OpenCore

Reference Manual (0.9.4.5)

[2023.08.31]

Failsafe: `false`
Description: Use PciRootBridgeIo for `ResizeGpuBars` and `ResizeAppleGpuBars`

The quirk makes `ResizeGpuBars` and `ResizeAppleGpuBars` use `PciRootBridgeIo` instead of PciIo. This is needed on systems with a buggy `PciIo` implementation where trying to configure Resizable BAR results in `Capability I/O Error`. Typically this is required on older systems which have been modified with ReBarUEFI.

13. `ShimRetainProtocol`
Type: `plist boolean`
Failsafe: `false`
Description: Request Linux shim to keep protocol installed for subsequent image loads.

This option is only required if chaining OpenCore from shim. It must be set in order to allow OpenCore to launch items which are verified by certificates present in shim, but not in the system Secure Boot database.

14. `ResizeGpuBars`
Type: `plist integer`
Failsafe: `-1`
Description: Configure GPU PCI BAR sizes.

This quirk sets GPU PCI BAR sizes as specified or chooses the largest available below the `ResizeGpuBars` value. The specified value follows PCI Resizable BAR spec. Use `0` for 1 MB, `1` for 2 MB, `2` for 4 MB, and so on up to `19` for 512 GB.

Resizable BAR technology allows to ease PCI device programming by mapping a configurable memory region, BAR, into CPU address space (e.g. VRAM to RAM) as opposed to a fixed memory region. This technology is necessary, because one cannot map the largest memory region by default, for the reasons of backwards compatibility with older hardware not supporting 64-bit BARs. Consequentially devices of the last decade use BARs up to 256 MB by default (4 remaining bits are used by other data) but generally allow resizing them to both smaller and larger powers of two (e.g. from 1 MB up to VRAM size).

Operating systems targeting x86 platforms generally do not control PCI address space, letting UEFI firmware decide on the BAR addresses and sizes. This illicit practice resulted in Resizable BAR technology being unused up until 2020 despite being standardised in 2008 and becoming widely available in the hardware soon after.

Modern UEFI firmware allow the use of Resizable BAR technology but generally restrict the configurable options to failsafe default (`OFF`) and maximum available (`ON`). This quirk allows to fine-tune this value for testing and development purposes.

Consider a GPU with 2 BARs:

- `BAR0` supports sizes from 256 MB to 8 GB. Its value is 4 GB.
- `BAR1` supports sizes from 2 MB to 256 MB. Its value is 256 MB.

Example 1: Setting `ResizeGpuBars` to 1 GB will change `BAR0` to 1 GB and leave `BAR1` unchanged.
Example 2: Setting `ResizeGpuBars` to 1 MB will change `BAR0` to 256 MB and `BAR0` to 2 MB.
Example 3: Setting `ResizeGpuBars` to 16 GB will change `BAR0` to 8 GB and leave `BAR1` unchanged.

Note 1: This quirk shall not be used to workaround macOS limitation to address BARs over 1 GB. `ResizeAppleGpuBars` should be used instead.

Note 2: While this quirk can increase GPU PCI BAR sizes, this will not work on most firmware as is, because the quirk does not relocate BARs in memory, and they will likely overlap. In most cases it is best to either update the firmware to the latest version or customise it with a specialised driver like ReBarUEFI.

15. `TscSyncTimeout`
Type: `plist integer`
Failsafe: `0`
Description: Attempts to perform TSC synchronisation with a specified timeout.

The primary purpose of this quirk is to enable early bootstrap TSC synchronisation on some server and laptop models when running a debug XNU kernel. For the debug kernel the TSC needs to be kept in sync across the cores before any kext could kick in rendering all other solutions problematic. The timeout is specified in microseconds and depends on the amount of cores present on the platform, the recommended starting value is `500000`.