Chrultrabook: A preview of the challenges with porting Windows to x86-64 chromebook


coolstar


coolstar@mastodon.social


coolstar@bsky.app

Notes about this presentation

- Porting Windows to chromebooks is very complex
- Many of the topics in these slides can be full length presentations of their own
  - (e.g. edk2 as coreboot payload, debugging Chrome EC, porting/writing audio drivers, Intel Smart Sound, Sound Open Firmware, Reverse engineering Windows drivers, debugging ACPI in Windows)
  - Many of the advancements detailed happened between the CFP for OSFC and today
  - Expect a full length talk next year – hopefully for OSFC 2024 ☺
- Some slides mention macOS on chromebooks, but that has been abridged from this talk
- These slides are heavily simplified and abridged for a lightning talk

A (very) brief overview of x86 ChromeOS devices

- Ship with coreboot (open source bootloader) and depthcharge
  - ACPI is often not fully complaint in Google's firmware
- Can be reflashed to run a custom-built UEFI payload after entering developer mode, disabling Write Protect, and flashing a custom coreboot firmware
  - End user can disable WP using a screw / jumper, or via USB-C suzyq cable
- Often have non-standard devices (trackpads, touch screens, embedded controller, audio codecs, sometimes webcams)
  - Drivers for these did not exist on Windows originally

Why port Windows to chromebooks?

- Chromebooks could often (historically) be found for much cheaper than their non-chrome counterparts
  - Can still be found for way less cost on the second-hand market
- Millions of chromebooks are used every year in education
  - Many of these are given to students when they graduate
- Chromebooks no longer get Chrome OS updates once they hit AUE
  - Can be a way to keep the hardware working past its ChromeOS expiration date
  - Several recyclers and resellers install Windows on chromebooks to get more value from the hardware
- Windows has much better software compatibility than Linux
  - Linux also faces challenges with drivers on newer chromebooks (often worse than Windows now)
- Many users don’t want to deal with maintaining Linux

Which chromebook to buy for modding?

- Ignore the model name for branding
  - “HP Chromebook 11” and “HP Chromebook 14” are too generic
- Look for the CPU model instead, and pick out based on which CPU fits your needs, just like a normal PC
- Chromebooks with NVMe SSDs and Fans will generally perform better and be upgradable
- RAM is soldered (except Acer C710 and Framework Chromebook)
- Check the Chrome OS board name against coolstar.org/chromebook and mrchromebox.tech for the full compatibility details

Chrultrabook history: 2014 - 2015

- Technically Originally started on Acer Forums in 2014
  - "**GUIDE** Install Windows 8 on the Acer C720"
  - Officially started in 2015
- Intel Haswell chromebooks could boot Windows via RW_LEGACY
  - Used SeaBIOS at the time
  - Upgradable M.2 SATA SSD
  - Tends to be an exception rather than the rule
  - Booting does not mean it runs well
  - Missing: GPU Acceleration on eDP, Keyboard, I2C, Trackpad, Touch Screen, Sleep / Wake, Brightness Controls, Battery Indicator, HDMI Audio

Intel 4th gen

Chrultrabook history: 2015 - 2016

- Haswell / Broadwell chromebooks
  - Coreboot and EC bugs were pretty prevalent in 2015
    - EC had broken 8042 emulation that didn't enable key scanning
    - LPC power management bug broke eDP in Windows
    - ACPI Battery query didn't append a NULL byte to the end of strings
    - Brightness controls already fixed in upstream coreboot!
    - Sleep/wake fixed by removing vboot!
    - Haswell I2C controllers were in PCI instead of ACPI mode
    - Non-cypress trackpads had Edge interrupts instead of Level
    - ACPI OpRegion for Intel GPU was not implemented
  - Trackpad and touch screen did not conform to I2C HID spec

Intel 4th–5th gen

Chrultrabook history: 2016 - 2017

- Bay Trail chromebooks
  - Similar but slightly different issues from Haswell
    - ACPI for I2C, GPIO and Smart Sound (I2S) controllers broken
    - Touch devices had Edge interrupts instead of Level
    - Same EC issues as Haswell
      - Keyboard interrupt was on GPIO here
    - ACPI OpRegion for Intel GPU was not implemented
  - Trackpad and touch screen also did not conform to I2C HID spec
  - Maxim 98090 I2S Codec not used in any windows machine
    - https://www.analog.com/media/en/technical-documentation/data-sheets/max98090.pdf
- Cherry Trail / Braswell chromebooks
  - Largely similar to Bay Trail
  - Standard Realtek I2S Codec fortunately

Intel 4th–5th gen

Custom windows drivers: 2015 - 2017

- New to drivers – so where to start?
- Keyboard
  - Originally was ripped from ReactOS to get started (i8042prt-cros)
- Battery info
  - Port parts of ectool to Windows with Microsoft's ioctl sample
  - Could run a tool from Command Prompt to read battery info from the driver as a workaround
- Haswell GPU
  - Luckily restarting the Intel GPU driver worked around eDP bug
- Haswell I2C bus
  - Shoutout VoodooI2C (macOS project) for assistance with debugging the ACPI
  - Referenced a lot of ACPI dumps from hackintoshers

