**# Title:**

Clarify Time and Alarm Device methods requirements

**# Status:**

Draft

**# Document:**

ACPI 6.4 specification

**# License:**

SPDX-License-Identifier: CC-BY-4.0

**# Submitters:**

* Samer El-Haj-Mahmoud, ARM
* Jacob Smith, Microsoft
* Yinghan Yang, Microsoft
* Brian Nezvadovitz, Microsoft
* TianoCore Community ([https://www.tianocore.org](https://www.tianocore.org/))

**# Summary of the change**

Clarify some of the interface requirements of the Time and alarm methods [https://uefi.org/specs/ACPI/6.4/09\_ACPI-Defined\_Devices\_and\_Device-Specific\_Objects/ACPIdefined\_Devices\_and\_DeviceSpecificObjects.html#time-and-alarm-device](https://uefi.org/specs/ACPI/6.4/09_ACPI-Defined_Devices_and_Device-Specific_Objects/ACPIdefined_Devices_and_DeviceSpecificObjects.html)

**# Benefits of the change**

Gives clarity to implementations to meet the specification goals. Some background:

* \_GWS/\_CWS/\_STP/\_STV/\_TIP/\_TIV no longer required, but dependent on if bit 0 is set or not. It doesn’t seem to make sense that the \_GRT/\_SRT methods are only required if bit 2 is set, but all the alarm related ones are required unconditionally when time only TADs are possible.
* The wording of the spec is not definitive, but it does seem to hint DC wake as an optional/add-on to the more basic AC only wake. This implies that DC wake should be supported only if AC wake is supported.
* It makes sense to require that if wake from some Sx state is supported, then the device also supports wake from the S(x-1) state. Along with if DC wake is supported on some Sx state, then AC wake is also supported for that Sx state.
* Another clarification is renaming bits 0 and 1 to specific “supports wake from S3”. Right now they are documented as generic “AC/DC wake implemented”. Prior description mentions that if these are set then the device must support wake from S3, but S4/S5 is optional. Since there are specific bits for S4/S5 support, the first two bits only effectively describe support for wake from S3.
* It does not make sense to support setting a wake state and not getting the status

**# Impact of the change**

* No impact to existing OSes that support Time and Alarm Device. OSes may want to take advantage of the new language to detect the support for the various TAD object elements using \_GCP.
* Platform firmware implementations may need to follow the new guidance if there is mismatch based on prior interpretation of the spec.
* ACPI compliance tests (e.g. FWTS) need to be updated to applyh the clarified rules

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

* Insertions in **green**
* Removals in ~~red~~

**9.18 Time and Alarm Device**

…

**9.18.2 \_GCP (Get Capability)**

This object is required and provides the OSPM with a bit mask of the device capabilities. The device can implement

the time function in addition to the wake function. The capabilities bitmask will indicate to the OSPM what support is implemented. If the platform implements both AC and DC timers then it is capable of waking up based on the power source.

**Arguments:(0)**

**Return Value:**

A 32-bit integer containing a result bitmask as follows:

Bit [0] - 1 = AC wake implemented, 0 = not supported

Bit [1] - 1 = DC wake implemented, 0 = not supported

Bit [2] - 1 = Get/Set real time features implemented, 0 = not supported

Bit [3] - 1 = Real time accuracy in milliseconds, 0 = Real time accuracy in seconds

Bit [4] - 1 = \_GWS returns correct values for wakes from S4/S5 caused by timer. 0 = not supported

Bit [5] - 1 = Wake supported from S4 on AC, 0 = Wake not supported from S4 on AC

Bit [6] - 1 = Wake supported from S5 on AC, 0 = Wake not supported from S5 on AC

Bit [7] - 1 = Wake supported from S4 on DC, 0 = Wake not supported from S4 on DC

Bit [8] - 1 = Wake supported from S5 on DC, 0 = Wake not supported from S5 on DC

Bit [9] to Bit [31] are reserved and must be 0.

**Note:** The following rules apply for the \_GCP returned value:

* If wake on DC is supported (bit 1), then wake from AC (bit 0) must be supported
* If wake on AC from S5 is supported (bit 6), then wake on AC from S4 must be supported (bit 5)
* If wake on AC from S4 is supported (bit 5), then wake on AC must be supported (bit 0)
* If wake on DC from S5 is supported (bit 8), then wake on DC from S4 must be supported (bit 7)
* If wake on DC from S4 is supported (bit 7), then wake on DC must be supported (bit 1)
* If wake on DC from S4 is supported (bit 7), then wake on AC from S4 must be supported (bit 5)
* If wake on DC from S5 is supported (bit 8), then wake on AC from S5 must be supported (bit 6)
* If wake from S4/S5 is supported (bits 5-8), then \_GWS must be supported (bit 4)

…

**9.18.5 \_GWS (Get Wake alarm status)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ enables the OSPM to read the status of wake alarms.

…

**9.18.6 \_CWS (Clear Wake alarm status)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ enables the OSPM to clear the status of wake alarms.

…

**9.18.7 \_STP (Set Expired Timer Wake Policy)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ sets the expired timer wake policy.

…

**9.18.8 \_STV (Set Timer Value)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ sets the timer to the specified value.

…

**9.18.9 \_TIP (Expired Timer Wake Policy)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ returns the current expired timer wake policy setting of the specified timer.

…

**9.18.10 \_TIV (Timer Values)**

This object is required if the capabilities bit 0 is set to 1. It ~~and~~ returns the remaining time of the specified timer before it expires.

…