**# Title:**

ACPI Device Configuration Object - Registers Access Restriction

**# Status:**

Draft

**# Document:**

ACPI Specification Version 6.next

**# License:**

SPDX-License-Identifier: CC-BY-4.0

**# Submitter:**

* Samer El-Haj-Mahmoud, ARM
* TianoCore Community (<https://www.tianocore.org>)

**# Summary of the change**

Add new device configuration object method to return register access type

**# Benefits of the change**

Some systems have devices that do not support 64-bit MMIO transactions. Access to 64-bit registers needs to be done using separate 32-bit transfers.  
  
One examples is the XHCI controller on the RPi4. But this exists on other types of controllers on other systems.  
  
One solution is to define an ACPI AML for the firmware to describe to the OS this device register access limitation in a generic way.

**# Impact of the change**

New ACPI method needs support in Firmware, OSes, and ACPICA.

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

* Changes in **yellow**
* Insertions in **green**
* Removals in **~~red~~**
* References that need fixup in blue

6.2 Device Configuration Objects

…

Table 6-193 Device Configuration Objects

|  |  |
| --- | --- |
| **Object** | **Description** |
| … | … |
| \_RAR | Object that specified the register access restrictions for a device |

**6.2.19 \_RAR (Register Access Restriction)**

An optional object that returns register access restrictions of a device. On platforms which may have register access restrictions for some devices, \_RAR can be used to communicate to the OS the register access restriction type. For example, a value of 1 is used to indicate to the OS that 32-bit register access must be used for 64-bit addressed registers.

Arguments

None

Return Value

An **Integer** indicating the device’s register access type:

1 – Must use 32-bit register access (for 64-bit registers)

Other values– Reserved

**6.2.19.1 \_RAR Example ASL:**

Scope (\\_SB) {

…

Device (XHCI) {

…

Name (\_RAR, ONE) // Must use 32-bit register access

…

}

}