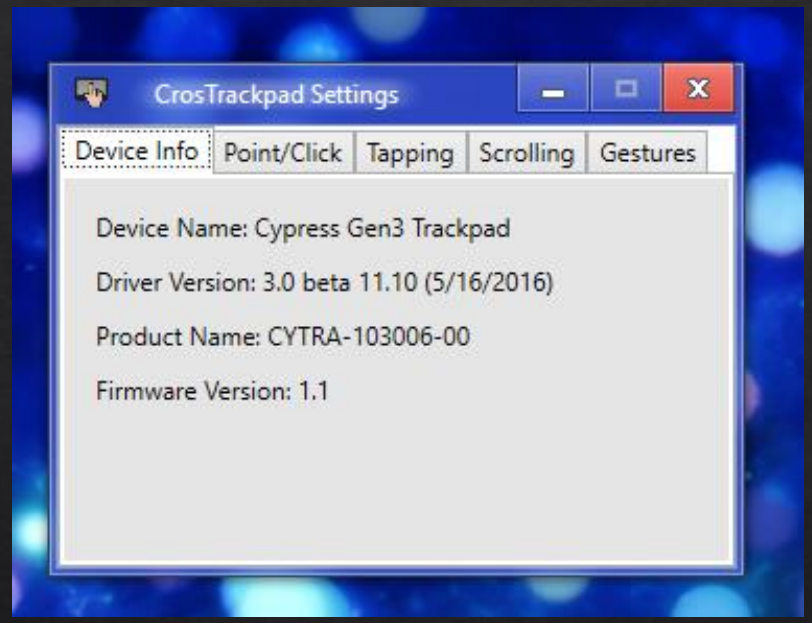
Intel 4th gen

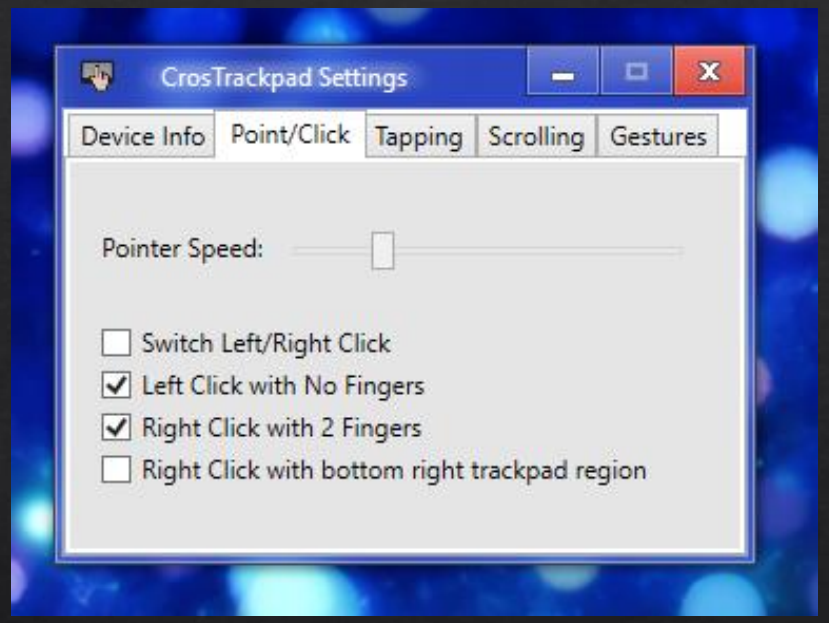
Custom windows drivers: 2015 - 2017

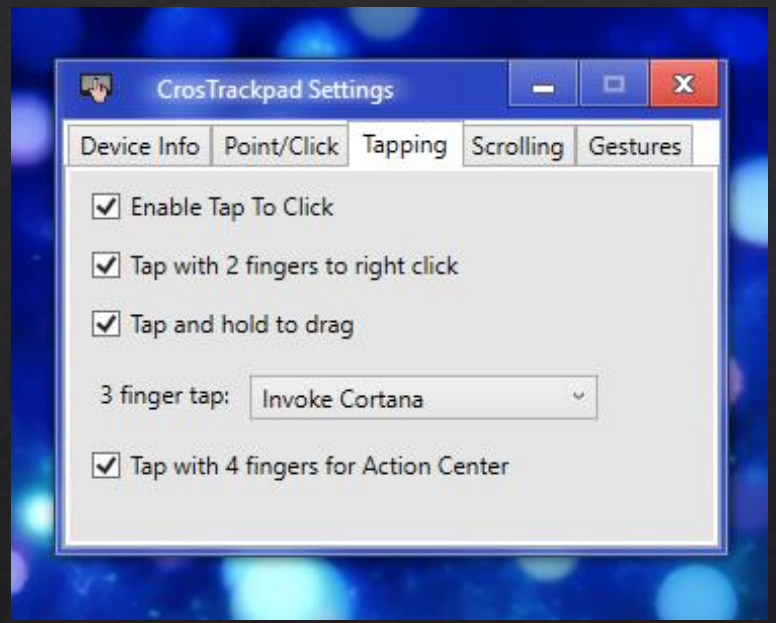
- Trackpad & Touch Screen
  - SPBTestTool from Microsoft
  - Atmel and Cypress code originally referenced from DragonFlyBSD
  - Initially drivers implemented in userspace (crostrackpad and crostouchscreen)
  - Later based off vmulti with bits from SPBTestTool
    - Fully kernel driver
  - crostrackpad2 and crostouchscreen2 first released in June 2015
    - Made the mistake of testing the trackpad driver against video games
  - CSGesture in crostrackpad3 first released in October 2015
  - Microsoft Precision Touchpad implemented in crostouchpad4 in November 2016
    - Fully native windows gestures

Intel 4th-5th gen

Custom windows drivers: 2015 - 2017







Intel 4th-5th gen

Custom windows drivers: 2015 - 2017

- Sandy / Ivy Bridge chromebooks (except Pixel 2013)
  - Trackpad I2C over SMBus
  - Custom driver attaches to entire PCIe device and consolidates entire I2C + SMBus + trackpad stack
  - crostouchpad4-smbus first released November 2016
- Maxim 98090 Amplifier (Bay Trail)
  - Luckily Intel SST driver initializes I2S bus and audio endpoints
    - Only Missing codec I2C configuration
  - https://www.analog.com/media/en/technical-documentation/data-sheets/max98090.pdf
  - Use SPBTestTool to check registers and program them based off datasheet
  - I2c-tools in crouton could dump registers
  - Read GPIO to check if jack plugged in

