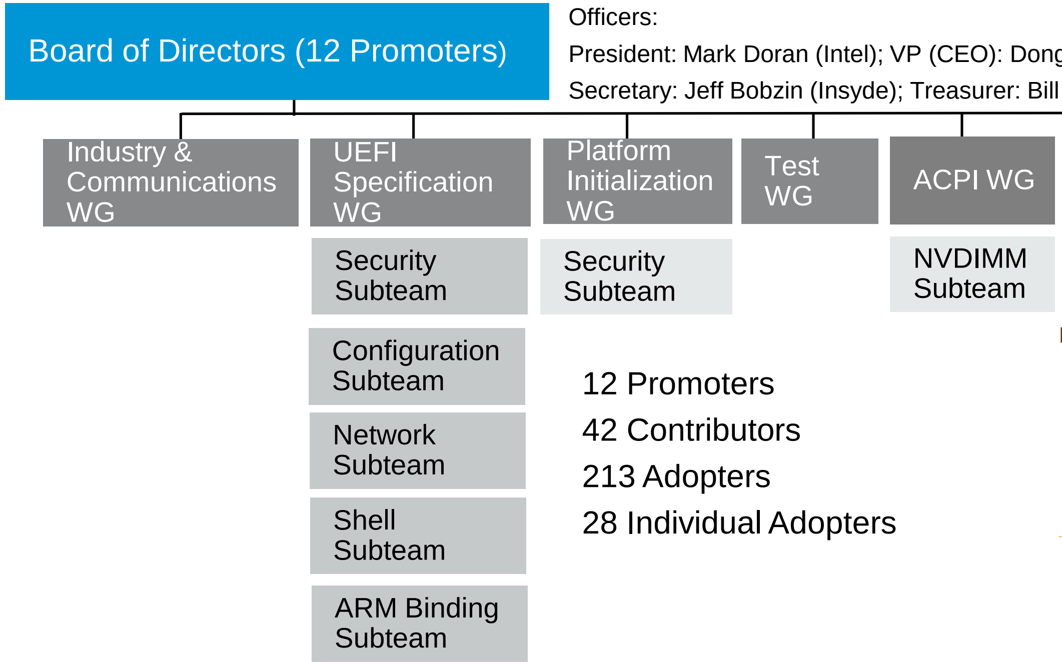
Hewlett Packard
Enterprise

UEFI and RISC-V

Abner Chang, Dong Wei

The UEFI Forum Organization



Board of Directors (12 Promoters)

Officers:

President: Mark Doran (Intel); VP (CEO): Dong Wei (HPE)

Secretary: Jeff Bobzin (Insyde); Treasurer: Bill Keown (Lenovo)

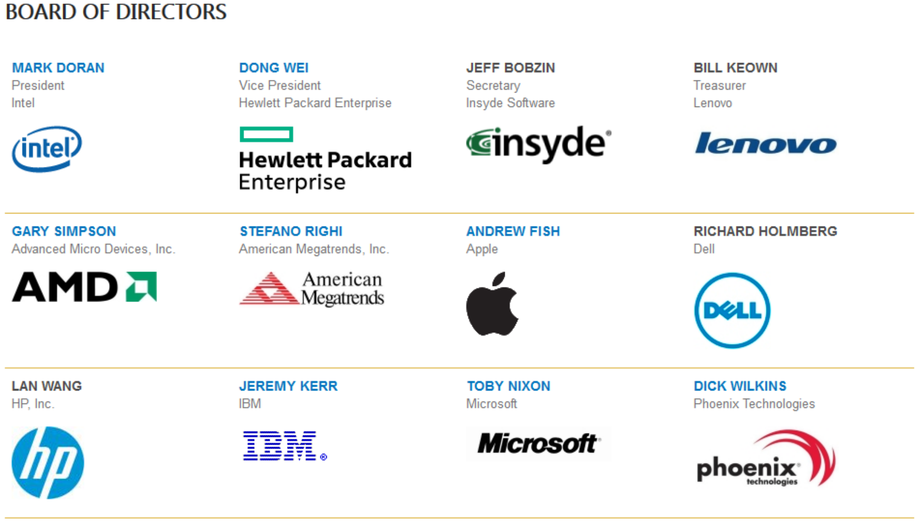
- Industry & Communications WG
- UEFI Specification WG
  - Security Subteam
  - Configuration Subteam
  - Network Subteam
  - Shell Subteam
  - ARM Binding Subteam
- Platform Initialization WG
  - Security Subteam
- Test WG
- ACPI WG
  - NVDIMM Subteam
- Security Response Team

12 Promoters

42 Contributors

213 Adopters

28 Individual Adopters



BOARD OF DIRECTORS

MARK DORAN President Intel	DONG WEI Vice President Hewlett Packard Enterprise	JEFF BOBZIN Secretary Insyde Software	BILL KEOWN Treasurer Lenovo
GARY SIMPSON Advanced Micro Devices, Inc.	STEFANO RIGHI American Megatrends, Inc.	ANDREW FISH Apple	RICHARD HOLMBERG Dell
LAN WANG HP, Inc.	JEREMY KERR IBM	TOBY NIXON Microsoft	DICK WILKINS Phoenix Technologies

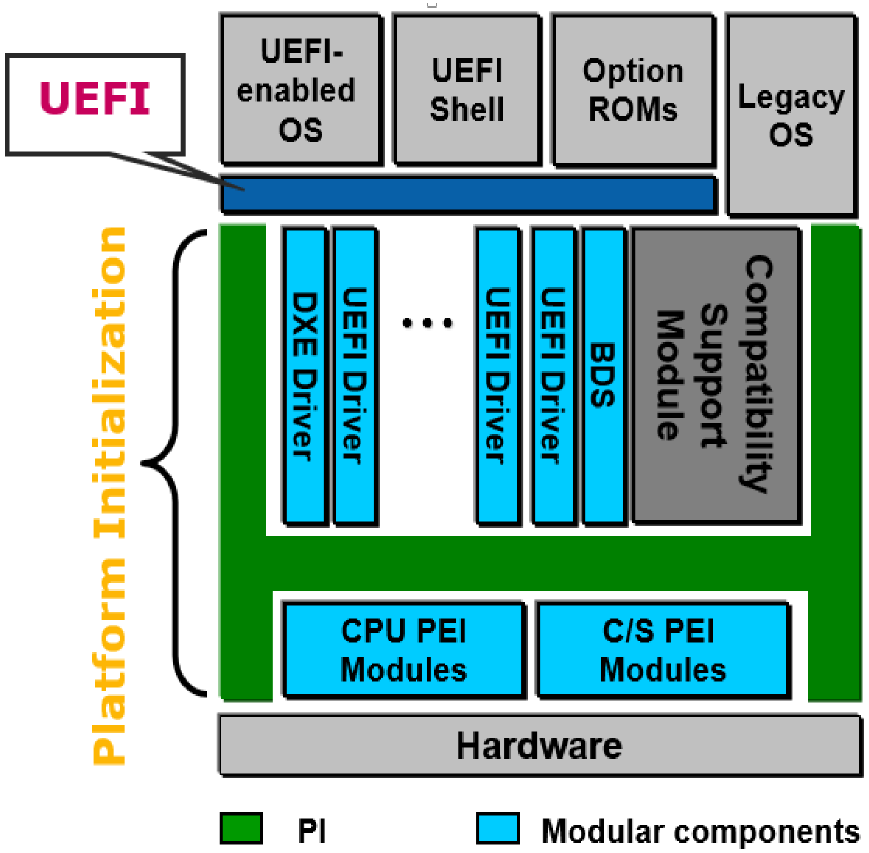
UEFI Technology

Platform Initialization (PI)

– Interfaces produced & consumed by firmware only; promote interoperability between firmware components

