

成都海光集成电路设计有限公司

系统平台部

DramEccErr Injector 操作手册

App	DRAMErrorInjector V1.0
Author	黄胜熙
Email	huangshengxi@higon.com
Version	1.3
LastUpdate	2020/03/07

修订历史

修订前版本	修订内容	完成日期	修订人	修订后版本
	创建文档	2018-05-08	黄胜熙	1.0
	文档内容优化	2019-04-04	张凯	1.1
	注 错 成 功 率 提 升	2019-12-14	张凯	1.2
	环境设置	2020-03-07	张凯	1.3

注：“草稿”状态的文档版本为 0.Y.Z, $Y \geq 0$, $Z > 0$, Y、Z 的数值不断累加；
“正式发布”状态的文档版本为 X.Y, $X \geq 1$, $Y \geq 0$, 且 X、Y 值不断累加；
“正在修改”状态的文档指对“正式发布”后的文档进行修改，文档版本为 X.Y.Z, 其中 X.Y 同修改之
前的文档版本号, $Z > 0$, Z 的数值不断累加

1. 引言

本文档是海光处理器 ECC 错误注入工具 DramEccErrInjector 的操作说明书，本文读者应为软件的使用人员。通过阅读本文档，使用者可理解系统的功能特征，便于进行软件操作。

2. 工具介绍

2.1 目的

本软件产品主要实现了以下功能：

1. 本工具提供了访问、设置 UMC 某些寄存器的功能。
2. 3 种不同的 ECC 错误注入方式。通过这几种不同的错误注入方式，可以在不同情况下，验证 CPU 对于 DRAM 错误的处理机制。
3. ECC 报错的分析验证，确认 CPU 的错误验证机制是否有效。
4. DRAM RAS 相关的设置，可以调整 DRAM ECC 的相关错误处理机制。

2.2 原理

注入工具主要依赖 UMC 中的错误向量寄存器和控制寄存器，实现了如下三种形式的 ECC 错误注入：

- Explicitly Addressed Error Injection
- Persistent Error Injection
- One-shot Error Injection

2.2.1 Explicitly Addressed Error Injection

Explicitly addressed error injection allows diagnostic firmware to inject errors in to DRAM and test the ECC logic.

Explicitly addressed error injection only works on addresses mapped as "WB" memory. The injection process

ensures that the error is injected into the most recent copy of the cacheline (i.e., the injection is coherent).

Explicitly addressed error injection is configured by the following registers:

- DF::DramScrubErrAddrLo
- DF::DramScrubErrAddrHi
- UMC::CH::EccErrInj

- UMC::CH::EccErrInjCtrl

The steps to enable explicitly addressed error injection are as follows:

- Program the desired error pattern into UMC::CH::EccErrInj for each beat of data.
- Enable address-based injection in UMC::CH::EccErrInjCtrl.
- Program DF::DramScrubErrAddrHi with the upper bits of the address to be injected.
- Program DF::DramScrubErrAddrLo with the lower bits of the address to be injected and DF::DramScrubErrAddrLo[ErrInjEn] set to 1. The address must be 64-byte aligned.
- The error injection is complete when DF::DramScrubErrAddrLo[ErrInjDone] is set.

2.2.2 Persistent Error Injection

Persistent error injection allows diagnostic firmware to inject errors into future DRAM writes in a memory

controller. To enable persistent error injection:

- Program the desired error pattern into UMC::CH::EccErrInj for each beat of data.
- Enable persistent error injection using UMC::CH::EccErrInjCtrl.

2.2.3 One-shot Error Injection

One-shot error injection allows diagnostic firmware to inject errors into the next DRAM write in a memory

controller. To enable persistent error injection:

- Program the desired error pattern into UMC::CH::EccErrInj for each beat of data.
- Enable one-shot error injection using UMC::CH::EccErrInjCtrl.