Intel 2nd-5th gen

Custom windows drivers: 2015 - 2017

- Realtek 5677 Amplifier (Pixel 2 Chromebook [Broadwell])
  - Intel Smart Sound driver did not initialize audio endpoints
    - Requires API call from codec driver
    - API call behind NDA
    - Linux driver also broken
- Skylake chromebooks
  - Soldered eMMC
  - I2S Audio
    - Maxim 98357a or Analog SSM4567 Speaker Amp + Nuvoton Headphone Codec
    - No audio endpoints from Windows driver here either
    - Upstream Linux driver broken
  - Suddenly rather expensive

Intel 5th-6th gen

Chrultrabook history: 2017 - 2020

- Haswell and Broadwell had full support (except Pixel 2015)
  - Lulu missing Elan touchscreen driver
- Sandy and Ivy Bridge had buggy trackpad driver (except Pixel 2013)
- Bay Trail and Cherry Trail had almost full support
  - Various models missing Elan and Melfas touchscreen driver
  - Cyan (CHT) missing Audio driver
- Drivers were cross-signed. Worked so long as no Secure Boot
- Skylake – Comet Lake got full ROMs but missing drivers
  - Shoutout to MrChromebox for fixing ACPI and shipping full ROMs
- Pixelbook got drivers from Google
  - Project Campfire – Killed By Google™

Intel 2nd–5th gen

The return of the Windows drivers: Late 2021

- Loose ends up to Braswell tied up (except Pixel 2013 / Pixel 2015)
  - Elan and Melfas touch screen drivers
    - Used sleep functions to have reliable touch screen init
  - Acer R11 audio driver
  - Sandy / Ivy Bridge SMBus driver rewritten – bugs fixed!
    - Callback functions so driver is async!
  - Drivers now signed by Microsoft Hardware Portal!
    - Ready for Secure Boot

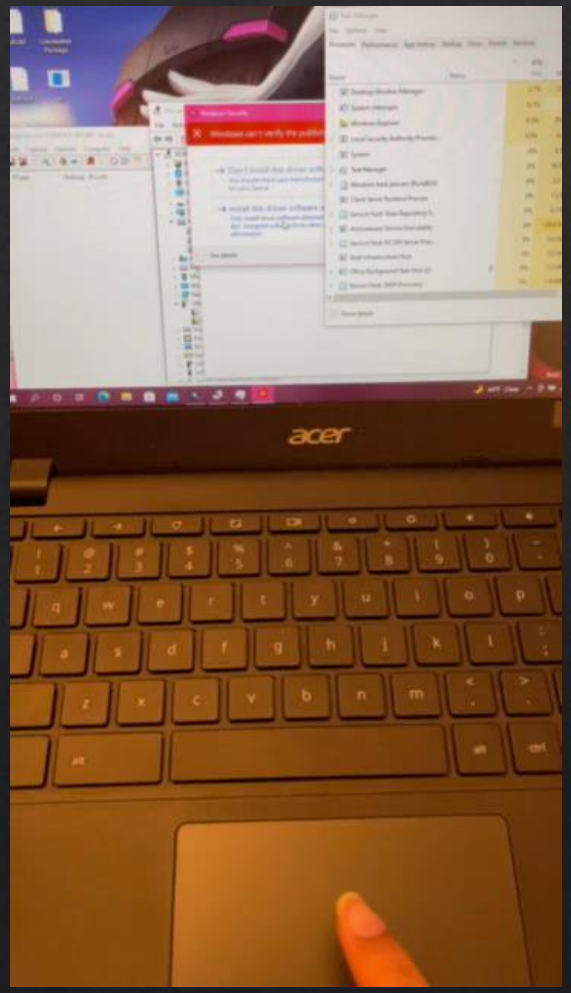
Intel 2nd–5th gen

Early 2022: Reverse engineering Intel's SST driver

- Skylake and newer still missing Audio in Windows
  - Smart Sound mixed with HD Audio Controller, despite SST not being HDA
  - Depthcharge somehow creates beep noises in the stock firmware
    - Bitbangs I2S on Skylake over GPIO
    - Writes directly to the SSP ports on Apollo Lake / Gemini Lake
    - Maybe no firmware needed on the sound card after all?
      - Sound driver should be simpler?

