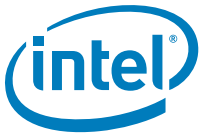


Gemini Lake Platform - Intel® Trusted Execution Engine (Intel® TXE) Firmware Bring-Up Guide

User Guide

Revision 1.0

July 2017

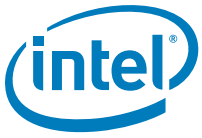
Intel Confidential

Contents

Figures

Revision History

Revision Number	Description	Revision Date
1.0	• Aligning revision number	July 2017
0.8	• Minor typos	May 2017
0.6	• Chapter 2: Updated switch mapping to initiate either a SPI or eMMC flashing procedure • Section 2.6.1.4 was removed. Cable detection is not supported as detection method for DnX.	March 2017
0.5	• Initial Release	January 2017



1 Introduction

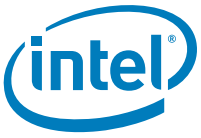
This document covers the Gemini Lake Platform Intel®Trusted Execution Engine (Intel® TXE) Firmware bring-up procedure.

Please notes that this guide only contains the SMIP configuration procedure for the critical boot settings.

1.1 Terminology

Title	Definition
GLK	Gemini Lake
Intel® FIT	Intel® Flash Image Tool
MEU	Manifest Extension Utility
DnX	Download and Execute
SMIP	Signed Master Image Profile
ROT	Root of Trust
ISS	Intel® Integrated Sensor Solution
GPIO	General Purpose Input/Output
Intel®PTT	Intel® Platform Trusted Technology
IFWI	Integrated Firmware Image. The new Firmware image layout used in GLK platforms
SPD	Storage Proxy Driver
VR	Voltage Regulator

§ §



2 Image Creation/ Flashing Procedure

2.1 Prerequisites

2.1.1 IFWI Image Components, Tools and Drivers

In order to build the image, the following image components are required:

Requirements	Require tool/component	Description
Tools	FIT	Flash image tool that is used to create the image
	MEU	Manifest Extension Utility that is used to create manifests
	OpenSSL	Freeware, used to sign the manifests
Image Components (critical for platform boot)	IAFW (BIOS) SMIP Binary	Available in the BKC
	PMC binary	Available in the PMC FW Kit
	uCode patch 1	Available in the BKC
	uCode patch 2	Available in the BKC
	TXE FW binary	Available in TXE FW kit
	ROT Key manifest	Available in TXE FW kit
	OEM Key Manifest	Available in TXE FW kit or created using MEU
	Full IAFW(BIOS) binary	Generated by OEM/Available in the BKC
Additional Image Components	iUnit binary	Available in the BKC
	ISS image	Available in ISS Kit
	ISS PDT File	Available in ISS Kit
Signing Keys	Private key for SMIP signing	OEM generated
	Private key for DnX signing	
Drivers	TXEI, SPD	Available in TXE FW kit

2.1.2 MEU Configurations

2.1.2.1 Configuring MEU Signing Settings

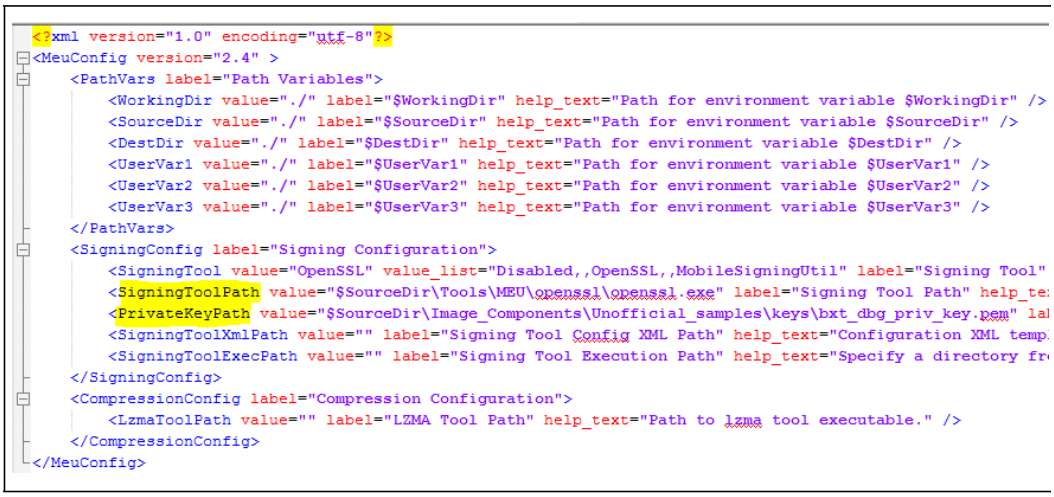
FIT will use MEU in order to create the SMIP and DnX manifests (as part of the image creation process).Therefore, the signing settings will have to be configured in MEU prior to building the image.

Generate the MEU configuration file and run: MEU -gen meu_config

Edit the MEU configuration xml (meu_config.xml) which was created in the previous step, and set the following:

1. “SigningToolPath” - path to the signing tool (the OpenSSL tool)
2. “PrivatekeyPath” - path to the private key that used to sign the SMIP/DnX.

Figure 2-1. MEU Configurations Example



```
<?xml version="1.0" encoding="utf-8"?>
<MeuConfig version="2.4" >
    <PathVars label="Path Variables">
        <WorkingDir value="./" label="$WorkingDir" help_text="Path for environment variable $WorkingDir" />
        <SourceDir value="./" label="$SourceDir" help_text="Path for environment variable $SourceDir" />
        <DestDir value="./" label="$DestDir" help_text="Path for environment variable $DestDir" />
        <UserVar1 value="./" label="$UserVar1" help_text="Path for environment variable $UserVar1" />
        <UserVar2 value="./" label="$UserVar2" help_text="Path for environment variable $UserVar2" />
        <UserVar3 value="./" label="$UserVar3" help_text="Path for environment variable $UserVar3" />
    </PathVars>
    <SigningConfig label="Signing Configuration">
        <SigningTool value="OpenSSL" value_list="Disabled,,OpenSSL,,MobileSigningUtil" label="Signing Tool"
        <SigningToolPath value="$SourceDir\Tools\MEU\openssl\openssl.exe" label="Signing Tool Path" help_te
        <PrivateKeyPath value="$SourceDir\Image_Components\Unofficial_samples\keys\bxt_dbg_priv_key.pem" la
        <SigningToolXmlPath value="" label="Signing Tool Config XML Path" help_text="Configuration XML temp
        <SigningToolExecPath value="" label="Signing Tool Execution Path" help_text="Specify a directory fr
    </SigningConfig>
    <CompressionConfig label="Compression Configuration">
        <LzmaToolPath value="" label="LZMA Tool Path" help_text="Path to lzma tool executable." />
    </CompressionConfig>
</MeuConfig>
```

2.2 Flashing ROM Bypass

For GLK platform the ROM bypass needs to be flashed prior to the bring-up process, Please follow “Appendix A: ROM Bypass” to flash the ROM bypass image, before the image creation procedure.

2.3 Start FIT

Start the FIT tool by navigating to: \\Tools\FIT folder and running fit.exe

2.4 Creating the Binary Image

2.4.1 Configuring and Building the Image

Please follow the procedure below in order to configure and build the IFWI image.

2.4.1.1 Flash Layout Configurations

In the flash layout section in FIT, the following regions will be defined: TXE, BIOS, SMIP, iUnit, PMC, uCode.

Please note that the first region that needs to be configured is the TXE region since loading it will reset the existing image configurations.

