

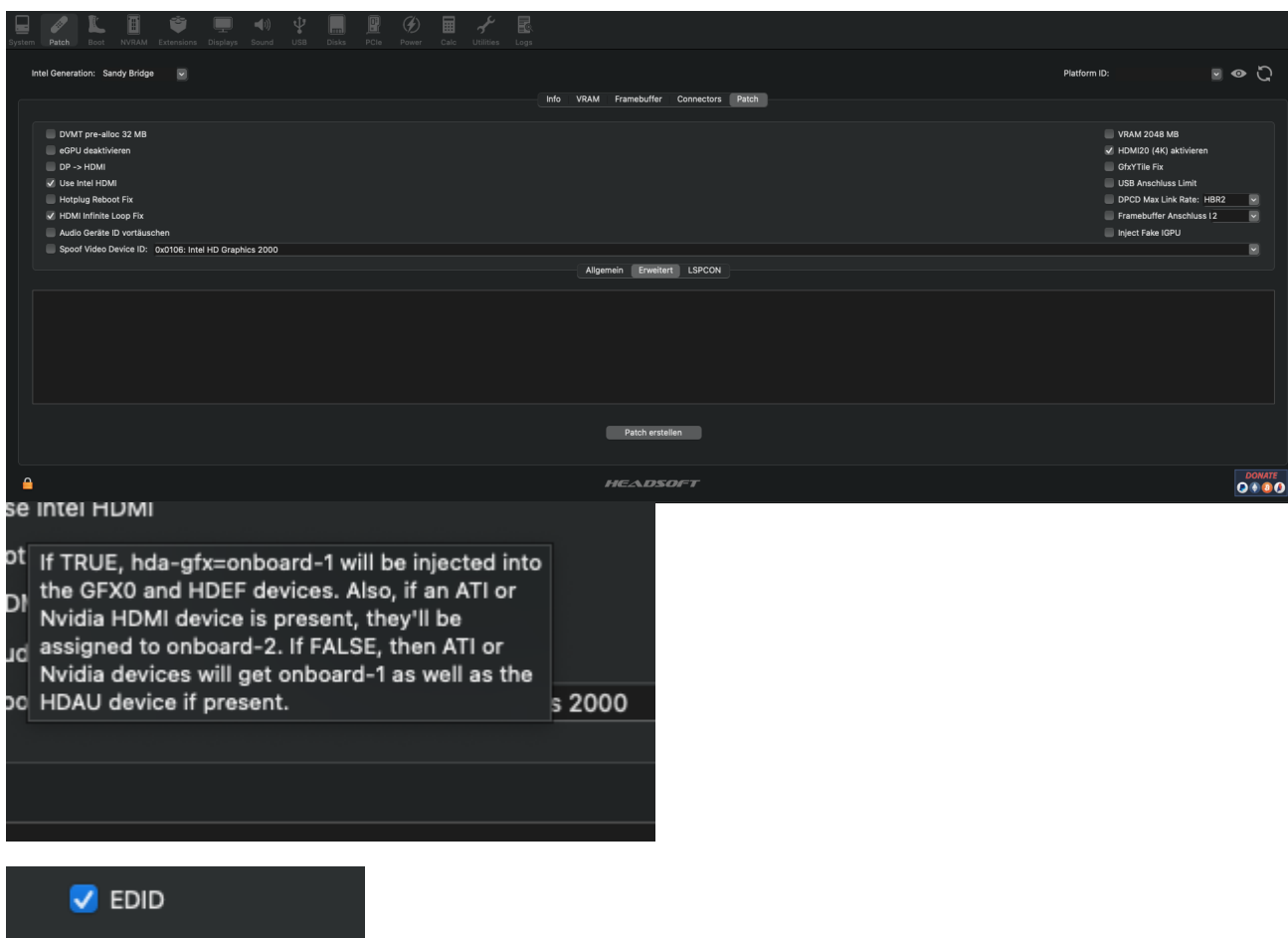
In Arbeit

Monterey Intel HD 530 DVI

Beitrag von „Hecatomb“ vom 27. Mai 2022, 16:20

Hast du mal im Hackintool die Infos angeschaut? Eventuell steht da ja was nützliches dabei. Bekommst du angezeigt wenn du mit etwas Geduld über manche Sachen gehst.

