Device Graphs
Using _DSD to represent arbitrary graphs

Oct 2018
Revision 1.0
The material contained herein is not a license, either expressly or impliedly, to any intellectual property owned or controlled by any of the authors or developers of this material or to any contribution thereto. The material contained herein is provided on an "AS IS" basis and, to the maximum extent permitted by applicable law, this information is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses and of lack of negligence, all with regard to this material and any contribution thereto. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The Unified EFI Forum, Inc. reserves any features or instructions so marked for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION AND ANY CONTRIBUTION THERETO.

IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR ANY CONTRIBUTION THERETO BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Copyright 2018 Unified EFI, Inc. All Rights Reserved.
## Contents

1  Introduction................................................................................................................................. 6  
1.1 Terms ........................................................................................................................................ 6  
1.2 Conventions used in this document.......................................................................................... 6  
1.2.1 Typographic conventions ..................................................................................................... 6  
1.3 References.................................................................................................................................. 8  

2  _DSD- graphs UUID ..................................................................................................................... 9  
2.1 Format definition....................................................................................................................... 9  
2.2 Example ...................................................................................................................................... 10  

3  Known Graph UUIDs...................................................................................................................... 13  
3.1 Arm CoreSight ACPI specification............................................................................................. 13  
3ECBC8B6-1D0E-4FB3-8107-E627F805C6CD << Link to Arm CoreSight ACPI spec>> 13
## Revision history

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Description</th>
<th>Revision Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1.0&gt;</td>
<td>First revision</td>
<td>October 2018</td>
</tr>
</tbody>
</table>

§
1 Introduction

_DSD (Device specific data) [1,2] is an ACPI device configuration object that can be used to convey information on hardware features, which are not specifically covered by the ACPI specification.

Graphs are a concept that is often observed in computing. A graph is a set of nodes, that are connected together through links that represent logical relationships. Often, hardware components relate to each other functionally, or have physical connections to each other, forming a graph. An example is shown in Figure 1.

![Example graph](image)

**Figure 1** Example graph. A Debug Trace Graph includes Trace Masters (Trace 1 and Trace 2) that produce traces, multiplexors that combine traces from more or more masters, and multiple Sinks to which traces are fed. The Traces could be generated by debug logic attached to devices or logical blocks in a system.

In order to function, software drivers need to comprehend these graphs. This document provides a specification for describing such graphs to operating systems using ACPI.

This document specifies the data format associated with UUID:

```
ab02a46b-74c7-45a2-bd68-f7d344ef2153
```

(Device Graph UUID) for the _DSD (Device Specific Data) ACPI device configuration object.

1.1 Terms

The following terms are used throughout this document to describe varying aspects of input localization:

**ACPI**

Advanced Configuration and Power Interface specification.

**Device**

Hardware component or set of interrelated hardware registers.

1.2 Conventions used in this document

1.2.1 Typographic conventions
This document uses the typographic and illustrative conventions described below:

Plain text  The normal text typeface is used for the vast majority of the
descriptive text in a specification.
Introduction

Computer code, example code segments, and all prototype code segments use a **BOLD Monospace** typeface with a dark red color. These code listings normally appear in one or more separate paragraphs, though words or segments can also be embedded in a normal text paragraph.

### 1.3 References

[1] _DSD (Device Specific Data) Implementation Guide._
[http://www.uefi.org/sites/default/files/resources/_DSD-implementation-guide-toplevel-1_1.htm](http://www.uefi.org/sites/default/files/resources/_DSD-implementation-guide-toplevel-1_1.htm)


2 Device graphs UUID

2.1 Format definition

The _DSD graphs UUID

\texttt{ab02a46b-74c7-45a2-bd68-f7d344ef2153}

defines the format of a package (data structure), which is as follows:

The package is then constructed as follows:

\begin{verbatim}
Package () {
    Revision,   // Integer must be zero
    NumberOfGraphs, // N in the list below
    Graph[1],   // N graphs that this device belongs to
    ...
    Graph[N]
}
\end{verbatim}

Each Graph entry in turn is a package with the following format:

\begin{verbatim}
Package () {
    GraphID,   // INTEGER, identifies a graph the
              // parent device belongs to
    UUID,     // 16-byte buffer UUID for
              // specification that governs this
              // graph
    NumberOfLinks, // INTEGER for number of links on this
                   // node
    Links[1],   // List of graph links, with
                 // NumberOfLinks entries.
    ...
    Links[N]
}
\end{verbatim}

