Introduction

This start-up guide explains R-Car H3/M3/M3N/E3/D3 Yocto recipe package files, the system environments, the make method of kernel, the operating of U-Boot and so on.

This product R-Car H3/M3/M3N/E3/D3 Yocto recipe is a basic package to operate built-in Linux and basic middleware on the R-Car H3/M3/E3/D3 System Evaluation Board. Please contact Renesas Electronics person who provided this product to you in case of questions.

1.   R-Car H3/M3/M3N/E3/D3 Linux BSP package files

This Yocto recipe will be taken

The U-Boot source code from:

　　git://github.com/renesas-rcar/u-boot.git

R-Car H3/M3/M3N/E3/D3 Linux source code from:

　　https://github.com/renesas-rcar/linux-bsp.git

1.1   Reference (R-Car H3/M3/M3N/E3/D3)

No.	Document name	Version
1	R-Car Series, 3rd Generation User's Manual: Hardware	Rev.1.50
2	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7795SIPB0011S	Rev.1.09
3	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7796SIPB0011S	Rev.0.04
4	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Hardware Manual	Rev.2.04
5	R-CarE3 System Evaluation Board Ebisu Hardware Manual RTP0RC77990SEB0010S	Rev.0.03
6	R-CarE3 System Evaluation Board Ebisu-4D (E3 board 4xDRAM) Hardware Manual	Rev.1.01
7	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Setup Manual	Rev.2.04
8	R-CarD3 System Evaluation Board Draak Setup Manual	Rev.1.20

Note: # is for latest version.

2.   Environmental Requirement

2.1   Setting of parts

Host PC and terminal software are necessary for the operation of this product. Furthermore, Ethernet cable is required to use NFS mount function. Please refer to Table 1.

Table 1 R-Car H3/M3/M3N/E3/D3 Linux BSP Environmental Requirement

Equipment	Explanation
Linux Host PC	Ubuntu 20.04 LTS (64bit) is recommended as OS. 32bit version is not supported. It is used as building and debugging environment. It is used as TFTP server and NFS server.
Windows Host PC	Windows 10 is recommended as OS. It is used as debugging environment. Terminal software and VCP driver are executed.
Terminal software	Please use following software. 1) Tera Term (Confirmed with Japanese version of Tera Term 4.88 Available at http://sourceforge.jp/projects/ttssh2)
VCP driver	Please install in Windows Host PC. Execute CP210xVCPInstaller_x86/x64.exe for install before connect. USB become virtual COM port on terminal software. Please connect to CN25 (Serial-USB Bridge CP2102) on H3/M3/E3/D3 System Evaluation Board. (Available at http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx)
TFTP server software	It is used when Hyper Flash is written by U-Boot or Image is downloaded.
NFS server software	It is used when File system is mounted by NFS.

Recommended Environment

The following shows a Recommended Environment.
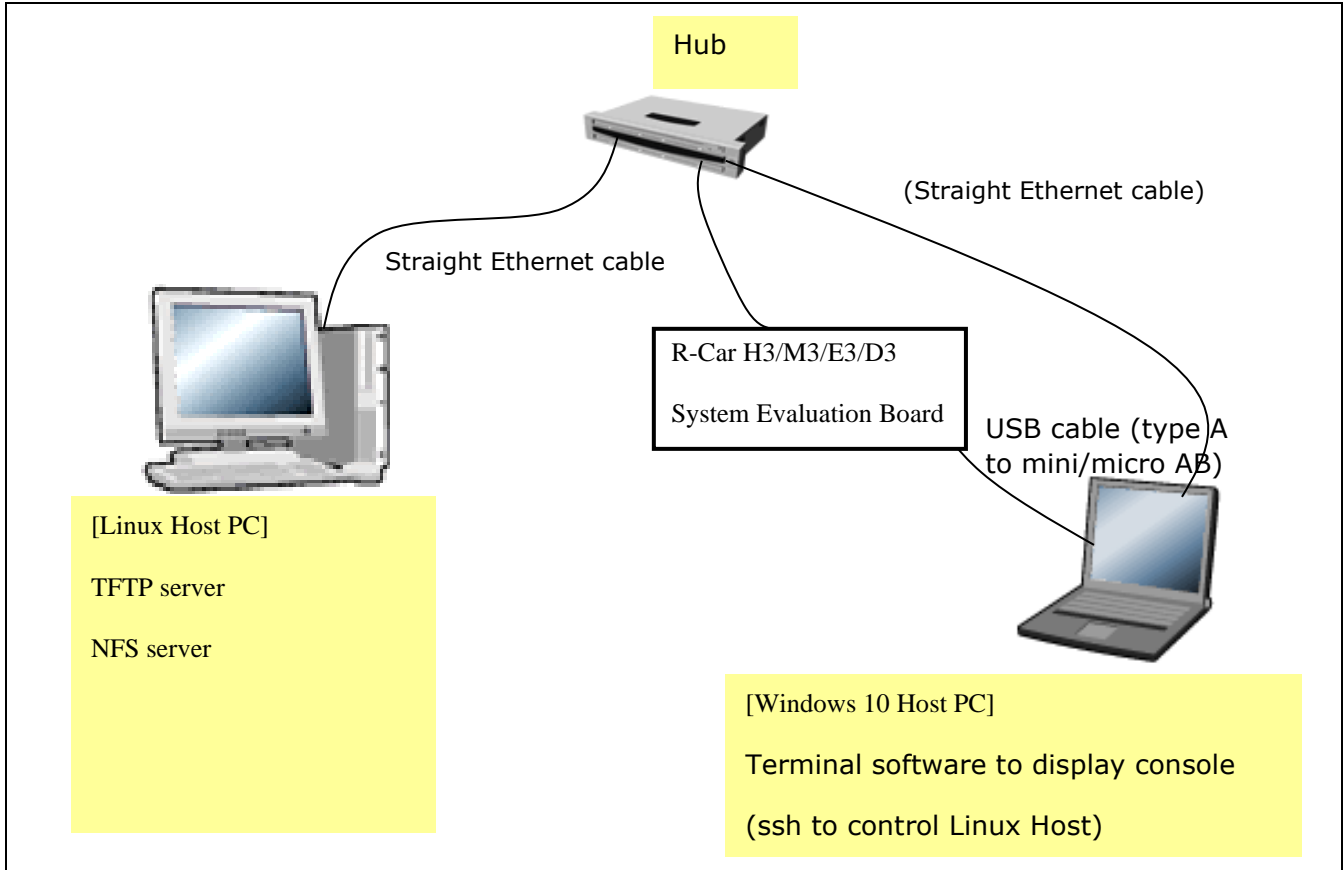
Figure 1. Recommended Environment for R-Car H3/M3/M3N/E3/D3 Linux BSP

