文章编号:1671-4598(2018)04-0264-04 DOI:10.16526/j.cnki.11—4762/tp.2018.04.068 中图分类号:TN913.6 文献标识码:A

基于 VCS 的固存坏块仿真系统设计与应用

刘国斌，祝周荣，宁 静，刘 攀，陈恩耀
（上海航天电子技术研究所，上海 201109）

摘要：NAND Flash 固态存储器（固存）广泛应用于航天工程，受限于微电子特性及制造工艺，固存在出厂及使用过程中均会产生坏块，通常由固存控制 FPGA（现场可编程门阵列）来管理并标记坏块；为保证固存控制 FPGA 对坏块管理的正确性、健壮性，必须对其进行严格验证；提出了基于 VCS 的固存坏块仿真验证系统，为固存控制 FPGA 提供了所需的外围接口，特别是提供了固存坏块反馈机制，令固存坏块产生时机受控；实时向 FPGA 反馈固存读写过程及产生的坏块信息；将坏块表建立、维护和固存响应过程记录到数据文件；实现了坏块分布的可配置性和仿真系统的闭环性、可记录性；仿真系统可有效发现坏块管理的设计缺陷，进而优化设计，提高航天固存产品可靠性。

关键词：VCS；仿真；固存；坏块管理

Design and Application of Simulation System for Solid—state Memory Invalid Blocks Based on VCS

Liu Guobin, Zhu Zhourong, Ning Jing, Liu Pan, Chen enyao
(Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China)

Abstract: Solid—state Memory based on NAND Flash has been widely applied in aerospace engineering. Limited by its microelectronic characteristic and manufacturing technics, Solid—state Memory may generate invalid blocks during the manufacturing or working process. Usually, FPGA is used to manage and tag invalid blocks of Solid—state Memory. To guarantee the correctness and robustness of invalid blocks management provided by FPGA, strict verification must be performed. Simulation System for Solid—state Memory Invalid Blocks Based on VCS was proposed. It provides necessary peripheral IOs for FPGA, especially the feedback mechanism of invalid blocks which making the generation of invalid blocks under control; reading—writing process and the information of invalid blocks are fed back to FPGA in real time; the establishment and maintenance procedure of invalid blocks as well as the response of Solid—state Memory are recorded in data file simutaneously . The simulation system is closed—loop, recordable , and configurable for the distribution of invalid blocks. The simulation system shows validity in verifying and optimizing design of invalid blocks management, and the reliability of Solid—state Memory used in aerospace has been enhanced accordingly.

Keywords: VCS, simulation; solid—state memory; invalid block management

0 引言

NAND Flash[1]固存由于其具有的优点广泛应用于航天数据处理系统，这些优点包括访问速度快、低功耗、高密度、大容量、抗震性强等[2]。由于 NAND Flash 的制造工艺不能保证其存储区域在生命周期内保持性能的可靠，因此在 NAND Flash 生产及使用过程中都有可能产生坏块[3]。在数据处理系统中使用固存控制 FPGA 对坏块进行管理，为保证坏块管理的正确性，必须对固存控制 FPGA 进行严格验证。随着坏块管理设计复杂性越来越高，对坏块的验证难度也随之变得日益复杂。传统上，一个完整的设计，往往需要花费 70%的时间在验证上，即便如此，也很难保证坏块管理的完全正确性[4]。传统的对固存控制 FPGA 坏块管理的验证方法有模块级仿真和硬件联测。前者单独测试固存的擦写、读、全擦除等流程，无法模拟固存工作过程中坏块的产生及其对固存控制过程的影响，难以保证验证的充分性。后者在硬件上宏观上测试系统功能，被动地等待固存坏块产生，难以主动控制固存坏块产生位置，且测试周期长，无法保证坏块在不同分布工况下的功能正确性。为此有必要设计一种坏块受控的仿真系统来保证坏块验证的充分性，并提高坏块验证效率。

