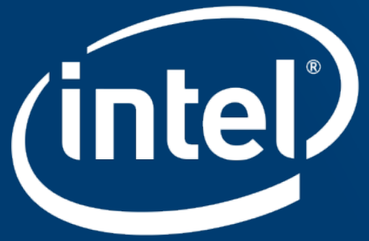


Tiger Lake Platform Controller Hub (PCH) Chipset Initialization (Chipset_Init) Format 2.0 Details

Technical Guidance

October 28, 2019

Legal Disclaimer

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel® products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel® technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel® disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

All information provided here is subject to change without notice. Contact your Intel® representative to obtain the latest Intel® product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

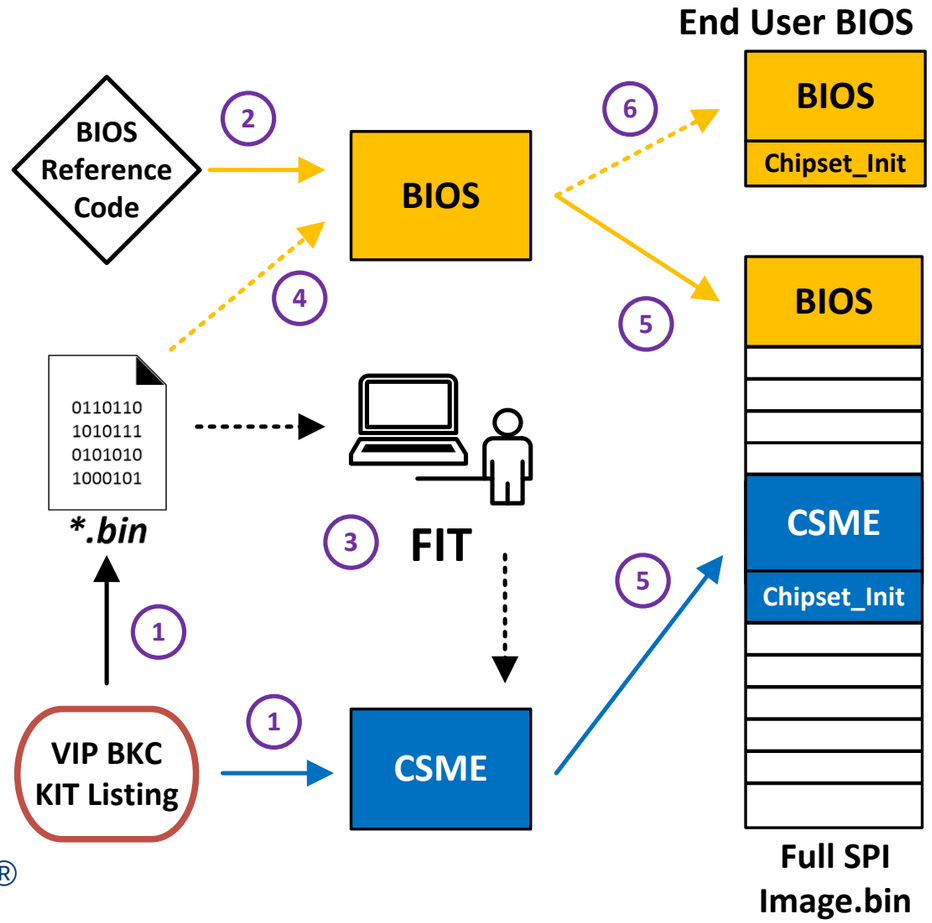
PCH Chipset Initialization (Chipset_Init)

- Chipset_Init is used to set the PCH High Speed Input/Output (HSIO), USB2, DMI/OPI, Flex I/O, and GPIO interface PHY's to get them stable early in the boot flow
  - o These PHY's have HW defaults which may not establish enough margin to get these interfaces stable

- Complex (BIOS → CSME → PMC) ingredient flow used to execute Chipset_Init during any system Cold Boot, Warm Reset, or SUS Power Gating/Sx/S0ix system event
  - o Non executed Chipset_Init events could result in unexpected system/interface behavior or system boot issues

Chipset_Init Format 2.0 Build Flow Steps – (Standard)