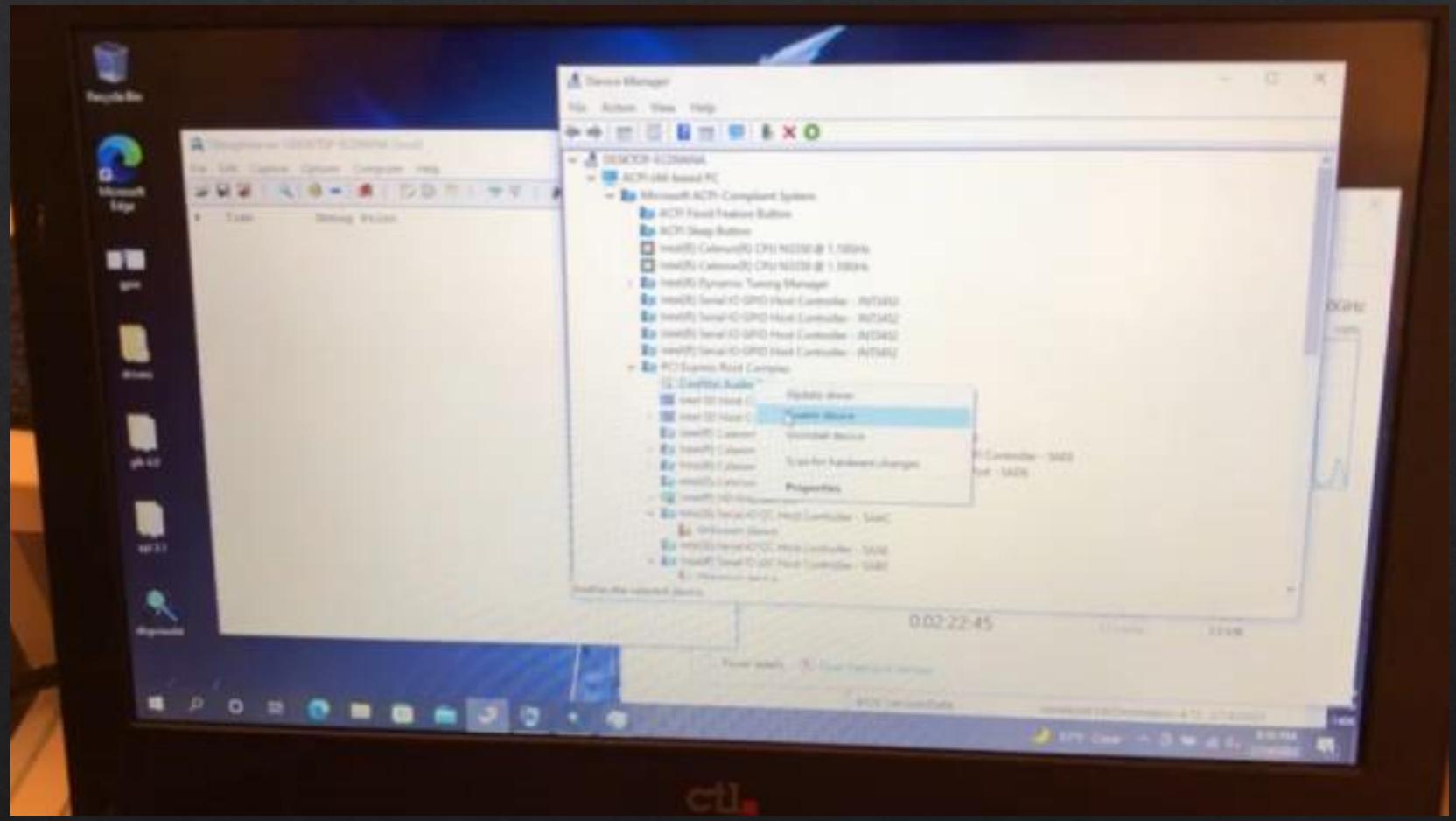
Intel 6th–8th gen

Bitbanging I2S on Skylake



Intel 6th gen

SSP ports on Apollo Lake



Intel 7th gen

Reverse engineering Intel's SST driver

- Skylake and newer still missing Audio in Windows
  - Depthcharge somehow creates beep noises in the stock firmware
    - Bitbangs I2S on Skylake over GPIO
    - Writes directly to the SSP ports on Apollo Lake / Gemini Lake
    - Maybe no firmware needed on the sound card after all?
      - Sound driver should be simpler?
      - Could only target speakers on Skylake. Headphones were doable on Apollo Lake and Gemini Lake
      - 100% CPU usage. Quality on Skylake is terrible due to GPIOs. Both platforms also require pre-encoding PCM using ffmpeg when doing this
      - Could only target speakers on Skylake. Headphones were doable on Apollo Lake and Gemini Lake
    - Guess we do need firmware after all unfortunately ☹
- … Project Campfire had a driver for the Pixelbook?

Intel 6th–8th gen

Reverse engineering Intel's SST driver

- Project Campfire driver
  - Google and Intel made special API for Pixelbook
  - Uses Windows callback APIs
    - \\CallBack\\IntcAudioSSTMultiHwCodecAPI
  - Could snoop these easily from a running system and replay calls to enable speaker
    - Combine with disassembly to get near complete structure
  - Headphone API was not so straightforward ☹
    - Maybe more undocumented calls that only Google, Realtek and Intel will ever get to know

Intel 6th–8th gen

Reverse engineering Intel's SST driver

```
#pragma pack(push,1)
typedef struct _IntcSSTArg
{
        int32_t chipModel;
        int32_t sstQuery;
        int32_t caller;
        int32_t querySize;

#ifdef __GNUC__
        char EndOfHeader[0];
#endif

        uint8_t deviceInD0;
#ifdef __GNUC__
        char EndOfPowerCfg[0];
#endif

        int32_t dword11;
        GUID guid;

#ifdef __GNUC__
        char EndOfGUID[0];
#endif
        uint8_t byte25;
        int32_t dword26;
        int32_t dword2A;
        int32_t dword2E;
        int32_t dword32;
        int32_t dword36;
        int32_t dword3A;
        int32_t dword3E;
        uint8_t byte42;
        uint8_t byte43;
        char padding[90]; //idk what this is for
}  IntcSSTArg, * PIntcSSTArg;
#pragma pack(pop)
```

Intel 6th–8th gen

Reverse engineering Intel's SST driver

- Project Campfire driver
  - Only speakers
  - Certain models had volume scaling issues (4% volume was 100% volume)
  - Closed source, and limitations to disassembly
  - API did not exist outside of Skylake and Kaby Lake because it was only made by Intel for Google
  - Implementing firmware loading from a driver would be tricky

Intel 6th–8th gen

Mid-2022: A new player shows up!