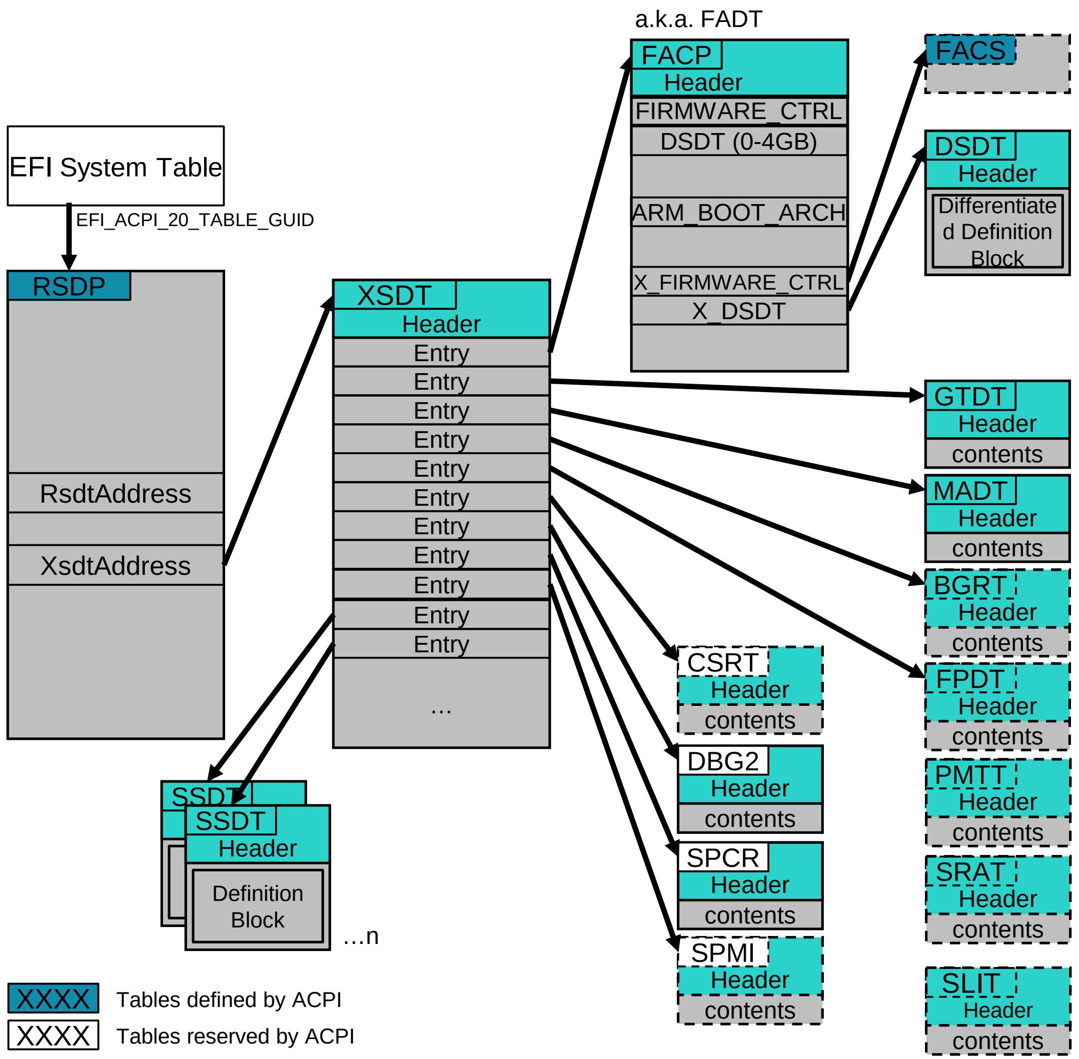
UEFI

– Pre-OS (and limited runtime program interfaces) between UEFI Applications (incl. OSes)/UEFI Drivers and system firmware



ACPI Technology

- Static tables and primary runtime interpreted control methods provided by system firmware to the OS for system configuration, power management and error handling
- Processor architecture agnostic



UEFI & ACPI History

UEFI History

1995 – HP/Intel needed a boot architecture for Itanium servers that overcame BIOS PC-AT limitations

1997 - 2000 – Intel created EFI with HP and others in the industry, made it processor agnostic (x86, ia64)

2004 – **tianocore.org**, open source EFI community launched

2005 – **Unified EFI (UEFI)** The UEFI Forum, with 11 promoters, was formed to standardize EFI, extended to x64

2009 – UEFI extended to ARM AArch32

2012 – Windows 8 and ubiquitous native UEFI adoption for client PCs (Boot Performance, Secure Boot focused)

2013 – Linux Distros extended support for UEFI Secure Boot. First Linux Foundation hosted UEFI Plugfest. UEFI v2.4 extended to ARM AArch64.

ACPI History

1996 – Intel/Microsoft/Toshiba created ACPI 1.0 for 16 and 32 bit PC client devices

2000 – Compaq/Intel/Microsoft/Phoenix/Toshiba publishes ACPI 2.0 for 64-bit support as well as support for multiprocessor workstations and servers

2004 – HP/Intel/Microsoft/Phoenix/Toshiba published ACPI 3.0 further enhancing the spec to support both client and server systems

2009 – ACPI 4.0 is published providing additional support for both client and server systems

2011 – Hardware-reduced ACPI model was introduced into the published ACPI 5.0 spec to include the support for SoC devices. ARM specific descriptions are also introduced

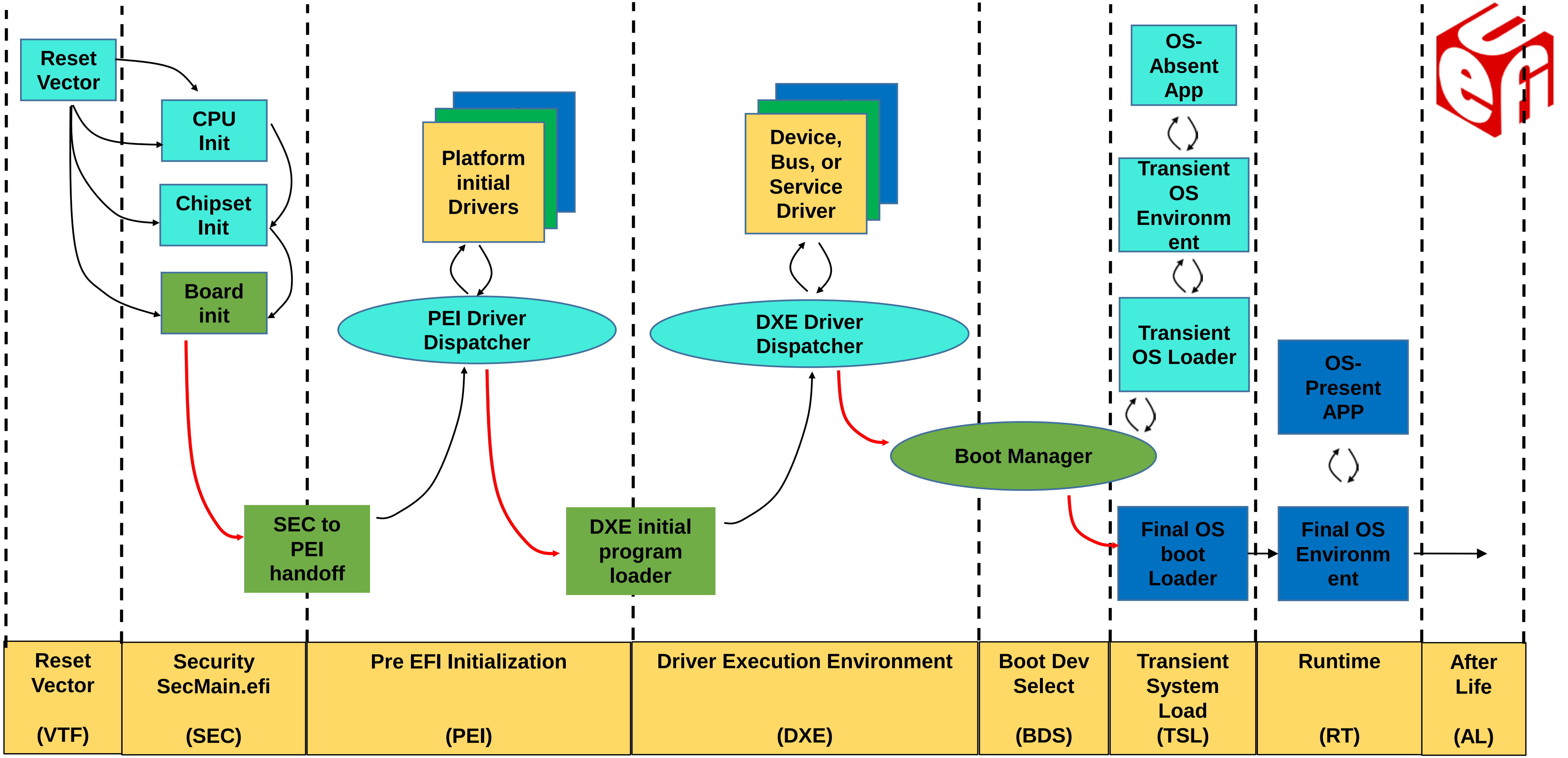
2013 – ACPI Asset transferred to the UEFI Forum.

UEFI as the converged firmware infrastructure

2014 – ACPI v5.1 for ARM AArch64 support
(e.g., ARM SBSA/SBBR servers)

2015 – UEFI v2.5, PI v1.4, ACPI v6.0 for NVDIMM support

2016 – Ready for RISC-V?



Reset Vector

CPU Init

Chipset Init

Board init

SEC to PEI handoff

Platform initial Drivers

PEI Driver Dispatcher

DXE initial program loader

Device, Bus, or Service Driver

DXE Driver Dispatcher

Boot Manager

OS-Absent App

Transient OS Environm ent

Transient OS Loader

Final OS boot Loader

OS-Present APP

Final OS Environm ent

Reset Vector (VTF)	Security SecMain.efi (SEC)	Pre EFI Initialization (PEI)	Driver Execution Environment (DXE)	Boot Dev Select (BDS)	Transient System Load (TSL)	Runtime (RT)	After Life (AL)

Power on → [....Platform initialization..] → [....OS boot...] → Shutdown

UEFI/PI Execution Phases

RISC-V UEFI Port on EDKII (EFI Development Kit II)

