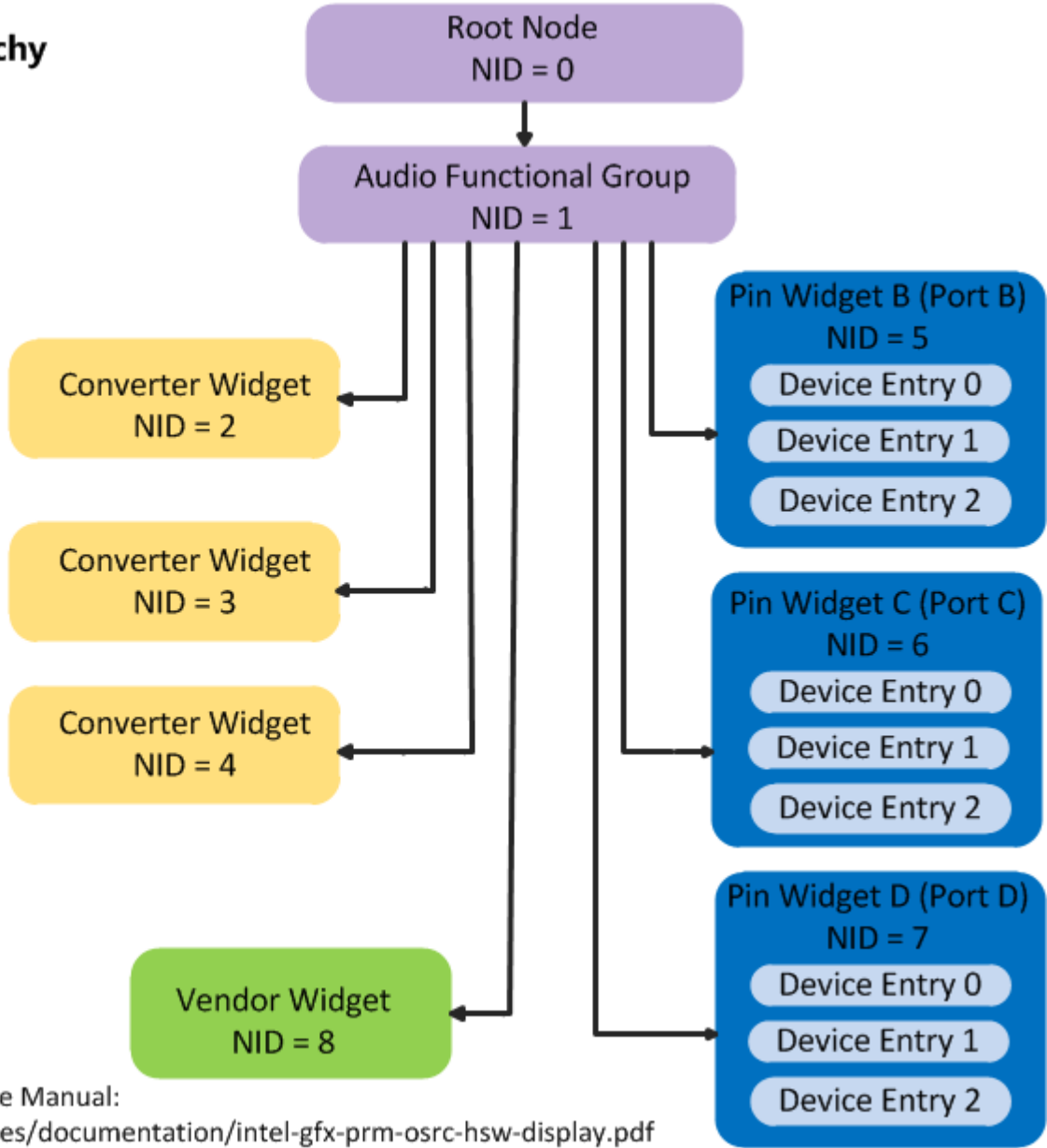


Add Support for DP 1.2 MST Audio to HD-A driver

- Mengdong Lin, Libin Yang, October 14, 2015

- **DisplayPort 1.2** added support for **Multi-Stream Transport (MST)** - enabling multiple monitors to be used via a single DisplayPort, by use of a DP MST Hub or a monitor capable of daisy-chaining.
- On Intel Haswell and later processors, a DP port can support up to 3 streams (3 device entries, 1 stream per device entry). The total number of streams is also 3, equal to the number of display pipelines & audio codec converters.