Finally, each link is a package that contains the following format:

\begin{verbatim}
Package () {
    SourcePortAddress,   // INTEGER
    DestinationPortAddress, // INTEGER
    DestinationDeviceName, // REFERENCE to another
                            // device in the name space
}
\end{verbatim}

Devices represent vertices in a graph. A device can indicate that it participates in a graph by providing a _DSD graph object in its scope. The object primarily lists the connections the device has to other devices in a given graph. The _DSD graph object has the following fields:
Revision: must be zero

NumberOfGraphs: Specifies the number of graphs that the current device is a part of.

Graph[N]: list of packages that specifies the characteristics of each graph that the current device is a member of.

The Graph package itself is composed of the following fields:

GraphID: An identifier for the graph. Each graph has a unique identifier and it illegal for a given device to provide more than in _DSD graph object with the same identifier value.

UUID: A UUID that identifies the specification that governs the behavior of the graph. Known UUIDs and associated specifications are listed at the end of this document.

NumberOfLinks: the number of links the parent device has to the other devices on the graph identified by the _DSD graph object.

Links: Each link entry is in turn a package, that contains, a source port address, a destination port address and a destination device reference.

The behavior of a graph and its meaning is determined by the devices themselves and their drivers. Therefore, the properties of a link, ports, and directionality, are determined by the devices themselves and their drivers. To identify the meaning, each graph in a device graph _DSD object carries a UUID which in turns links to a vendor provided specification for the graph. The specification determines how the graph is to be understood by device drivers.

A link package may be extended with additional vendor defined data. The graph specification determines how that data is to be interpreted.

2.2 Example

The following ASL shows an illustrative example of four devices which are connected in two device graphs. The graphs are illustrated in Figure 2. In this example, devices ABC, DEF and GHI are interconnected in a first graph topology. Devices ABC, GHI and JKL are interconnected in a second graph topology.

Figure 2 four devices interconnected in two graphs
Scope (_SB) {
  Device (ABC) {
    ... 
    Name (_DSD, Package () {
      ToUUID("ab02a46b-74c7-45a2-bd68-f7d344ef2153"),
      Package() {
        0, // Revision 
        2, // NumberOfGraphs 
        Package() {
          1, // GraphID - Graph 1 
          ToUUID("UUID_For_Graph_Arch"),
          2, // Number of links 
          Package (3) {0,3,\_SB.DEF},
          Package (3) {1,10,\_SB.GHI}
        },
        Package() {
          2, // GraphID - Graph 2 
          ToUUID("UUID_For_Graph_Arch"),
          1, // Number of links 
          Package (3) {2,380,\_SB.JKL}
        }
      }
    }
  }
  ... 
}

Device (DEF) {
  ... 
  Name (_DSD, Package () {
    ToUUID("ab02a46b-74c7-45a2-bd68-f7d344ef2153"),
    Package() {
      0, // Revision 
      1, // NumberOfGraphs 
      Package() {
        1, // GraphID - Graph 1 
        ToUUID("UUID_For_Graph_Arch"),
        1, // Number of links 
        Package (3) {5,20,\_SB.GHI}
      }
    }
  })
  ...
}

Device (GHI) {
  ... 
  Name (_DSD, Package () {
    ToUUID("ab02a46b-74c7-45a2-bd68-f7d344ef2153"),
Package() {
  0, // Revision
  2, // NumberOfGraphs
  Package () {
    1, // GraphID - Graph 1
    ToUUID("UUID_For_Graph_Arch"),
    1, // Number of links
    Package (3) {10,1,"SB.ABC"}
  },
  Package() {
    2, // GraphID - Graph 2
    ToUUID("UUID_For_Graph_Arch"),
    1, // Number of links
    Package (3) {30,210,"SB.JKL"},
  }
}
...
}
3 Known Graph UUIDs

3.1 Arm CoreSight ACPI specification

3ECBC8B6-1D0E-4FB3-8107-E627F805C6CD

https://developer.arm.com/docs/den0067/latest/acpi-for-coresighttm-10-platform-design-document