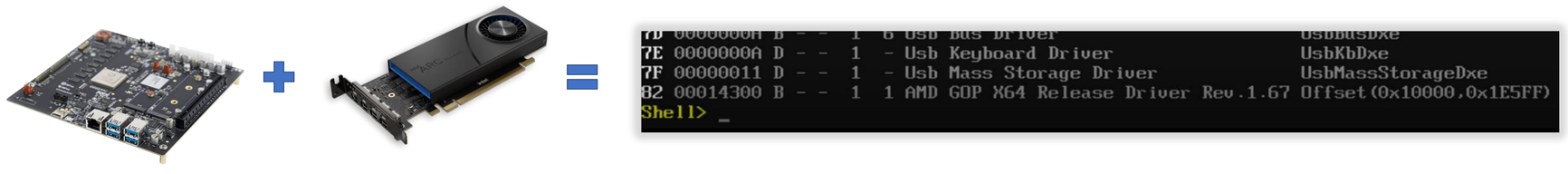
Multi-ISA Firmware Compatibility

Bringing RISC-V and IHV Ecosystems Together

Andrei Warkentin (Intel)



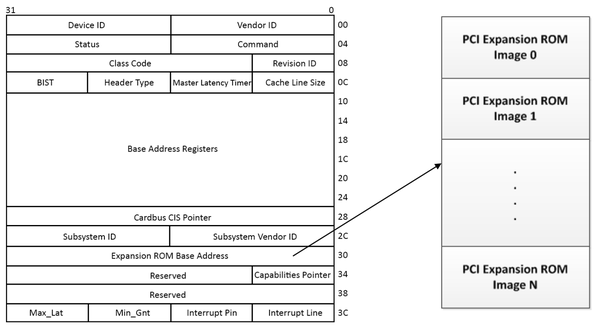
❖**Today ➔** Non-standard RISC-V platforms, that don't enable horizontal market segments where interoperability is key.
- ❖Single-vendor.
- ❖Embedded hardware with little extensibility.
- ❖Embedded firmware and OS with all the drivers baked in.
- ❖"I plugged a PCIe NIC in, but don't know how to network boot my OS"

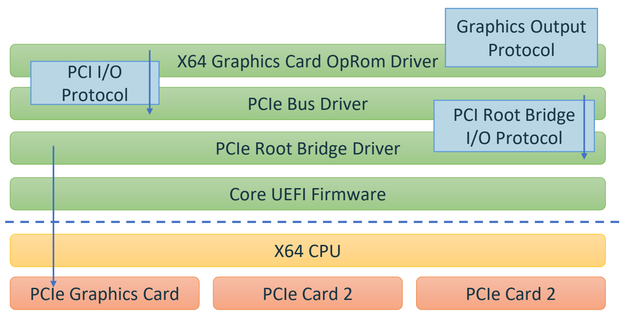
❖**Tomorrow ➔** An interoperable RISC-V ecosystem that allows building servers and PCs.
- ❖Many vendors come together for a solution.
- ❖PCIe/CXL connectivity for off-the-shelf devices.
- ❖Rich UEFI + ACPI firmware experience.
- ❖Same OS image can boot across different platforms, SoCs, IP blocks.

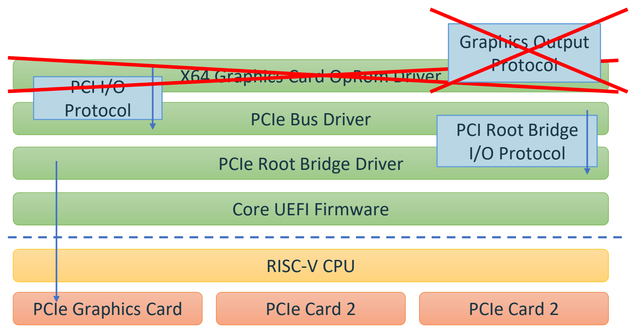
How to make existing PCIe devices work?

Will future PCIe devices ship with RISC-V firmware drivers?

PCIe Firmware Drivers and UEFI







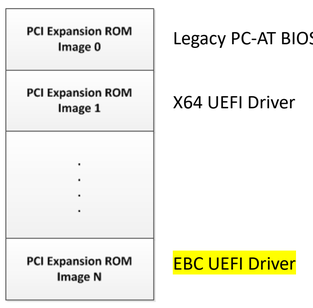
UEFI drivers stored in adapter Flash

☑ **Driver ISA == firmware ISA**

❌ **Driver ISA != firmware ISA**

What about EFI Byte Code?