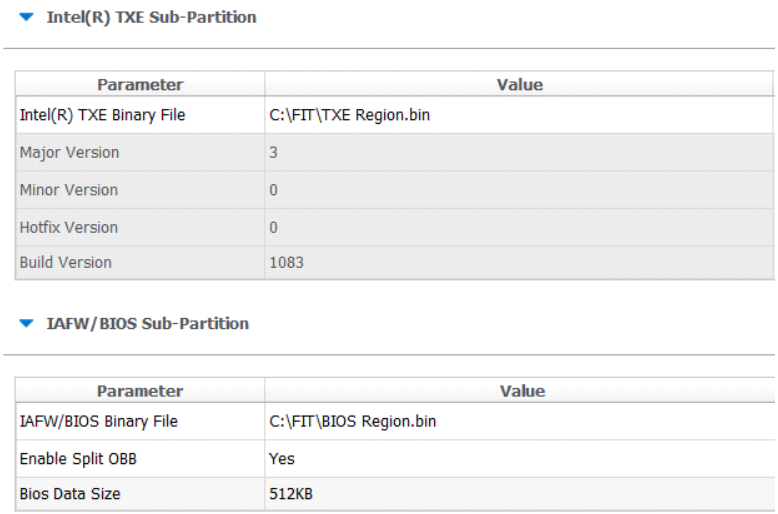
1. Configure Intel® TXE region:

- On the left panel select the Flash layout tab
- In the “Intel® TXE Sub-Partition” set the following:
  - “Intel® TXE Binary file”

2. Configure the BIOS region:

- in Flash Layout tab, IA/BIOS Sub-Partition, configure:
  - "BIOS Binary File"
  - "Enable Split OBB" - enable this to extend the OBB into the LBP2 in order to accommodate for a larger OBB.
  - "BIOS Data Size" - configure the BIOS data size, this can be configured to '0', '128KB', '256KB', '384KB', 512KB', this configuration will affect the maximum size of the OBB.

Figure 2-2. Intel® TXE and BIOS Region Configurations Example



3. Configuring the SMIP region:

- In the **flash layout** tab, **SMIP Sub-partition**, configure:
  - IAFW SMIP binary file (the BIOS SMIP).

Figure 2-3. SMIP Configurations Example

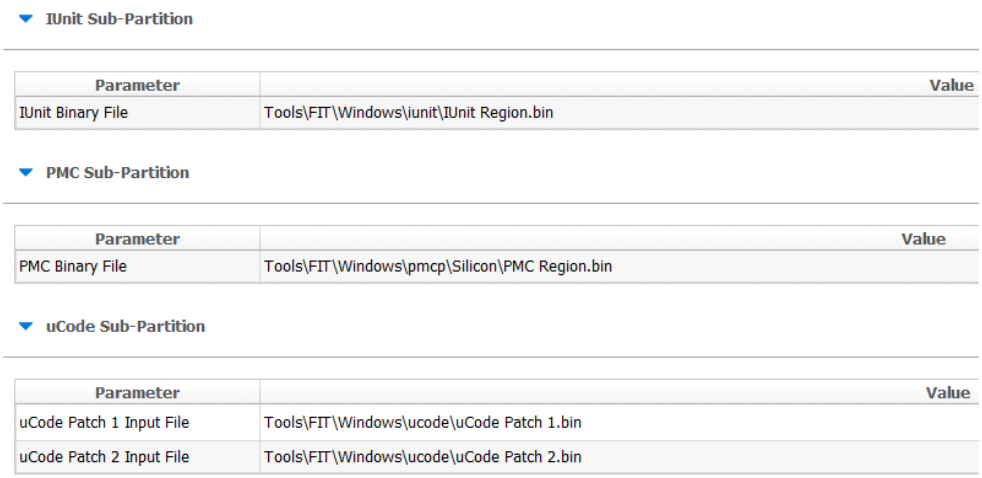
4. Configuring the PMC and uCode regions:

- in the **Flash layout** tab, **PMC Sub-Partition**, select:
  - PMC Binary file.
- In the **Flash layout** tab, **uCode Sub-Partition**, select:
  - uCode patch 1 Input file.
  - uCode patch 2 Input file.

5. Configuring the iUnit (optional)

- In the **Flash layout** tab, **iUnit Sub-Partition**, select:
  - iUnit Binary File.

Figure 2-4. iUnit, PMC, uCode Configuration Example



2.4.1.2 Flash Settings Configurations

In this section, the bootable device setting will be configured.

2.4.1.2.1 SPI Based Platform Configurations

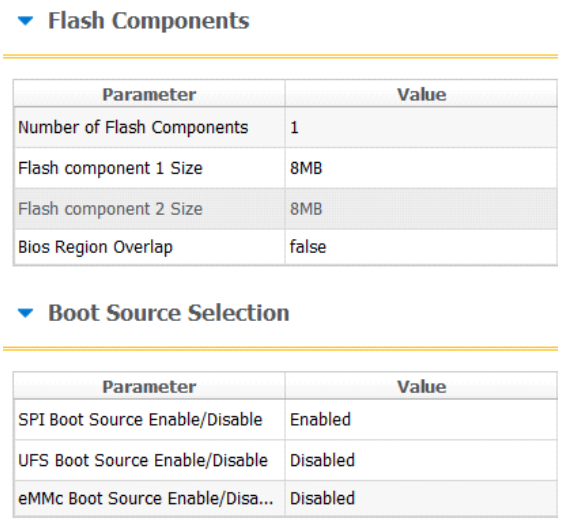
Under "**Flash Setting**" tab, "**flash component**" section set the following:

- "Number of Flash Components": should be configured to "1".
- Flash Component 1 size: should be configured to "8MB".
- BIOS region overlap: should be configured to "False".

Under "**Flash Setting**" tab, "**Boot Source Selection**" section, set the following:

- "SPI Boot Source Enable/Disable": should be set to "Enabled".
- "UFS Boot Source Enable/Disable": should be set to "Disabled".
- "eMMC Boot Source Enable/Disable": should be set to "Disabled".

Figure 2-5. SPI Flash Setting Configuration Example

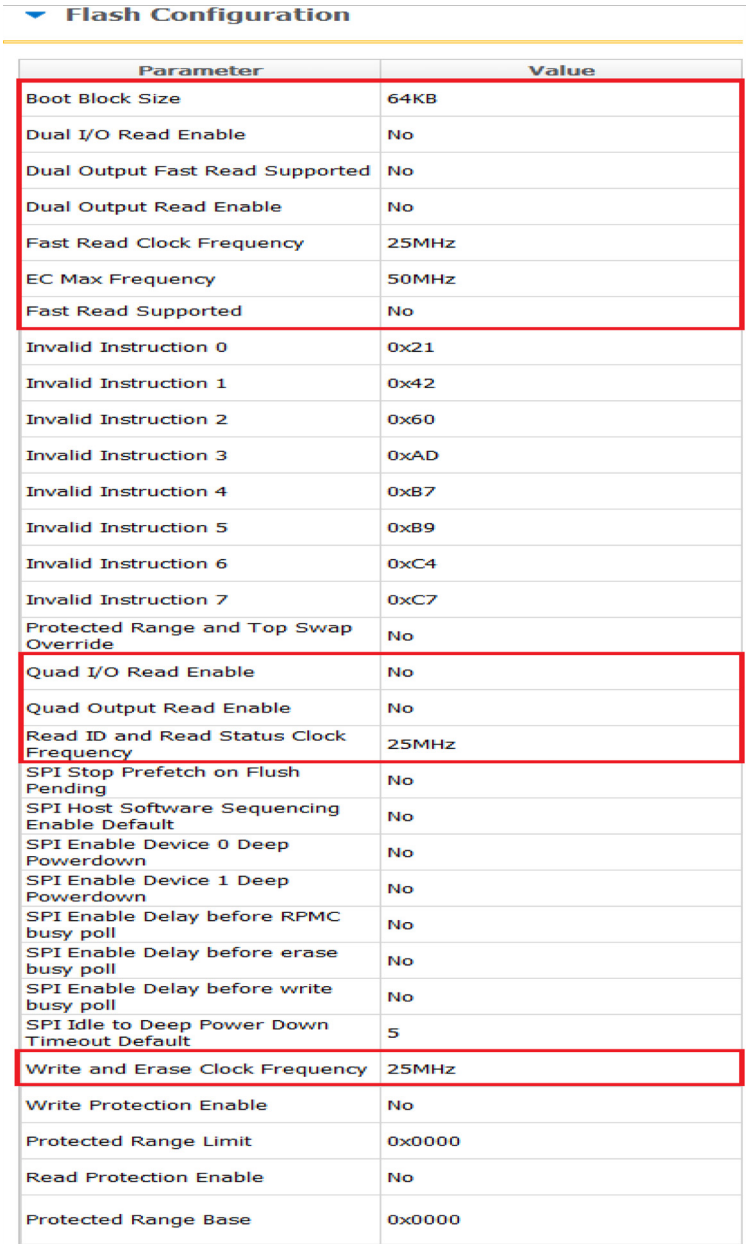


Under "**Flash Setting**" tab, "**Flash Configuration**" section set the following according to the SPI flash part support:

- Boot Block Size - Enable per Top Swap usage on platform.
- Dual I/O Read Enabled
- Dual Output Fast Read Support
- Dual Output Read Enabled
- Fast Read Clock Frequency
- Fast Read Supported
- Quad I/O Read Enabled - please refer to the note below.
- Quad Output Read Enabled - please refer to the note below.
- Read ID and Read Status Clock Frequency
- Write and Erase Clock Frequency

Note: When setting "Quad I/O Read Enabled" or "Quad Output Read Enabled" to "Yes", the "Quad Enabled" bit need to be set in the SPI, without it the platform will **NOT BOOT**, please refer to "Appendix C: Enabling Quad mode on SPI Part" for the procedure.