OVMF (Open Virtual Machine Firmware) RISC-V Package on QEMU

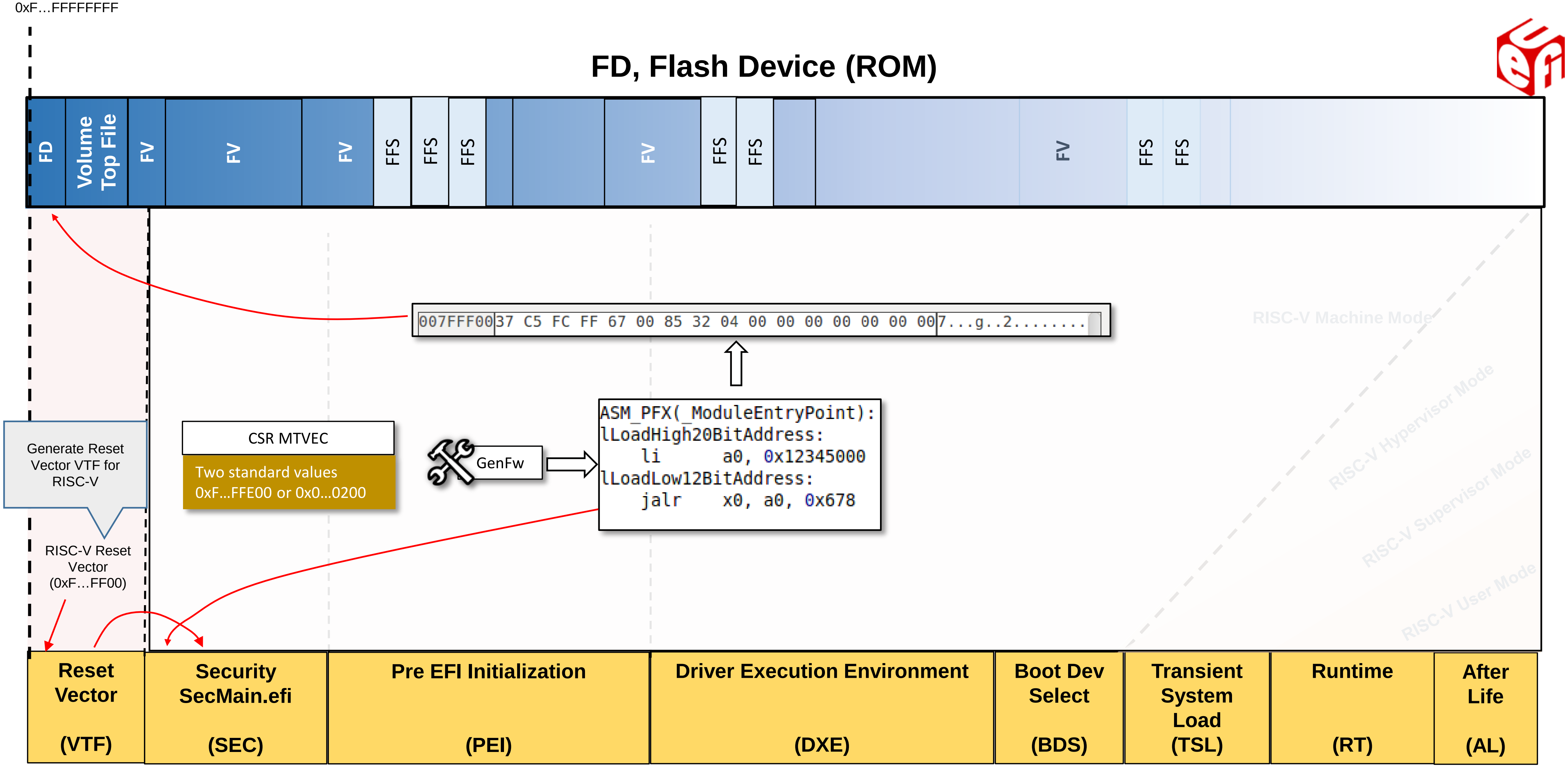
Reset Vector	Security SecMain.efi	Pre EFI Initialization	Driver Execution Environment	Boot Dev Select	Transient System Load	Runtime	After Life
(VTF)	(SEC)	(PEI)	(DXE)	(BDS)	(TSL)	(RT)	(AL)

Power on → [....Platform initialization..] → [....OS boot...] → Shutdown

UEFI/PI Execution Phases

Hewlett Packard
Enterprise

FD, Flash Device (ROM)



```
007FFF00 37 C5 FC FF 67 00 85 32 04 00 00 00 00 00 00 00 7...g..2........
```

```
ASM_PFX(_ModuleEntryPoint):
lLoadHigh20BitAddress:
    li       a0, 0x12345000
lLoadLow12BitAddress:
    jalr     x0, a0, 0x678
```

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

CSR MTVEC

Two standard values 0xF…FFE00 or 0x0…0200

Reset Vector (VTF)

Security SecMain.efi (SEC)

Pre EFI Initialization (PEI)

Driver Execution Environment (DXE)

Boot Dev Select (BDS)

Transient System Load (TSL)

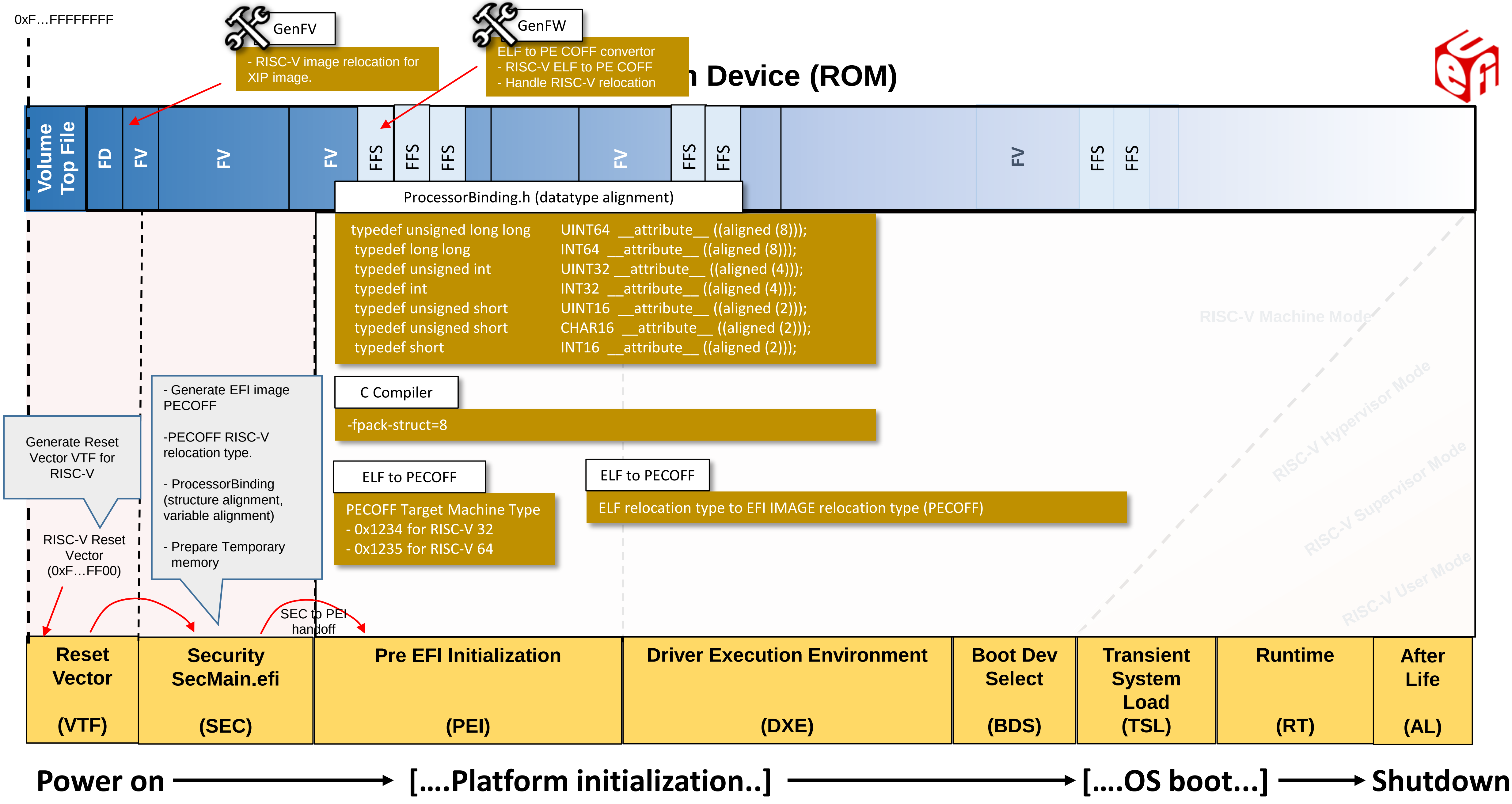
Runtime (RT)

After Life (AL)

Power on → [….Platform initialization..] → [….OS boot…] → Shutdown

UEFI/PI Execution Phases

Hewlett Packard Enterprise



GenFV

- RISC-V image relocation for XIP image.

GenFW

ELF to PE COFF convertor
- RISC-V ELF to PE COFF
- Handle RISC-V relocation

ProcessorBinding.h (datatype alignment)

```
typedef unsigned long long    UINT64  __attribute__ ((aligned (8)));
 typedef long long            INT64  __attribute__ ((aligned (8)));
 typedef unsigned int         UINT32 __attribute__ ((aligned (4)));
 typedef int                  INT32  __attribute__ ((aligned (4)));
 typedef unsigned short       UINT16  __attribute__ ((aligned (2)));
 typedef unsigned short       CHAR16  __attribute__ ((aligned (2)));
 typedef short                INT16  __attribute__ ((aligned (2)));
```

C Compiler

-fpack-struct=8

ELF to PECOFF

PECOFF Target Machine Type
- 0x1234 for RISC-V 32
- 0x1235 for RISC-V 64

ELF to PECOFF

ELF relocation type to EFI IMAGE relocation type (PECOFF)

