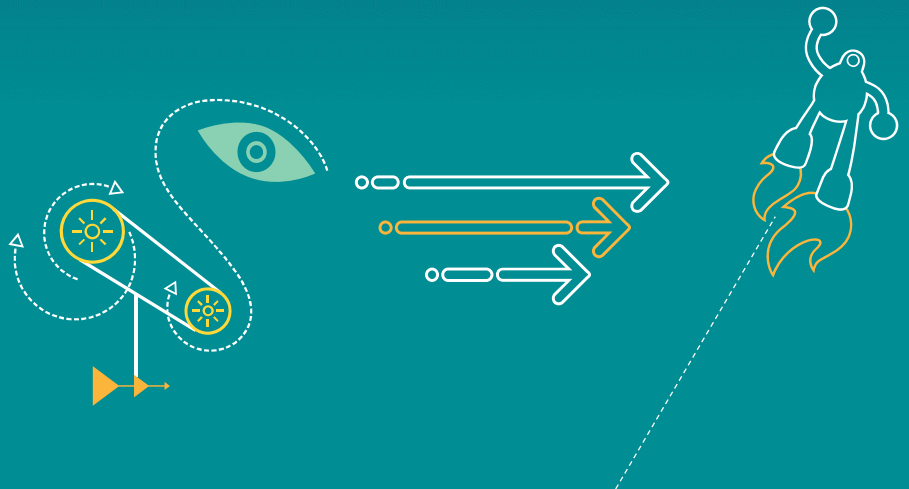
MSM8952 Boot Architecture Overview

Qualcomm Technologies, Inc.



80-NV610-3 B

Confidential and Proprietary – Qualcomm Technologies, Inc.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. All Qualcomm Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

© 2015 Qualcomm Technologies, Inc. and/or its affiliated companies. All rights reserved.

Revision History

Revision	Date	Description
A	Apr 2015	Initial release
B	May 2015	Added slides 29, 40, and 41

QUALCOMM
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

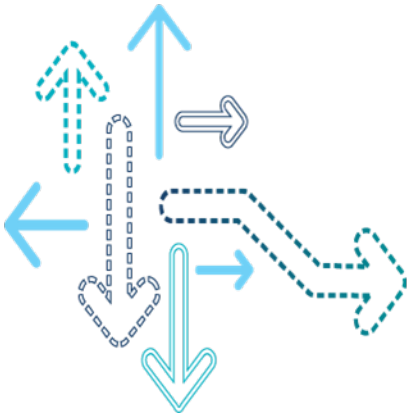
Contents

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Boot Architecture



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Address for Processors

- There are different processors in the MSM8952 chipset. The table lists the processor types and boot addresses.

Subsystem	Processor	Boot address	
APPS	Cortex-A53	0x00100000*	
RPM	Cortex-M3	0x00200000$_{(Subsystem\ view)}$	0x0$_{(System\ view)}$
Modem	MSS_QDSP6	Configurable*	
Pronto	ARM9™	0x0 or 0xFFFF0000 or hardware remap*	
LPASS	LPASS_QDSP6	Configurable*	

* No change in boot address in System and Subsystem views

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Call Stack

Component	Based on processor	Loaded from	Executes in		Function
Application Processor Primary Boot Loader (APPS PBL)	Cortex-A53 (AArch32)	NA	APPS ROM		Boot device and interface detection, Emergency Download mode support, load and authenticate SBL1 ELF segments across L2TCM, and RPM code RAM
Secondary Boot Loader stage 1 (SBL1)	Cortex-A53 (AArch32)	eMMC	L2 TCM (segment1)	L2 TCM	Initial memory subsystem (buses, DDR, clocks, and CDT), load/authorize TZ, Hyperviser, RPM_FW, APPS BL images, memory dump via USB2.0 and Sahara, Watchdog debug retention (e.g., L2 flush), RAM dump to eMMC/SD support, USB driver support, USB charging, thermal check, PMIC driver support, configure DDR, and flush L1/L2/ETB to crash debug support related configuration
			OCIMEM	–	
			RPM code RAM (segment2)	RPM code RAM	
QSEE/TZ(TrustZone)	Cortex-A53 (AArch64)	eMMC	LPDDR3		Equivalent to TZBSP; setup secure run time environment, configure xPU, support fuse driver, authenticates any subsystem images. Abnormal RESET debug functionality is added
QHEE(Hypervisior)	Cortex-A53 (AArch64)	eMMC	LPDDR3		Hypervisor image is responsible for VMM setup, SMMU configurations, and xPU access control

QUALCOMM
2016-10-04 05:42:47 PST
rbw46@zeus4.qti.qualcomm.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Call Stack (cont.)

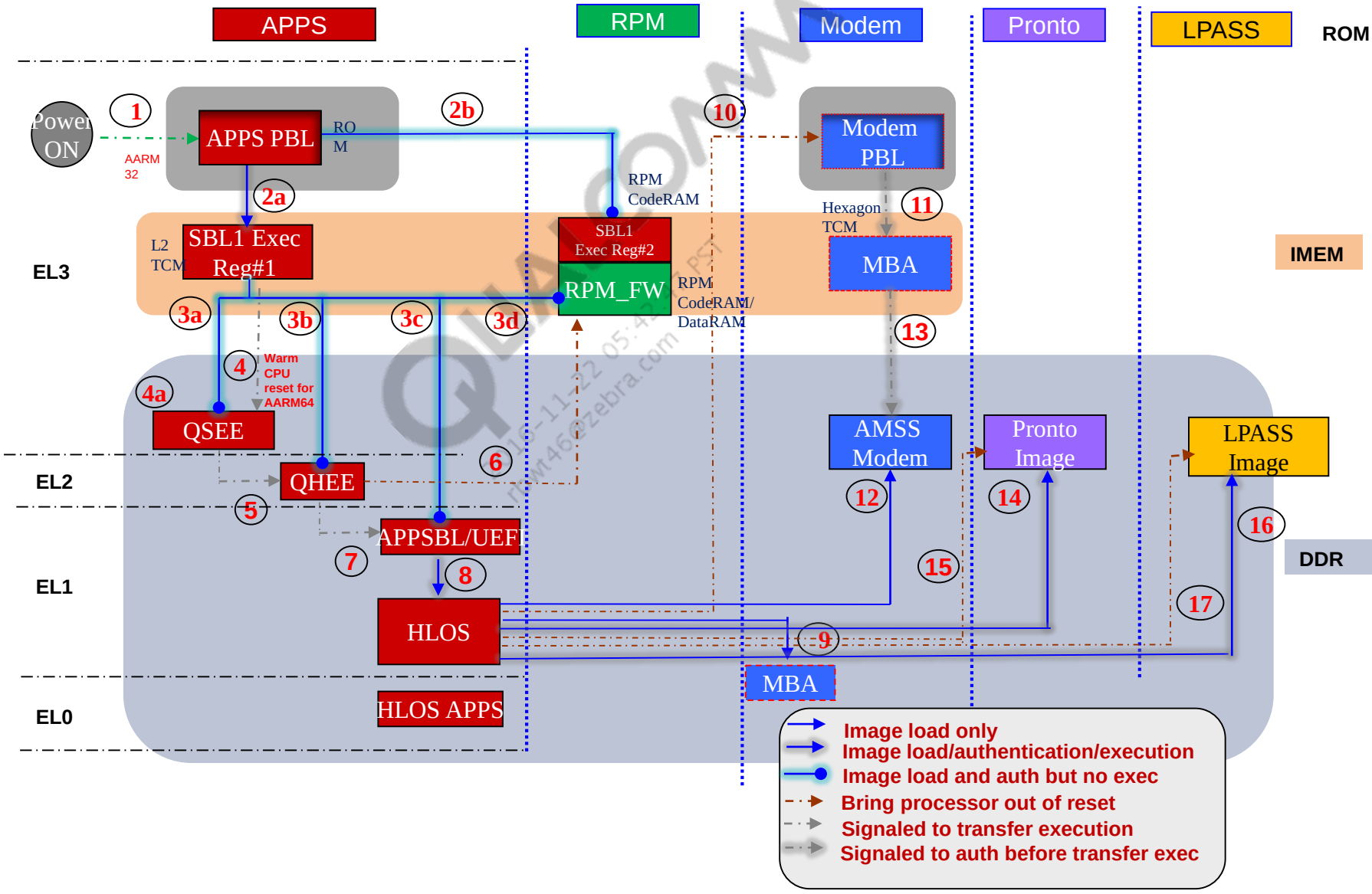
Component	Based on processor	Loaded from	Executes in	Function
Resource Power Manager Firmware (RPM_FW)	Cortex-M3	eMMC	RPM code RAM	Resource power management
APPSBL/boot manager and OS loader	Cortex-A53* (AArch32/ AArch64)	eMMC	LPDDR3	Splash screen, loads and authenticates the kernel, and provides HLOS-specific BL features using UEFI
High-Level Operating System (HLOS)	Cortex-A53 (AArch32/ AArch64)	eMMC	LPDDR3	Boots HLOS images, e.g., A53 HLOS kernel image, Pronto image, etc.
Modem Primary Boot Loader (Modem PBL)	MSS_QDSP6	NA	Modem ROM Qualcomm® Hexagon™ TCM (Data and stack)	Set up Hexagon TCM, copies MBA from LPDDR/3 into Hexagon TCM, and authenticates MBA in Hexagon TCM
Modem Boot Authenticator (MBA)	MSS_QDSP6	eMMC	Hexagon TCM	Authenticates the modem image, xPU protects the DDR regions for modem, and memory dump

