**# Title:**

RASF Gen2

**# Status:**

Draft

**# Document:**

ACPI Specification Version 6.next

**# License:**

SPDX-License-Identifier: CC-BY-4.0

**# Submitter:**

* Harb Abdulhamid (Ampere)
* Thanu Rangarajan (Arm)
* Samer El-Haj Mahmoud (Arm)
* TianoCore Community (<https://www.tianocore.org>)

**# Summary of the change**

This ECR is related to a proposal to upgrade the current RASF table. The upgraded RASF table offers a richer set of interfaces that enables the OS to perform improved PFA and scrub control of system memory. Additionally, the upgraded RASF table:

* includes provisions for extending to other types of RAS features that might apply to other components in the system such as caches.
* fixes issues that were present in RASF Gen 1.

The enhanced scrub controls are required to address RAS requirements of modern scalable platforms that have complex memory systems with a multitude of memory controllers that are in turn associated with NUMA domains. It is also common for RAS errors related to memory to be associated with NUMA domains, where the NUMA domain functions as a FRU identifier. As a result, PFA tools and error recovery algorithms that such tools might use could benefit from being able to control memory scrubbing at a NUMA domain granularity. The updated RASF table enables this capability.

Specifically, the ECR includes the following changes:

1. Updated signature of the shared memory of the subchannel in accordance with the definitions in Chapter 14 (Platform Communication Channel) of the ACPI specification, which dictate that the shared memory must have a signature of type “PCCx”, where x is the subchannel index in the PCCT. RASF Gen1 violates this rule.

2. The ability to scale RAS functions to multiple instances of the same underlying component, which is offered in the form of more than one PCC channel per RAS function, where a channel can be dedicated to a given component instance.

3. Independent memory scrubbing controls for each NUMA domain, based on #2.

4. Generalization to system components other than memory.

5. Provision for background (or patrol) scrubbing, distinct from on-demand scrubbing for a specific region.

The changes are done in a manner to ensure backward compatibility with RASF Gen1.

**# Benefits of the change**

The use of RASF Gen2 enables OSPM to perform improved PFA in a given system that supports it.

**# Impact of the change**

Platforms firmware will have to support the new format of RASF. OS’s require a driver to support RASF. These are both fundamentally new code.

**# Detailed description of the change [normative updates]**

Existing text

New text

Deleted Text

**5.2.20 ACPI RAS Feature Table (RASF)**

…

**Table x.xx RASF Table Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Header | | | |
| Signature | 4 | 0 | Signature is set to ‘RASF’ for RAS Feature Table. |
| … | … | … | … |
| Revision | 1 | 8 | ~~1~~ 2 |
| … | … | … | … |
| Reserved | 2 | 36 | Reserved, should be zero. |
| Number of PCC descriptors | 2 | 38 | Number of PCC descriptors. |
| RASF Platform Communication Channel Identifier (PCC) Descriptor List | 12N\*8 | 3640 | List of PCC descriptors. |

**5.x.x.x Common Definitions**

**5.x.x.x.x RASF Platform Communication Channel Descriptor**

The RASF PCC descriptor specifies the PCC sub-space associated with a specific RAS function. The RAS function type specifies the RAS function.

**Table x.xy RASF Platform Communication Channel Descriptor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| PCC Identifier | 1 | 0 | Identifier of the RASF Platform Communication Channel. OSPM should use this value as an index into the subspace array within the PCCT table. |
| Reserved | 2 | 1 | Reserved, must be zero. |
| Function Type | 1 | 3 | RASF function type. RAS function types are defined in **5.x.jk**. |
| Instance | 4 | 4 | Identifier for the system component instance that this RAS function is associated with. |

**Table 5.x.jk RASF Function types**

|  |  |
| --- | --- |
| **RAS Function Type** | **Description** |
| 0x00 | RAS functions related to memory or memory-specific RAS features. |
| 0x01-0x7F | Reserved for future standard RAS function types defined by this specification. |
| 0x80-0xFF | Vendor-defined RAS function types. |

**Table x.xy RASF Platform Communication Channel Shared Memory Region**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Signature | 4 | 0 | The PCC Signature of 0x52415346 (corresponds to ASCII signature of RASF)  The PCC signature. The signature of a subspace is computed by a bitwise-or of the value 0x50434300 with the subspace ID. For example, subspace 3 has the signature 0x50434303. |
| Command | 2 | 4 | … |
| … | … | … | … |