1. Download PCH Chipset_Init Package and Intel® CSME from associated Intel® Validation Internet Portal (VIP) platform BKC Kit
   - PCH Chipset_Init Package includes Chipset_Init binary files for N and/or N-1 PCH SKUs
2. Download the Intel platform BIOS Reference Code
3. CSME Chipset_Init Intel® Flash Image Tool (FIT) Binary Stitching
   - Stitch the Chipset_Init binary associated with the PCH SKU you require into CSME using FIT
4. If desired, the BIOS can be compiled with the Chipset_Init binary file
   - For single image N and N-1 PCH SKU support the N and N-1 Chipset_Init binary files can be placed in BIOS
   - The BIOS placed in the full SPI image does not require the Chipset_Init binary for system Chipset_Init execution
5. Build Full SPI Image Binary with BIOS and CSME
6. For end user system Chipset_Init updates provide BIOS with compiled Chipset_Init binary
   - Will result in a one time global reset
   - End user system Chipset_Init updates can not be done through CSME updates using the Intel® ME Firmware Update (FWUpdate) Tool



Build Flow Step Options

- Full SPI Image Chipset_Init Build Flows = Steps 1 + 2 + 3 + 5 or Steps 1 + 2 + 3 + 4 + 5
- End User System Chipset_Init Update Flow = Steps 1 + 2 + 4 + 6

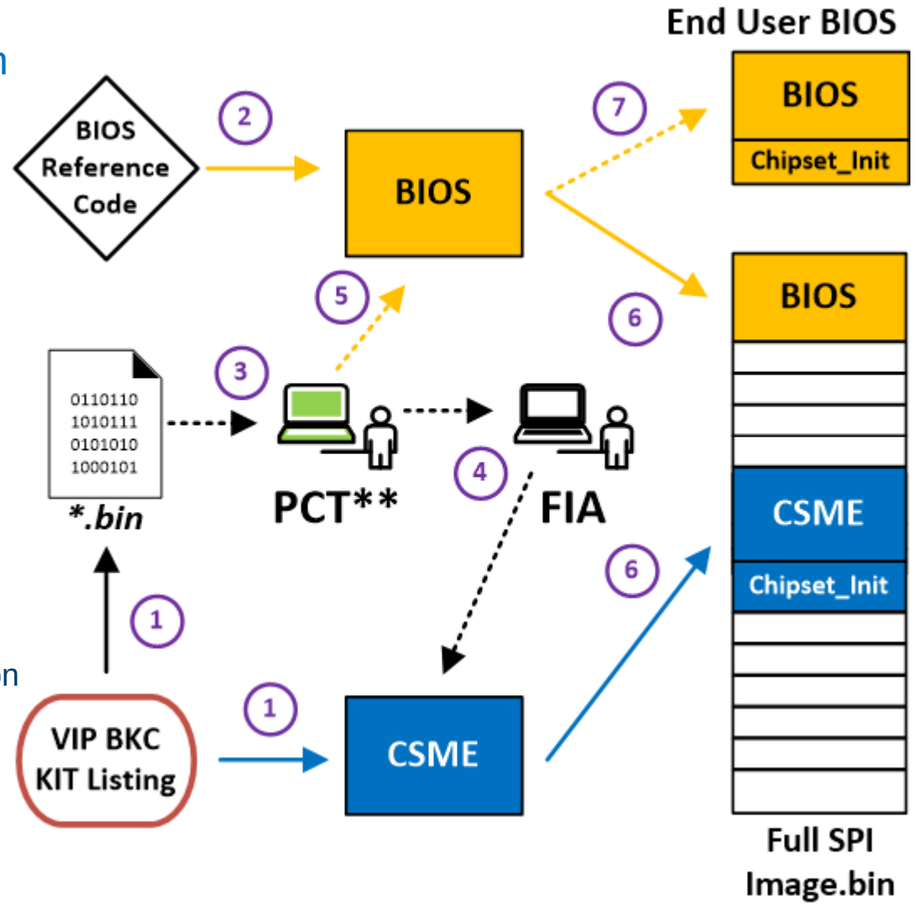
Chipset_Init Format 2.0 Build Flow Steps – (OEM Tuned)