Generate Reset Vector VTF for RISC-V

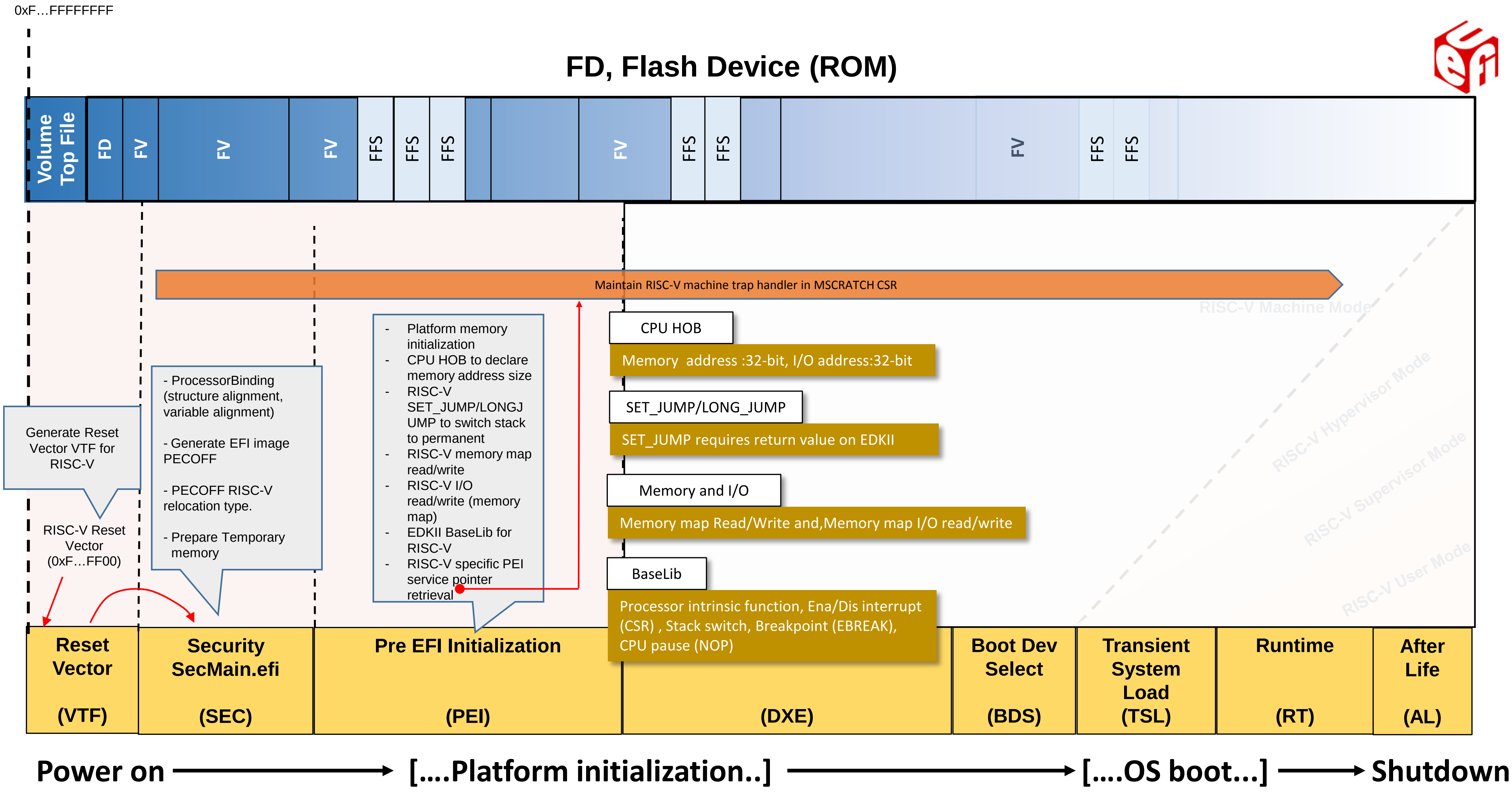
- Generate EFI image PECOFF

-PECOFF RISC-V relocation type.

- ProcessorBinding (structure alignment, variable alignment)

- Prepare Temporary memory

UEFI/PI Execution Phases



Maintain RISC-V machine trap handler in MSCRATCH CSR

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)
- Generate EFI image PECOFF
- PECOFF RISC-V relocation type.
- Prepare Temporary memory

- Platform memory initialization
- CPU HOB to declare memory address size
- RISC-V SET_JUMP/LONGJUMP to switch stack to permanent
- RISC-V memory map read/write
- RISC-V I/O read/write (memory map)
- EDKII BaseLib for RISC-V
- RISC-V specific PEI service pointer retrieval

CPU HOB

Memory address :32-bit, I/O address:32-bit

SET_JUMP/LONG_JUMP

SET_JUMP requires return value on EDKII

Memory and I/O

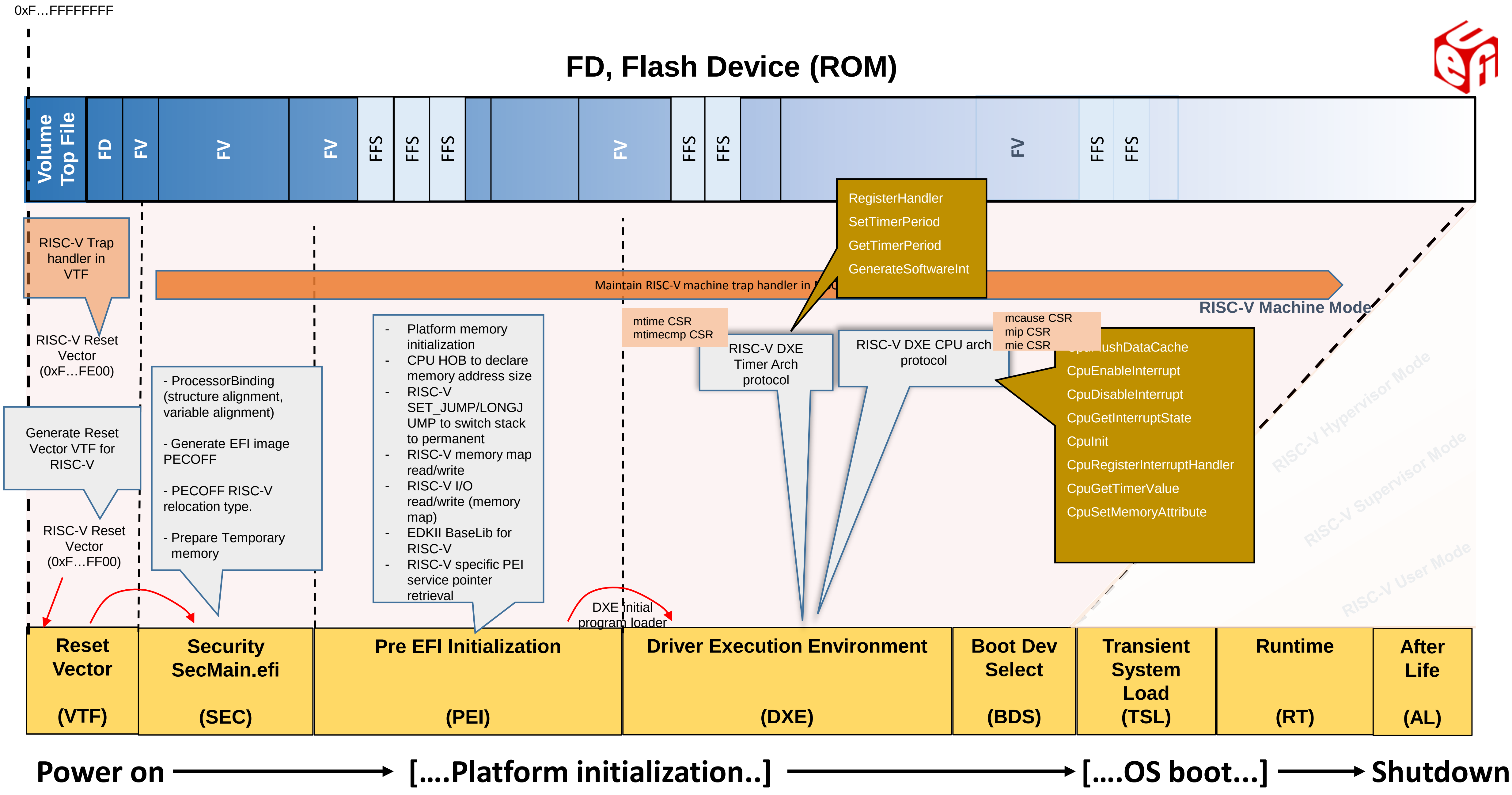
Memory map Read/Write and,Memory map I/O read/write

BaseLib

Processor intrinsic function, Ena/Dis interrupt (CSR) , Stack switch, Breakpoint (EBREAK), CPU pause (NOP)

Reset Vector (VTF)	Security SecMain.efi (SEC)	Pre EFI Initialization (PEI)	(DXE)	Boot Dev Select (BDS)	Transient System Load (TSL)	Runtime (RT)	After Life (AL)

UEFI/PI Execution Phases



RISC-V Trap handler in VTF

RISC-V Reset Vector (0xF…FE00)

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)
- Generate EFI image PECOFF
- PECOFF RISC-V relocation type.
- Prepare Temporary memory

- Platform memory initialization
- CPU HOB to declare memory address size
- RISC-V SET_JUMP/LONGJUMP to switch stack to permanent
- RISC-V memory map read/write
- RISC-V I/O read/write (memory map)
- EDKII BaseLib for RISC-V
- RISC-V specific PEI service pointer retrieval

DXE initial program loader

mtime CSR
mtimecmp CSR

RISC-V DXE Timer Arch protocol

RegisterHandler
SetTimerPeriod
GetTimerPeriod
GenerateSoftwareInt

RISC-V DXE CPU arch protocol

mcause CSR
mip CSR
mie CSR

CpuFlushDataCache
CpuEnableInterrupt
CpuDisableInterrupt
CpuGetInterruptState
CpuInit
CpuRegisterInterruptHandler
CpuGetTimerValue
CpuSetMemoryAttribute

UEFI/PI Execution Phases

0xF…FFFFFFFF

FD, Flash Device (ROM)

Volume Top File | FD | FV | FV | FV | FFS | FFS | FFS | FV | FFS | FFS | FV | FFS

RISC-V Trap handler in VTF

RISC-V Reset Vector (0xF…FE00)

User Mode Trap Handler

Supervisor Mode Trap Handler

- ProcessorBinding (structure alignment, variable alignment)