Kannst ja auch mal die die EDID vom Monitor mit in den patch packen...



Watch out, this is really messy (see `AppleIntelFramebufferController::MapFBToPort`).

I am not fully sure why this exists, and recommend setting index to array index (i.e. the sequential number from 0).

The only accepted values are 0, 1, 2, 3, and -1 (0xFF). When index is equal to array index the logic is simple:

Port with index 0 is always considered built-in (of LVDS type) regardless of any other values.

Ports with indexes 1~3 are checked against type, HDMI will allow the use of digital audio, otherwise DP is assumed.

Port with index 0xFF is ignored and skipped.

When index != array index port type will be read from `connector[index].type`.

Say, we have 2 active ports:

0 - [1] busID 4 type LVDS

1 - [2] busID 5 type DP

2 - [3] busID 6 type HDMI

3 - [-1] busID 0 type Dummy

This will result in 2 framebuffers which types will be shifted:

0 - busID 4 type DP

1 - busID 5 type HDMI

In fact busID values are also read as `connector[index].busID`, but are later mapped back via

`AppleIntelFramebufferController::getGMBusIDfromPort` by looking up a connector with the specified index.

The lookup will stop as soon as a special marker connector (-1) is found. To illustrate, if we have 2 active ports:

0 - [1] busID 4 type LVDS

1 - [2] busID 5 type DP

2 - [-1] busID 6 type HDMI

3 - [-1] busID 0 type Dummy

The result will be 2 framebuffers which types and the second busID will be shifted:

0 - busID 4 type DP

1 - busID 6 type HDMI

It is also used for port-number calculation.

- LVDS displays (more precisely, displays with `CNConnectorAlwaysConnected` flag set) get port-number 0.

- Other displays go through index - port-number mapping: 1 - 5, 2 - 6, 3 - 7, or fallback to 0.

☒ HDMI Infinite Loop Fix



Audio



Speaker

Fix the infinite loop when the graphics driver tries to establish a HDMI connection with a higher pixel clock rate, for example connecting to a 2K/4K display with HDMI 1.4, otherwise the system just hangs (and your builtin laptop display remains black) when you plug in the HDMI cable.

- For those who want to have "limited" 2K/4K experience (i.e. 2K@59Hz or 4K@30Hz) with their HDMI 1.4 port, you might find this fix helpful.

- For those who have a laptop or PC with HDMI 2.0 routed to IGPU and have HDMI output issues, please note that this fix is now succeeded by the LSPCON driver solution, and it is still recommended to enable the LSPCON driver support to have full HDMI 2.0 experience.

(You might still need this fix temporarily to figure out the connector index of your HDMI port.)

Convert Display Port connector types to HDMI

Convert Display Port connector types to HDMI

otplug Reboot Fix

Proven by AppleIntelFramebufferController::MapFBToPort, by a call to AppleIntelFramebufferController::getGMBusIDfromPort.
This is GMBUS (Graphic Management Bus) ID described in https://01.org/sites/default/files/documentation/intel-gfx-prm-osrc-hsw-display_0.pdf.
The use could be found in Intel Linux Graphics Driver source code:
https://github.com/torvalds/linux/blob/6481d5ed076e69db83ca75e751ad492a6fb669a7/drivers/gpu/drm/i915/intel_i2c.c#L43
https://github.com/torvalds/linux/blob/605dc7761d2701f73c17183649de0e3044609817/drivers/gpu/drm/i915/i915_reg.h#L3053
However, it should be noted that Apple identifiers are slightly different from Linux driver. In Linux 0 means disabled, however, for Apple it has some special meaning and is used for internal display.
Other than that the values are the same:
- GMBUS_PIN_DPC (4) HDMIC
- GMBUS_PIN_DPB (5) SDVO, HDMIB
- GMBUS_PIN_DPD (6) HDMID
- GMBUS_PIN_VGADDC (2) VGA until Broadwell inclusive.
So basically you could use 4, 5, 6 for arbitrary HDMI or DisplayPort displays.
Since 5 supports SDVO (https://en.wikipedia.org/wiki/Serial_Digital_Video_Out), it may also be used to support DVI displays.
Starting with Skylake VGA works via SDVO too (instead of a dedicated GMBUS_PIN_VGADDC id).