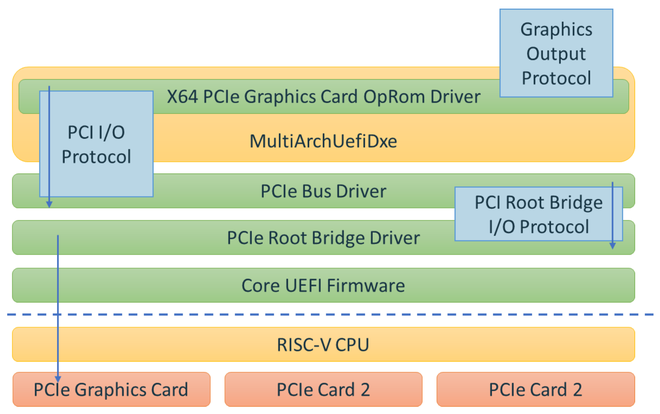
☑ Specifically made for this scenario!
- ❖Processor Independence
  - ❖sizeof(VOID*) is a runtime operation.
  - ❖VM takes care of 32 vs 64 vs 128-bit issues.
- ❖TianoCore comes with an interpreter.



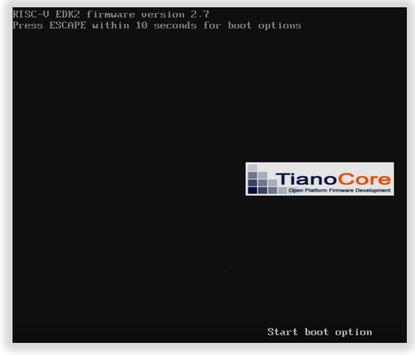
❌ Not used by the industry!
- ❖No tooling – the only supported and proprietary C language compiler has been retired.
  - ❖Some OSS now exists
    - ❖https://github.com/yabits/ebcvm / ELVM
    - ❖https://github.com/pbatard/fasmg-ebc
- ❖Different performance profile - interpreted code.
- ❖Didn't make a come-back when the Arm ServerReady/SystemReady ecosystem explored this space.

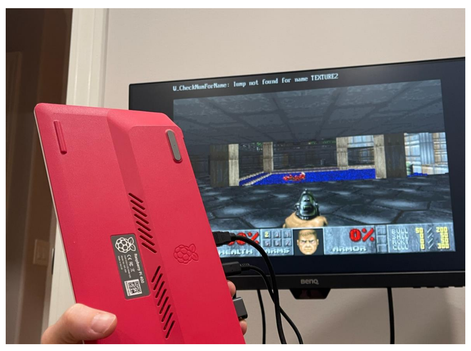
MultiArchUefiPkg

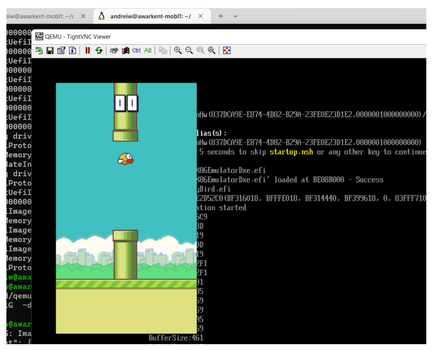
- ❖Rewrite of Linaro's X86EmulatorPkg
- ❖Portable: Supports AArch64 and 64-bit RISC-V UEFI hosts.
- ❖64-bit x64 and AArch64 UEFI Boot Service emulation.
- ❖Clean: Abstracts Qemu/TCG with Unicorn Engine API.
- ❖https://github.com/intel/MultiArchUefiPkg
- ❖RISE Project in the Firmware WG.
- ❖2/3rds the binary size on AArch64, compared to X86EmulatorPkg.
- ❖Comes with regression suite, improved emulated environment modeling.



- ❖Possible entirely due to narrowly-defined EFI ABI
- ❖Models Boot Services environment, with certain services filtered or disabled.
- ❖Tiano support for foreign binaries - **EDKII_PECOFF_IMAGE_EMULATOR_PROTOCOL**
- ❖Emulation is only interesting if thunking goes both ways!
  - ❖RISC-V No-Execute handler traps for native → emulated.
  - ❖Unicorn No-Execute handler traps for emulated → native.







Long term solutions?

- ❖Existing ISAs are a moving target.
- ❖Multiple ISA support in COTS adapters unlikely
- ❖Embrace emulation, but how?
  1. Resurrect EBC.
     - ❖Never been more relevant with 4 64-bit ISAs and 128-bit ISAs following.
     - ❖Tooling
     - ❖Performance
  2. Constrain x64 OpRom environment.
     - ❖Meet IHVs half-way.
     - ❖Pick a subset of x64– ring3, basic SSE, etc.
     - ❖Generate code that will guarantee to run via MultiArchUefiPkg.
  3. WASM for EFI.
     - ❖Great tooling.
     - ❖Sandboxing?
     - ❖128-bit support?.

intel®

RISE
RISC-V Software Ecosystem