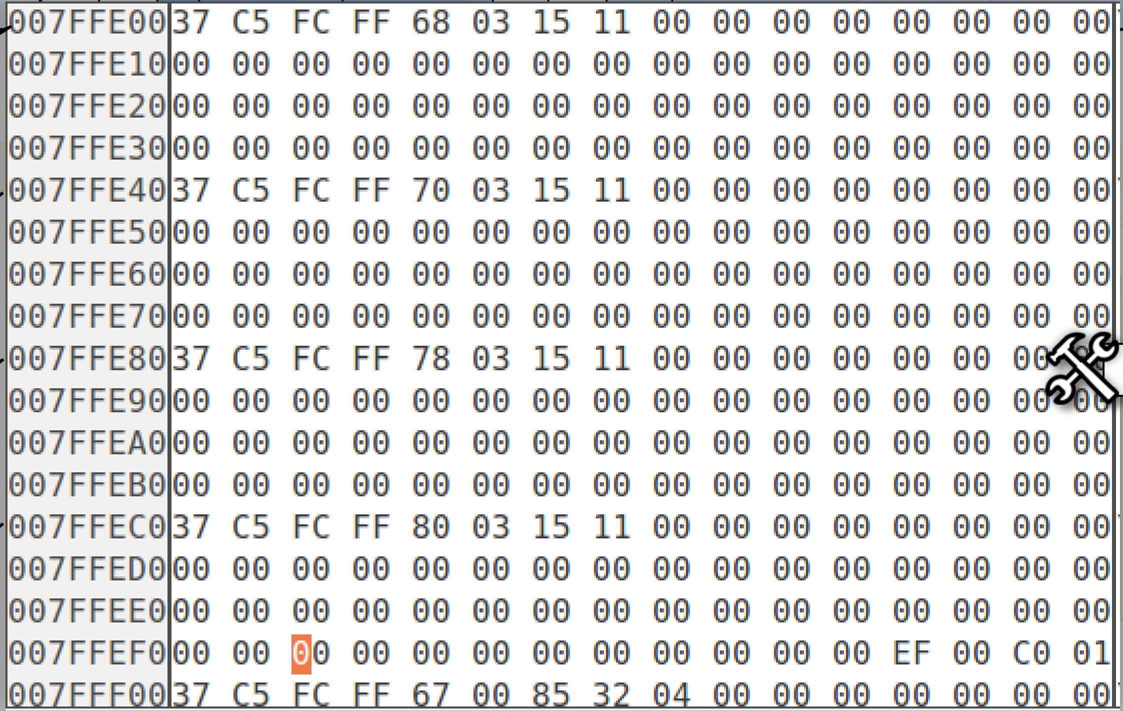
Hypervisor Mode Trap Handler

- PECOFF RISC-V relocation type.

Machine Mode Trap Handler

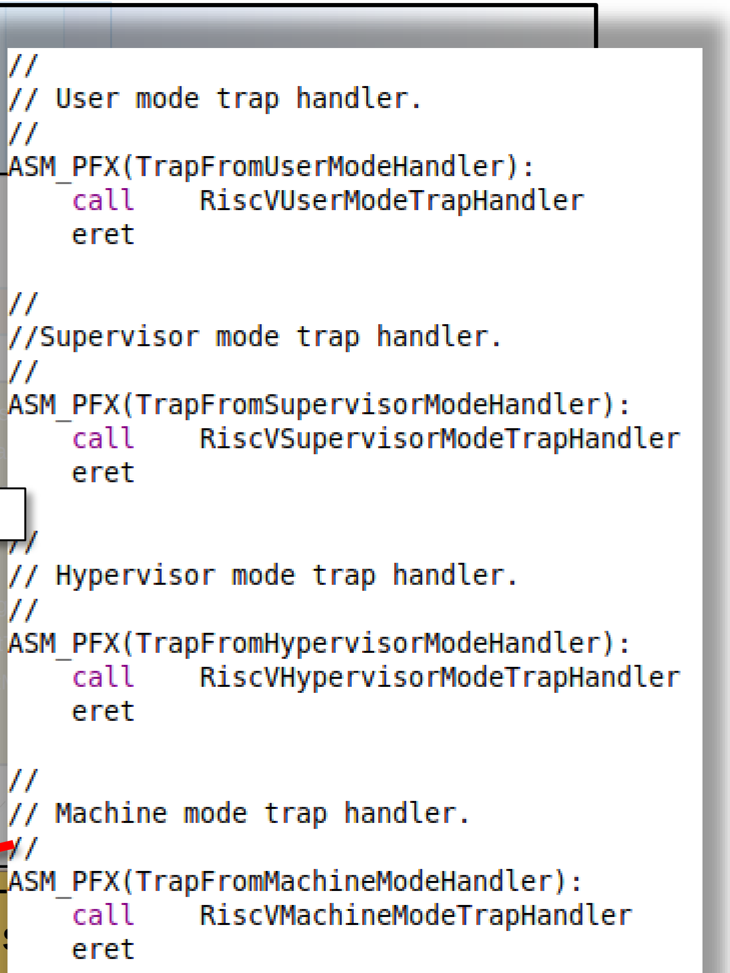
Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)



```
007FFE00 37 C5 FC FF 68 03 15 11 00 00 00 00 00 00 00 00
007FFE10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE40 37 C5 FC FF 70 03 15 11 00 00 00 00 00 00 00 00
007FFE50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFE80 37 C5 FC FF 78 03 15 11 00 00 00 00 00 00 00 00
007FFE90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFEA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFEB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFEC0 37 C5 FC FF 80 03 15 11 00 00 00 00 00 00 00 00
007FFED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFEE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007FFEF0 00 00 00 00 00 00 00 00 00 00 00 00 EF 00 C0 01
007FFF00 37 C5 FC FF 67 00 85 32 04 00 00 00 00 00 00 00
```

GenFw



```
//
// User mode trap handler.
//
ASM_PFX(TrapFromUserModeHandler):
    call    RiscVUserModeTrapHandler
    eret

//
//Supervisor mode trap handler.
//
ASM_PFX(TrapFromSupervisorModeHandler):
    call    RiscVSupervisorModeTrapHandler
    eret

//
// Hypervisor mode trap handler.
//
ASM_PFX(TrapFromHypervisorModeHandler):
    call    RiscVHypervisorModeTrapHandler
    eret

//
// Machine mode trap handler.
//
ASM_PFX(TrapFromMachineModeHandler):
    call    RiscVMachineModeTrapHandler
    eret
```

program loader

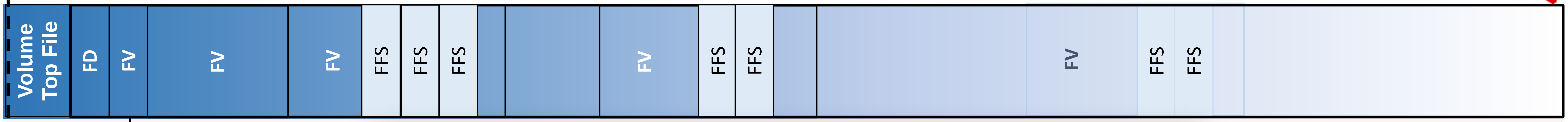
Reset Vector (VTF)	Security SecMain.efi (SEC)	Pre EFI Initialization (PEI)	Driver Execution Environment (DXE)	Boot Dev Select (BDS)	T Load (TSL)	(RT)	(AL)

Power on → [….Platform initialization..] → […OS boot…] → Shutdown

UEFI/PI Execution Phases

0xF…FFFFFFFF

FD, Flash Device (ROM)



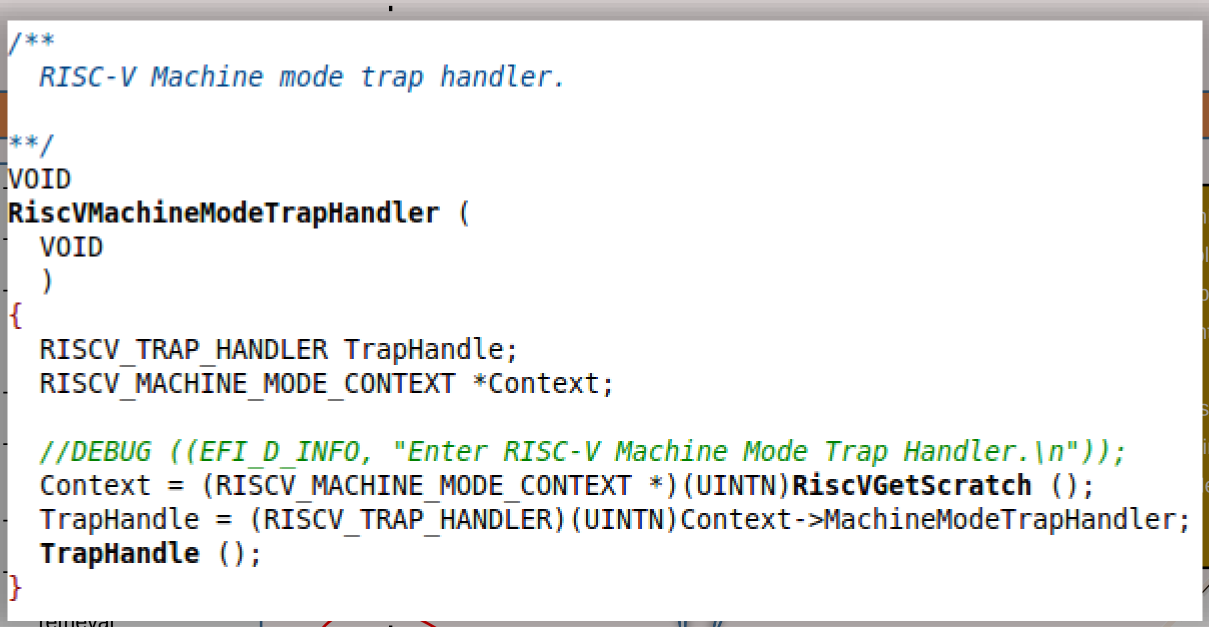
RISC-V Trap handler in VTF

RISC-V Reset Vector (0xF…FE00)