GPIO/SoundWire Only

1. Download PCH Chipset_Init Package and Intel® CSME from associated Intel® Validation Internet Portal (VIP) platform BKC Kit
   - PCH Chipset_Init Package includes Chipset_Init binary files for N and/or N-1 PCH SKUs
2. Download the Intel platform BIOS Reference Code
3. Intel® Platform Configuration Tool (PCT) Chipset_Init Binary OEM Tuning
   - Import Chipset_Init binary associated with the PCH SKU you require, make PCH GPIO/SoundWire buffer changes, and build a OEM tuned Chipset_Init binary
4. CSME Chipset_Init Intel® Flash Image Tool (FIT) Binary Stitching
   - Stitch the OEM tuned Chipset_Init binary into CSME using FIT
5. If desired, the BIOS can be compiled with the OEM tuned Chipset_Init binary file
   - For single image N and N-1 PCH SKU support the N and N-1 Chipset_Init binary files can be placed in BIOS
   - The BIOS placed in the full SPI image does not require the Chipset_Init binary for system Chipset_Init execution
6. Build Full SPI Image Binary with BIOS and CSME
7. For end user system Chipset_Init updates provide BIOS with compiled OEM tuned Chipset_Init binary
   - Will result in a one time global reset
   - End user system Chipset_Init updates can not be done through CSME updates using the Intel® ME Firmware Update (FWUpdate) Tool

Build Flow Step Options

- Full SPI Image Chipset_Init Build Flows = Steps 1 + 2 + 3 + 4 + 6 or Steps 1 + 2 + 3 + 4 + 5 + 6
- End User System Chipset_Init Update Flow = Steps 1 + 2 + 3 + 5 + 7



**** Intel® Platform Configuration Tool (PCT) is available for download off VIP**

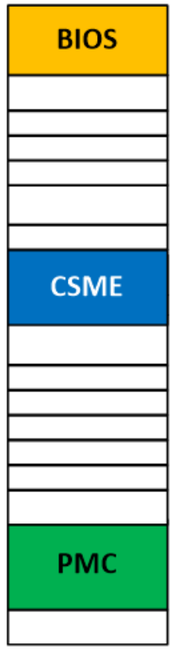
- Run the PCT "Chipset_Init Module" for Step #3 and follow the detailed notes and instructions

Chipset_Init Format 2.0 Execution Flow Details

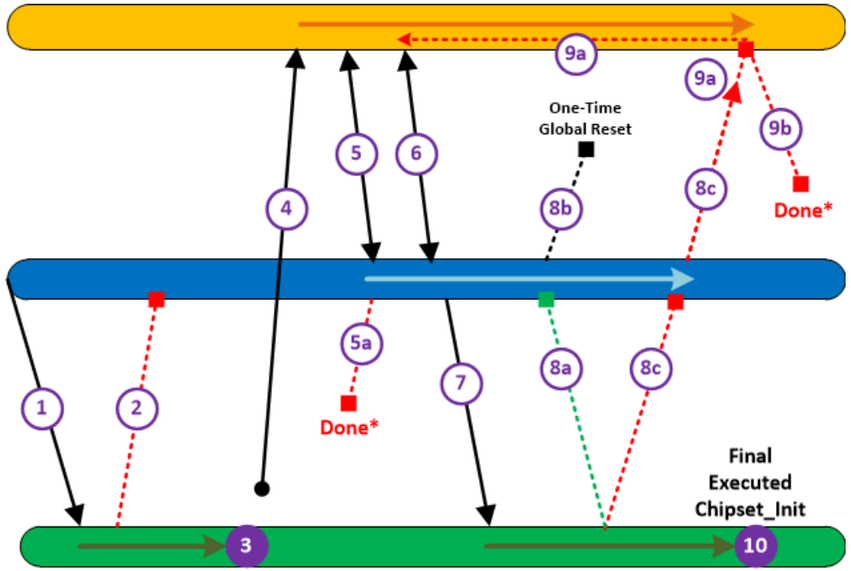
1. PMC pulls Chipset_Init from CSME
2. PMC filter check on the pulled CSME Chipset_Init
   If corrupt PMC reports this to CSME, it does not store the pulled CSME Chipset_Init, and it moves to Step 4
   If not corrupt PMC stores the pulled CSME Chipset_Init and it moves to Step 3
3. PMC executes its stored Chipset_Init
4. BIOS loads and starts executing
5. BIOS/CSME Chipset_Init information retrieval
   First it determines if BIOS and CSME have a stored Chipset_Init
   - If BIOS and CSME do not have a stored Chipset_Init then the Chipset_Init execution flow is complete