Figure 2-6. GLK RVP Flash Configuration Example



Flash Configuration

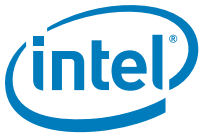
Parameter	Value
Boot Block Size	64KB
Dual I/O Read Enable	No
Dual Output Fast Read Supported	No
Dual Output Read Enable	No
Fast Read Clock Frequency	25MHz
EC Max Frequency	50MHz
Fast Read Supported	No
Invalid Instruction 0	0x21
Invalid Instruction 1	0x42
Invalid Instruction 2	0x60
Invalid Instruction 3	0xAD
Invalid Instruction 4	0xB7
Invalid Instruction 5	0xB9
Invalid Instruction 6	0xC4
Invalid Instruction 7	0xC7
Protected Range and Top Swap Override	No
Quad I/O Read Enable	No
Quad Output Read Enable	No
Read ID and Read Status Clock Frequency	25MHz
SPI Stop Prefetch on Flush Pending	No
SPI Host Software Sequencing Enable Default	No
SPI Enable Device 0 Deep Powerdown	No
SPI Enable Device 1 Deep Powerdown	No
SPI Enable Delay before RPMC busy poll	No
SPI Enable Delay before erase busy poll	No
SPI Enable Delay before write busy poll	No
SPI Idle to Deep Power Down Timeout Default	5
Write and Erase Clock Frequency	25MHz
Write Protection Enable	No
Protected Range Limit	0x0000
Read Protection Enable	No
Protected Range Base	0x0000

2.4.1.3 Platform SMIP Configurations

2.4.1.3.1 Voltage Regulator Depended SMIP Configurations

In the "**CPU Straps**" tab, under "**PUNIT**" configure the following according to the board VR:

- Rail 0 Alert polling enable:
  - "Enabled" = SVID OR Whiskey Cove PMIC VR Type



  - "Disabled" = I2C VR Type
- Rail 0 SVID ID:
  - 0x0 = SVID OR I2C VR Type
  - 0x5 = Whiskey Cove PMIC VR Type
- Rail 1 Alert polling enable:
  - "Enabled" = SVID OR Whiskey Cove PMIC VR Type
  - "Disabled" = I2C VR Type
- Rail 1 SVID ID:
  - 0x0 = I2C VR Type
  - 0x1 = Whiskey Cove PMIC VR Type
  - 0x2 = SVID VR Type
- Rail 2 Alert polling enable:
  - "Enabled" = Whiskey Cove PMIC VR Type
  - "Disabled" = SVID OR I2C VR Type
- Rail 2 SVID ID:
  - 0x0 = SVID OR I2C VR Type
  - 0x2 = Whiskey Cove PMIC VR Type
- Rail 3 Alert polling enable:
  - "Enabled" = Whiskey Cove PMIC VR Type
  - "Disabled" = SVID OR I2C VR Type
- Rail 3 SVID ID:
  - 0x0 = I2C VR Type
  - 0x1 = SVID VR Type
  - 0x6 = Whiskey Cove PMIC VR Type

Note: Please refer to the example below for the GLK Intel® RVP configuration example.

Figure 2-7. Configuration Example according to GLK Intel® RVP VR

▼ PUNIT

Parameter	Value
Thermal Throttle Unlock	Locked
Extended Reliability Enable	Disabled
Soft SVID Enable	Enabled
Rail 0 Alert Polling Enable	Enabled
Rail 0 SVID ID	0x00000000
Rail 1 Alert Polling Enable	Enabled
Rail 1 SVID ID	0x00000002

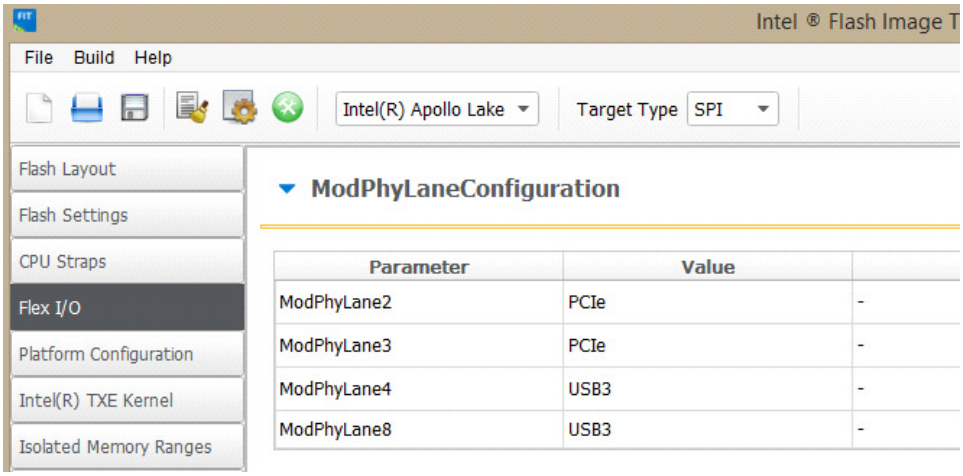
2.4.1.3.2 Mod-Phy Lane Depended SMIP Configurations

The following configurations need to be set according to the platform SMIP Mod-Phy lane configurations. Platform SMIP are fully configurable via FIT UI (XML or GUI).

Refer to the relevant FIT tab/section for configuring SMIP. SPI and SMIP programming guide (part of TXE kit) has further details of each SMIP configuration.

Configure the Platform SMIP via FIT of the platform Mod-Phy configurations according to the screenshot below.

Figure 2-8. Mod-Phy Lane Ownership FIT Configuration

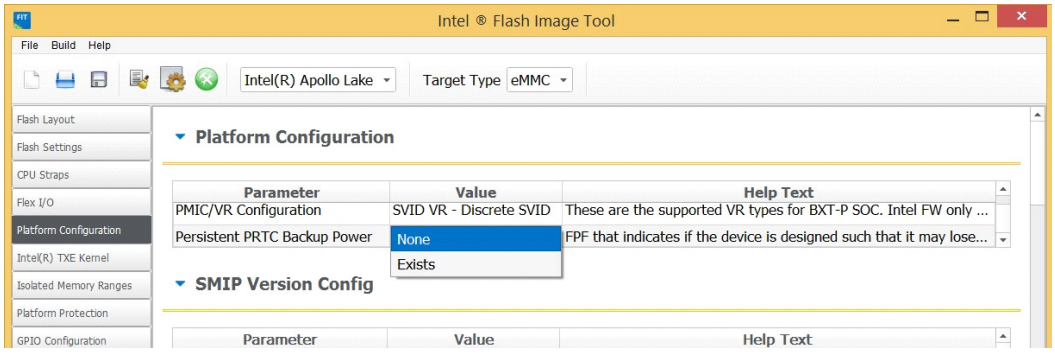


2.5 Coinless Platform Configuration

If your product design does not have Persistent RTC power (i.e. no coin battery), you may set the below configuration of "Persistent PRTC Backup Power" to "None" (Default is "Exists" = Coin Battery exists). Note that this configuration will be **permanently set** in FPF fuses and cannot be reversed. Setting this option, your system will lose some TXE features that depend on PRTC; like Anti-Replay Protection, PTT-Anti Hammering (PTT-AH), and DAL persistent time.