*LK boot loader will start in 32-bit.

QUALCOMM®
2016-11-22 05:42:47 PST
rbwa@cbra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Code Flow



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Flowchart

1. The system powers on and takes MSM8952 AP CPU out of reset.
2. In Cortex-A53 APPS PBL executes the following:
   (a) Loads and authenticates the SBL1 segment #1 from the boot device to L2 (as TCM)
   (b) Loads and authenticates SBL1 segment #2 (SDI equivalent) to RPM code RAM, and then jumps to SBL1
3. SBL1#1 initializes DDR
   (a) Loads and authenticates the QSEE/TZ image from the boot device to DDR
   (b) Loads and authenticates the QHEE image from the boot device to DDR
   (c) Loads and authenticates the RPM firmware image from the boot device to RPM code RAM
   (d) Loads and authenticates the HLOS APPSBL image from the boot device to DDR
4. SBL1 transfers execution to QSEE/TZ. QSEE/TZ sets up a secure environment, configures xPU, and supports the fuse driver.
   (a) SBL1 runs in AArch32 mode. QSEE/TZ runs in AArc64 mode. For AArch64 mode switch SBL1 sets boot re-mapper for QSEE entry and writes to RMR register and then triggers warm-reset. Now, QSEE starts in AArch64 mode.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

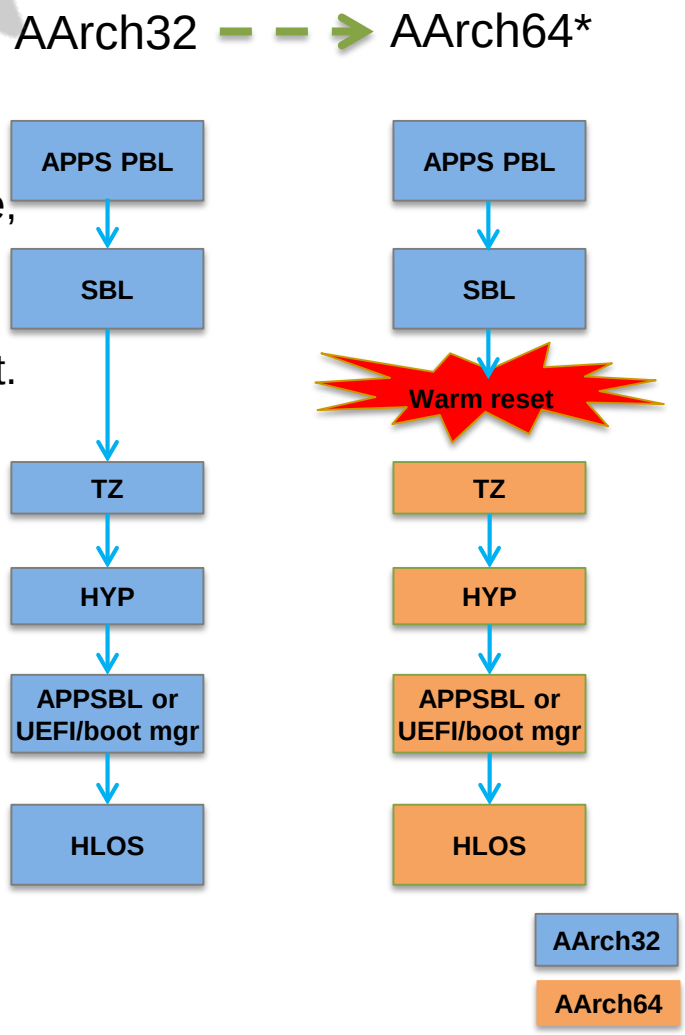
Boot Flowchart (cont.)

5. QSEE transfers execution to QHEE. QHEE* is responsible for VMM setup, SMMU configurations, and xPU access control.
6. QHEE notifies RPM to start the RPM firmware execution.
7. QHEE transfers execution to HLOS APPSBL to initialize the system.
   (a) The Linux APPS boot loader (HLOS APPSBL) starts the execution in AArch32 mode-only.
   (b) This is done by QHEE by looking at the ELF header for HLOS APPSBL, which indicates that it uses 32-bit instruction set architecture. QHEE changes to 32-bit mode and starts Linux APPS boot loader (HLOS APPSBL) execution in 32-bit mode.
8. HLOS APPSBL loads and authenticates the HLOS kernel. The Linux APPS boot loader (HLOS APPSBL) will indicate about the HLOS kernel AArch64 mode by making an SCM call to secure the monitor before exiting. LK does not jump into kernel directly as it used to do previously.
9. HLOS kernel loads the Modem Boot Authenticator (MBA) to DDR via PIL.
10. HLOS kernel brings Modem DSP Hexagon out of reset.
11. Modem PBL then continues its boot process.
12. HLOS kernel loads the AMSS modem image to DDR via PIL
13. Modem PBL authenticates MBA and then jumps to it.
14. HLOS loads the Pronto image to DDR via PIL.
15. HLOS brings the Pronto image out of reset so that the Pronto image starts executing.
16. HLOS loads the LPASS image to DDR via PIL.
17. HLOS brings the LPASS image out of reset so that the LPASS image starts executing.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