**5.x.x.x.x Platform RAS Capabilities**

…

**Table 5.x.xy Platform RAS Capabilities Bitmap**

|  |  |  |
| --- | --- | --- |
| **Bit** | **RAS Feature** | **Description** |
| 00 | Hardware based patrol scrub supported  Feature 1 | Indicates that the platform supports hardware based patrol scrub of DRAM memory  RAS Feature 1 |
| 11 | Hardware based patrol scrub supported and exposed to software  Feature 2 | Indicates that the platform supports hardware based patrol scrub of DRAM memory and platform exposes this capability to software using this RASF mechanism  RAS Feature 2 |
| … | … | … |
| 127  2-127 | *Reserved*  Feature 128 | *Reserved for future use*  RAS Feature 128 |

**5.x.x.x Memory-specific RASF – Function Type 0**

Memory-specific RASF functions apply to RAS capabilities, features and operations that are specific to memory. These features may be provided through one or more PCC sub-spaces. RASF sub-spaces for memory-specific RASF functions have a Function Type of 0x00 (Memory).

**Table x.ww Memory-specific RASF Platform Communication Channel Descriptor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| PCC Identifier | 1 | 0 | Identifier of the RASF Platform Communication Channel. OSPM should use this value as an index into the subspace array within the PCCT table. |
| Reserved | 2 | 1 | 0 |
| Function Type | 1 | 3 | 0x00: Memory. See **5.x.jk**. |
| Instance | 4 | 4 | Proximity domain that this RAS function is associated with. This field must match the ACPI SRAT table definitions. See **5.2.16**. |

**Table 5.x.xy Platform RAS Capabilities Bitmap for Memory-specific RASF**

|  |  |  |
| --- | --- | --- |
| **Bit** | **RAS Feature** | **Description** |
| 0 | Hardware based patrol scrub feature | Indicates that the platform supports hardware based patrol scrub of DRAM memory |
| 1 | Hardware based patrol scrub feature with software controllability | Indicates that the platform supports hardware based patrol scrub of DRAM memory and platform exposes this capability to software using this RASF mechanism |
| 2-127 | *Reserved* | *Reserved for future use* |

**Table x.xw Parameter Block Structure for PATROL\_SCRUB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Type | 2 | 0 | 0x00001 – Hardware-based Patrol scrub RAS feature with software controllability |
| Version | 2 | 2 | Byte 0 – Minor Version  Byte 1 – Major Version  For this format of the parameter block, this field should be set to 0x0001. |
| Length | 2 | 4 | Length, in bytes of the entire parameter block structure |
| Patrol Scrub Command  (INPUT) | 2 | 6 | 0x01 - GET\_PATROL\_PARAMETERS  0x02 - START\_PATROL\_SCRUBBER  0x03 – STOP\_PATROL\_SCRUBBER |
| Requested Address Range  (INPUT) | 16 | 8 | OSPM Specifies the BASE (Bytes 7-0) and SIZE (Bytes 15-8) of the address range to be patrol scrubbed. If OSPM requests default scrubbing through Bit 0 of the Configure patrol scrubbing field, then this field must be ignored by the platform.  OSPM sets this parameter for the following commands:  GET\_PATROL\_PARAMETERS  START\_PATROL\_SCRUBBER |
| Actual Address Range  (OUTPUT) | 16 | 24 | The platform returns this value in response to GET\_PATROL\_PARAMETERS. The platform calculates the nearest patrol scrub boundary address from where it can start. This range should be a superset of the Requested Address Range.  This field must be ignored by the OSPM if it is being returned in response to a request to enable default scrubbing through Bit 0 of the Configure patrol scrubbing field.  BASE (Bytes 7-0) and SIZE (Bytes 15-8) of the address |
| Flags (OUTPUT) | 2 | 40 | The platform returns this value in response to GET\_PATROL\_PARAMETERS  Bit [0]: Will be set if patrol scrubber is already running for address range specified in “Actual Address Range”  Bits [3:1]: Current Patrol Speeds scrub rate index, if Bit [0] is set  000b – Slow  100b – Medium  111b – Fast  All other combinations are reserved.  Bits[4:11]: Bitmap of supported scrub rates, where Bit 4+n corresponds to scrub rate index n:  Bit[4]: Scrub rate = Slow  Bit[8]: Scrub rate = Medium  Bit[11]: Scrub rate = Fast  Other bit values are reserved.  Bits [15:4]: RESERVED  Bits[15:12]: RESERVED |
| Requested Speed Configure Scrub Parameters (INPUT) | 1 | 42 | The OSPM sets this field as follows, for the START\_PATROL\_SCRUBBER command:  Bit [0]: Will be set if patrol scrubber is already running for address range specified in “Actual Address Range”  Bit[0]: Request default patrol scrubbing.  Bits [3:1]: Requested Patrol Speeds scrub rate, specified as index into the supported patrol scrub rates list.  000b – Slow  100b – Medium  111b – Fast  All other combinations are reserved.  Bits [7:3]: RESERVED |