M Saleh, Juan J Serrano, Janak H Patel. Reliability of scrubbing recovery-techniques for memory systems [J]. IEEE Transactions on Reliability, 1990, 39(1): 114-122.
- [7] 辛明瑞, 时晨, 张伟功, 等. 空间计算机抗单粒子翻转的存储器纠错检错与自动回写方法: 中国, ZL200510041617[P]. 2005-06-29.
- [8] John L. Hennessy, David A. Patterson, Computer architecture, a quantitative approach (fourth edition) [D]. BeiJing: Publishing House of Electronics Industry, 2007: 72-88.
- [9] 陈庆宇, 盛廷义, 吴龙胜, 等. 面向 AHB 总线从单元的可复用接口设计[J]. 微电子学与计算机, 2012, 29(12): 59-63.

作者简介:



陈庆宇 男, (1988-), 博士研究生. 研究方向为空间应用高性能处理器设计.

吴龙胜 男, (1968-), 博士生导师、研究员、总工程师. 研究方向为高性能 SoC 设计.