Important: With "Persistent PRTC Backup Power = Exists", RTC power must not lose power 10 times in the lifetime of the product. PTT-AH feature uses RTC to detect physical attacks. PTT-AH counts RTC power loss in FPF to detect this. Once PTT-AH FPFs reach count 10, user will be locked out for 120 minutes when it boots. Every subsequent RTC power loss, user will also be locked out for 120 minutes. If you think your system, according to its design, will lose RTC power more than 10 times in its lifetime, then select "Persistent PRTC Backup Power = None" to avoid this lock out.

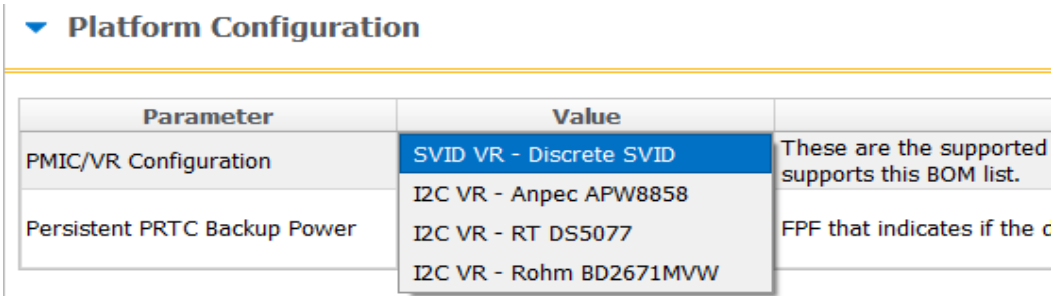
Figure 2-9. Coinless Platform Configuration



2.6 Voltage Regulator Configurations

When configuring customer platform with PMIC/VR setup (discrete SVIT/Rohm/RT/ TI), please use the below dropdown to make the selection:

Figure 2-10. Voltage Regulator Configurations



2.6.1 PCIe SMIP Configurations

In the "**Flex I/O**" tab under "**PCIe (x2)**" and **PCIE (x4)** sections set the "Root Port Configuration (RPCFG)" according to the platform schematics.

Figure 2-11. GLK Intel® RVP Root Port Configuration Example

▼ PCIe (x2)

Parameter	Value
Root Port Configuration (RPCFG)	2x1
Lane Reversal (LNREV)	No
PCIe Port 0 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 1 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 2 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 3 Non-Common Clock With SSC Mode Enable	Disabled

▼ PCIe (x4)

Parameter	Value
Root Port Configuration (RPCFG)	1x2, 2x1
Lane Reversal (LNREV)	No
PCIe Port 0 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 1 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 2 Non-Common Clock With SSC Mode Enable	Disabled
PCIe Port 3 Non-Common Clock With SSC Mode Enable	Disabled

2.6.2 Platform Protection Configurations

2.6.2.1 Platform Integrity and Boot Guard Configurations

In this section the configurations that are related to the boot guard authentication flow will be set, these settings need to be aligned with the OEM Key manifest settings.

There are 3 available Boot Guard profiles:

- Boot Guard Profile 0 - Legacy: in this profile Boot Guard boot block verifications and measurement protection is off.
- Boot Guard Profile 1 - V: Strict Verification Enforcement. Prevents unverified bios components from running.
- Boot Guard Profile 1 - VM: Strict Verification and Measurement enforcement. Prevents unverified Bios components from running.

When using the other Boot Guard profiles (Legacy/V/VM), and for complete information about signing and manifesting, please note that even when using "Boot Guard Profile 0 - Legacy" each component still needs to be manifested and signed.

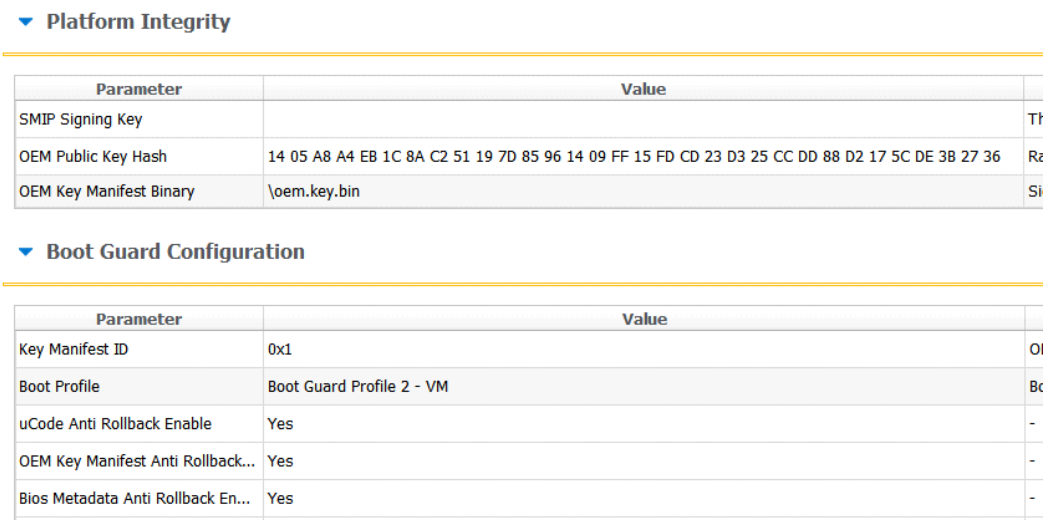
Note: When building an image for Intel® RVP, the required files for each of the boot guard profiles can be found in the TXE FW kit.

Once the necessary files were created according to the Boot Guard profile, in the "**platform protection**" tab, under "**Platform Integrity**" set:

- "SMIP Signing Key" - this will be the private key that will be used to sign the SMIP manifest, please note that as part of the OEM key manifest procedure, the SMIP public key (which is paired with this private key) will need to be configured for the SMIP manifest authentication.
- "OEM Public key Hash" - the hash of the public key that is used to authenticate the OEM key manifest.
- "OEM Key Manifest Binary" - the OEM Key manifest binary that was created using the MEU tool.
- "Key Manifest ID" - needs to be set according to the KMID in the OEM Key Manifest.
- "Boot Profile" - set to according to the boot guard profile.

When choosing not to sign the image, the above files does not need to be set, and 'Boot Profile' should be set to 'Boot Guard profile 0 - legacy'.

Figure 2-12. Platform Integrity and Boot Guard Configurations Example

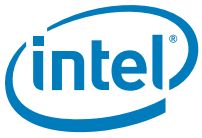


2.6.2.2 Intel® PTT and TPM Configurations

This settings needs to be set according to the TPM devices that is used on the platform.

When using fTPM the following configurations needs to be set:

- In the **platform protection** tab, under **Intel® PTT configurations**, set:
  - Intel® PTT initial power-up state to "Enable".
  - Intel® PTT Supported to "Yes".
  - Intel® PTT Supported [FPF] to "Yes".
- In the **platform protection** tab, under **TPM Over SPI Bus Configurations**, set:

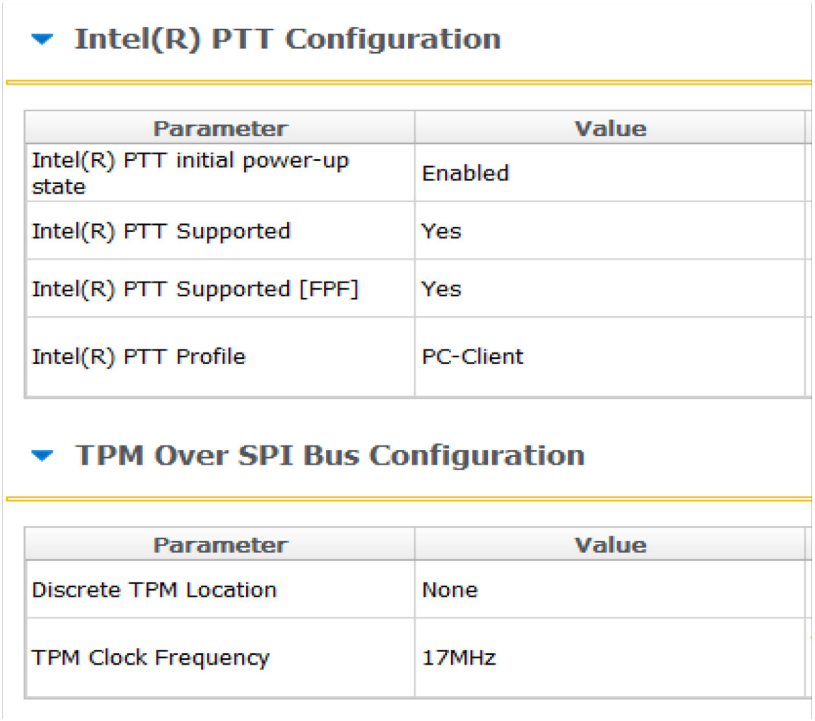


— Discrete TPM Location to “None”.

