ACPI Specification for Status Override Table

Specification: LINARO-0002

Authors: Al Stone <al.stone@linaro.org>, Graeme Gregory <graeme.gregory@linaro.org>, Parth Dixit<parth.dixit@linaro.org>

Revision History:

Version	Date	Comments
0.1	28 March 2014	Initial draft.
0.2	4 December 2014	First revision
0.3	6 January 2015	Second revision

Title:

Define a list of ACPI namespace names that are to be ignored by the OSPM

License:

Problem Statement:

Hypervisors control which devices are allocated to which virtual machines. If ACPI tables created by platform firmware are passed directly from a hypervisor to a new virtual machine, the virtual machine can see all of the devices available to the hypervisor. This would be an incorrect view for the new virtual machine.

By the same token, embedding the ability to create ACPI tables on the fly within a hypervisor adds more complexity and increases the resources needed by the hypervisor to run. To be clear, this is not a problem for the straightforward tables – such as the FADT, MADT, SRAT, or others – but is a problem for the DSDT and SSDT that contain AML byte codes. Generating AML byte

codes on the fly would require the use of an ASL compiler, the most complex part of the tools needed to create ACPI tables.

Hence, what we would like to be able to do is modify the ACPI tables used by virtual machines without having to use an ASL compiler to do so.

The Status Override Table:

This specification lays out the structure and content of a new ACPI table – the Status Override Table (STAO). This table, while not currently part of the ACPI specification, may or may not become part of the specification at a later date.

The purpose of the STAO is to list devices in the ACPI namespace that are to be ignored by the OSPM. For example, if the STAO contains the name "_SB.BUS0.DEV3", the OSPM must operate as if that device does not exist – searches for the name must fail, and any use of methods provided by that device must fail as if the device does not exist.

The structure of the STAO is defined in Table 1.

Table 1. Status Override Table

Field	Byte Length	Byte Offset	Description
Signature	4	0	STAO' to indicate that this is the Status Override Table.
Length	4	4	Length in bytes of the entire STAO
Revision	1	8	1 – current revision level of the STAO
Checksum	1	9	Standard ACPI table checksum; i.e., the entire table must sum to zero
OEMID	6	10	OEM ID
OEM Table ID	8	16	OEM model ID
OEM Revision	4	24	OEM Revision of the STAO for the OEM Table ID

Creator ID	4	28	Vendor ID of the utility that created the table
Creator Revision	4	32	Revision of the utility that created the table
UART	1	36	Flag to indicate that OSPM should ignore UART defined in SPCR table.
Name List [n]	-	37	A list of ACPI strings, where each string is the full path name in the ACPI namespace of the device that the OSPM is to ignore.

The STAO structure is very simple: a standard ACPI table header, followed by a list of ACPI namespace names.

Example:

Since the STAO is not a standardized ACPI table, creating one with the ACPICA tool iasl looks a little different than it would for other tables. Below, we define a simple table with a few names to be ignored by the OSPM:

```
/*
 * Template for [STAO] ACPI Table
 * Format: [ByteLength]  FieldName : HexFieldValue
 */

[0004]                    Signature : “STAO” // Status Override Table
[0004]                 Table Length : 00000000
[0001]                     Revision : 01
[0001]                     Checksum : 00
[0006]                       Oem ID : "LINARO"
[0008]                 Oem Table ID : "TEMPLATE"
[0004]                 Oem Revision : 00000000
[0004]              Asl Compiler ID : "INTL"
```

[0004] Asl Compiler Revision : 20140214

[0001] UART :1
Label : BeginStao
String : "_SB0.BUS0.DEV1"
String : "_SB0.BUS0.DEV2
String : "_SB0.BUS1.DEV1.DEV2"
String : "_SB0.BUS1.DEV2.DEV2"