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)
- Generate EFI image PECOFF
- PECOFF RISC-V relocation type.
- Prepare Temporary memory



```
/**
  RISC-V Machine mode trap handler.

**/
VOID
RiscVMachineModeTrapHandler (
  VOID
  )
{
  RISCV_TRAP_HANDLER TrapHandle;
  RISCV_MACHINE_MODE_CONTEXT *Context;

  //DEBUG ((EFI_D_INFO, "Enter RISC-V Machine Mode Trap Handler.\n"));
  Context = (RISCV_MACHINE_MODE_CONTEXT *)(UINTN)RiscVGetScratch ();
  TrapHandle = (RISCV_TRAP_HANDLER)(UINTN)Context->MachineModeTrapHandler;
  TrapHandle ();
}
```

retrieval

RISC-V Machine Mode

nDataCache
leInterrupt
bleInterrupt
nterruptState

sterInterruptHandler
imerValue
emoryAttribute

RISC-V Hypervisor Mode

RISC-V Supervisor Mode

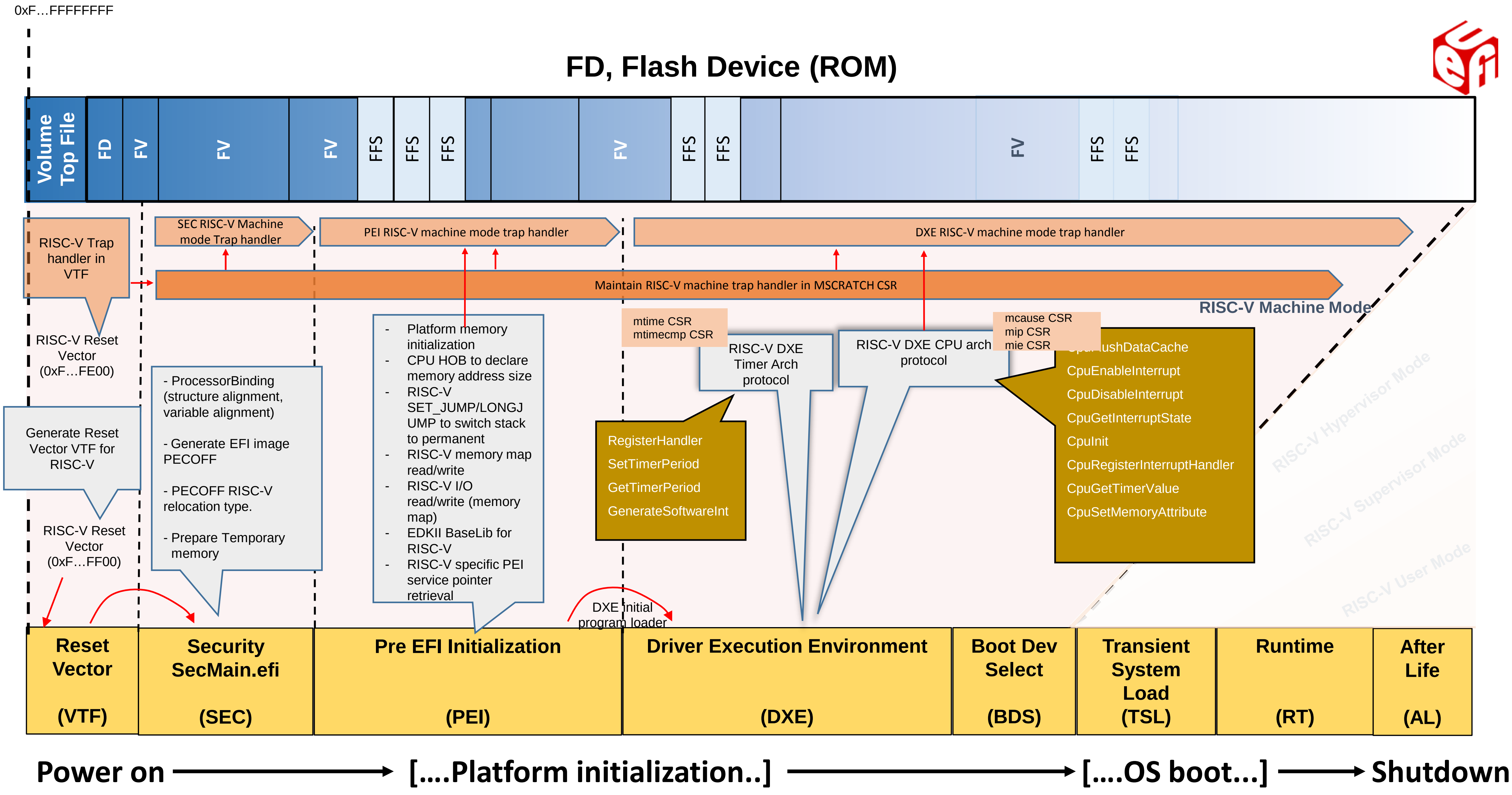
RISC-V User Mode

DXE initial program loader

Reset Vector (VTF)	Security SecMain.efi (SEC)	Pre EFI Initialization (PEI)	Driver Execution Environment (DXE)	Boot Dev Select (BDS)	Transient System Load (TSL)	Runtime (RT)	After Life (AL)

Power on → [….Platform initialization..] → […OS boot…] → Shutdown

UEFI/PI Execution Phases



0xF…FFFFFFFF

FD, Flash Device (ROM)

Volume Top File

FD

FV

FV

FV

FFS

FFS

FFS

FV

FFS

FFS

FV

FFS

FFS

RISC-V Trap handler in VTF

SEC RISC-V Machine mode Trap handler

PEI RISC-V machine mode trap handler

DXE RISC-V machine mode trap handler

Maintain RISC-V machine trap handler in MSCRATCH CSR

RISC-V Machine Mode

RISC-V Hypervisor Mode

RISC-V Supervisor Mode

RISC-V User Mode

RISC-V Reset Vector (0xF…FE00)

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)
- Generate EFI image PECOFF
- PECOFF RISC-V relocation type.
- Prepare Temporary memory

- Platform memory initialization
- CPU HOB to declare memory address size
- RISC-V SET_JUMP/LONGJUMP to switch stack to permanent
- RISC-V memory map read/write
- RISC-V I/O read/write (memory map)
- EDKII BaseLib for RISC-V
- RISC-V specific PEI service pointer retrieval

DXE initial program loader

mtime CSR
mtimecmp CSR

RISC-V DXE Timer Arch protocol

RegisterHandler
SetTimerPeriod
GetTimerPeriod
GenerateSoftwareInt

RISC-V DXE CPU arch protocol

mcause CSR
mip CSR
mie CSR

CpuFlushDataCache
CpuEnableInterrupt
CpuDisableInterrupt
CpuGetInterruptState
CpuInit
CpuRegisterInterruptHandler
CpuGetTimerValue
CpuSetMemoryAttribute

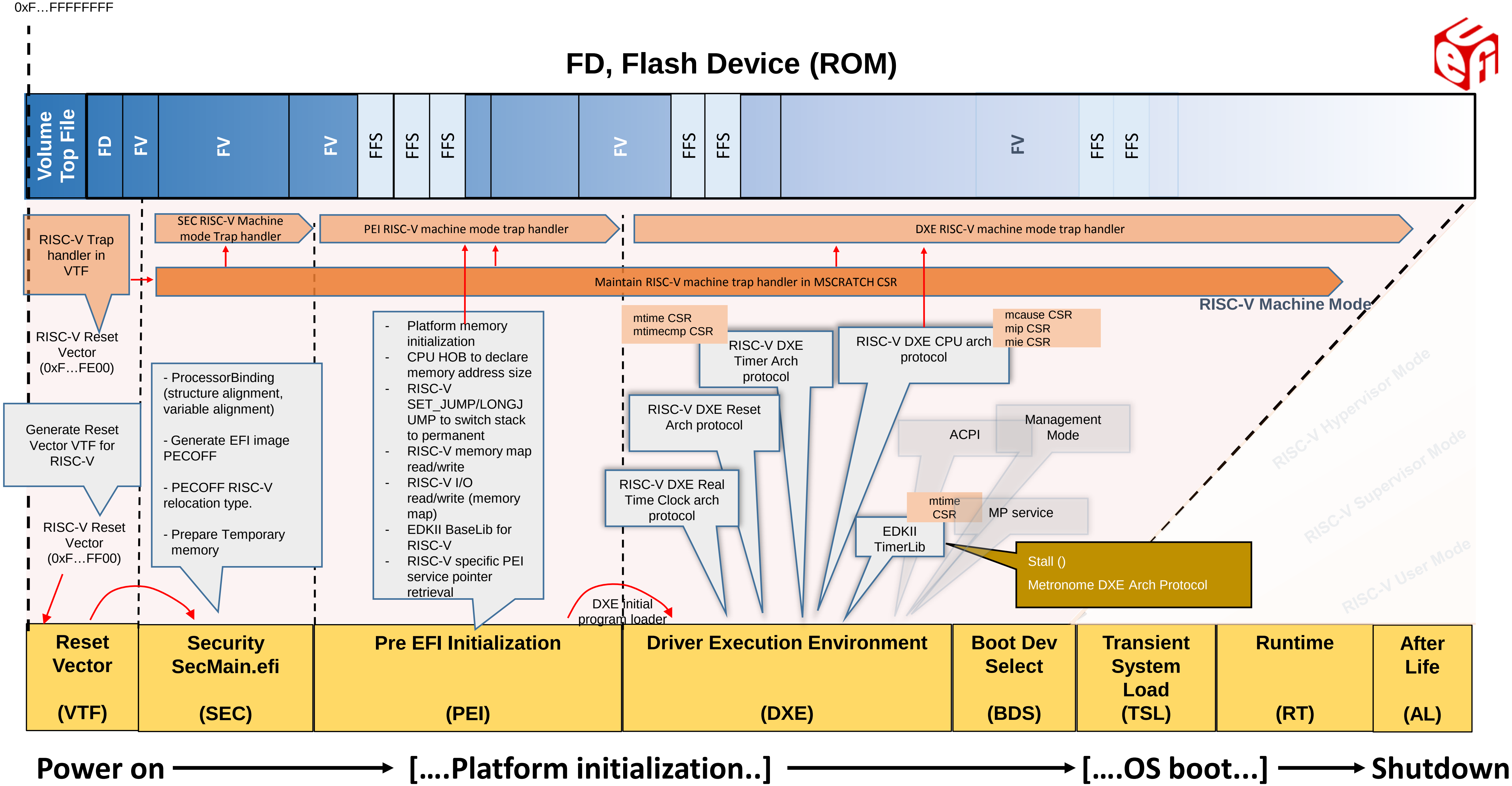
Reset Vector (VTF)	Security SecMain.efi (SEC)	Pre EFI Initialization (PEI)	Driver Execution Environment (DXE)	Boot Dev Select (BDS)	Transient System Load (TSL)	Runtime (RT)	After Life (AL)