Ryzen 3000

Windows on Ryzen 3000 chromebooks

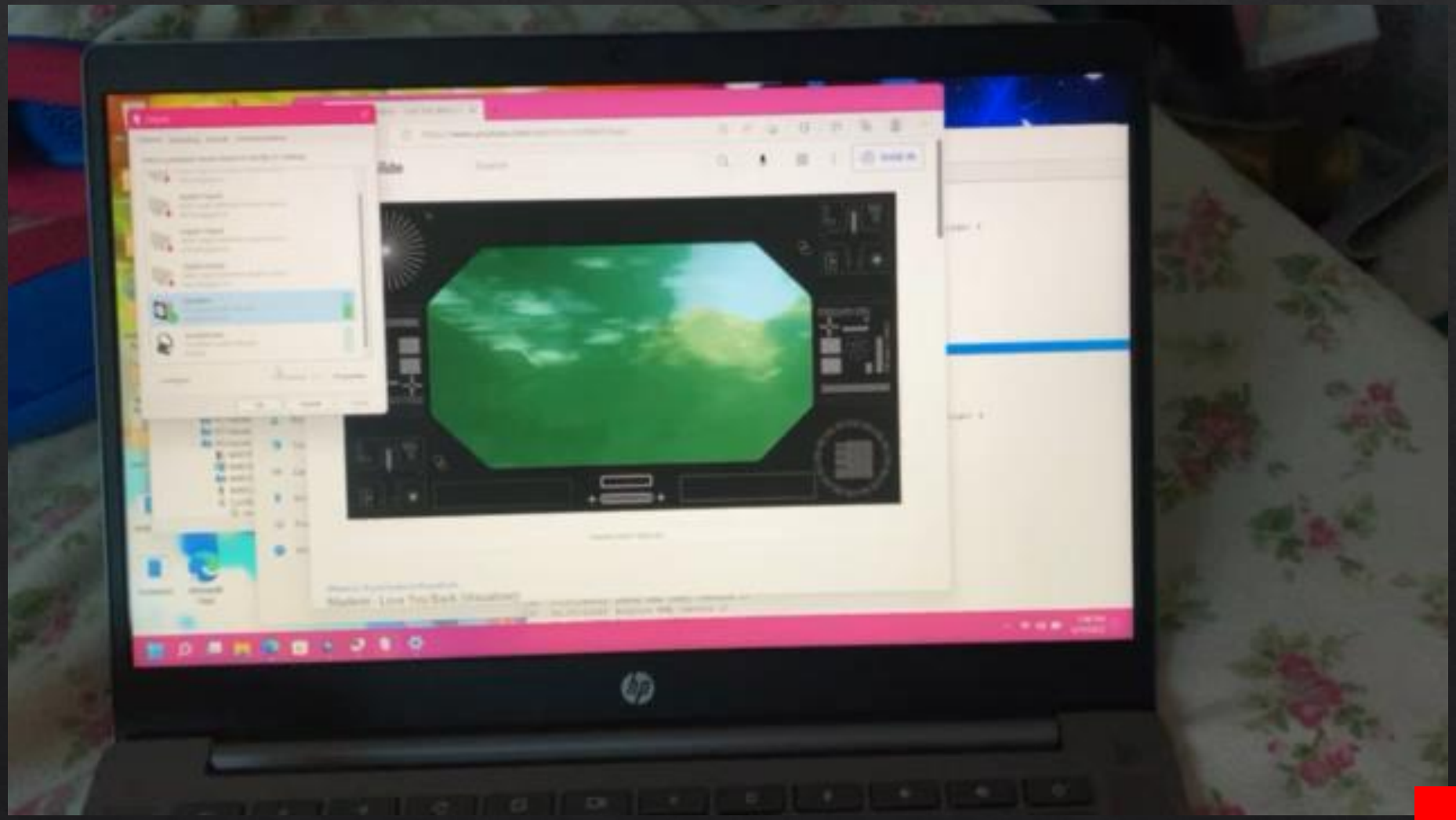
- Stock firmware shipped with edk2 in RW_LEGACY!
  - Broken ACPI ☹
  - edk2 means OpenCore can be used
    - Override ACPI without replacing original coreboot firmware
  - Windows largely worked once ACPI was override (only missing I2S Audio)
  - Depthcharge has driver for ACP (AMD's Audio CoProcessor)
  - ChromeOS doesn't upload firmware to ACP!

Ryzen 3000

Writing a new driver for AMD Audio CoProcessor

- Amplifier could be switched on with a single GPIO
  - Driver already existed for this since Skylake chromebooks had the same amplifier
- ACP is separate PCIe device
- Linux driver very small and relatively easy to read
  - Power on + Reset ACP
  - Program DMA pages to ACP
  - Start playback
- Could use Microsoft's SimpleAudioSample to start
  - Simply program the memory buffer to ACP for DMA
  - Remove all logic for writing to a file

Writing a new driver for AMD Audio CoProcessor



Ryzen 3000

Writing a new driver for AMD Audio CoProcessor

- Amplifier could be switched on with a single GPIO
  - Driver already existed for this since Skylake chromebooks had the same amplifier
- ACP is separate PCIe device
- Linux driver very small and relatively easy to read
  - Power on + Reset ACP
  - Program DMA pages to ACP
  - Start playback
- Could use Microsoft's SimpleAudioSample to start
  - Simply program the memory buffer to ACP for DMA
  - Remove all logic for writing to a file