When using a dTPM the following configurations needs to be set:

- In the **platform protection** tab, under **TPM Over SPI Bus Configurations**, set:
  — Discrete TPM location according to board configurations to SPI/LPC.

Figure 2-13. TPM Configuration Example



Intel(R) PTT Configuration

Parameter	Value
Intel(R) PTT initial power-up state	Enabled
Intel(R) PTT Supported	Yes
Intel(R) PTT Supported [FPF]	Yes
Intel(R) PTT Profile	PC-Client

TPM Over SPI Bus Configuration

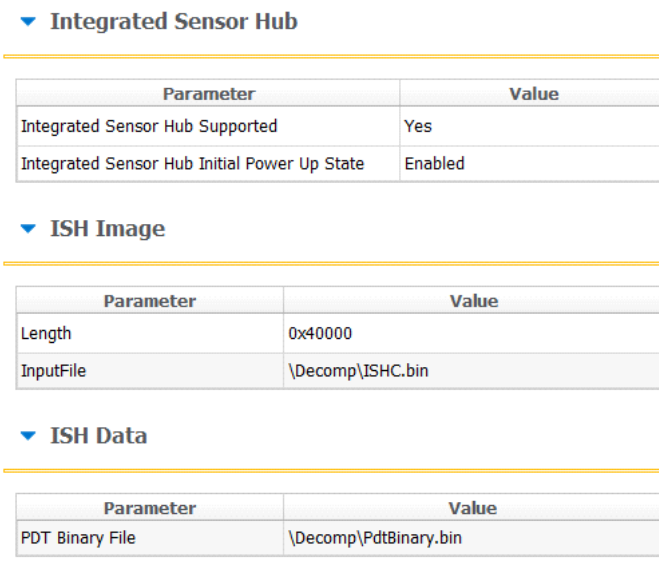
Parameter	Value
Discrete TPM Location	None
TPM Clock Frequency	17MHz

2.6.3 Intel® Integrated Sensor Solution Configurations

To enable Intel® Integrated Sensor Solution, the following configurations needs to be set in the “**Integrated Sensor Hub**” tab:

- Under “integrated Sensor Hub” section, set “Integrated Sensor Hub Supported” as “Yes”.
- Under “ISH Image” section, select the ISH binary location in “InputFile” field.
- Under “ISH Data” section, select the PDT file location in “PDT Binary File” field.

Figure 2-14. ISS Configurations Example



▼ Integrated Sensor Hub

Parameter	Value
Integrated Sensor Hub Supported	Yes
Integrated Sensor Hub Initial Power Up State	Enabled

▼ ISH Image

Parameter	Value
Length	0x40000
InputFile	\Decomp\ISHC.bin

▼ ISH Data

Parameter	Value
PDT Binary File	\Decomp\PdtBinary.bin

2.6.4 Configuring Intel® FIT build Settings

In the main menu select Build→ Build settings Edit your configuration as shown below.

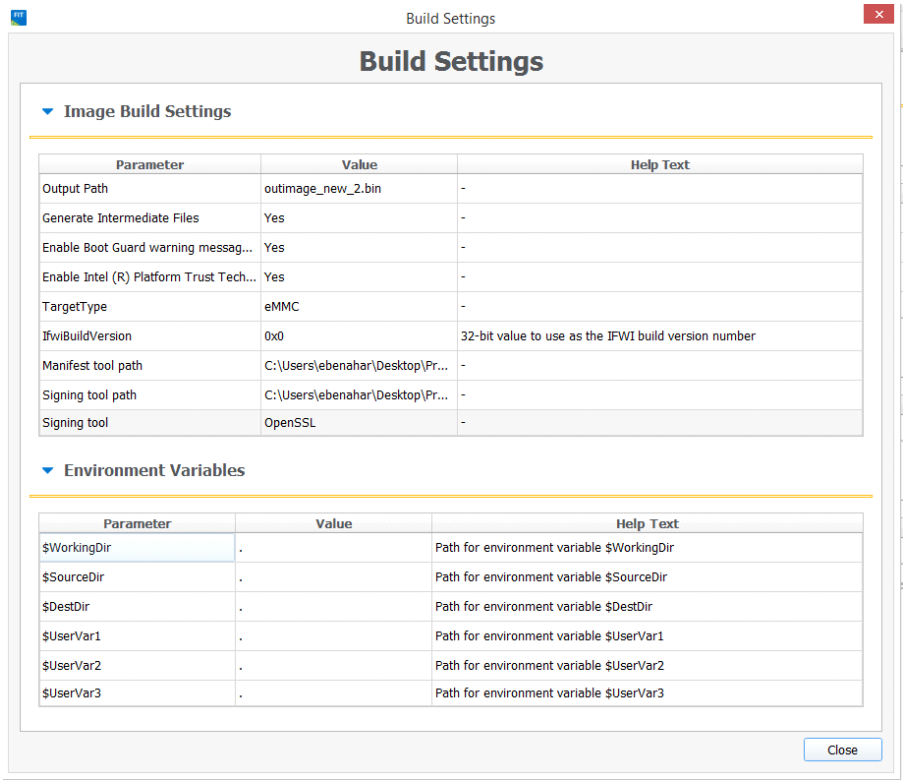
Image build setting:

1. **Output path:** the location and name of the image that will be created.
2. **Target Type:** the bootable device type SPI/eMMC/UFS.
3. **Manifest tool path:** the path to the MEU tool.
4. **Signing tool path:** the path to the signing tool.
5. **Signing tool:** the signing tool that is going to be used.

Environment Variables: (optional)

1. **$SourceDir:** The location where FIT will look for binary images during the image creation process.
2. **$DestDir:** The location where FIT will save the binary image.

Figure 2-15. Build Configuration Settings Example



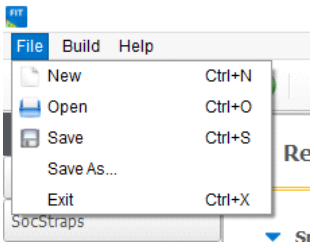
2.6.5 Save/Load Intel® FIT XML Configuration

Once the IFWI setting have been configured, it's **highly** recommended to save these setting into a FIT xml, these settings can be loaded to simplify future image creations.

To save/load FIT configurations xml, from the FIT menu select: File →

"open"/"save"/"save as".

Figure 2-16. Saveing/Loading Intel® FIT Configurations



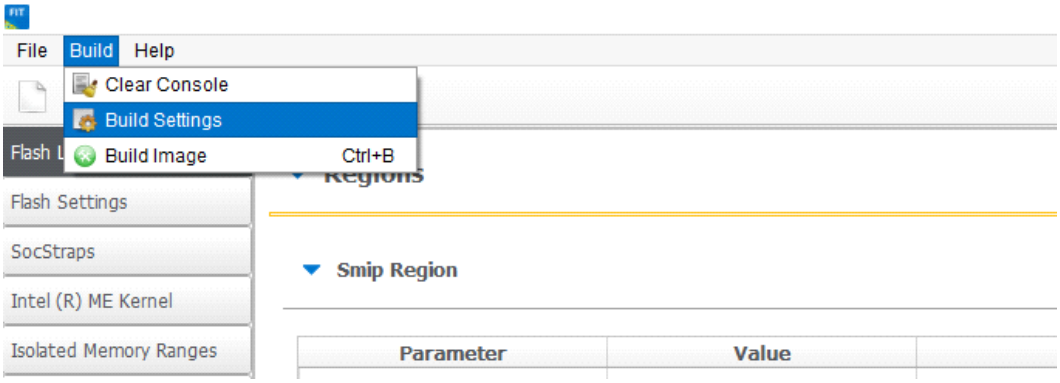
2.6.5.1 Building the Firmware Flash Image

Note: Before building the FW image please make sure that the MEU setting are configured (procedure in the prerequisites section), without this the image creation, the process will **FAIL**.