欲获取更详细的参考信息，可查阅 AMD PPR 文档 section 3.7 DRAM RAS Features。

2.3 适用范围

该工具适用于当前海光 CPU 的所有规格，规格如下：

四 die 单路：die id 0,1,2,3(socket 0)

四 die 双路：die id 0,1,2,3(socket 0), 4,5,6,7(socket 1)

单 die 单路：die id 0(socket 0)

单 die 双路：die id 0(socket 0), 4(socket 1)

双 die 双路：die id 0,1(socket 0), 4,5(socket 1)

备注：目前该工具要求 UMC 内部 CS hash 关闭的情况下才能使用基于地址的注入方式。

2.4 环境设置

2.4.1 BIOS 设置

当注入 UCE 时，有两种处理方式：hang 机或者重启，这两种方式可以在 BIOS 中 setup 中进行设置：

Hang 机设置方法：

1. HYGON CBS > UMC Common Options > DDR4 Common Options > Common RAS >Data Poison ->Auto(或 Enable)
2. HYGON CBS > Moksha Common Option > Platform UCE Handling Mode -> Custom Mode

重启设置方法：

1. CBS > UMC Common Options > DDR4 Common Options > Common RAS >Data Poison->Disable
2. HYGON CBS > Moksha Common Option > Platform UCE Handling Mode -> Normal Mode

2.4.2 GRUB 设置

对于目前主流 Linux 操作系统，系统会默认开启 MCE 服务。系统 MCE 处理机制与内存注错工具会产生冲突，因此，在 linux 系统启动之前，不要开启 MCE 服务。以 CentoOS 为例，设置方法如下：

1. 开机至 grub 界面，键入“E”进入 grub 编辑模式
2. 在内核命令行参数中添加 mce=ignore_ce
3. 键入组合键 Ctrl+X 继续引导内核

2.5 运行

本程序为 ELF 格式，运行的软件环境为 linux OS。在 linux commandline 中，执行如下操作：./DramErrorInjector

DramErrorInjector 工具的相关选项信息如下：

选项名字	有无参数	选项含义
-h	无	帮助信息
-E	无	打印示例
-T	无	System address 与 chip info 转换
-V	无	版本信息
-B	无	Verbose 模式
-f	√	配置文件路径

-D	√	Scurbbber 模式设置
-P	√	Poison 模式设置
-R	√	Scrubber rate 设置
-C	无	清楚某些 UMC 寄存器状态
-S	无	展示某些 UMC 寄存器状态
-N	无	显示 ECC error count
-d	√	Die id
-u	√	Umc id
-M	√	注入模式选择, 手动注入
-a	√	System address
-g	无	System assress 由 chip info 生成
-s	√	Chip select
-b	√	Bank
-r	√	Row
-c	√	Column
-z	√	Rank multiple
-A	√	注入模式选择, 自动注入
-t	√	错误类型
-m	√	自动注入的模式
-v	√	注入次数
-p	√	注入向量的生成模式
-0	√	注入的错误向量, 序号 0
-1	√	注入的错误向量, 序号 1
-2	√	注入的错误向量, 序号 2
-3	√	注入的错误向量, 序号 3

表格 1 DramErrorInector 选项说明

3. 使用示例

3.1. 打印帮助信息

获取帮助指令，向用户展示软件支持的功能，并提示软件使用的命令行格式。

指令信息

指令格式	指令功能
./DRAMErrorInjector -h	获取帮助

Table 1 打印帮助信息

3.2. 显示使用示例

显示使用的例子。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -E</code>	显示本软件的使用示例

Table 2 打印帮助信息

3.3. 设置 scrubber redirection

启用或禁用某个 UMC 的 scrubber redirection 功能。

指令信息：

指令格式	指令功能
<code>./DRAMErrorInjector -d 0 -u 0 -e 0</code>	禁用 die 0 umc 0 的 scrubber redirection
<code>./DRAMErrorInjector -d 0 -u 0 -e 1</code>	启用 die 0 umc 0 的 scrubber redirection

Table 3 设置 scrubber redirection

3.4. 设置 data poison

启用或禁用系统的 data poison 功能。

指令信息：

指令格式	指令功能
<code>./DRAMErrorInjector -P 0</code>	禁用系统的 data poison
<code>./DRAMErrorInjector -P 1</code>	启用系统的 datapoison

Table 4 设置 scrubber redirection

3.5. 设置 scrubber rate

设置某个 die 内某个 umc 的 scrubber rate。scrubber rate 的有效值为 0~15，对应的含义如下

<pre>enum DRAM_SCRUB_RATE { RATE_640_NS, RATE_1_28_US, RATE_2_56_US, RATE_5_12_US, RATE_10_2_US,</pre>
--

```
RATE_20_5_US,  
RATE_41_0_US,  
RATE_81_9_US,  
RATE_163_8_US,  
RATE_327_7_US,  
RATE_655_4_US,  
RATE_1_31_MS,  
RATE_2_62_MS,  
RATE_5_24_MS,  
RATE_10_49_MS,  
RATE_20_97_MS,  
SCRUB_RATE_MAX_CNT,//16  
};
```

