

Technical Disclosure Commons

Defensive Publications Series

October 2019

POST ERROR INDICATION IN UEFI DRIVER MODEL DRIVERS

HP INC

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

INC, HP, "POST ERROR INDICATION IN UEFI DRIVER MODEL DRIVERS", Technical Disclosure Commons, (October 29, 2019)

https://www.tdcommons.org/dpubs_series/2609



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Post Error Indication in UEFI Driver Model Drivers

Abstract:

- In general, when UEFI DXE Core dispatcher starts to load DXE and UEFI drivers, BIOS sends out generic post code for each DXE/UEFI drivers.
- After a UEFI driver is loaded via the boot services, this UEFI driver can be started anywhere to connect from a controller. This UEFI driver can be stopped to disconnect from a controller as well.
- Basically, BIOS has no way to issue post code when starting/stopping a UEFI driver, especially UEFI driver is protected in binary format. Once system gets hang up problem when starting/stopping UEFI driver, it is hard to identify the problem root cause.
- UEFI defines Component Name protocol and Load Image protocol. Both can be used to get a readable name and a globally unique ID (GUID) of a UEFI driver.
- Before starting a UEFI driver on a controller, BIOS can read driver name and driver GUID in advance and store them into ECMOS.
- After the call complete of starting a UEFI driver, both UEFI driver name and GUID will be cleared from ECMOS.
- Once the call of starting a UEFI driver doesn't complete, BIOS keeps ECMOS content for UEFI driver name and GUID.
- In next boot, BIOS can issue a private WMI command to read UEFI post error code from ECMOS and reports it to windows application for record. After that, BIOS can issue a private WMI command to clear UEFI post error code from ECMOS.

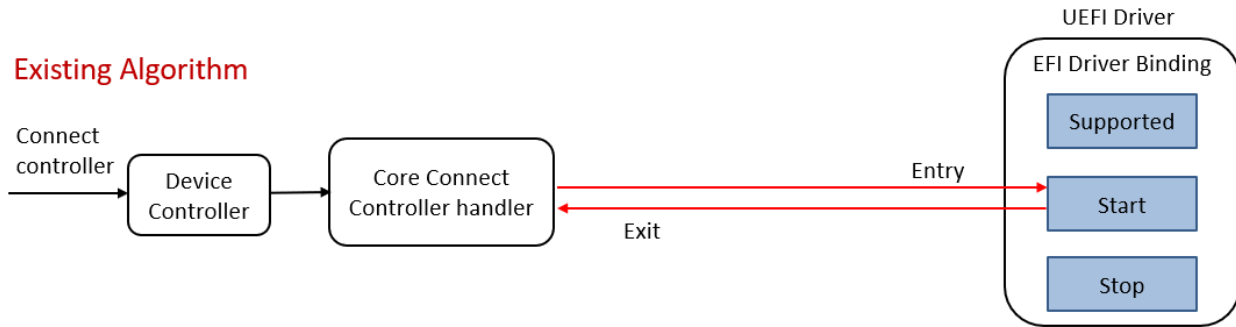
Design Construction:

- HW: No HW need.
- SW: No SW need.
- BIOS: Before starting a UEFI driver, BIOS gets useful information of a UEFI driver via UEFI protocol services and stores them into ECMOS. BIOS clears ECMOS content after a call of starting a UEFI driver is complete.