   Second it determines if a Chipset_Init PGR section needs to be built, updated, or requires no change
6. BIOS Chipset_Init push or pull
   If PGR section needs to be built or updated:
   - BIOS with Chipset_Init → BIOS builds/updates PGR, PGR is appended to its stored BIOS Chipset_Init, and then the Chipset_Init is pushed to CSME
   - BIOS without Chipset_Init → BIOS pulls the CSME Chipset_Init, it builds/updates PGR, PGR is appended to the pulled Chipset_Init, and then the Chipset_Init is pushed back to CSME

   If PGR section does not need to be built or updated:
   - BIOS and CSME have different Chipset_Init versions = BIOS Chipset_Init pushed to CSME
   - All other BIOS/CSME Chipset_Init cases = No BIOS action taken
7. CSME pushes Chipset_Init to PMC
   Pushes Chipset_Init from Step 6 or its stored Chipset_Init
8. PMC filter check on the Chipset_Init pushed by CSME
   8a. If not corrupt PMC notifies CSME and CSME/PMC both store the Chipset_Init and move to Step 10
   8b. If PGR Section was updated or BIOS has a different Chipset_Init version than CSME then a one time global reset will occur (Result of Steps 5 and 6)
   Note: Initially built PGR section does not result in a one-time global reset
   8c. If corrupt PMC notifies CSME, CSME notifies BIOS, CSME/PMC both do not store the Chipset_Init, and move to Step 9
9. Chipset_Init flow retry
   9a. Move back to Step 6 and retry the execution flow three more times. On the third retry BIOS pulls and stores CSME Chipset_Init before going back to Step 6
   9b. After a third failed retry then Done* - system proceeds without Chipset_Init execution
10. PMC executes its stored Chipset_Init
    Chipset_Init stored during Step 8a



Full SPI Image.bin



Notes:
- Cold Boot, Warm Boot, and Reset Execution Flow = Steps 1 to 10
- SUS Power Gating/Sx/S0ix Execution Exit Flow = Step 10
- Done*: Non executed Chipset_Init cases could result in unexpected system/interface behavior or system boot issues

Tiger Lake PCH Chipset_Init Format 2.0 Quick Reference

Chipset_Init Deliverables

- BIOS Reference Code Releases → Do not incorporate any Chipset_Init format 2.0 binary file(s)
- CSME BKC Releases → Do not incorporate any Chipset_Init format 2.0 binary file
- BKC PCH Chipset_Init Packages → Include Chipset_Init format 2.0 binary file(s) for multiple PCH SKUs where applicable (N and N - 1)

CSME Details

- Stitch the Chipset_Init format 2.0 binary, from the latest BKC PCH Chipset_Init package, into CSME using FIT
- Details covered in pages 4 and 5

BIOS Details

- Option #1: Don't place any Chipset_Init Format 2.0 binary file(s) in BIOS and leave the BIOS Reference Code Chipset_Init format 2.0 Binary Pointer and Length Policy settings as NULL (default)
- Option #2: For single PCH SKU support and end user system Chipset_Init updates place the Chipset_Init format 2.0 binary for the latest PCH SKU (N) in BIOS making sure the BIOS Chipset_Init Format 2 Binary Pointer and Length Policy settings are set for the binary file
- Option #3: For multiple PCH SKU support place N and N-1 Chipset_Init Format 2.0 binary files in BIOS making sure the BIOS Chipset_Init Format 2 Binary Pointer and Length Policy settings are set for these binary files
- See BIOS Chipset_Init format 2.0 Binary Pointer and Length Polices

BIOS Chipset_Init format 2.0 Binary Pointer and Length Polices

```
/**
  The PCH_HSIO_CONFIG block provides HSIO message related settings.
**/
typedef struct {
  CONFIG_BLOCK_HEADER   Header;       ///< Config Block Header
  /**
  Policy used to point to the Base (+ OEM) ChipsetInit binary format v2 used to sync between BIOS and
CSME
  **/
  UINT32    ChipsetInitBinPtr;
  /**
  Policy used to indicate the size of the Base (+ OEM) ChipsetInit binary format v2 used to sync between
BIOS and CSME
  **/
  UINT32    ChipsetInitBinLen;
} PCH_HSIO_CONFIG;
```