指令信息

指令格式	指令功能
./DRAMErrorInjector -d 0 -u 0 -R 7	设置 die 0 umc 0 的 scrubber rate 为 7

Table 5 scrubber rate

3.6. 清除 UMC Status registers

在系统运行过程中，UMC 的 MCA 以及 IP 寄存器会记录机器的相关状态信息。本指令会清除这些信息，把寄存器重新初始化。

指令信息

指令格式	指令功能
./DRAMErrorInjector -C	清除 die[0:7] umc[0:1]的 mca status 寄存器状态以及 Dram Ecc Error Count 寄存器 Ecc 计数

Table 6 清除 UMC MCA Status registers

3.7. 展示 RAS 相关的寄存器信息

系统中有诸多与 RAS 相关的状态寄存器，本指令可显示他们的实时状态。

指令信息

指令格式	指令功能
./DRAMErrorInjector -S	展示 die[0:7] umc[0:1]的 mca status 寄存器状态

Table 7 展示 RAS 相关的寄存器信息

3.8. 显示 UMC 的 ECC 错误数目

本指令可显示某个 UMC 上发生的 ECC 错误数量。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -N</code>	展示 die[0:7] umc[0:1]的 mca status 寄存器状态以及 Dram Ecc Error Count 寄存器 Ecc 计数

Table 8 显示 UMC 的 ECC 错误数目

3.9. 手动注入 ECC 错误

- 注入的类型依靠-M 参数控制，参数值与注入模式对应关系为:
- 0 : persistent mode
 - 1 : one-shot mode
 - 2 : address-based mode

3.9.1 手动注入 ECC 错误(one-shot mode)

本指令向 DRAM 注入一次 ECC 错误向量。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -d 0 -u 1 -M 1 -0 0x00010001 -1 0x00010001 -2 0x00010001 -3 0x00010001</code>	向 die 0 umc 1 注入一次 ECC 错误

Table 9 DRAM ECC 错误 One-shot 注入

3.9.2 手动注入 ECC 错误(persistent mode)

本指令向 DRAM 持续注入 ECC 错误向量。如果错误向量注入到内核上下文，有可能造成系统崩溃。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -d 0 -u 1 -M 0 0x00010001 -1 0x00010001 -2 0x00010001 -3 0x00010001</code>	向 die 0 umc 1 持续 ECC 错误

Table 10 DRAM ECC 错误 persistent 注入

3.9.3 手动注入 ECC 错误(address-based mode)

本指令可指令一个明确的地址，向 DRAM 中的该地址注入一次 ECC 错误向量。如果该地址位于内核上下文，有可能造成系统崩溃。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -d 0 -u 1 -M 2 -a 0x11110000 -0 0x00010001 -1 0x00010001 -2 0x00010001 -3 0x00010001</code>	向 die 0 umc 1 的地址 0x11110000 注入错误

Table 11 DRAM ECC 错误 address-based 注入

3.10. 自动注入 DRAM ECC 错误 one-shot

注入模式依靠 -A 参数控制，参数值与注入模式对应关系为：

- 0 : persistent mode
- 1 : one-shot mode
- 2 : address-based mode

本指令可自动向 DRAM 中的该地址注入多次 ECC 错误向量。如果注入地址位于内核上下文，有可能造成系统崩溃。可通过命令行参数，选择控制注入模式、注入错误类型等因素。

指令信息

指令格式	指令功能
<code>./DRAMErrorInjector -d 0 -u 0 -A 1 -t 0 -m 0 -v 100 -g 0</code>	向 die 0 umc 0 注入 one-shot 错误 100 次