Windows on Ryzen 3000 chromebooks

- Stuck with stock firmware
  - Booting Windows with custom coreboot ROM yielded a broken GPU
  - Full coreboot rom wouldn't boot without vboot
  - Later discovered this was due to AMD PSP
    - PSP verstage needs to be signed
    - vboot ABI changed since signed verstage, so didn't work with upstream coreboot until a shim was made
    - Present in newer Ryzen 5000 / 7000 chromebooks
    - Full ROM finally working as of July, 2023
  - Setup for booting working Windows on Ryzen 3000 for a while:
    Coreboot -> Depthcharge -> edk2 -> OpenCore -> rEFInd -> Windows

Challenges back on Intel Smart Sound

- Broadwell and newer need Smart Sound firmware
- Broadwell Smart Sound
  - Linux driver was cleaned up compared to 2017
    - Fortunately no topologies in Linux
  - Implemented firmware loading for Broadwell DSP and ported rest of the driver to Windows
  - Pixel 2015 finally had working audio as of August 2022!

Intel 5th gen

Challenges back on Intel Smart Sound

- Skylake / Kaby Lake / Apollo Lake
  - Sound firmware signed and checked by Management Engine ☹
  - PCIe device also used for HD Audio. Had to replace HDA driver
    - Microsoft documents the OS APIs, but still had to make replacement bus driver
    - HD Audio didn't work alongside custom SST driver until February 2023
  - Linux driver extremely messy and was broken for years
    - Finally started getting cleaned up in 2022
    - Still has the potential to burn speakers, so is not safe to use
    - Used topology files that were obscure and nearly impossible to read
  - Amplifier has no volume controls
  - A speaker was burned in the process of getting developing this driver for Windows – RIP ☹
  - Driver was finally released in November 2022

Intel 6th–8th gen

Challenges back on Intel Smart Sound

- Gemini Lake
  - Theoretically a refresh of Apollo Lake
  - Management Engine now uses Community Keys instead of Intel's
    - Community private key provided since Chrome OS uses Sound Open Firmware here instead
  - Driver written for Apollo Lake works on Gemini Lake (with Intel's firmware) once firmware binary was re-signed with community key
  - Intel firmware almost worked on Gemini Lake chromebooks
    - Unfortunately burned speakers of multiple beta testers until the experimental Gemini Lake support was removed
    - Should've used Sound Open Firmware to begin with

Intel 8th gen

Sound Open Firmware

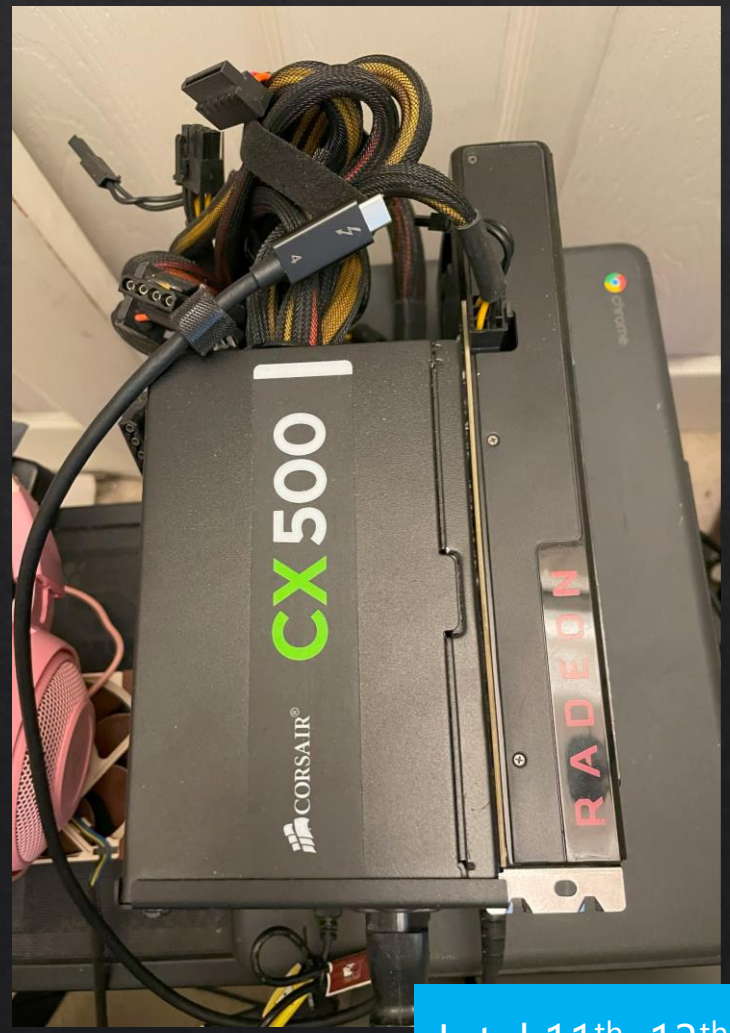
- Open Source firmware for Audio DSPs!
  - Based on Zephyr
  - Community keys mean you can compile your own firmware
- Used on Intel Gemini Lake / Comet Lake chromebooks and newer
- Used on AMD Ryzen 5000 chromebooks and newer
- Linux driver still uses topologies ☹
  - Fortunately topologies are open source
  - mostly readable m4's on github
- No burned speakers so far!
- Certain chromebook models ship with proprietary blobs
  - need to be extracted from Chrome OS for decent audio quality on these

USB4 / Thunderbolt 4 on Intel

- 11th generation and newer chromebooks can have thunderbolt / usb4 controller
  - Controller is on Intel PMC
- Controller itself is standard, but Chrome EC expects an OS driver for TCPC
  - Normally implemented in ACPI on windows pc's
  - Chrome EC TCPC uses structs – would be tricky for ACPI
- Chrome OS has logic in userspace for handling thunderbolt devices
  - Had to port to kernel mode for Windows driver
- Coreboot's PCIe hotplug I/O and Memory spaces relatively small
  - Thunderbolt can hotplug a GPU
  - No one probably expected an eGPU on a chromebook