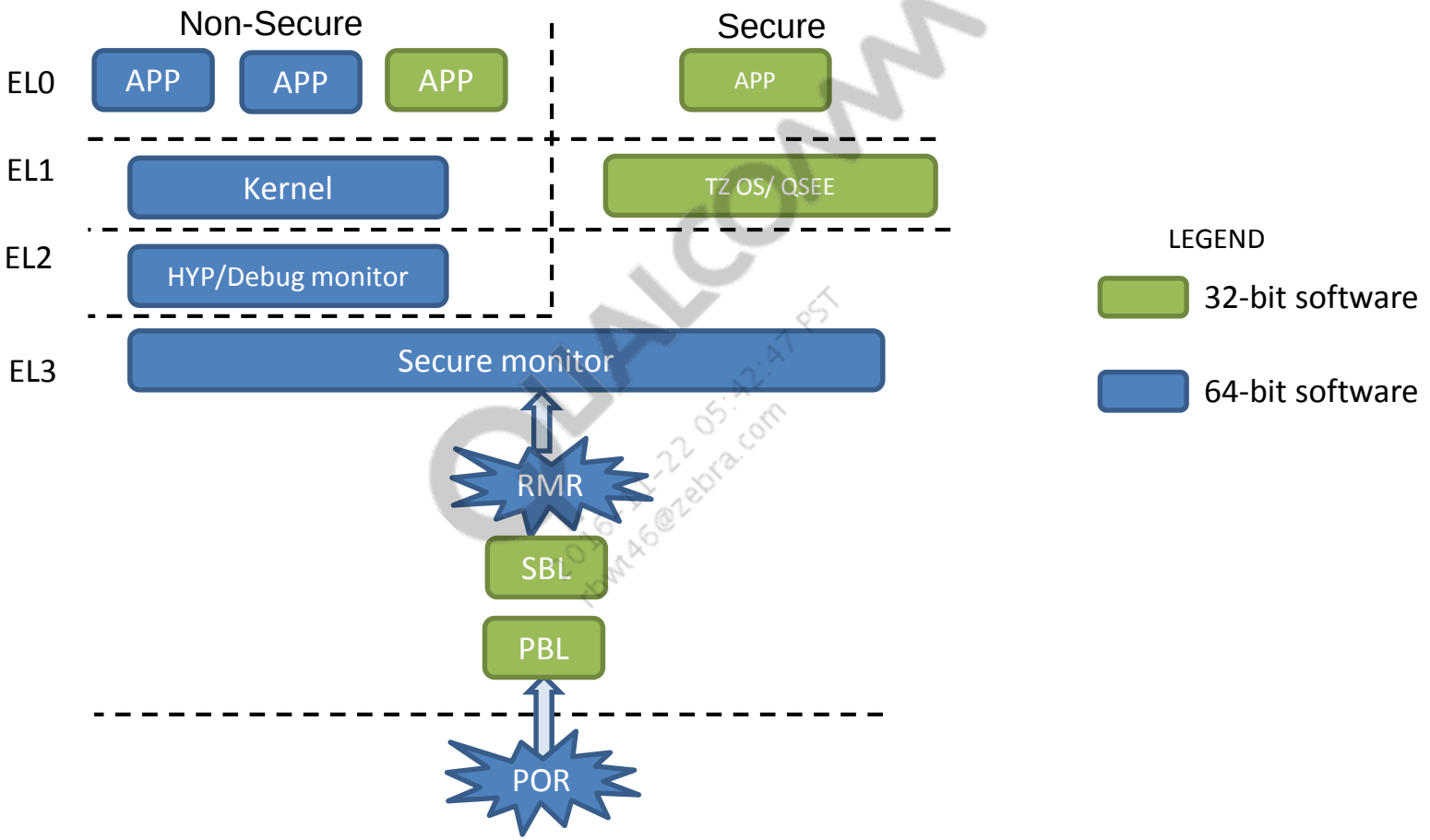
Boot – AArch64 High-Level Flow

- PBL and SBL run in AArch32 mode only.
- SBL downloads and authenticates images for subsequent boot flow: QSEE, QHEE, RPM firmware, and APPSBL.
- PBL and SBL configure QSEE image architecture mode, pass the images information to TZ, and transfer the execution into TZ.
- If AArch64 is not supported, jump into QSEE entry point.
- If AArch64 is supported, update reset address, remap secure path to QSEE entry, and then trigger AP CPU warm reset (single-core reset). The CPU will then reset from QSEE at AArch64* mode.
- The remaining AP software stages are supported in AArch64.



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

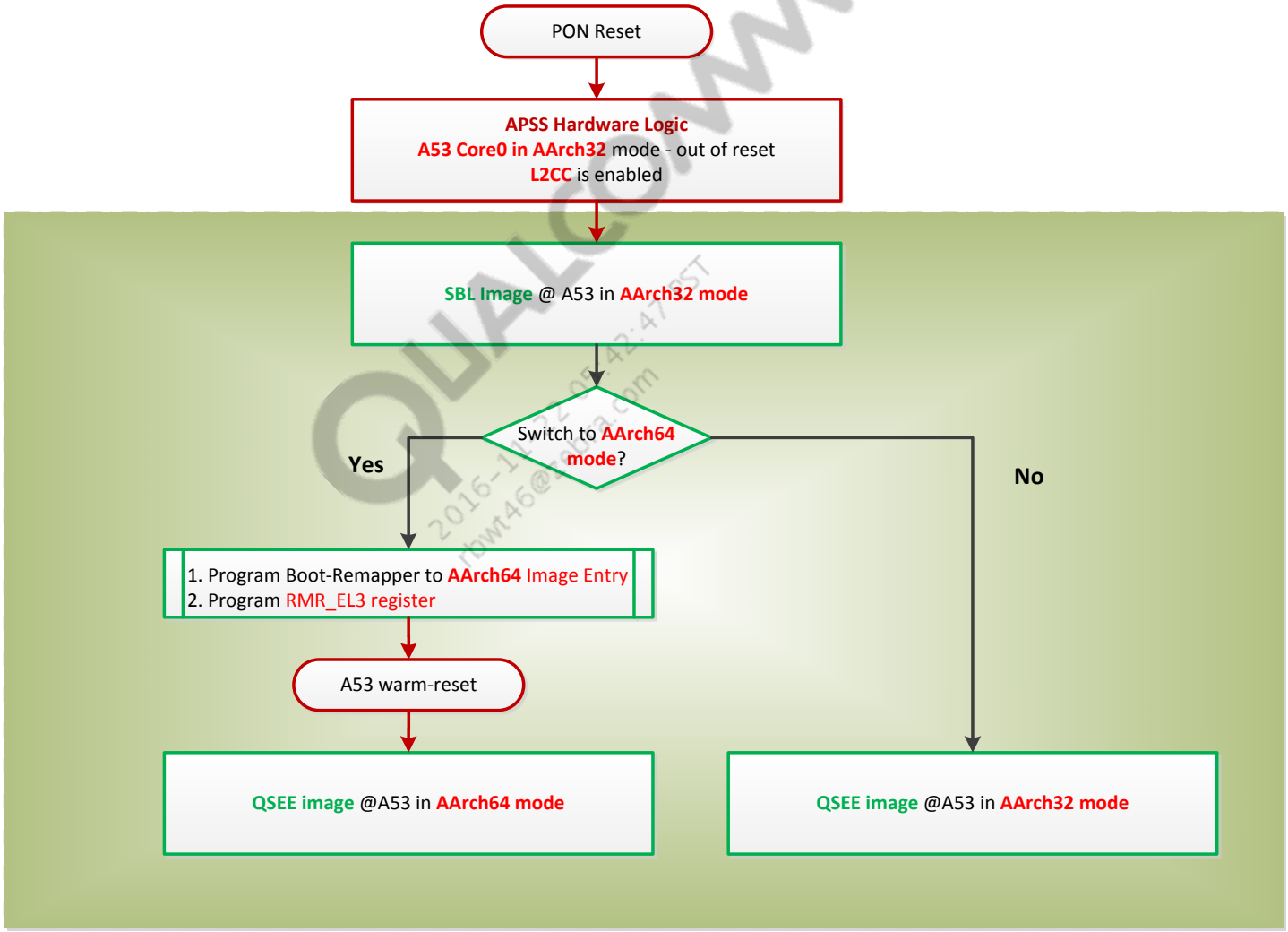
Boot – AArch64 High-Level Flow (cont.)