After the IFWI configurations and the Build setting are set, build the image: FIT setting select build → "Build image".

The output will be the two images, one for DnX flashing (on eMMC based platform), and the other for external programmer/FPT flashing.

Figure 2-17. Saving/Loading Intel® FIT Configurations



2.7 IFWI Flashing Procedure

2.7.1 Prerequisites

The following equipment and setup are required in order to complete IFWI flashing with DnX:

- Management console (a.k.a. Recovery host). Can be any PC, running Windows 7/ 8.1 OS
- Phone Flash Tool (PFT) should be installed on the recovery host. (Link to PFT location available in TXE kit Release Notes)
- DnX module (can be found in TXE kit) and the recovery image should be downloaded to the recovery host.
- eMMC needs to be selected as the boot source for the platform, on GLK RVP set SW7D1 S#4 to on.

2.8 Flashing Procedure for a SPI Based Platform

Please note that on Intel® GLK the boot source needs to be set as SPI, to do so set SW7D1 S#3 to on.

2.8.1 Flashing an Image using Intel® FPT Tool

Flashing the SPI image can be done on the target platform from OS/EFI Shell using the Flash Programing Tool, the tool is located in the FW Kit under tools\Flash_Programming_tool.

To flash the image:

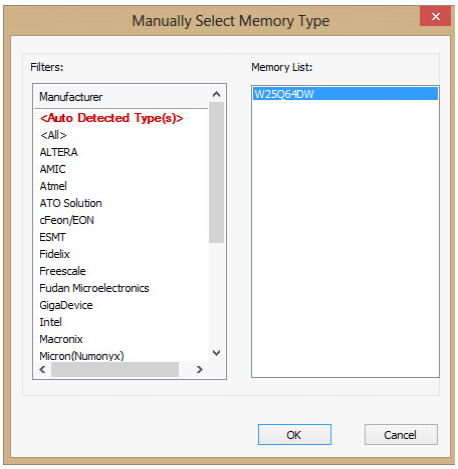
- Copy the FPT tool and the SPI image to the target platform
- From the FPT tool run: FPT -f "image_name.bin"

The expected output from the flashing procedure is "FPT Operation Passed".

2.8.2 Flashing the Image using Dediprog

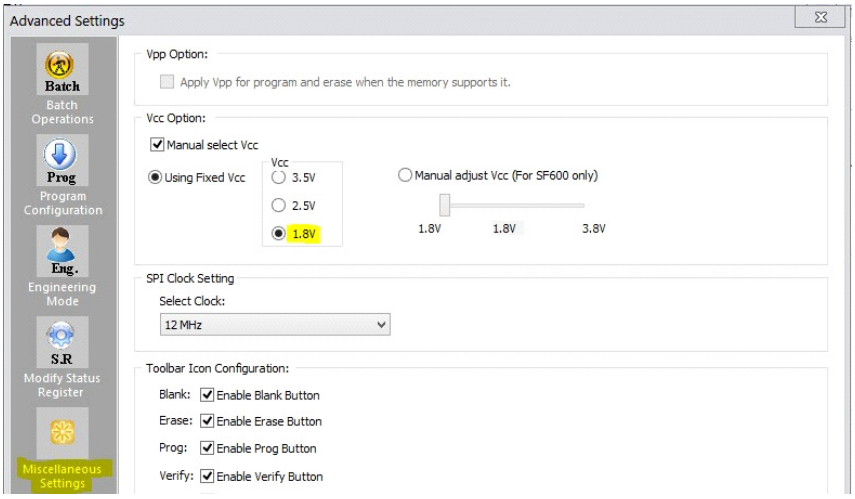
- Connect the Dediprog to the platform and run the Dediprog software.
- Click "Detect".
- Under "Manually

Figure 2-18. Selecting the SPI Component



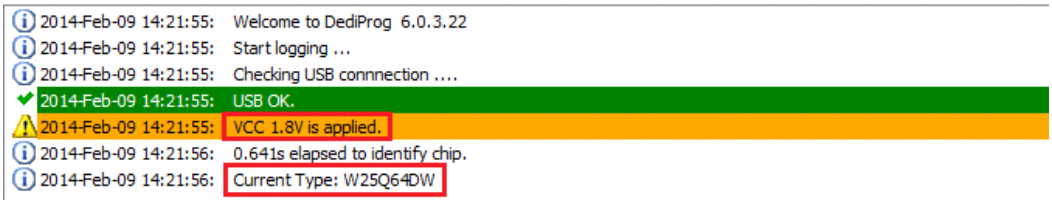
- Click: Config → Miscellaneous Settings, under "Vcc Option", configure Vcc voltage to 1.8V.

Figure 2-19. Setting VCC Voltage



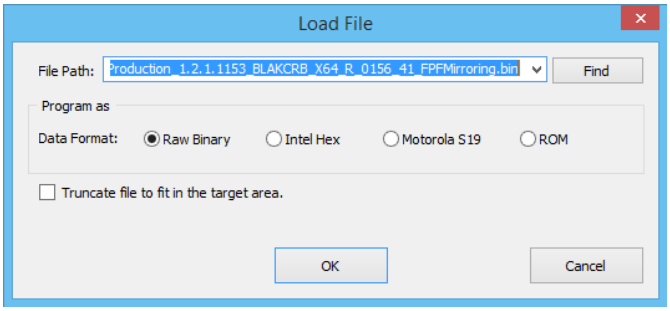
- Under DediProg main window, the VCC voltage will be set to 1.8V, and the SPI component will be selected.

Figure 2-20. Main Window after the Configurations



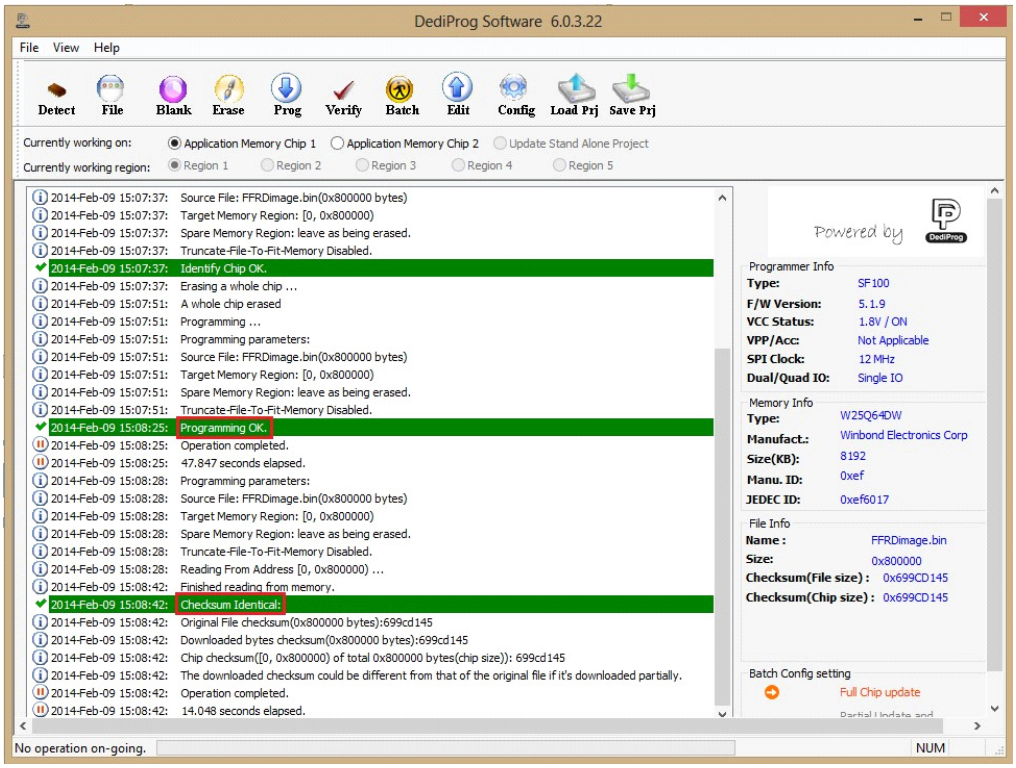
- Click "File", select the SPI image that was built in section 2.4" Creating the Binary Image.
- Under "Program as", set data format as "Raw binary".