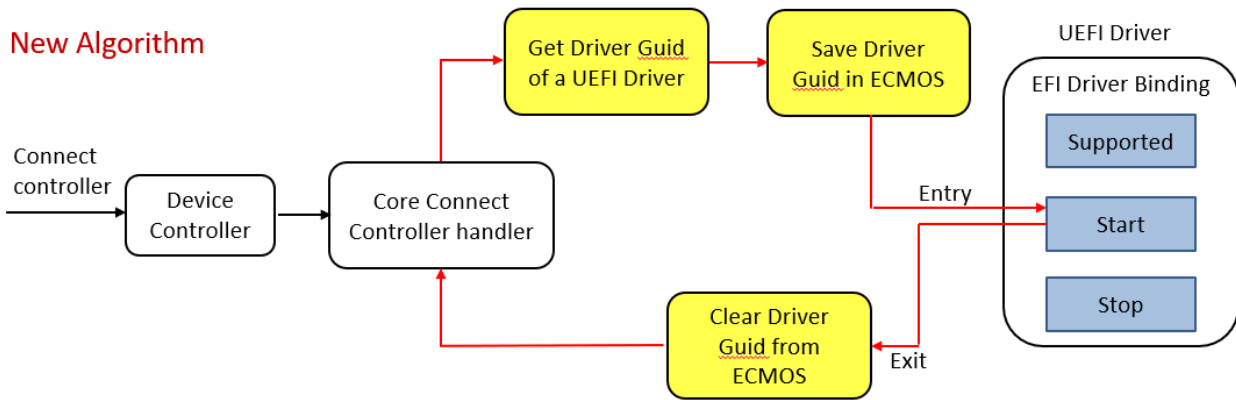
< Block Flow Diagram >

Feature Model

Existing Algorithm

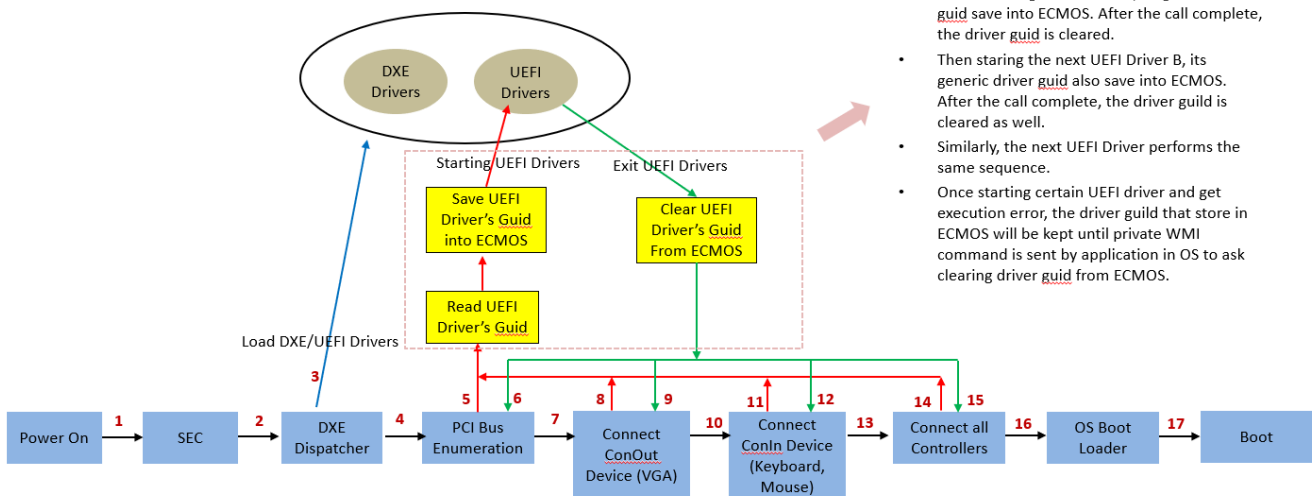


New Algorithm



< Flow Chart >

Boot Flow



- When starting UEFI Driver A, its generic driver guid save into ECMOS. After the call complete, the driver guid is cleared.
- Then starting the next UEFI Driver B, its generic driver guid also save into ECMOS. After the call complete, the driver guid is cleared as well.
- Similarly, the next UEFI Driver performs the same sequence.
- Once starting certain UEFI driver and get execution error, the driver guid that store in ECMOS will be kept until private WMI command is sent by application in OS to ask clearing driver guid from ECMOS.

- **Business Strategy/Advantages**

1. Identify post error problem in UEFI drivers easily, including IP license protection UEFI drivers with binary format.
2. Provide more useful debug message when system get hang up problem at pre-boot phase.

Disclosed by Chia-Cheng Lin, Neo Chou, Kyle Chang, Nung-Kai Chen, HP Inc.