AArch64 has four exception levels (0-3), that replace eight processor modes found in ARM7™. The least privileged, EL0, is equivalent to User mode and EL1 to Kernel mode with EL2 for hypervisors, while the highest privilege level, EL3, is used for the TrustZone security monitor.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot – AArch32 to AArch64 Switch Flowchart



In MSM8952 QSEE/TZ will always run in AArch64 mode.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Switch from AArch32 to AArch64

- SBL runs in AArch32 mode
  - Set up boot remapper to AArch64 entry which is the QSEE image
  - Write to RMR_EL3 register to set AArch64 mode and trigger a warm-reset
- Upon warm-reset, A53 successfully warm boots in QSEE with AArch64 mode

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

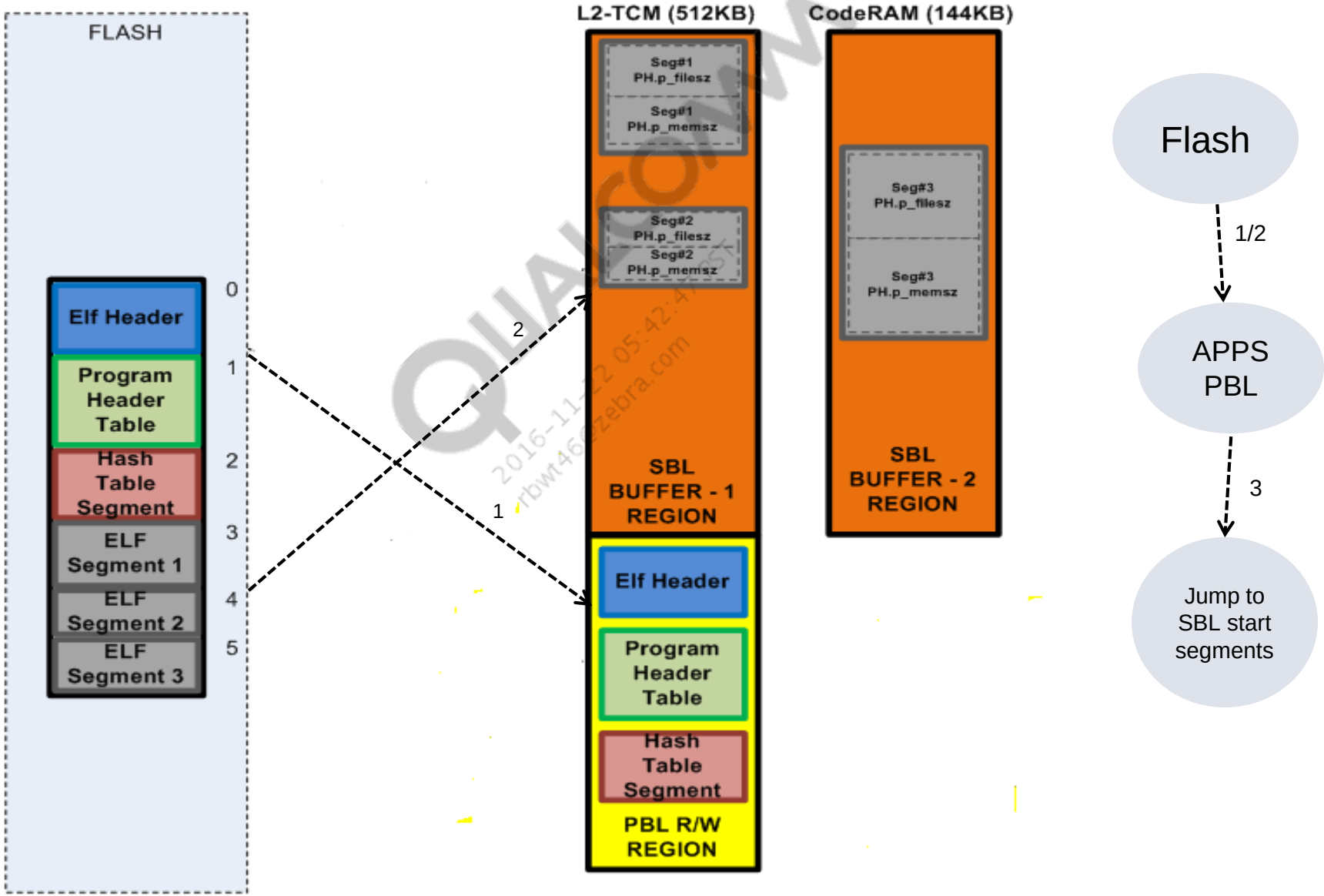
Debug Assistant for AArch64

- Abnormal reset causes A53 to boot up in Core0 AArch32 mode
- To debug, take a backup of all the alive cores before abnormal reset
- Arch
  - PBL saves X0 and X1, and passes the control to SBL_Debug_Entry
  - SBL_Debug saves all Core0 (except x0, x1) GPRs in AArch32 mode and passes control to AArch64 QSEE
  - AArch64 QSEE then saves all Core0 GPRs in AArch64 mode

QUALCOMM
2016-11-22 08:42:47 PST
rbwt46@zebra.com

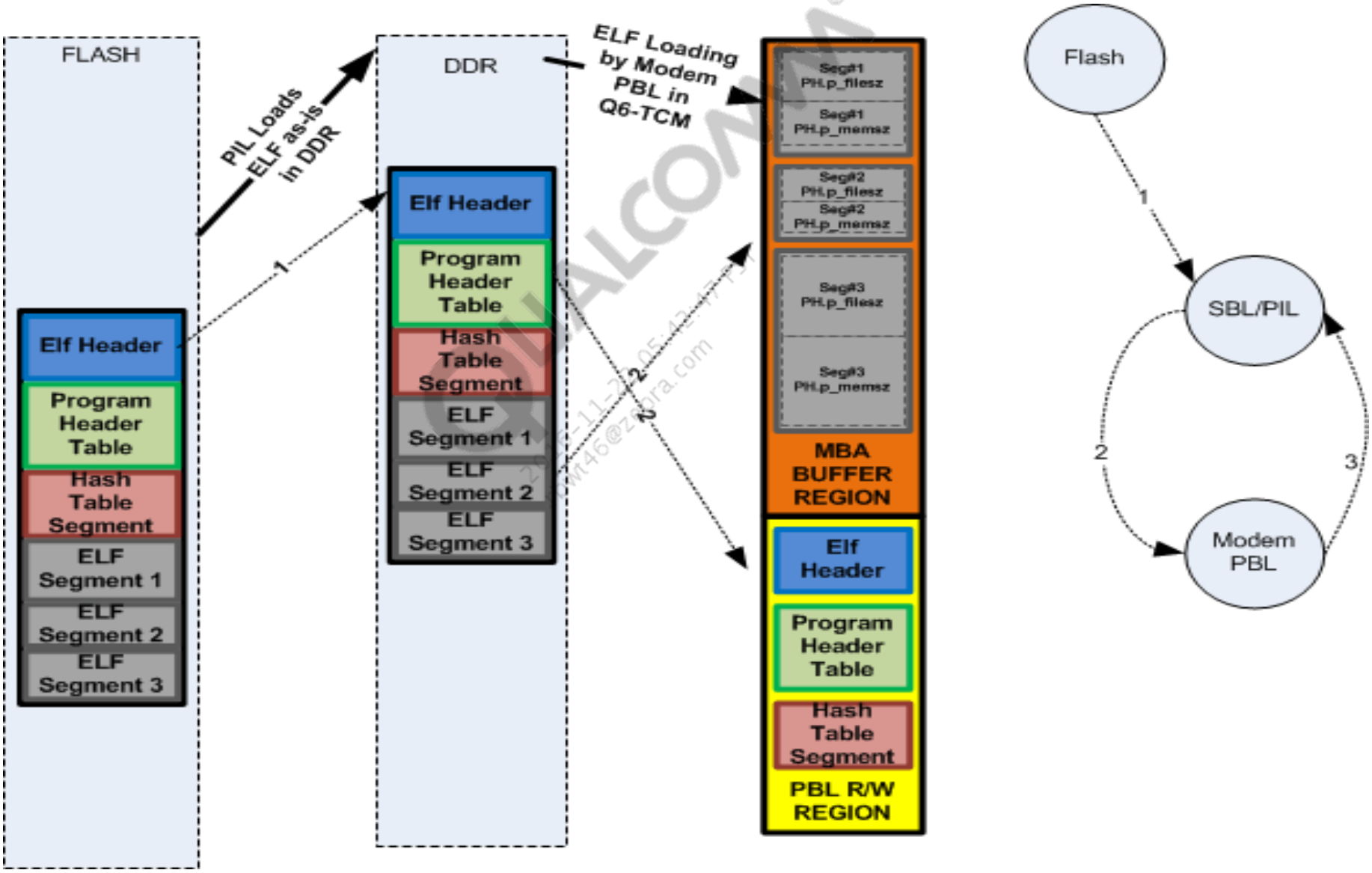
Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot – SBL ELF Loading