Figure 2-21. Load File Settings



- Click “Batch” to flash the file. When the procedure is over, click “Verify” to verify that the flashing was performed correctly.

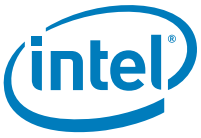
Figure 2-22. Flashing Procedure Expected Result



2.9 Windows Drivers Installation

Once the platform boots up to OS, install the TXEI and SPD using the SetupTXE.exe file that can be located in the kit under the “Installers” folder.

Note: The TXEI and SPD standalone drivers can be found under the same folder.



A Appendix A—ROM Bypass

For GLK based platform ROM bypass needs to be flashed to the platform prior to the bring-up procedure.

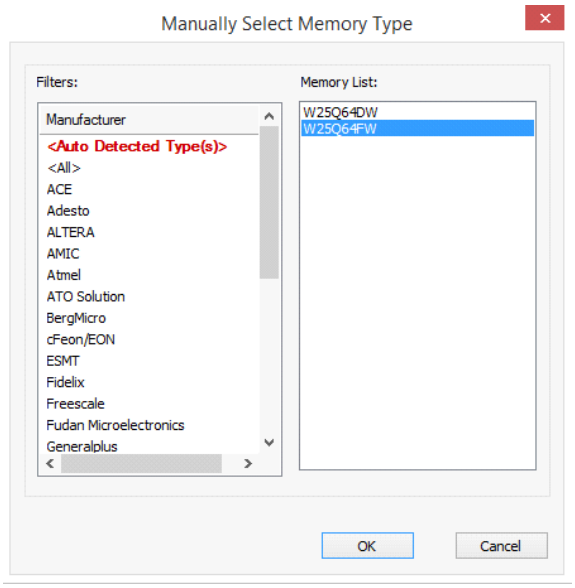
The ROM bypass SPI image can be found in the TXE FW kit, under "Image_Components\TXE"

A.1 Flashing the ROM bypass

- Connect the Dediprog to the platform and run the Dediprog software.
- Click "Detect".
- In the "Manually Select Memory Type" window, select the SPI flash and click OK

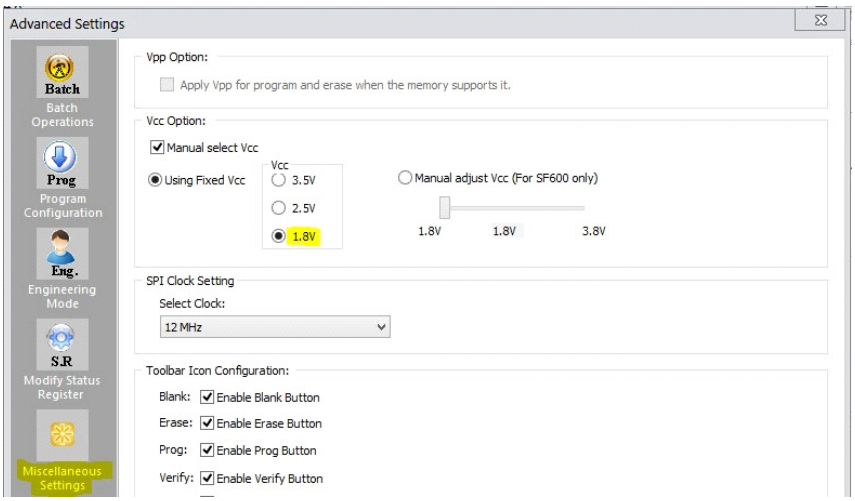
Note: On Intel RVP choose: "W25Q64FW"

Figure A-1. Selecting the SPI Component



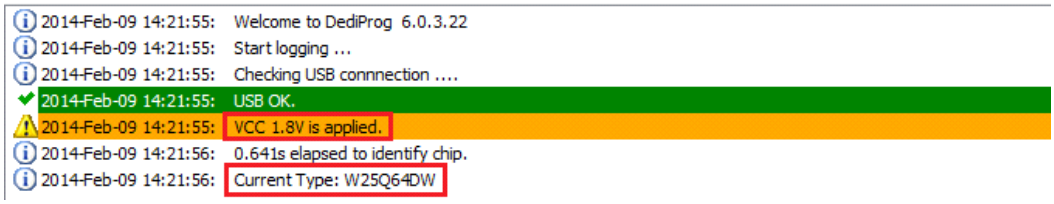
- Click: Config → Miscellaneous Settings, under "Vcc Option" configure Vcc voltage to 1.8V.

Figure A-2. Set VCC Voltage



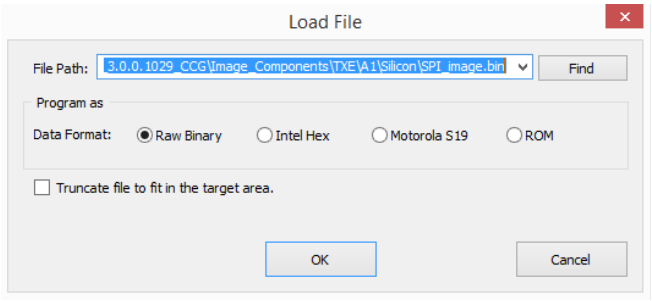
- In the DediProg main window the VCC voltage will be set to 1.8V, and the SPI component will be selected.

Figure A-3. Main Window after the Configurations



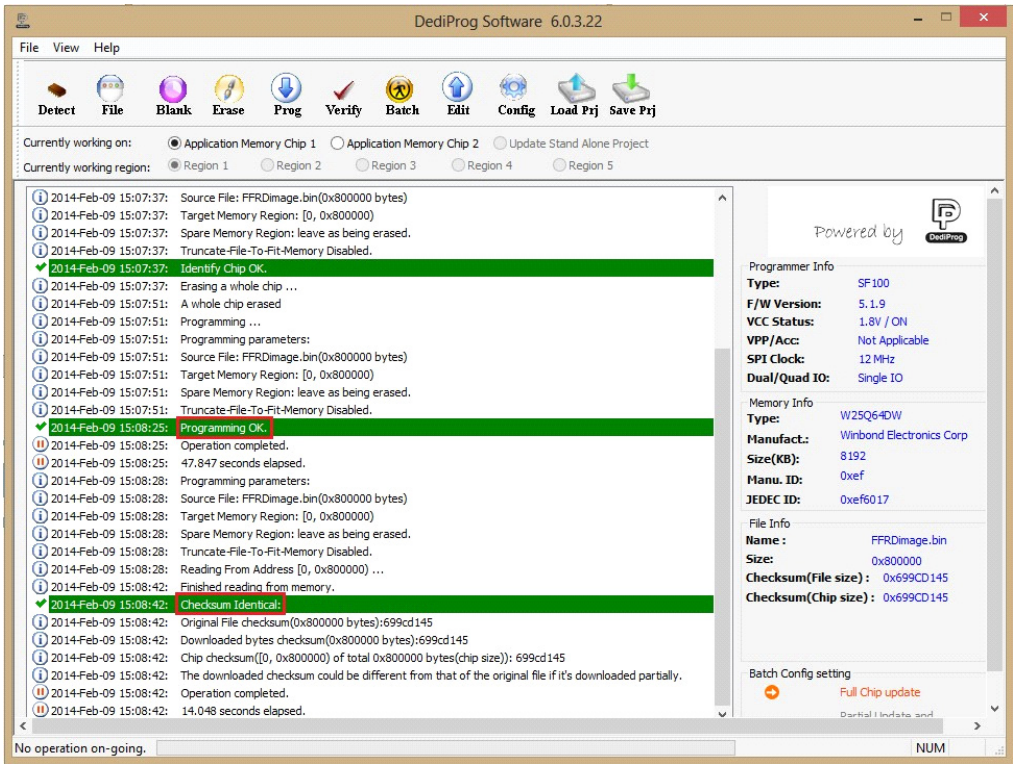
- Click “File”, select the SPI image that was built in section 2.4“Creating the Binary Image. Under “Program as”, set data format as “Raw binary”.