Power on → [….Platform initialization..] → […OS boot…] → Shutdown

UEFI/PI Execution Phases

Hewlett Packard Enterprise

FD, Flash Device (ROM)



0xF…FFFFFFFF

Volume Top File | FD | FV | FV | FV | FFS | FFS | FFS | FV | FFS | FFS | FV | FFS | FFS

RISC-V Trap handler in VTF

SEC RISC-V Machine mode Trap handler

PEI RISC-V machine mode trap handler

DXE RISC-V machine mode trap handler

Maintain RISC-V machine trap handler in MSCRATCH CSR

RISC-V Machine Mode

RISC-V Reset Vector (0xF…FE00)

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)
- Generate EFI image PECOFF
- PECOFF RISC-V relocation type.
- Prepare Temporary memory

- Platform memory initialization
- CPU HOB to declare memory address size
- RISC-V SET_JUMP/LONGJUMP to switch stack to permanent
- RISC-V memory map read/write
- RISC-V I/O read/write (memory map)
- EDKII BaseLib for RISC-V
- RISC-V specific PEI service pointer retrieval

DXE initial program loader

mtime CSR
mtimecmp CSR

RISC-V DXE Timer Arch protocol

RISC-V DXE CPU arch protocol

mcause CSR
mip CSR
mie CSR

RISC-V DXE Reset Arch protocol

RISC-V DXE Real Time Clock arch protocol

ACPI

Management Mode

mtime CSR

MP service

EDKII TimerLib

Stall ()

Metronome DXE Arch Protocol

RISC-V Hypervisor Mode

RISC-V Supervisor Mode

RISC-V User Mode

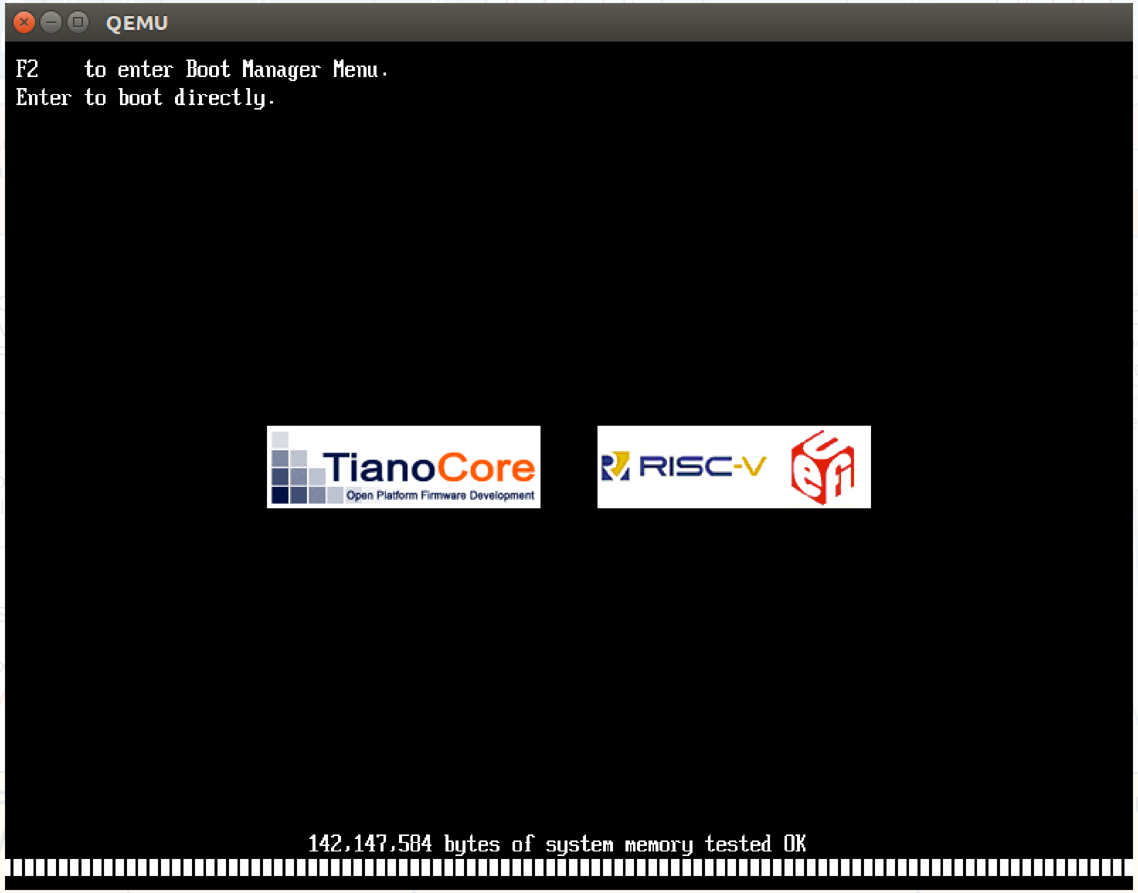
Reset Vector (VTF) | Security SecMain.efi (SEC) | Pre EFI Initialization (PEI) | Driver Execution Environment (DXE) | Boot Dev Select (BDS) | Transient System Load (TSL) | Runtime (RT) | After Life (AL)

Power on → [....Platform initialization..] → [....OS boot...] → Shutdown

UEFI/PI Execution Phases

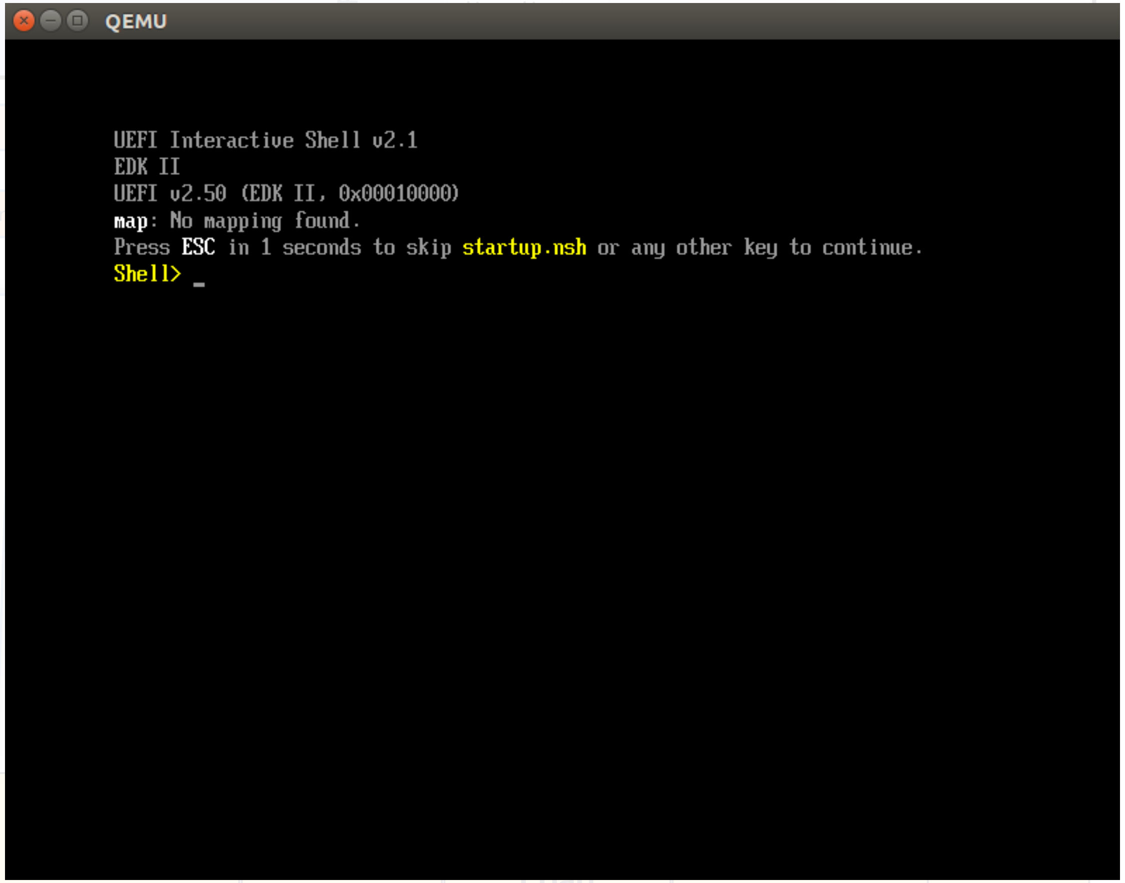
0xF…FFFFFFFF

FD, Flash Device (ROM)



```
F2     to enter Boot Manager Menu.
Enter  to boot directly.

              142,147,584 bytes of system memory tested OK
```