收稿日期:2018-02-02； **修回日期**:2018-03-01。

作者简介:刘国斌(1986-),男,上海人,工学硕士,工程师,主要从事 FPGA 第三方验证方向的研究。

1 仿真系统搭建

选用 VCS 作为搭建坏块仿真环境的平台。VCS 是 Synopsys 公司提供的 FPGA/ASIC 仿真验证平台，支持 VHDL、Verilog、SystemVerilog 等语言[5]。

1.1 VCS 环境准备

编写执行脚本 Makefile 文件对仿真模型和固存控制 FPGA 进行分析、编译、仿真。观察结果输出，检查坏块管理的正确性。在 Makefile 中对 VCS 做仿真设置，VCS 工作过程分为：分析（Analysis）、编译（Elaboration）、仿真（Simulation）[6]。

分析（Analysis）是 VCS 仿真的第一阶段。该阶段对库文件、固存控制 FPGA、仿真模型进行分析、进行语法检查。编译（Elaboration）是 VCS 仿真的第二阶段。该阶段 VCS 使用在分析（Aanlysis）阶段产生的中间文件，建立例化层次并生成可执行的二进制文件。仿真（Simulation）是 VCS 仿真第三阶段。该阶段可通过命令 ./ {OUTPUT} —gui—cm {CM _ COMMAND} —cm _ dir {OUTPUT} 打开仿真界面，观察仿真波形[7]。

1.2　仿真系统结构

在 VCS 平台中通过 Verilog 语言建立仿真模型，如图 1 所示。仿真模型情况如表 1 所示。核心仿真模型是 EEPROM 模型和固存模型。前者用于验证坏块表的维护，并将维护过程记录在数据文件中（eeprom0. dat～eeprom3. dat），后者用于模拟固存的全擦除、擦写、读流程，并在上述工作流程中模拟产生坏块并反馈给固存控制 FPGA，将固存响应过程记录在固存响应过程文件文件中（flashctrl. dat）。遥控指令发送模型用于发送读、写、停等遥控指令，其中指令保存在文件 remote _ control. txt 中。

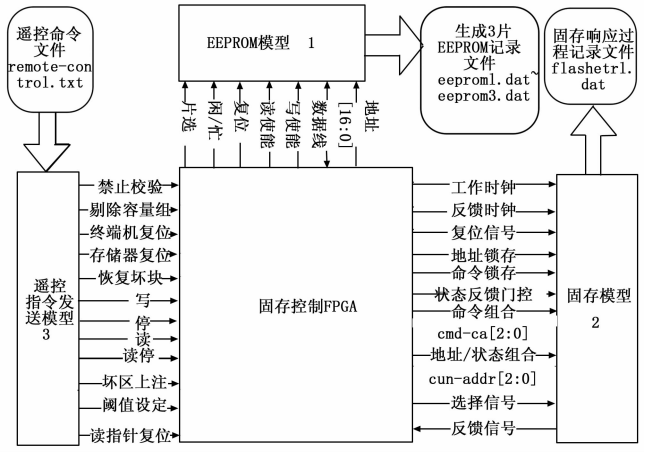

图 1　仿真系统结构

仿真模型详情如表 1 所示，对 EEPROM 模型和固存模型做详细阐述。

表 1　仿真模型表

序号	名称	功能
1	EEPROM 模型	模拟 EEPROM 的 3 个基片，分别模拟初始坏块表、备份坏块表、工作坏块表
2	固存模型	接收信号，并反馈固存工作状态和坏块信息
3	遥控指令发送模型	按照波特率 38400bps，发送串行遥控指令，包括写、读、停等指令

1.2.1　EEPROM 模型

EEPROM 模块包括 3 片 EEPROM，其中 EEPROM1 为初始坏块表，EEPROM2 为备份坏块表，EEPROM3 为工作坏块表。其中，工作坏块表是固存工作时的日常用表，初始坏块表和备份坏块表是工作坏块表出现异常时，作为备用的应急表。三种表的作用将在其维护过程的验证中做详细说明。