Codec Node Hierarchy



- **Target**

HD-A Driver should be able to route audio to a specified monitor as before, no matter if it's a HDMI or DP monitor, directly connected or bridged by hub or another monitor.

- **Gap**

Without DP MST, a monitor is **connected** to a pin. So current display audio codec driver handles the ELD info, PCM, convertor selection, channel map on a pin (struct `hdmi_spec_per_pin`) not on a device entry.

With DP MST, a monitor is **connected** to a device entry. So the above operations/info need to be oriented to a device entry..

Solution

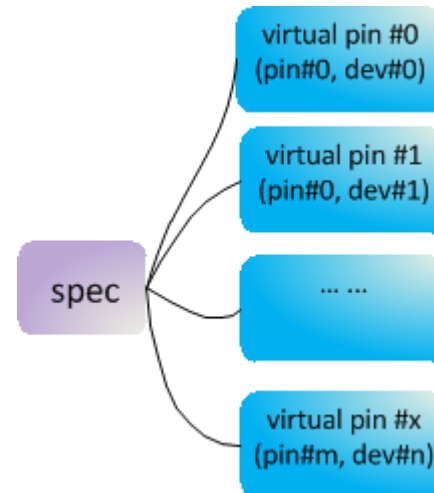
- Reuse and enhance current hdmi codec driver to support HDMI/DP SST/DP MST audio.
- Both non-mst audio and mst audio will use virtual pin and dynamic PCM binding

How to Represent a Device Entry?

- Virtual pin

- struct `hdmi_spec_per_pin` evolves to a virtual pin by adding a `dev_id` field.

```
+typedef u8 hdma_dev_t;  
struct hdmi_spec_per_pin {  
    hdma_nid_t pin_nid;  
+    hdma_dev_t dev_idx;  
    ...  
}
```



- a virtual pin can represent either a unique pin (non-mst mode) or a unique device entry (mst mode).

PCM

- Dynamic PCM assignment
 - Statically create the PCM based on the number of converter.
 - When the monitor is connected, attach a free PCM to the pin.
 - When the monitor is disconnected, the PCM is detached from the pin; Stop the PCM
- How to assign the PCM to pin?
 - Each pin has its preferred PCM, for example, the pin of nid 5 prefer PCM#3.
 - If the preferred PCM is not available, it will try to find a free PCM to attach to the pin.

PCM

```
struct hdmi_spec_per_pin {
    ... ..
+   struct hda_pcm *pcm;
    ... ..
};

struct hdmi_spec {
    ... ..
    struct hda_pcm *pcm_rec[16];
+   ul6 pcm_bitmap;
+   int pcm_used;
    ... ..
};
```

- The `pcm_bitmap` is used to indicate the corresponding PCM is bound or not. Each bit for each PCM.

ELD

- ELD info to userspace will be:
`/proc/asound/cardn/eld#x.y.z`
 - x: codec id;
 - y: pin id;
 - z: device entry id
- Each ELD info for each device entry

Jack

- Each device entry has its own Jack and unique jack tag.

```
struct hda_jack_tbl {  
    hda_nid_t nid;  
+    hda_dev_t dev_idx;  
    unsigned char tag;  
    ... ..  
};
```

SPDIF

- The number of SPDIF is the same as PCM number.
- SPDIF will not bound to the pin in initialization.