



Error INjection support for RAS on RISC-V architecture

Himanshu Chauhan

Engineer, Senior Staff, Qualcomm India Pvt. Ltd.

@qualcomm

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries.
Qualcomm patents are licensed by Qualcomm Incorporated.



APEI - RAS Error Injection (EINJ)

- A generic mechanism through which OS can inject hardware errors
- OS is not required to have platform specific knowledge to inject the errors
- EINJ can be used to test the error handling stack of the OS
- RAS Agents - software entities running on platform microcontroller or application processor have all details of handling error sources
- EDK2 dynamically creates the EINJ table with the help of the RAS agents
- The EINJ table is made up of injection actions like BEGIN_INJECTION_OPERATION, GET_ERROR_TYPE
- Each action consists of one or more injection instruction like READ_REGISTER, WRITE_REGISTER
- READ_REGISTER, WRITE_REGISTER instructions have an ACPI GAS associated to it.

The Problem:

1. RAS agent can be present at various places within a platform.
2. The error sources may not be RERI compliant and would need RAS agent assisted error injection.
3. So there is a need to standardize the way to communicate with the RAS agent.

Proposed Solution:

1. Add FFH as an ASID for EINJ
2. When the RAS agent is not accessible directly, FFH entry can encode a way to reach it.
3. FFH entry, similar to other actions, allows read/write on a given register encoded within the FFH entry
4. EINJ Table can also keep a mix of the ASIDs. For example, set/get error type can be MMIO but execution and Injection can be FFH to demand action from firmware

BEGIN_INJECTION_OPERATION INSTRUCTION: NO-OP
GET_ERROR_TYPE INSTRUCTION: READ_REGISTER ASID: AML_AS_SYSTEM_MEMORY/FFH
SET_ERROR_TYPE/WITH_ADDR INSTRUCTION: WRITE_REGISTER ASID: AML_AS_SYSTEM_MEMORY/FFH
EXECUTE_OPERATION INSTRUCTION: WRITE_REGISTER ASID: AML_AS_SYSTEM_MEMORY/FFH
CHECK_BUSY_STATUS INSTRUCTION: READ_REGISTER ASID: AML_AS_SYSTEM_MEMORY/FFH
INJECTION_ERROR_OPERATION INSTRUCTION: READ_REGISTER ASID: AML_AS_SYSTEM_MEMORY/FFH
END_OPERATION INSTRUCTION: NO-OP

APEI – EINJ Table

RISC-V FFH Encoding for EINJ

- A Generic Address Structure (GAS) is used to describe an EINJ read/write operation
- The GAS points to an FFH address
- A new FFH encoding has been proposed in RISC-V FFH Specification

Bits[63:60] (Type)	Bits[59:32]	Bits[31:0]	Description
0x0	SBI MPXY Channel ID	REGISTER_ID	Read/write operation on a GAS entry. The REGISTER_ID is as per the RPMI RAS Agent service group.

APEI EINJ – Linux Kernel Support

- Linux kernel has support for parsing EINJ table and do instructions as mentioned in the injection entries
- It publishes EINJ actions as files under /sys/kernel/debug/apei/einj/
- A simple cat/echo on these files can be used to inject an error

```
/sys/kernel/debug/apei/einj # echo 1 > error_type
/sys/kernel/debug/apei/einj # echo 1 > param1
/sys/kernel/debug/apei/einj # echo 0 > nottrigger
/sys/kernel/debug/apei/einj # echo 1 > error_inject
/sys/kernel/debug/apei/einj # [ 2154.462290] {1}[Hardware Error]: Hardware error from APEI Generic Hardware Er
ror Source: 1
[ 2154.462713] {1}[Hardware Error]: event severity: recoverable
[ 2154.462965] {1}[Hardware Error]: Error 0, type: recoverable
[ 2154.463215] {1}[Hardware Error]: section_type: general processor error
[ 2154.463481] {1}[Hardware Error]: processor_type: 3, RISCv
[ 2154.463678] {1}[Hardware Error]: processor_isa: 6, RISCv64
[ 2154.463863] {1}[Hardware Error]: error_type: 0x08
[ 2154.464118] {1}[Hardware Error]: micro-architectural error
[ 2154.464356] {1}[Hardware Error]: operation: 3, instruction execution
```

Thank you

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

© Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm and Snapdragon are trademarks or registered trademarks of Qualcomm Incorporated.
Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patents are licensed by Qualcomm Incorporated.

Follow us on: [in](#) [X](#) [@](#) [v](#) [f](#)

For more information, visit us at qualcomm.com & qualcomm.com/blog