建立工作坏块表包括初始化流程和受固存 FPGA 控制的读/写流程。初始化流程的意义是建立初始坏块表，读写流程的意义是在存储板工作流程中，根据坏块变化情况维护 EEPROM 中的工作坏块表。EEPROM 地址范围为 0 ～ 131071，其存储的数据位宽为 8bit，每个 EEPROM 地址对应固存中的一个块地址。每个 EEPROM 地址中的数据位宽为 8bit，判断坏块的阈值为 5，即 EEPROM 地址中“1”的个数小于 5，则认为该 EEPROM 地址对应的固存块为坏块。通过 Verilog 语言可以随机产生坏块也可精确产生坏块。EEPROM 的地址范围为 0～131071，对应固存中 131072 个块。对工作坏块表的维护过程存储在记录文件 eeprom3. dat 中。建立工作坏块表流程如图 2 所示。

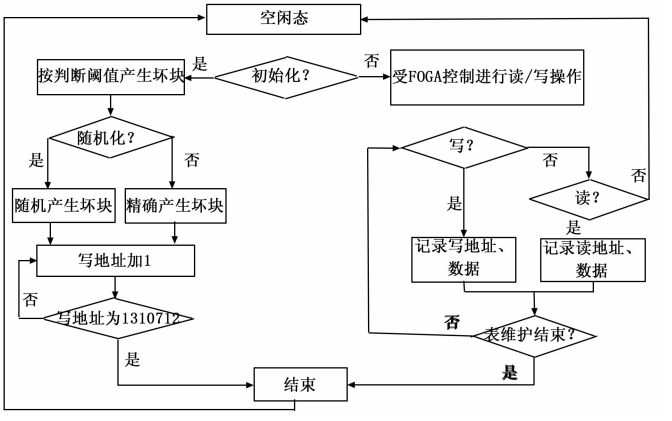

图 2　建立工作坏块表流程

1.2.2　固存模型

使用 Verilog 语言建立固存模型，在模型参数端口可精确设置坏块发生的位置（设置固存的块地址、层地址、页地址）。由固存芯片资料可知，擦写流程中产生坏块的方式与读流程中产生坏块的方式不同。故在固存模型中设置三种模式的坏块反馈机制：第一，擦写固存时坏块判断标准：查询 6 次未收到准备好信息，或收到准备好信息但存储板反馈为坏块时，则认为出现坏块[8-9]。第二，读固存初始坏块表坏块判断标准：查询 1 次收到读出厂标志为低电平（坏块），则认为出现坏块。第三，读固存时坏块判断标准：读操作时当读完一个区块数据后，对收到的汉明校验计数结果进行判断，若汉明校验信号低电平脉冲个数大于遥控指令设定的阈值，则认为该区块为坏块。其中第一、第三种情况出现在固存工作过程中，是固存自发行为。第二种情况受遥控指令控制，当遥控指令发送模型发送“读初始坏块表指令”后，固存读 EEPROM 第 1 层中存储的初始坏块信息。固存坏块产生机制如图 3 所示。

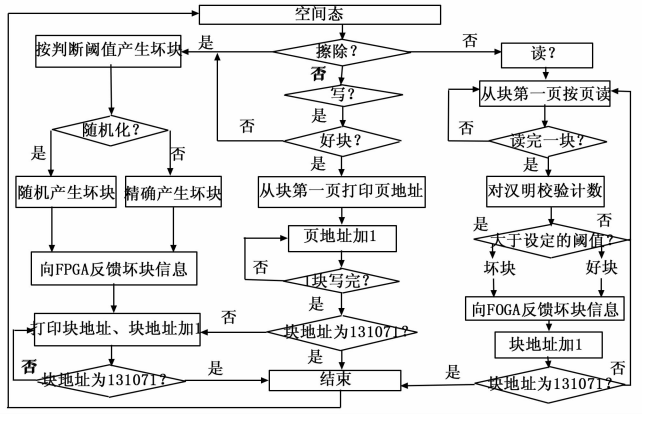

图 3　固存工作流程中坏块产生流程图

2　坏块管理验证

对固存坏块管理的验证包括坏块表维护过程的验证和坏块分布对固存读写流程影响的验证。

2.1　坏块表维护过程验证

仿真系统对固存坏块表维护过程的验证包括建立初始坏块

表、恢复初始坏块表、恢复备份坏块表、建立备份坏块表、坏块取消、坏块上注、工作坏块表下传、写固存时坏块查询及标记、读固存时坏块查询及标记、擦除固存时坏块查询及标记。