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot – MBA ELF Loading



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot – ELF Loading

- SBL ELF loading
  - APPS PBL starts ELF loading once the ELF headers are verified. Then APPS PBL loads the program headers, and then authenticates the hash segments
  - APPS PBL loads the loadable segments and verifies hashes for each of those segments
- MBA ELF loading
  - Before the Hexagon processor is out of RESET, the PIL loads MBA ELF as is in the DDR and loads RMB0 with start address
  - Modem PBL starts to load MBA ELF from DDR to Hexagon L2 TCM
  - Modem PBL first loads and verifies ELF headers, loads program headers and hash segments, and then authenticates hash segments
  - Modem PBL loads loadable segments and verifies hashes for those segments

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

SBL-QSEE Interface

SBL loads and authenticates different boot images and hands over the control to QSEE. SBL populates image information as captured in the below structure for all images loaded by SBL, and then passes to QSEE.

```
typedef struct boot_sbl_qsee_interface
{
  uint32 magic_1;
  uint32 magic_2;
  uint32 version;
  uint32 number_images;
  uint32 reserved_1;
  boot_image_entry[BOOT_IMAGES_NUM_ENTRIES];

} boot_sbl_qsee_interface;
```

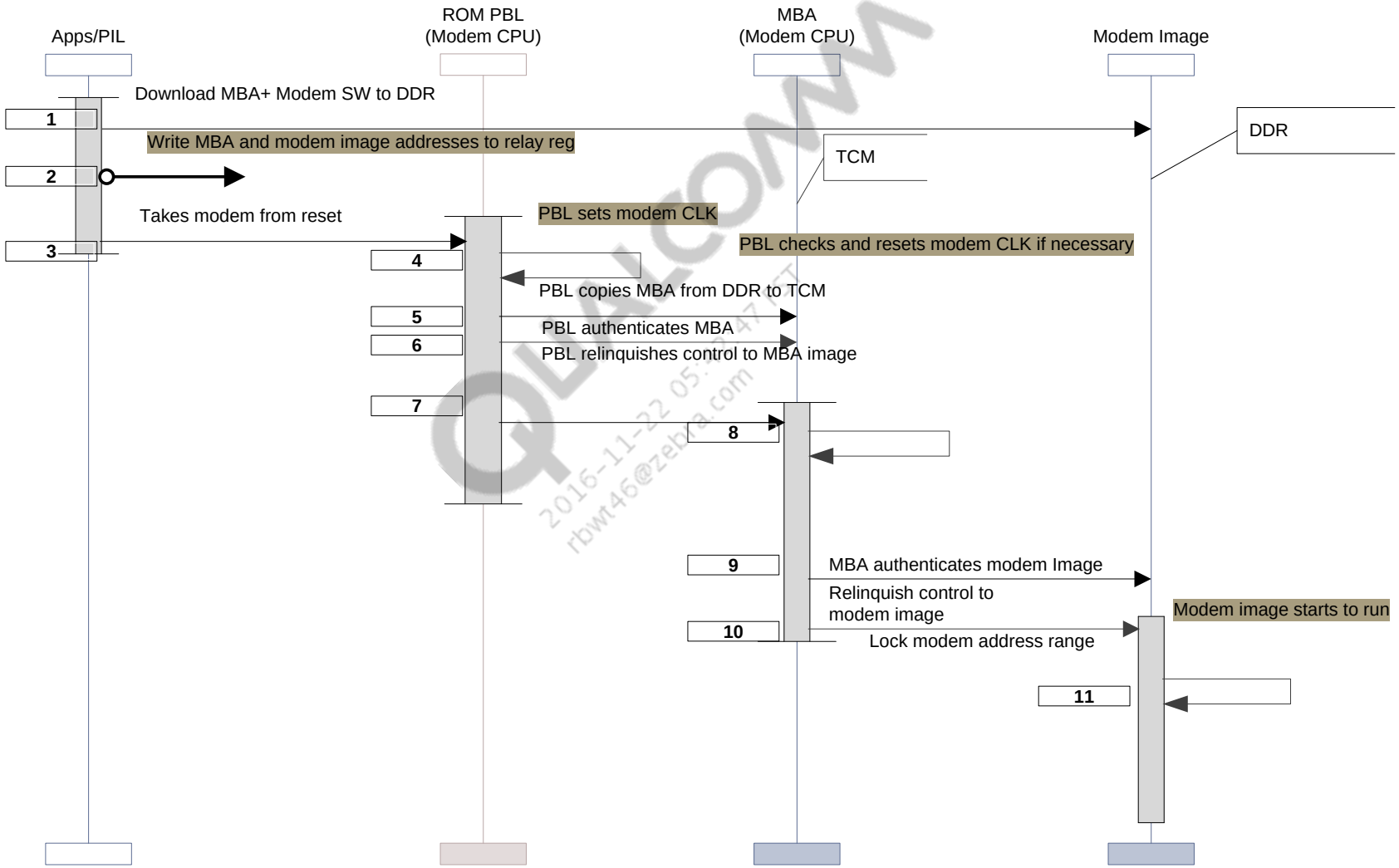
```
boot_images_entry
{
  secboot_sw_type image_id;
  uint32 e_ident;
  uint64 entry_point;
  secboot_verified_info_type  image_verified_info;
  uint32 reserved_1;
  uint32 reserved_2;
  uint32 reserved_3;
  uint32 reserved_4;

} boot_images_entry;
```

```
typedef struct
secboot_verified_info_type
{
  uint32      version_id;
  uint64      sw_id;
  uint64      msm_hw_id;
  uint32      enable_debug;
  secboot_image_hash_info_type
  image_hash_info;
  uint32 enable_crash_dump;
} secboot_verified_info_type
```

QUALCOMM 2016-1[illegible]22 05:42:47 PST rbwt46@[illegible]ora.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Independent Modem Authentication Flow



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

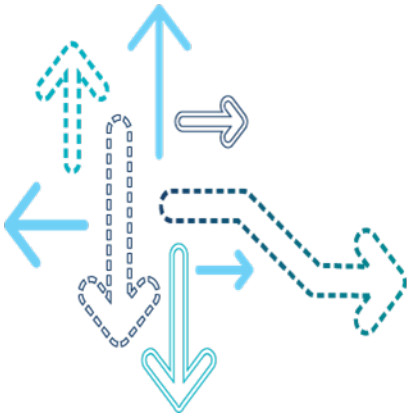
Boot Flowchart – Independent Modem Authentication

1. APPS HLOS downloads the MBA and modem image to DDR
2. APPS HLOS writes MBA and modem image address to RMB registers
3. APPS HLOS takes the modem out of reset
4. Modem PBL executes and sets its own clock
5. Modem PBL copies the MBA from DDR to modem TCM
6. Modem PBL authenticates the MBA
7. Modem PBL relinquishes control to the MBA
8. MBA closes the public domain and locks the modem address range
9. MBA authenticates the modem image
10. MBA relinquishes control to the modem image
11. Modem image starts to run

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

MSM8952 PBL



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

PBL Error Logging

- APPS PBL − Saves the error information in RPM code RAM
  - Once a PBL error occurs, it goes to the error handler.
  - PBL logs the necessary log information into the RPM code RAM.
  - PBL goes to Emergency Download mode if it is not disabled by the fuse.
  - PBL tries SDC Port 2 recovery first. If it fails, PBL goes to HS-USB Emergency Download mode.
  - In Emergency Download mode, PBL enters the Sahara protocol to receive and authenticate the flash programmer from the host.
  - Once loaded and authentications are passed, the system jumps to the flash programmer start address.
  - The flash programmer executes and downloads the necessary boot images from the host side.
- Modem PBL – Updates PBL-logs' error status and error details on RMB-status registers