Intel 11th–13th gen

USB4 / Thunderbolt 4 on Intel



Intel 11th–13th gen

Debugging AMD Ryzen again

- Ryzen 3000 GPU broken on full ROM
  - Fixed by shimming vboot API in coreboot and using signed verstage
  - PSP now reported “Production” mode and worked
- Ryzen 5000 and 7000 GPUs also broken on full ROM
  - PSP reported developer mode on stock firmware
  - Had to flash externally to get it to report Production mode
    - Seems PSP update fixed the PSP mode later on
    - GPU driver still broken
  - AMD GPUs use VFCT ACPI table on UEFI
    - No documentation from AMD

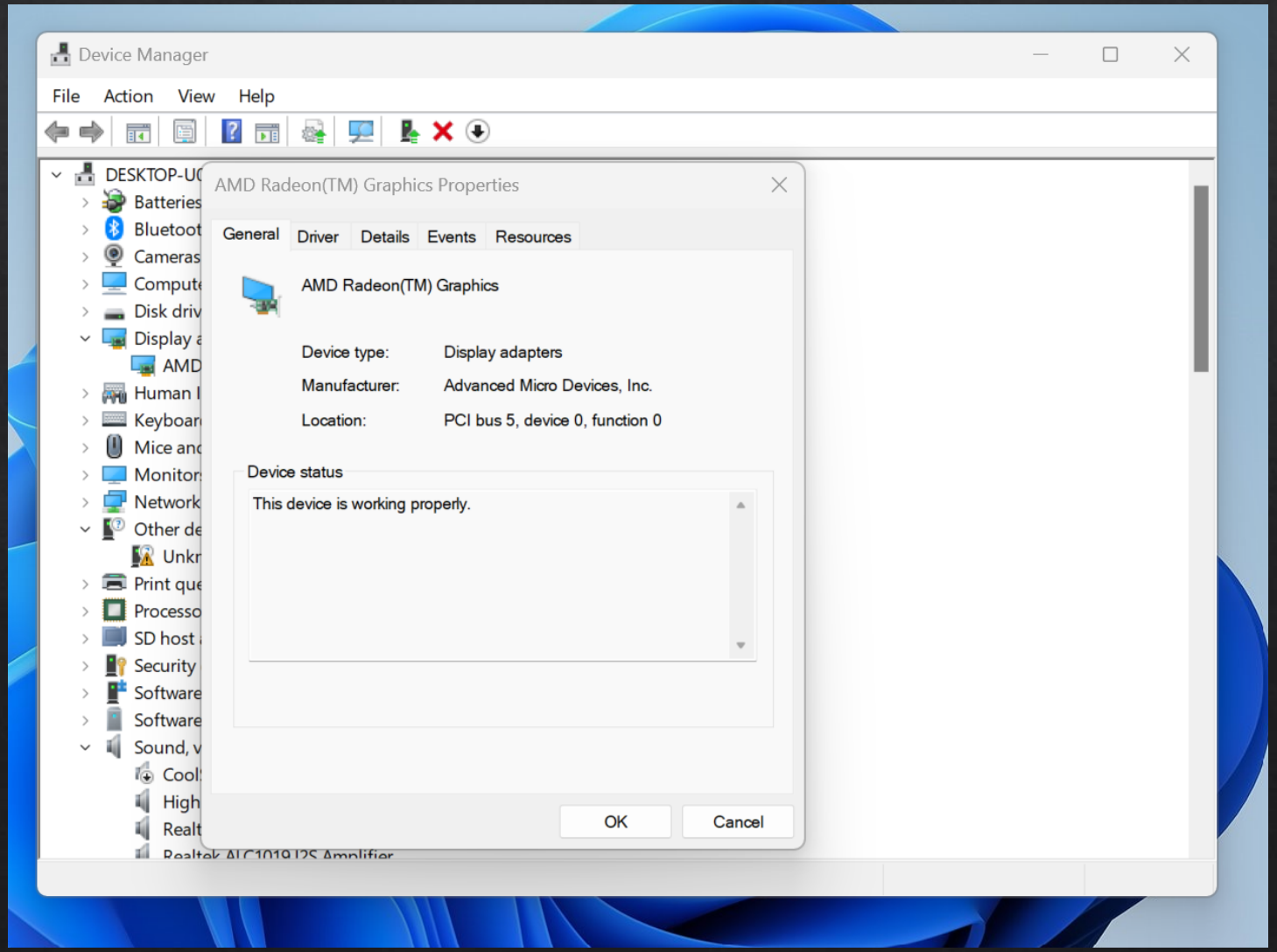
Ryzen 3000-7000

Debugging AMD Ryzen again

- Ryzen 5000 and 7000 GPUs broken on full ROM
  - Reverse engineered the driver for hours
  - “nop” and “int 3” instructions to the rescue
    - Binary patch the driver to spray this all over the place and pull the resulting crash
  - Error seemed to be from the vbios parser
    - CRC check was failing because of incomplete FSP GOP implementation
- Hopefully no more PSP breakage on newer AMD CPUs