1）建立初始坏块表：仿真系统发送“读初始坏块表指令(32’h000018C8)”。仿真系统根据固存模型反馈，将固存中的初始坏块情况写入记录文件 eeprom1.dat。Data 为 8bit 数据，若数据的二进制形式含“1”的个数小于 5，则是坏块，若“1”的个数大于等于 5，则是好块。固存好块地址在 eeprom1.dat 中记录情况（部分）如下所示：

top_tb.MU5：Addr = 000b,Data = 7f；top_tb.MU5：Addr = 001c,Data = df；

top_tb.MU5：Addr = 001f,Data = 7f；top_tb.MU5：Addr = 0024,Data = fd；

2）恢复初始坏块表：仿真系统发送“恢复初始坏块表指令（32’h000018C7)”。FPGA 将初始坏块表 eeprom1.dat 的数据写入工作坏块表 eeprom3.dat。导入数据时进行阈值为 5 的判断，即若数据“1”的个数小于 5，认为是坏块，若数据“1”的个数大于 5，认为是好块。将好块写为全 1，将坏块写为全 0。eeprom3.dat 记录了恢复初始坏块表的全部过程。固存共 131072 个块，仿真系统将整个恢复坏块表流程记录，发现初始坏块表最后一个数据无法写入工作表。

经分析原因如下：在读初始坏块表最后一个地址时，在状态机 state _ prom 的状态 PROM _ RDPROM _ FIRST _ OELOW 态将 wrprom _ nxt _ end 置为高电平，导致写工作坏块表最后一个地址时片选信号无效，无法完成对工作表最后一个地址的写入。针对该问题的修改措施为：将对初始坏块表判最后一个地址的位置由状态 PROM _ RDPROM _ FIRST _ OELOW 移状态 PROM _ WRPROM _ NXT _ END，使得判到初始坏块表最后一个地址时，对工作表的片选信号有效，从而完成对最后一个地址的写入。

3）恢复备份坏块表：仿真系统发送“恢复备份坏块表指令（32’h000018C6)”。仿真系统将备份坏块表中的信息写入工作坏块表，即通过读 eeprom2.dat 写 eeprom3.dat 记录了恢复备份坏块表过程。由于备份表关于地址的控制逻辑与初始表关于地址的逻辑相同，故备份表最后一个地址（131071）也无法被选中，及备份坏块表的最后一个地址信息复发恢复到工作坏块表。修改措施与恢复初始坏块表的修改方法相同。

4）建立备份坏块表：仿真系统发送“建立备份坏块表指令（32’h000018C5)”。仿真系统将工作坏块表信息写入备份坏块表，即读 eeprom3.dat 写 eeprom2.dat。仿真系统同样发现工作坏块表最后一个地址无法写入备份坏块表。修改措施与恢复初始坏块表的修改方法相同。

5）坏块上注：当固存数据传输异常，地面控制站认为固存某地址是坏块时，需要进行坏块上注，在工作坏块表中将固存的坏块信息进行标注。仿真系统发送“坏块上注指令 32’hbf000005”，即向固存地址 05h 上注信息，将其标记为坏块。观察 eeprom3.dat，可见仿真系统对指定的坏块进行了上注。

6）坏块取消：当地面控制站认为固存某地址是好块时，需要进行坏块取消。其工作流程类似于坏块上注。仿真系统发送“坏块取消指令 32’hb5000013”，即向固存地址 13h 上注信息，将其标记为好块。观察 eeprom3.dat，地址 13 h 由坏块(00h) 转变为好块（ffh)。

7）工作坏块表下传：仿真系统发送“坏块表下传指令”，工作坏块表信息会通过遥测下传。

8）写固存时坏块查询及标记：仿真系统发送“写指令”。仿真系统可根据工作坏块表对固存坏块标记情况跳过对应的坏块，只写好块。通过查看仿真系统产生的 eeprom3.dat 中坏块分布信息及 flashctrl.dat 中固存写过程，可以验证 FPGA 跳过了坏块，只写好块。