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

APPS PBL Error Log Format

140 Bytes	APPS PBL last error info x 5 (The APPS PBL will have four last error entries to be able to debug multiple error re-entrant case

```
/* Error log structure to store data describing error  */

typedef struct boot_pbl_err_type
{
   uint32                  pbl_err_code_start;
   uint32                  pbl_err_details;
   uint32                  timestamp;
   uint32                  pbl_id;
   uint32                  patch_id;
   const char*             filename;
   uint32                  line_num;
   uint32                  pbl_err_code_end;
} boot_pbl_err_type;
```

IMP Details of PBL last error structure

- Error code (4 bytes)
- Error details (4 bytes)
- Time stamp (4 bytes)
- PBL ID (4 bytes)
- Line number (4 bytes)

ERROR code details (4 bytes)

For normal errors (except exceptions) ERROR DETAILS are used to get additional information about error. Driver specific errors are logged here

- 0xEFAABBCC
- EF: Error signature or 7F: Pass signature
- AA: PBL function block ID
- BB: Sub-error in functional block
- CC: Error count number, start from 0x1

Indicates Sub-Error functional block, classifieds into two types.
Recoverable – Hardware stable, error code starts from 0x1 to 0xDF.
Non Recoverable – Hardware is not stable, error code starts from 0xE0 to 0xFF.

Current list of PBL function block IDs (128 max):
Refer next page
pbl_log__block_type

QUALCOMM® 2016-11-22 05:42:47 PST

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

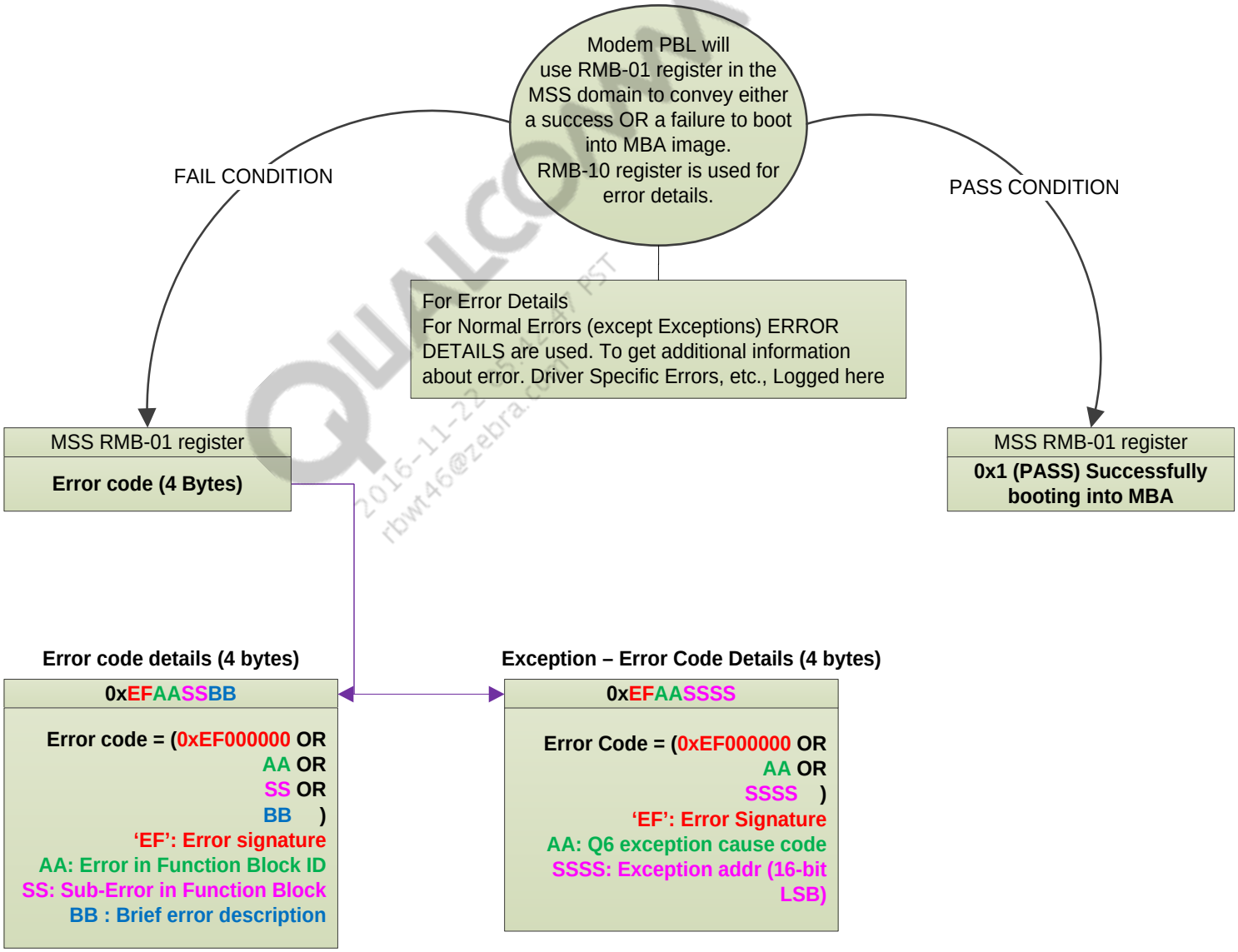
APPS PBL Error Code Definitions

```
typedef enum
{
  PBL_LOG_GENR           = 0x010000,
  PBL_LOG_PROC           = 0x020000,
  PBL_LOG_LOADER         = 0x030000,
  PBL_LOG_FUSE           = 0x040000,
  PBL_LOG_AUTH           = 0x050000,
  PBL_LOG_TIMER          = 0x060000,
  PBL_LOG_CLOCK          = 0x070000,
  PBL_LOG_SEC_HW         = 0x080000,
  PBL_LOG_SECBOOT        = 0x090000,
  PBL_LOG_SEC_IMG_AUTH   = 0x0A0000,
  PBL_LOG_SDCC           = 0x0B0000,
  PBL_LOG_SAHARA         = 0x0C0000,
  PBL_LOG_NAND           = 0x0D0000,
  PBL_LOG_PCIe           = 0x0E0000,
  PBL_LOG_UFS            = 0x0F0000,
  PBL_LOG_USB            = 0x100000,
  PBL_LOG_EXCEPTION      = 0x110000,
  PBL_LOG_ELF             = 0x120000,
  PBL_LOG_FORCE32BITS    = 0x7FFFFFFF /* To ensure it's 32 bits wide */
}pbl_log_block_type;
```

QUALCOMM®
2015-11-22 05:42:47 PST
wm46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Modem PBL Error Log Format



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Common PBL Error Log

Recoverable errors
- Authentication failure
- Failure due to bad image or image type
- Failures due to Sahara
- Failed to detect a device

Non-recoverable errors
- NULL pointer checks
- Any clock/PLL failures
- Timer failures
- Unsupported SDCC spec version or card type
- Failures related to MMU
- Invalid boot option configured
- Invalid boot speed configured (Proc or MCLK)
- Failures during memcpy or memzi
- Hardware configurations/device state machine failures
- Enumeration timeouts
- Failures during stack copy
- Stack check failures
- FEC/fuse errors
- Exceptions