Ryzen 5000-7000

Debugging AMD Ryzen again



Ryzen 5000-7000

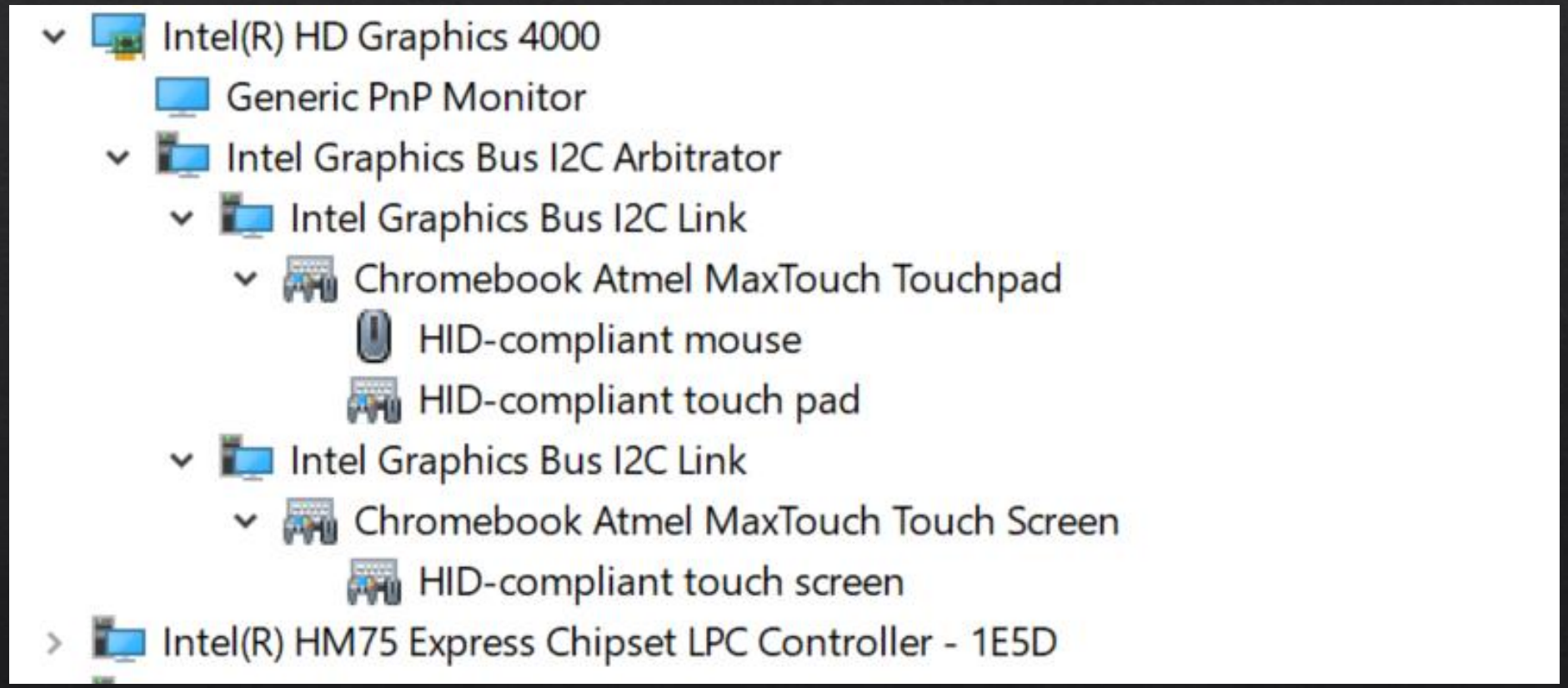
Google wiring hardware to GPUs

- Pixel 2013 chromebook still missing trackpad and touch screen support
  - Ivy Bridge CPU that was nearing a decade old
  - Has long been AUE
  - Trackpad and Touch Screen themselves very similar to the Acer C720 and Pixel 2015
    - Those are supported in Windows
    - Wired to GPU on Pixel 2013
- Lots of ACPI hacks
- 2 drivers (Intel's GPU driver and my I2C driver) accessing the same PCIe device's MMIO at once

Intel 3rd gen

Google wiring hardware to GPUs

- Pixel 2013 chromebook



Intel 3rd gen

Wall of shame: AMD Stoneyridge

- PSP bootloader in coreboot is broken
  - Does not boot PSP SecureOS. Known to break HDCP
  - Windows GPU driver takes 30 minutes to boot
    - Driver patched only 2 days ago
      - Required reverse engineering
      - Binary patched driver is not signed and requires disabling secure boot and signature enforcement
    - System idled at 90 C without a GPU driver (fanless chromebook)
      - Made it annoying to debug without a fan pointed at the machine
- AMD built the Audio CoProcessor into the GPU
  - Linux driver is messy (doesn't even work without patches)
  - Windows driver for this is still a work in progress

AMD A4/A6

Intel IPU webcams

- Certain intel chromebooks use MIPI webcams
  - Relies on Intel's Image Processor unlike normal USB webcams
  - Intel IPU runs proprietary intel firmware and has a proprietary Windows driver
  - Pixelbook Go / Pixel Slate use ipu3
    - Cameras not found in Windows laptops, so will need a custom driver
  - Certain 11th and 12th gen chromebooks use ipu6
    - Should be fixable with ACPI patches in coreboot
    - Some use normal USB, some use MIPI. Hard to tell before purchase as this varies depending on the SKU
  - Not currently working in either Windows or Linux

Intel 8th–13th gen

Current status of x86-64 chromebooks in Windows

- Intel
  - All devices supported in Windows!
  - Certain chromebooks that use MIPI webcams have no webcam support
    - Most still use USB, so simply check “lsusb” in Chrome OS, or try it in Windows and see if it works
- AMD
  - All Ryzen chromebooks supported in Windows!
    - Hopefully it stays this way, as PSP was recently fixed
  - Stoney Ridge is still work in progress
    - PSP may forever be broken on this platform

- Latest information and guides are available on https://coolstar.org/chromebook/

Demo