Chipset_Init Format 2.0 Verification Guidance

If customers need to verify which Chipset_Init binary they are working with, which Chipset_Init binary is stored in CSME, and which Chipset_Init binary executed on a booted platform they can do the following:

1. Determine the version and PCH Sku support of a Chipset_Init format 2.0 binary
2. Determine which Chipset_Init format 2.0 binary is stored in CSME
3. Determine which Chipset_Init format 2.0 binary executed on a booted platform

How to Determine the Version and PCH SKU Support of a Chipset_Init Format 2.0 Binary

Method #1: Use Intel® Flash Image Tool (FIT)

- Open Chipset_Init binary file in FIT
  - FIT.exe → “Flash Layout” tab → Load Chipset_Init binary file in “Chipset Initialization Binary” cell
- Read Chipset_Init binary information results in FIT
  - “Flash Layout” tab → “Chipset Initialization Version” parameter = xx.yy.zz.x
    - yy is the “PCH Stepping Identifier”
      - See BKC PCH Chipset_Init Release Notes for the yy mapping and details
    - zz identifies the Chipset_Init Version in Hex

Method #2: Use a Hex Editor Based Application

- Open Chipset_Init binary file in a Hex Editor Based Application
  - Byte 6 = Chipset_Init Version in Hex
  - Byte 8 = “PCH Stepping Identifier”
    - See BKC PCH Chipset_Init Release Notes for the mapping and details

How to Determine Which Chipset_Init Format 2.0 Binary is Stored in CSME Binary

Use Intel® Flash Image Tool (FIT)

- Method #1: Open CSME binary in FIT
  - FIT.exe → “File” tab → Select “Open” → Browse and Select the CSME binary file to open

  or
  - FIT.exe → “Flash Layout” tab → Load CSME Binary in “Intel® ME Binary File” cell

- Method #2: Open a non platform booted Full SPI Image binary in FIT
  - FIT.exe → “File” tab → Select “Open” → Browse and Select the Full SPI Image binary to open

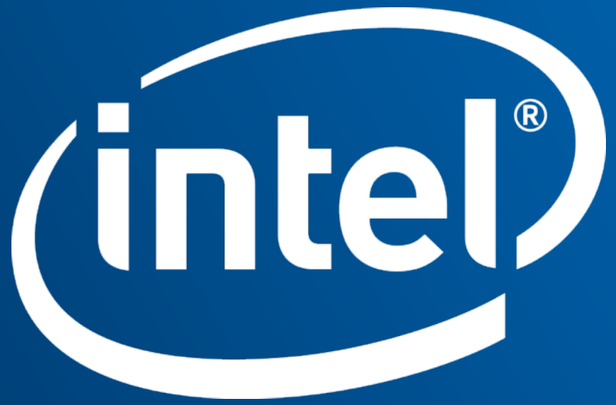
Results: For Method #1 or #2 read the Chipset_Init binary information results in FIT

- “Flash Layout” tab → “Chipset Initialization Version” parameter = xx.yy.zz.x
  - yy is the “PCH Stepping Identifier”
    - See BKC PCH Chipset_Init Release Notes for the yy mapping and details
  - zz identifies the Chipset_Init Version in Hex

How to Determine Which Chipset_Init Format 2.0 Binary Executed on a Booted Platform

If Platform Booted Determine Which Chipset_Init Format 2.0 was Executed by PMC and Stored in CSME

- Method #1: Open a platform booted Full SPI Image binary in Intel® Flash Image Tool (FIT)
  - FIT.exe → “File” tab → Select “Open” → Browse and Select the Full SPI Image binary to open
  - “Flash Layout” tab → “Chipset Initialization Version” parameter = xx.yy.zz.x
    - yy is the “PCH Stepping Identifier.” See BKC PCH Chipset_Init Release Notes for the yy mapping and details
    - zz identifies the Chipset_Init Version in Hex
- Method #2: Use Intel® Clock Commander (CCT) Tool found in BKC Intel® CSME kit
  - Use the following CCT command: cct gm > csi_dump.txt
    - In order for this command to run you must compile or configure the BIOS to disable EOP (End-of-POST) so that no EOP messages are sent to Intel® CSME
  - Open csi_dump.txt file
    - Byte 6 = Chipset_Init Version in Hex
    - Byte 8 = “PCH Stepping Identifier.” See BKC PCH Chipset_Init Release Notes for the yy mapping and details


intel