Sub-error per functional block

Example

General

PBL_GEN_REC_LOG_WDOG_RESET_DEBUG	0x0100
PBL_GEN_REC_LOG_RANGE_SIZE_ERR	0x0200
PBL_GEN_REC_LOG_FLASH_STATUS	0x0300
PBL_GEN_REC_LOG_MISC	0x0400
PBL_GEN_REC_RESERVED_1	0x0700
PBL_GEN_REC_RESERVED_2	0x0800
PBL_GEN_REC_RESERVED_3	0x0900
PBL_GEN_REC_RESERVED_4	0x0A00
PBL_GEN_NON_REC_LOG_NULL_PTR_ERR	0xE000
PBL_GEN_NON_REC_LOG_STACK_CPY_ERR	0xE100
PBL_GEN_NON_REC_LOG_BOOT_OPTION	0xE200
PBL_GEN_NON_REC_LOG_EXCEPTION_ABORT	0xE300
PBL_GEN_NON_REC_LOG_FUNC_EXEC	0xE400
PBL_GEN_NON_REC_LOG_MEMCPY	0xE500
PBL_GEN_NON_REC_LOG_MEMZI	0xE600
PBL_GEN_NON_REC_RESERVED_1	0xE700
PBL_GEN_NON_REC_RESERVED_2	0xE800
PBL_GEN_NON_REC_RESERVED_3	0xE900
PBL_GEN_NON_REC_RESERVED_4	0xEA00
PBL_GEN_LOG_FORCE32BITS	0x7FFFFFFF

PBL function block IDs

GENERAL	0x010000
SDCC	0x020000
NAND	0x030000
SPI	0x040000
RESERVED	0x050000
LOADER	0x060000
USB	0x070000
TIMER	0x080000
CLOCK	0x090000
SAHARA	0x0A0000
SECBOOT	0x0B0000
SEC_HW	0x0C0000
SEC_CE	0x0D0000
AUTHENTICATION	0x0E0000
FLCB	0x0F0000
SPMI	0x100000
FUSE	0x110000
PROCESSOR	0x120000

QUALCOMM Confidential and Proprietary

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

PBL Debugging

1. Obtain APPS PBL dump
   1. Open build <META ROOT>\common\t32\t32start.cmd
   2. Navigate to JTAG DAP mode > Podbus device chain > power trace ethernet
   3. Select Cortex A53 Cluster1 Core 0
   4. Click **Start**
      - Store binary images
        - `d.save.binary c:\Temp\APPS_PBL_ROM.bin 0x00100000--0x00117FFF`
        - `d.save.binary c:\Temp\RPM_CODE_RAM.bin 0x00200000--0x00223FFF`
        - `d.save.binary c:\Temp\APPS_PBL_L2.bin 0x08000000--0x0807FFFF`
      - Store CPU register information
        - `store c:\Temp\r_Apps.cmm R`
2. Upload the APPS PBL dump in Salesforce and open a case to analyze the issue

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

PBL Boot Option

- The Boot_config [0..4] GPIOs or BOOT_CONFIG fuses can be used to select the following boot options. Once the fuse is blown, the GPIOs used for the boot option are free to be used as common GPIOs.

BOOT_CONFIG	MSM8952	Comments
0x00	BOOT_DEFAULT_OPTION	eMMC@SDC1→SD@SDC2→USB
0x01	BOOT_SDC_PORT2_THEN_SDC_ PORT1_OPTION	SD@SDC2→eMMC@SDC1
0x02	BOOT_SDC_PORT1_OPTION	eMMC@SDC1
0x03	BOOT_USB_OPTION	

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Fuses Used in PBL

Fuse bits eFuse	Domain	GPIO	Description
FAST_BOOT (4 bits)	OEM	4	Used by boot code to specify which device has priority to be booted from; this helps speed up the boot flow.
VID (16 bits)	OEM		USB vendor ID
PID (16 bits)	OEM		USB product ID
E_DLOAD_DISABLE	OEM		Disables emergency downloader
USB_ENUM_TIMEOUT	OEM		BOOT ROM must support USB enumeration timeout. Timeout applies to USB Download mode. If USB is not enumerated within time (90 sec), quit USB enumeration. If USB suspends or is disconnected after enumeration, start timer again.
SDCC_MCLK_BOOT_FREQ	OEM		SDCC core clock
OEM_PK_HASH	OEM		Blows the hash of OEM public key
FORCE_USB_BOOT	No fuse	1 pin GPIO50	Forces boot from USB
FORCE_DLOAD_DISABLE	OEM		Force-download is disabled

QUALCOMM
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Boot Loader Customization

- If the licensee uses different memory parts than those used in the reference design, the boot loader needs to be updated
- For eMMC memory:
  - If it is compliant, then no customization is required
  - For partition change, see Application Note: eMMC Partition Tables and Raw Programming (80-N4584-1) for details
- For DDR memory:
  - CDT in .xml needs to be reviewed and updated
  - DDR device type, mode (interleaved or not), density (rank/bank/row/column), and JEDEC specification default timing data
  - Retrieved by SBL1 and stored in shared IMEM and accessed by RPM
- Hardware platform ID:
  - Defines PLATFORM_SUBTYPE based on the same chipset (MTP, CDP, etc.)
  - Controls the alternate functions on GPIOs
  - Stored in SMEM (DDR) and accessed by LK boot loader and the Linux kernel
  - For details, see DDR SDRAM HAL APIs and Customization Reference Guide (80-N1218-1) and Application Note: EEPROM Software Configuration Data Table (CDT) (80-N3411-1)

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

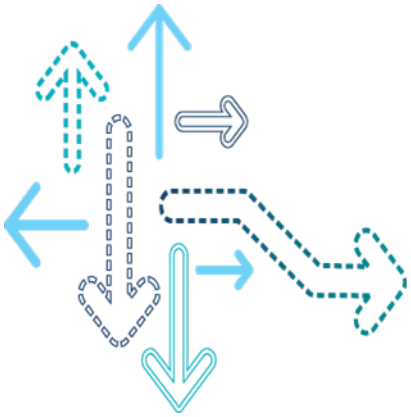
Single Instance DDR Driver

- Eliminate 3 instances of DDR driver (SBL1, RPM_FW, and SDI) to 1 instance
- RPM CodeRAM size increased by 16 KB (144 KB) to host a SBL1 ELF segment which includes DDR driver.
- Since RPM(M3) and APPS(Cortex-A7) are part of ARMv7 family, both SBL1 and RPM_FW plug into same driver instance loaded in RPM CodeRAM by PBL. The SDI watchdog reset path also uses the same instance to bring DDR out of self refresh.
- Rationale
  - Scale DDR driver sequence and setting maintenance
  - Image size optimization