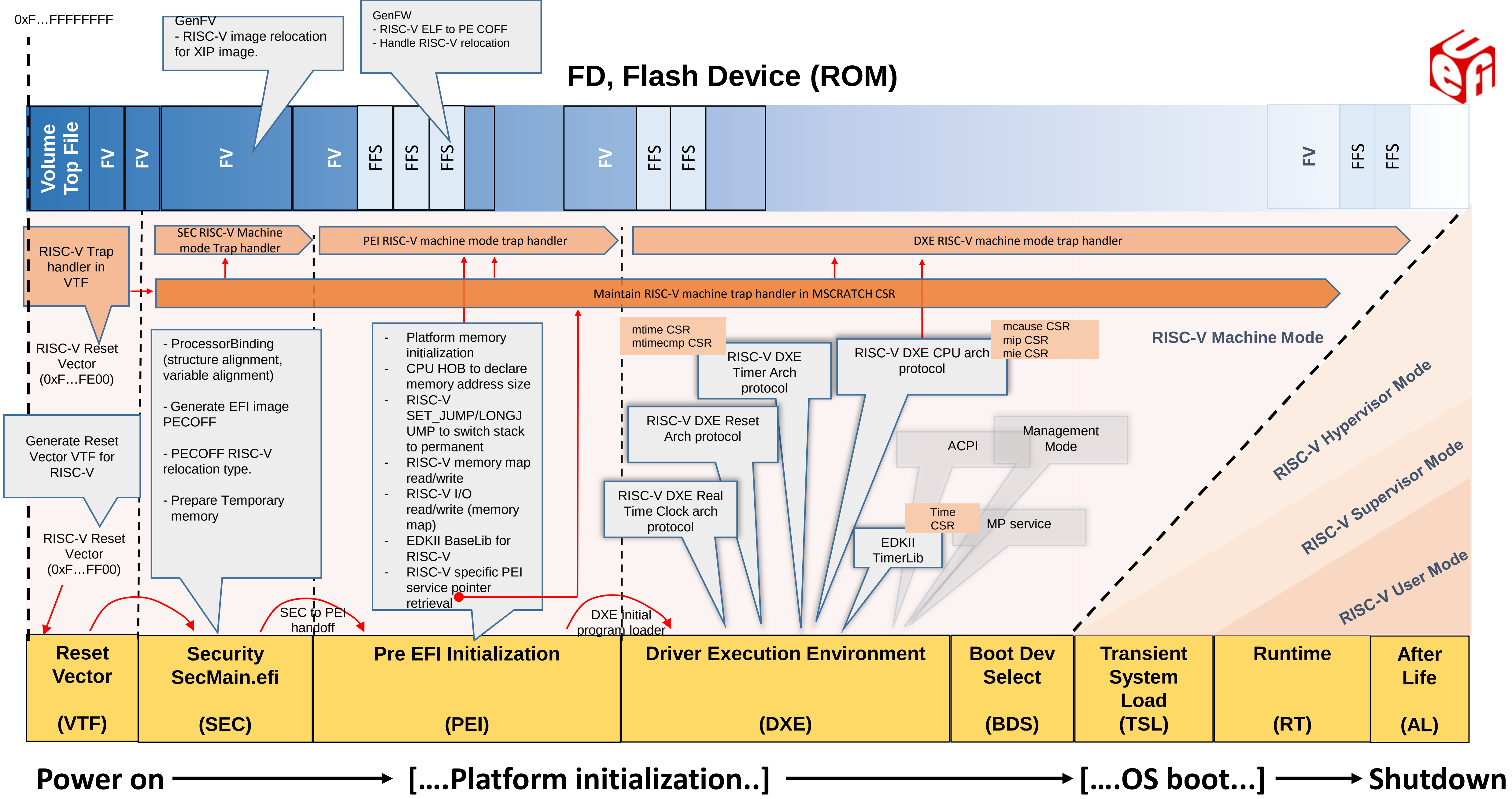


```
UEFI Interactive Shell v2.1
EDK II
UEFI v2.50 (EDK II, 0x00010000)
map: No mapping found.
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> _
```

(VTF) (SEC) (PEI) (DXE) (BDS) (TSL) (RT) (AL)

Power on → [....Platform initialization..] → [....OS boot...] → Shutdown

UEFI/PI Execution Phases



0xF…FFFFFFFF

GenFV
- RISC-V image relocation for XIP image.

GenFW
- RISC-V ELF to PE COFF
- Handle RISC-V relocation

FD, Flash Device (ROM)

Volume Top File

FV

FV

FV

FV

FFS

FFS

FFS

FV

FFS

FFS

FV

FFS

FFS

RISC-V Trap handler in VTF

SEC RISC-V Machine mode Trap handler

PEI RISC-V machine mode trap handler

DXE RISC-V machine mode trap handler

Maintain RISC-V machine trap handler in MSCRATCH CSR

RISC-V Reset Vector (0xF…FE00)

Generate Reset Vector VTF for RISC-V

RISC-V Reset Vector (0xF…FF00)

- ProcessorBinding (structure alignment, variable alignment)

- Generate EFI image PECOFF

- PECOFF RISC-V relocation type.

- Prepare Temporary memory

- Platform memory initialization
- CPU HOB to declare memory address size
- RISC-V SET_JUMP/LONGJUMP to switch stack to permanent
- RISC-V memory map read/write
- RISC-V I/O read/write (memory map)
- EDKII BaseLib for RISC-V
- RISC-V specific PEI service pointer retrieval

mtime CSR
mtimecmp CSR

RISC-V DXE Timer Arch protocol

RISC-V DXE CPU arch protocol

mcause CSR
mip CSR
mie CSR

RISC-V Machine Mode

RISC-V DXE Reset Arch protocol

ACPI

Management Mode

RISC-V DXE Real Time Clock arch protocol

Time CSR

MP service

EDKII TimerLib

RISC-V Hypervisor Mode

RISC-V Supervisor Mode

RISC-V User Mode

SEC to PEI handoff

DXE initial program loader

Reset Vector (VTF)

Security SecMain.efi (SEC)

Pre EFI Initialization (PEI)

Driver Execution Environment (DXE)

Boot Dev Select (BDS)

Transient System Load (TSL)

Runtime (RT)

After Life (AL)

Power on → [....Platform initialization..] → [....OS boot...] → Shutdown

UEFI/PI Execution Phases

Hewlett Packard Enterprise

RISC-V QEMU

- QEMU RISC-V PC/AT board

  Built up RISC-V PC/AT board on QEMU with some PC peripherals.

- QEMU PC/AT memory map devices (CMOS, PM, PCI and other devices)

  Changed these PC peripherals to memory map I/O device because RISC-V uses memory map I/O.

- RISC-V machine mode on RISC-V QEMU port

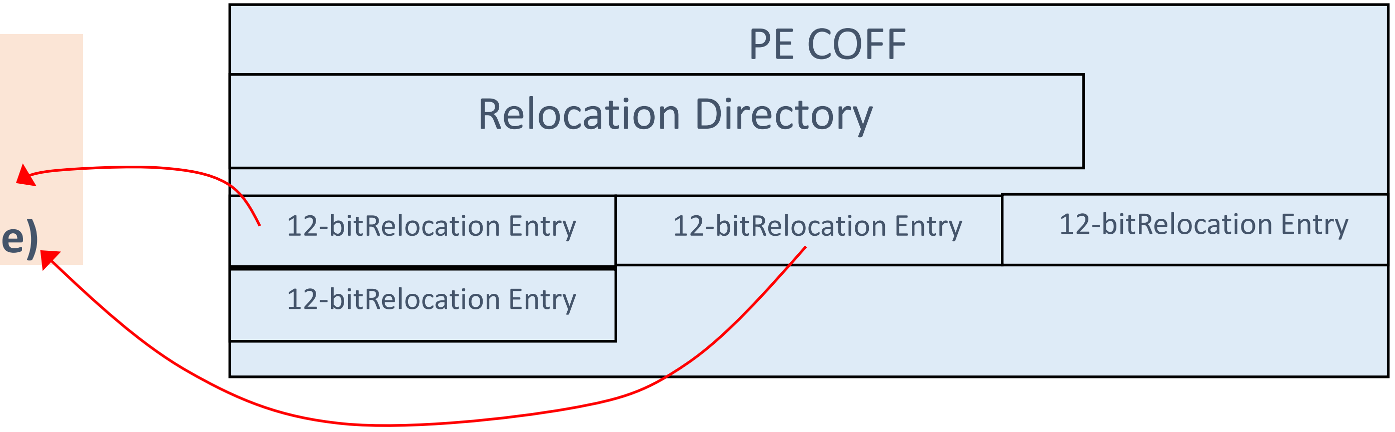
  Implemented RISC-V machine mode on RISC-V QEMU port.

Issues

- How PECOFF support High 20bit/ Low 12bit relocations

```
Temp = &mRootBridgeDevicePathTemplate

lui     a5, %hi (mRootBridgeDevicePathTemplate)
add     a1, a5, %lo (mRootBridgeDevicePathTemplate)
```



- RISC-V relocation in GNU link

  When relative offset < 0x800, it forces to use X0 (hard wired to 0) as base address. This results in inconsistent register usage when load the target address.

- GNU link Optimization (no-relax support)

  When relative offset < 0x800, it deletes AUIPC op-code.

  Call Function ->

```
auipc  t0, 20-bit // U-type integer
jalr t0, 12-bit       // I-type integer
```

We need more in RISC-V spec

- Timer, add periodical timer CSR
- RTC, provide date, time, year and alarm CSR
- PI Management Mode support
- MP support
- ACPI support
- Reset mechanism

UEFI/PI spec change for RISC-V

UEFI spec change for RISC-V

- 2.1.1. UEFI Images
- 2.3. Calling Conventions
- 2.3. RISC-V 32 (64) Platforms
- 17.2 EFI Debug Support Protocol

PI spec change for RISC-V

- Volume 1 : 5.4 RISC-V PEI Services Table Retrieval
- Volume 3 : PI Status code

Next step

- PE COFF image machine type for RISC-V
- PE COFF image relocation type for RISC-V

- EDKII RISC-V code review and commit
- QEMU RISC-V code review and commit
- GNU Link code change review and commit

- EDKII RISC-V OVMF: Add ACPI support
- QEMU : Keyboard/USB/ACPI on QEMU RISC-V PC/AT board
- QEMU : Boot to Linux on RISC-V UEFI port

Hewlett Packard
Enterprise

Thank you

abner.chang@hpe.com
dong.wei@hpe.com