Table 12 DRAM ECC 错误 address-based 注入

3.11. 向指定 dimm 注入错误

在测试中，可能需要想主板上的某根 dimm 注入 ECC 错误，此时依靠 chip select 参数 -s 来区分 dimm，对应关系如下：

对于 1ranks 的 dram

Dram ranks	Dimm0	Dimm1
-s	0	2

表格 2 1-rank DRAM chipselect 与 dimm 对应关系

对于 2ranks 的 dram

Dram ranks	Dimm0	Dimm1
-s	0, 1	2, 3

表格 3 2-rank DRAM chipselect 与 dimm 对应关系

3.12. 查看版本号以及更新 log

在工具使用过程中，如果想查看工具版本号、编译日期，以及本次更新 log，命令格式如下：

指令格式	指令功能
<code>./DramErrorInjector -V</code>	查看使用工具的版本信息

表格 4 查看工具的版本信息

4. 故障排查

4.1. 软件故障一：无法注错

某些情况下，使用 `DramErrorInjector` 可能无法注错，此时可能与 `MSR_MCG_CAP` 等寄存器设置有关。

使用 `-G` 选项，查看当前系统跟注错有关的寄存器设置。正确的设置应如下图所示：

```
+ DRAMErrorInjector git:(master) sudo ./DramErrorInjector -G
MSR_PFEH_CFG      : 0X0. MSR_PFEH_CFG.PfehEnable:0x0
MSR_PEEF_CLOAK_CFG : 0x0
MSR_HW_CFG        : 0x9040011, MSR_HW_CFG.MC_STATUS_WR_EN:1

MSR_MCG_CAP       : 0x117, MCG_CAP.McgCtIP:1, MCG_CAP.Count:23.
MSR_MCG_STAT      : 0x0

MSR_MCG_CTL       : 0xffffffffffffef
MCG BANK 15 [UMC] : Enable
MCG BANK 16 [UMC] : Enable

cpuid:0
MCA_CTL_UMC[ die0 umc0 ] : 0x0. MCA_CTL_UMC.DramEccErr : 0x0, MCA_CTL_UMC.WriteDramEccErr : 0x0 .
cpuid:0
MCA_CTL_UMC[ die0 umc1 ] : 0x0. MCA_CTL_UMC.DramEccErr : 0x0, MCA_CTL_UMC.WriteDramEccErr : 0x0 .
+ DRAMErrorInjector git:(master)
+ DRAMErrorInjector git:(master)
```

Figure 1 系统 MCA 相关寄存器设置

`MSR_MCG_CTL` 的值一般为： `0xffffffffffffef`。若 `MCG BANK 15、16` 没有 `ENABLE`，可使用如下指令将他们使能：

```
+ DRAMErrorInjector git:(master)
+ DRAMErrorInjector git:(master) sudo ./DramErrorInjector -L 0xffffffffffffef
```

Figure 2 Enable MCG BANKS

`MCA_CTL_UMC[die0 umc0]`一般为 `0x3f`。可使用如下指令设置：

```
+ DRAMErrorInjector git:(master) sudo ./DramErrorInjector -l 0x3f
set value:3f to MCA_CTL_UMC
+ DRAMErrorInjector git:(master)
```

Figure 3 Enable MCA_CTL_UMC

4.2. 软件故障二：注错故障之后没有 Ecc Error 产生

某些情况下，使用 DramErrorInjector 进行注错，但是 os 或者 Ecc Error 计数统计并没有中断打印或者错误计数产生。这种情况大概率并非注入失败，而是由于注入的内存位置没有被系统读写到，因此无法触发 Ecc。

对于这种情况，建议进行注错的时候，同时对相应内存区域进行压力读写测试，比如 memtester 工具等。这样更容易达到我们想要的测试结果。

4.3. 软件故障三：工具注错成功率低

注入工具目前实现了如下三种形式的 ECC 错误注入：

- Explicitly Addressed Error Injection
- Persistent Error Injection
- One-shot Error Injection。

其中，Persistent 和 One-shot 注错方式是通过跟随当前 Soc 对 DDR 的写操作进行注入的，具有随机性，因此 UMC 无法知道注入错误的位置。Addressed 方式是根据物理地址进行针对性注入，地址是可知的。

对于 One-shot 注错方式，在注错之前需要提前运行 memtester 工具，并且控制测试内存范围尽量比较小，测试次数多，如“memtester 20M 500”此类命令，同时使用参数 -v 注入尽量多的次数。

对于 Addressed 注错方式，注入成功率可以达到 100%，注入地址推荐 0x1ff00, 0x2ff00, 0x3ff000, 0x4ff00, 0x5ff00 等。由于工具内部涉及物理地址到虚拟地址的转换，部分物理地址是无法注错的，建议使用推荐地址。