9）读固存时坏块查询及标记：在仿真系统中令汉明校验开启，令固存模型在读流程中产生坏块，固存坏块信息保存在文件 flashctrl.dat 中，工作坏块表对固存读流程中坏块变化情况保存在文件 eeprom3.dat 中。通过比较 flashctrl.dat 与 eeprom3.dat 可以验证在读流程中坏块产生及标记的正确性。

10）擦除固存时坏块查询及标记：令仿真系统模拟固存在擦除过程中不产生坏块，固存操作记录 flashctrl.dat 中只对工作坏块表 eeprom3.dat 中的好块进行擦除；令固存在擦除过程中产生坏块，eeprom3.dat 根据固存坏块变化情况实时进行变化。

2.2 坏块分布对固存读停的影响

2.2.1 读停问题描述

固存控制 FPGA 要求读固存页地址到达 300 h 时，可自动读停。若不能读停而对相同地址重复读，会造成数据重复、紊乱。令仿真系统中工作坏块表第 0 块、第 4095 块为好块，其余块为坏块（块地址 12d 也为坏块），观察固存能否在页地址 300 h 时自动读停。工作坏块表设置方式如下所示：

```
initial begin
k = 0;
sram_cnt = 0;
repeat(131072) begin
sram[sram_cnt] = 8'h00; //备注:00h 表示坏块
sram_cnt = sram_cnt + 1;
end
for (k = 0;k < 4096; k = k + 1)begin
sram[0] = 8'hff; //备注:将第 0 块,第 4095 块设置为好块,其余均为坏块
sram[4095] = 8'hff; //备注:ffh 表示好块
end
end
```

固存控制 FPGA 在页地址为 300h（768d），即块地址为 768/64=12d（1 块包含 64 页）时配合页读使能信号 rd _ page _ adden 对信号 cntt 进行计数，当 cntt 计到“10”时，令固存读停，源代码下所示。

```
process(rst_n,clk)
  … …
  if(zs_rd_addreq='1')then
  cntt <=(others=>'0');
  elsif (rd_page_adden = '1')and(rd_addr(width downto 0)="000" &. x"0300")then
  cntt <=cntt+1;
  ……
  end process;
  process(rst_n,clk)
  begin
  … …
  elsif (cntt ="10")then
  rd_stop <='1';
```

```
else
rd_stop <='0';
… …
end process;
```

由于仿真系统已将第 1 块～第 4094 块均设置为坏块，故固存页地址跳过了这些地址，当页地址为 300h 时，固存页读地址使能信号 rd _ page _ adden 保持无效，cntt 计数失效，固存读停失败。仿真如图 4 所示。

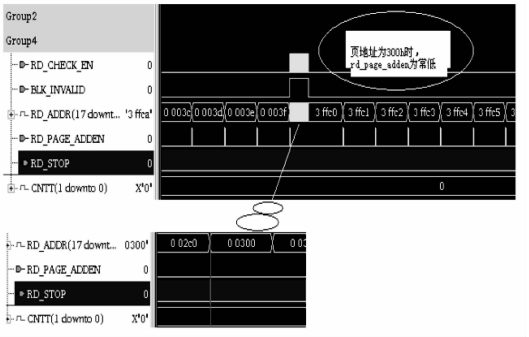

图 4　地址为 300 h 时，信号 RD _ PAGE _ ADDEN 为低电平

2.2.2　读停设计优化

固存读停失败的原因是未考虑坏块对判停位置的影响。针对读停失效的修改方法是增加对坏块的考量，增加坏块标志信号 blk _ invalid 的判断。若固存准备读停时遇到坏块，信号 cntt 正常进行计数，计数到达“10”时，信号 rd _ stop 有效，可令固存读停，优化后的代码如下所示：

```
process(rst_n,clk)
  … …
  if(zs_rd_addreq='1')then
cntt <=(others=>'0'); //备注:当 blk_invalid 为 1 时,表示遇到坏块。
elsif( (rd_page_adden='1')or (blk_invalid ='1'andrd_check_en = '1') )and(rd_addr(width downto 0)="000" & x"0300")then //备注:准备读停时,增加对坏块的考虑
  cntt <=cntt+1;
  … …
end process;
```