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Watchdog Reset Debug



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

SDI Handling changes

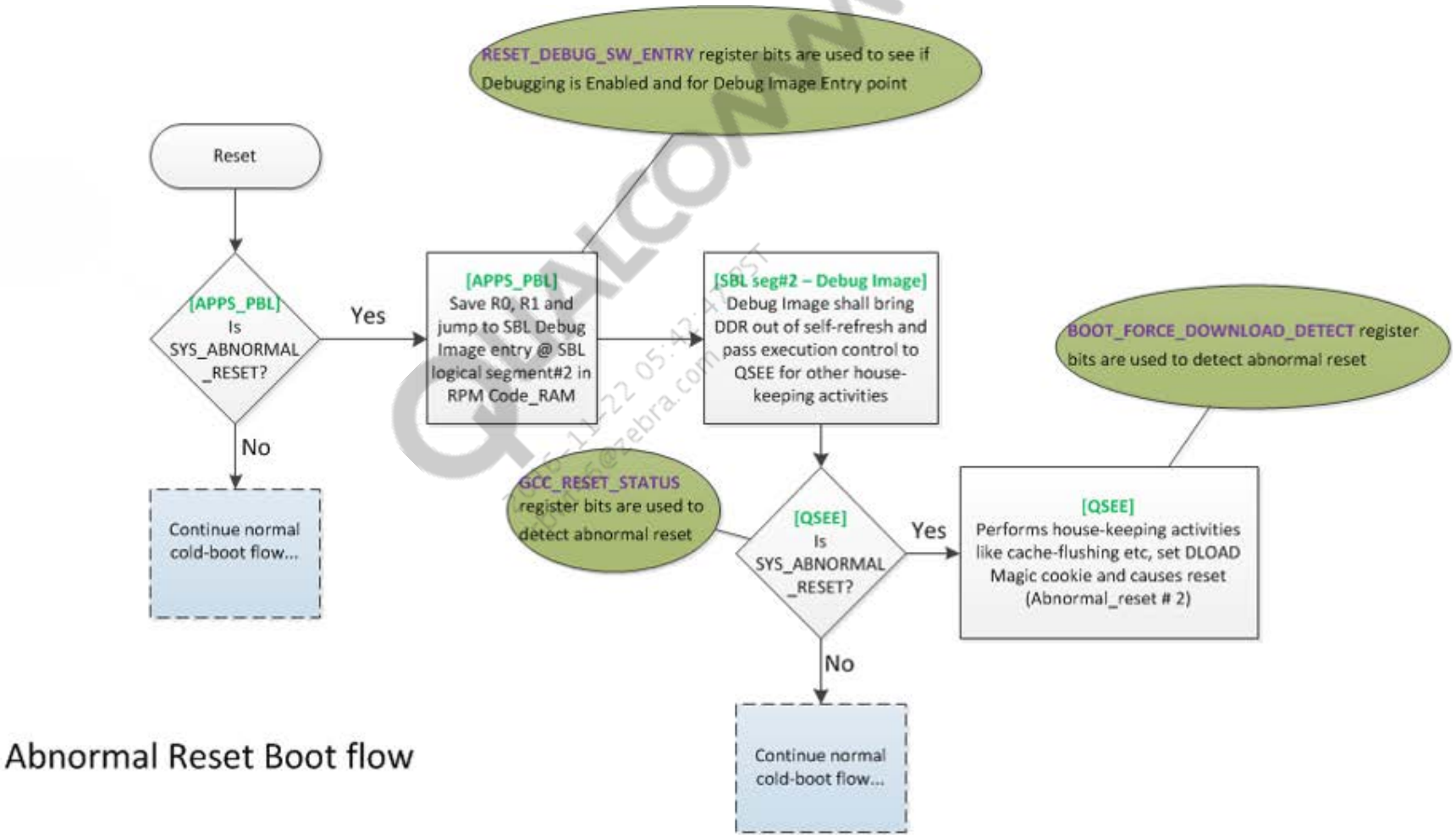
- MSM8952
  - No separate SDI image. A part (~20%) of SDI logic is in SBL1 RPM-code-ram segment and the remaining is in TZ image.
  - During cold boot, the SDI logic part of TZ enables the wdog-reset/SDI-path for non-production devices only.
  - DDR handling logic, during SDI, reuses SBL1 DDR driver segment in RPM CodeRAM. All the L2/secure cache flush logic, during SDI, is reused from the TZ image.
  - PBL on watchdog reset jumps to SBl1 segment#2 of RPM CodeRAM fixed offset.

Note: For information on general query to SDI watchdog reset debugging process, refer to Presentation: MSM8916/MSM8909 Linux Android Debug Overview (80-NL239-7)

.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Watchdog Debug Flow Diagram



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Watchdog Debug Flow

- Cold boot
  - APPS PBL → SBL1 → QSEE → RPM_FW* → APPSBL.

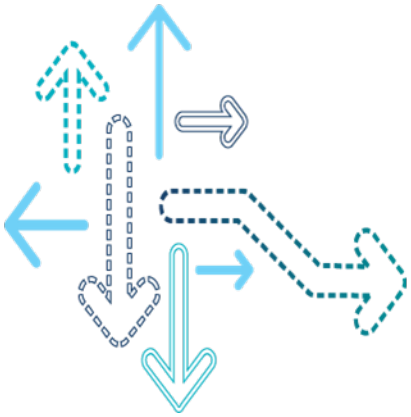
    *Logically, QSEE transfers the control to APPSBL (not RPM_FW).
- Watchdog reset (first reset)
  - APPS PBL → SBL1 segment #2, brings DDR out of self-refresh → QSEE
  - QSEE flushes the cache, saves on-chip debug buffer, before forcing second reset via PS_HOLD drop
- Second reset
  - APPS PBL → SBL1 → Sahara Download mode
  - SBL1 supports the default memory dump feature to dump the DDR via USB for non-production devices.

QUALCOMM®

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

MSM8952 Vs MSM8939



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

MSM8952 vs MSM8939

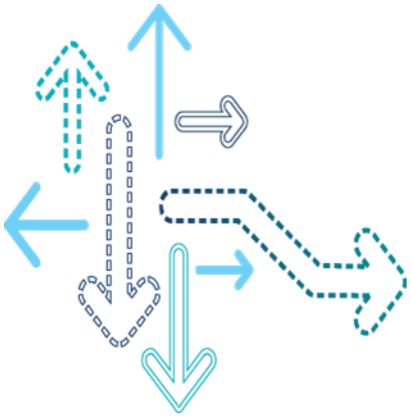
	MSM8952	MSM8939
CPU	4xA53 1.5 GHz to 1.7 GHz, 512 KB L2 4xA53 1.1 GHz to 1.2 GHz, 512 KB L2	4xA53 1.5 – 1.7 GHz 512 KB 4xA53 1.0 GHz 256 KB L2
Memory	1x933 MHz LPDDR3; eMMC5.1, SD 3.0	1x800 MHz LPDDR3; eMMC4.5, SD3.0
PBL	No major changes, PBL supports ELF load.	PBL supports ELF load.
SBL	frequency plan is changed for DDR is 933 MHz and Mass Storage mode is removed.	Frequency plan for DDR is 800 MHz.
DDR controller	BIMC controller and PHY as that used in MSM8939.	- NA -

Note: Refer MSM8952 Clock Plan (80-NV610-4) for detailed frequency information.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

SBL Debugging



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

SBL Debugging

- Run boot_debug.cmm scripts to debug the SBL image
  1. Open build <META ROOT>\common\t32\t32start.cmd
  2. Navigate to JTAG DAP mode > Podbus device chain > Power Trace Ethernet
  3. Select Cortex A53 Cluster1 Core 0
  4. Click **Start**
  5. Run <Boot build>\boot_images\core\boot\secboot3\scripts\ boot_debug.cmm
  6. Select the required option
- Manual debug
  1. Open build <META ROOT>\common\t32\t32start.cmd
  2. Navigate to JTAG DAP mode > Podbus device chain > Power Trace Ethernet
  3. Select Cortex A53 Core 0
  4. Click **Start**
  5. Run D.LOAD.ELF <Boot build>boot_images\core\boot\secboot3\hw\msm8952\sbl1\SBL1_ASIC.elf /nocode /noclear
  6. Insert break point at sbl1_main_ctl /o
  7. `Sys.up` and `go`.

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

References

Documents	
Qualcomm Technologies, Inc.	
Application Note: eMMC Partition Tables and Raw Programming	80-N4584-1
DDR SDRAM HAL APIs and Customization Reference Guide	80-N1218-1
Application Note: EEPROM Software Configuration Data Table (CDT)	80-N3411-1
Presentation: MSM8916/MSM8936/MSM8939 Boot Architecture Overview	80-NL239-1
Presentation: MSM8916/MSM8909 Linux Android Debug Overview	80-NL239-7

Qualcomm
2016-11-2 rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Acronyms

Acronyms	
Term	**Definition**
APSS PBL	Application Processor Primary Boot Loader
TZ	TrustZone
RPM FW	Resource Power Manager Firmware
HLOS	High-Level Operating System
Modem PBL	Modem Primary Boot Loader
MBA	Modem Boot Authenticator

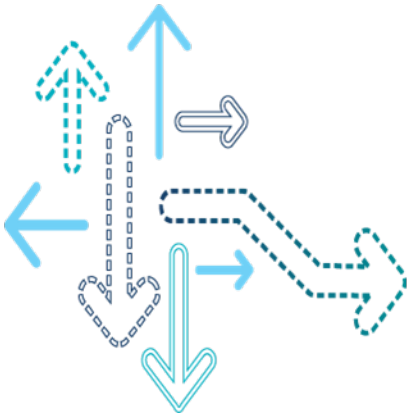
QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

QUALCOMM®
2016-11-22 05:42:47 PST
rbwt46@zebra.com

Questions?

https://support.cdmatech.com



Confidential and Proprietary – Qualcomm Technologies, Inc. | MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION