Graphs

_DSD

_DSD (Device Specific Data) [dsd-guide] is a predefined ACPI device configuration object that can be used to convey information on hardware features which are not specifically covered by the ACPI specification [acpi]. There are two _DSD extensions that are relevant for graphs: property [dsd-guide] and hierarchical data extensions. The property extension provides generic key-value pairs whereas the hierarchical data extension supports nodes with references to other nodes, forming a tree. The nodes in the tree may contain properties as defined by the property extension. The two extensions together provide a tree-like structure with zero or more properties (key-value pairs) in each node of the tree.

The data structure may be accessed at runtime by using the device_* and fwnode_* functions defined in include/linux/fwnode.h .

Fwnode represents a generic firmware node object. It is independent on the firmware type. In ACPI, fwnodes are _DSD hierarchical data extensions objects. A device's _DSD object is represented by an fwnode.

The data structure may be referenced to elsewhere in the ACPI tables by using a hard reference to the device itself and an index to the hierarchical data extension array on each depth.

Ports and endpoints

The port and endpoint concepts are very similar to those in Devicetree [devicetree, graph-bindings]. A port represents an interface in a device, and an endpoint represents a connection to that interface. Also see [data-node-ref] for generic data node references.

All port nodes are located under the device's "_DSD" node in the hierarchical data extension tree. The data extension related to each port node must begin with "port" and must be followed by the "@" character and the number of the port as its key. The target object it refers to should be called "PRTX", where "X" is the number of the port. An example of such a package would be:

```
Package() { "port@4", "PRT4" }
```

Further on, endpoints are located under the port nodes. The hierarchical data extension key of the endpoint nodes must begin with "endpoint" and must be followed by the "@" character and the number of the endpoint. The object it refers to should be called "EPXY", where "X" is the number of the port and "Y" is the number of the endpoint. An example of such a package would be:

```
Package() { "endpoint@0", "EP40" }
```

Each port node contains a property extension key "port", the value of which is the number of the port. Each endpoint is similarly numbered with a property extension key "reg", the value of which is the number of the endpoint. Port numbers must be unique within a device and endpoint numbers must be unique within a port. If a device object may only has a single port, then the number of that port shall be zero. Similarly, if a port may only have a single endpoint, the number of that endpoint shall be zero.

The endpoint reference uses property extension with "remote-endpoint" property name followed by a reference in the same package. Such references consist of the remote device reference, the first package entry of the port data extension reference under the device and finally the first package entry of the endpoint data extension reference under the port. Individual references thus appear as:

```
Package() { device, "port@X", "endpoint@Y" }
```

In the above example, "X" is the number of the port and "Y" is the number of the endpoint.

The references to endpoints must be always done both ways, to the remote endpoint and back from the referred remote endpoint node.

A simple example of this is show below:

```
Scope (\_SB.PCI0.I2C2)
{
    Device (CAM0)
    {
        Name (_DSD, Package () {
            ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
            Package () {
                Package () { "compatible", Package () { "nokia,smia" } },
            },
            ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
            Package () {
                Package () { "port@0", "PRT0" },
            }
        })
        Name (PRT0, Package() {
            ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
            Package () {
                Package () { "reg", 0 },
            },
            ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
            Package () {
                Package () { "endpoint@0", "EP00" },
            }
        })
```

```
            Name (EP00, Package() {
                ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
                Package () {
                    Package () { "reg", 0 },
                    Package () { "remote-endpoint", Package() { \_SB.PCI0.ISP, "port@4", "endpoint@0" } },
                }
            })
        }
    }

    Scope (\_SB.PCI0)
    {
        Device (ISP)
        {
            Name (_DSD, Package () {
                ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
                Package () {
                    Package () { "port@4", "PRT4" },
                }
            })

            Name (PRT4, Package() {
                ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
                Package () {
                    Package () { "reg", 4 }, /* CSI-2 port number */
                },
                ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
                Package () {
                    Package () { "endpoint@0", "EP40" },
                }
            })

            Name (EP40, Package() {
                ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
                Package () {
                    Package () { "reg", 0 },
                    Package () { "remote-endpoint", Package () { \_SB.PCI0.I2C2.CAM0, "port@0", "endpoint@0" }
                }
            })
        }
    }
```

Here, the port 0 of the "CAM0" device is connected to the port 4 of the "ISP" device and vice versa.

References

[acpi] Advanced Configuration and Power Interface Specification.
https://uefi.org/specifications/ACPI/6.4/, referenced 2021-11-30.

[data-node-ref] Documentation/firmware-guide/acpi/dsd/data-node-references.rst

[devicetree] Devicetree. https://www.devicetree.org, referenced 2016-10-03.

[dsd-guide] DSD Guide.
https://github.com/UEFI/DSD-Guide/blob/main/dsd-guide.adoc, referenced 2021-11-30.

[dsd-rules] _DSD Device Properties Usage Rules.
Documentation/firmware-guide/acpi/DSD-properties-rules.rst

[graph-bindings] Common bindings for device graphs (Devicetree).
https://github.com/devicetree-org/dt-schema/blob/main/schemas/graph.yaml, referenced 2021-11-30.