Figure A-4. Load File Settings



- Click “Batch” to flash the file, when the procedure is over, click “Verify” to verify that the flashing was performed correctly.

Figure A-5. Flashing Procedure Expected Result


DediProg Software 6.0.3.22
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Feb-09 15:07:37: Source File: FFRDimage.bin(0x800000 bytes)
2014-Feb-09 15:07:37: Target Memory Region: [0, 0x800000)
2014-Feb-09 15:07:37: Spare Memory Region: leave as being erased.
2014-Feb-09 15:07:37: Truncate-File-To-Fit-Memory Disabled.
2014-Feb-09 15:07:37: Identify Chip OK.
2014-Feb-09 15:07:37: Erasing a whole chip ...
2014-Feb-09 15:07:51: A whole chip erased
2014-Feb-09 15:07:51: Programming ...
2014-Feb-09 15:07:51: Programming parameters:
2014-Feb-09 15:07:51: Source File: FFRDimage.bin(0x800000 bytes)
2014-Feb-09 15:07:51: Target Memory Region: [0, 0x800000)
2014-Feb-09 15:07:51: Spare Memory Region: leave as being erased.
2014-Feb-09 15:07:51: Truncate-File-To-Fit-Memory Disabled.
2014-Feb-09 15:08:25: Programming OK.
2014-Feb-09 15:08:25: Operation completed.
2014-Feb-09 15:08:25: 47.847 seconds elapsed.
2014-Feb-09 15:08:28: Programming parameters:
2014-Feb-09 15:08:28: Source File: FFRDimage.bin(0x800000 bytes)
2014-Feb-09 15:08:28: Target Memory Region: [0, 0x800000)
2014-Feb-09 15:08:28: Spare Memory Region: leave as being erased.
2014-Feb-09 15:08:28: Truncate-File-To-Fit-Memory Disabled.
2014-Feb-09 15:08:28: Reading From Address [0, 0x800000) ...
2014-Feb-09 15:08:42: Finished reading from memory.
2014-Feb-09 15:08:42: Checksum Identical.
2014-Feb-09 15:08:42: Original File checksum(0x800000 bytes):699cd145
2014-Feb-09 15:08:42: Downloaded bytes checksum(0x800000 bytes):699cd145
2014-Feb-09 15:08:42: Chip checksum([0, 0x800000) of total 0x800000 bytes(chip size)): 699cd145
2014-Feb-09 15:08:42: The downloaded checksum could be different from that of the original file if it's downloaded partially.
2014-Feb-09 15:08:42: Operation completed.
2014-Feb-09 15:08:42: 14.048 seconds elapsed.
Powered by DediProg
Programmer Info
Type: SF100
F/W Version: 5.1.9
VCC Status: 1.8V / ON
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
Dual/Quad IO: Single IO
Memory Info
Type: W25Q64DW
Manufact.: Winbond Electronics Corp
Size(KB): 8192
Manu. ID: 0xef
JEDEC ID: 0xef6017
File Info
Name : FFRDimage.bin
Size: 0x800000
Checksum(File size) : 0x699CD145
Checksum(Chip size) : 0x699CD145
Batch Config setting
Full Chip update
No operation on-going.
NUM

B Appendix B—Platform Flash Tool DnX Commands

Please refer to the table below for the Platform Flash Tool DnX related commands, please note that this commands needs to be run from a CLI.

Description	CLI Command
Flashing IFWI image	dnxFwDownloader.exe --command downloadfwos -- fw_dnx DNXP_0x1.bin --fw_image <IFWI_DnX_Image> --flags 0
Clear GPP4/RPMB	dnxFwDownloader.exe --command clearrpmb -- fw_dnx DNXP_0x1.bin --device 2 --idx 0
Configure the GPPs on an eMMC based platform	dnxFwDownloader .exe --command configpart -- fw_dnx DNXP_0x1.bin --path cfgpart.xml --device 2 -- idx 0
Read token	dnxFwDownloader .exe --command readtoken -- fw_dnx DNXP_0x1.bin --path read.bin --slot 0
Write token	dnxFwDownloader .exe --command writetoken -- fw_dnx DNXP_0x1.bin --token test_token.bin --slot 0
Erase token	dnxFwDownloader .exe --command erasetoken -- fw_dnx DNXP_0x1.bin --slot 0
Read boot media contents - EMMC BP1	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path boot1.bin --device 2 -idx 0 --start 0 --blocks 4096 --part 0
Read boot media contents - EMMC BP2	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path boot2.bin --device 2 -idx 0 --start 0 --blocks 4096 --part 1
Read boot media contents - EMMC GPP4	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path gpp4.bin --device 2 -- idx 0 --start 0 --blocks 4096 --part 35
Read boot media contents - EMMC RPMB	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path rpmb.bin --device 2 -- idx 0 --start 0 --blocks 4096 --part 16
Read boot media contents - UFS BP1	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path boot1.bin --device 3 -idx 0 --start 0 --blocks 4096 --part 0
Read boot media contents - UFS BP2	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path boot2.bin --device 3 -idx 0 --start 0 --blocks 4096 --part 1
Read boot media contents - UFS GPP4	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path gpp4.bin --device 3 -- idx 0 --start 0 --blocks 4096 --part 22
Read boot media contents - UFS RPMB	dnxFwDownloader .exe --command readbootmedia -fw_dnx DNXP_0x1.bin --path rpmb.bin --device 3 -- idx 0 --start 0 --blocks 4096 --part 48

C Appendix C—Enabling Quad Mode on SPI Part

When enabling quad operations in the soft steps the Quad enable bit needs to be set accordingly within the SPI part, if not the platform will not boot.

The Quad Enable bit location is different for each SPI vendor model, please refer to the SPI Spec in order to get the Quad Enabled bit location for your SPI device.

C.1 Setting the Quad Enabled Bit using Dediprog

The following procedure uses the SPI part "MX25U6435FM2I-10G" as an example, please follow the procedure below with the settings that corresponds to the SPI device that is used on your platform.

Figure C-1. The Quad Enable information from the "MX25U6435FM2I-10G

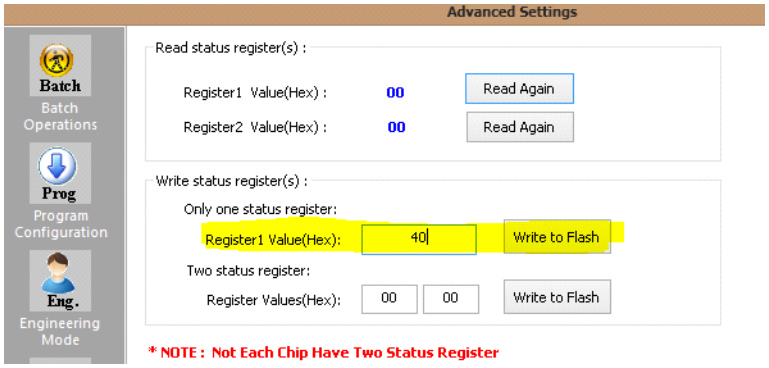
Status Register

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SRWD (status register write protect)	QE (Quad Enable)	BP3 (level of protected block)	BP2 (level of protected block)	BP1 (level of protected block)	BP0 (level of protected block)	WEL (write enable latch)	WIP (write in progress bit)
1=status register write disable	1=Quad Enable 0=not Quad Enable	(note 1)	(note 1)	(note 1)	(note 1)	1=write enable 0=not write enable	1=write operation 0=not in write operation
Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	volatile bit	volatile bit

To set the Quad enable bit:

- Attached Dediprog to SPI device & open Dediprog Software
- Go to Config → S.R. Modify Status Register
- Under "Write Status register(s)", write "0x40" to "Register1 Value(Hex)" as shown below,

Figure C-2. Writing the Quad Enable bit to the Flash



- Verify Register 1 has the value "40" as shown below

Figure C-3. Verifying the register new value