2.3　坏块分布对固存写停的影响

2.3.1　写停问题描述

固存在工作时，若写到最后一块，要求可以自动写停。令固存模型的最后一块为坏块，仿真发现固存写到最后一块时，未写停，而是对最后一块重复写，导致数据出错。原因在于，当固存最后一块为坏块时，固存控制 FPGA 会跳过最后一块，最后一页地址" 111" &x" fffff" 无法执行到，即 if 语句条件不成立，写溢出信号 wr _ overflow 无法置高，固存无法写停。出错的代码如下：

```
elseif (wr_addr = "111"&x"fffff") then //备注:32'h7fffff 即 8388608d
wr_overflow <= '1'; //备注:8388608/64 = 13072,即最后一块的末页
```

2.3.2　写停设计优化

针对固存最后一块为坏块的工况，不判最后一块的末页，改为判最后一块的首页，即页地址由 fffff 改为 fffc0，优化后的代码如下所示：

```
elseif (wr_addr = "111"&x"fffc0") then //备注: fffff-fffc0 = 3f,即 63
wr_overflow <= '1'; //备注:8388608/64 = 13072,即最后一块的末页
```

此时 if 语句条件满足，写溢出信号 wr _ overflow 有效(高电平)，固存可写停。

2.4　坏块分布对数据处理安全性的影响

2.4.1　坏块连续分布验证的必要性

固存一般应用在数据处理系统，数据处理系统对数据处理速率比较敏感，在全速率工作时，要求数据流畅传输，不阻塞。但由于固存本身物理特性，写固存前需要对其进行擦除，如果固存连续出现坏块，则需对其进行连续擦除，直到擦到好块为止[10-11]。数据处理系统的载荷输入端不间断地向固存输入载荷数据，如果坏块连续出现，固存一直处于擦除流程，不接收载荷数据，数据会阻塞在固存前端，如前端缓存 FIFO，造成 FIFO 写溢出，导致数据丢失。故有必要验证坏块连续分布对数据处理系统工作安全性的影响。

2.4.2　坏块连续分布验证过程

令固存模型在擦除时连续五次遇到坏块，仿真系统设置方法如下：

```
case (erase_block_addr)
16'h0000,16'h0001,16'h0002,16'h0003,16'h0004: force_sts_fail = 1'b1;
default: force_sts_fail = 1'b0; //备注:force_sts_fail 低电平表示擦除时遇到好块
end case
```

令固存连续出现 5 次坏块时，发现固存数据输出端丢失数据。输出数据记录文件内容（部分）如下所示，发现计数区丢失，表明固存丢失数据。

L1:1acf fc1d 540f 0000 00ff aaaa aaaa aaaa

L2: aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa

L3: aaaa aaaa aaaa aaaa aaaa 3fff 1234 1234

L4: aaaa aaaa aaaa aaaa aaaa 3fff 1234 1234

仿真如图 5 所示。期望的第一帧数据计数区应为“0001”，计数区位置应在数据区“1234”前，但因固存连续出现坏块，导致计数区丢失。

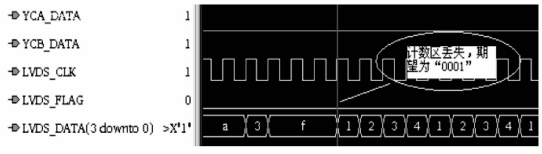

图 5　仿真波形显示丢失计数区

在仿真系统中令坏块连续出现的次数逐渐减小，直到固存输出端数据正常。此时坏块连续出现的次数为 3。

2.4.3　坏块连续分布对数据处理系统安全性的启示

仿真结果表明，当坏块连续出现 4 次及以上时，数据处理系统会丢失数据。为此，设计对坏块的控制需进行两方面的考虑。

1）研读固存芯片手册，向芯片厂商求证固存坏块是否会连续出现 4 次或 4 次以上。若固存芯片厂商不能保证坏块分布可靠性，则需对数据处理系统性能做出限制，如降低载荷输入速率。

(下转第 293 页)