Note) Functions in covered with () are optional.

2.2　　Setting of dip switch

The setting of R-Car H3/M3/E3/D3 System Evaluation Board's dip switches is shown in the following Table 2, Table 3, Table 4.

For Salvator-X board, please refer to "R-CarH3-SiP System Evaluation Board RTP0RC7795SIPB0011S (Salvator-X) Hardware Manual".

For Salvator-XS board, please refer to "R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board RTP0RC7795SIPB0012S (Salvator-XS) Setup Manual".

For Ebisu board, please refer to "R-CarE3 System Evaluation Board RTP0RC77990SEB0010S (Ebisu) Setup Manual".

For Ebisu-4D board, please refer to "R-CarE3 System Evaluation Board RTP0RC77990SEB0020S Ebisu 4xDRAM (Ebisu-4D) Setup Manual".

For Draak board, please refer to "R-CarD3 System Evaluation Board Draak RTP0RC77995SEB0010S Setup Manual".

Table 2 setting of Dip switches (R-Car H3 Ver.3.0, 2.0, M3 Ver.1.x, Ver.3.0, M3N Ver.1.1)

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	S	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW2	QSPI-B	S	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW3	QSPI-C	S		✔ (ON)	-	-	-	-	-	-
SW13	QSPI-D	C	✔	-		-	-	-	-	-
SW4	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW5	GPIO/PWM1	S			✔	-	-	-	-	-
SW6	GPIO/PWM2	S		✔		-	-	-	-	-
SW7	DDRBKUP-A	S	*1	✔		-	-	-	-	-
SW8	DDRBKUP-B	C	OFF	OFF	OFF	OFF	-	-	-	-
SW9	TRST#	S			✔	-	-	-	-	-
SW10	MODESW-A	C	ON	ON	ON	ON	ON	ON	**OFF**	**ON**
SW11	MODESW-B	C	OFF	ON	ON	ON*2	ON	ON	ON	ON
SW12	MODESW-C	C	OFF	ON	ON	ON	ON	ON	ON	ON
SW14	SSI78-M/S	S	✔	-		-	-	-	-	-
SW15	USB-SW	S	✔			-	-	-	-	-
SW16	SDHI0/DBG2-A	C	✔	-		-	-	-	-	-
JP2	SDHI0/DBG2-B	C	✔	-		-	-	-	-	-
SW28	VDDQVA_SD0	S	OFF	-	-	-	-	-	-	-
SW17	LVDS	C		✔	-	-	-	-	-	-
SW20	TACTSW0	C	Tactile SW	-	-	-	-	-	-	-
SW21	TACTSW1	C	Tactile SW	-	-	-	-	-	-	-
SW22	TACTSW2	C	Tactile SW	-	-	-	-	-	-	-
SW23	ACCSW	C		-	✔ (OFF)	-	-	-	-	-
SW27	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW29	MIPI-SW	S	ON	ON	-	-	-	-	-	-
SW30	PHYAD	C	OFF	OFF	-	-	-	-	-	-
SW31*3	GPIO/USB	S	ON	ON	OFF	OFF	OFF	OFF	-	-

C: Component side of the board, S: Solder side of the board

*1: From Yocto BSP v2.12.0, it is necessary to enable BKUP_TRG signal for Suspend to RAM, please set SW7 Pin-1.

*2: If debugger is connected, please set SW11-4 pin off.

*3: The SW31 exists in Salvator-XS board only.

Table 3 setting of Dip switches (R-Car E3 System Evaluation Board Ebisu, and Ebisu-4D)

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	C	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW2	QSPI-B	C	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW3	QSPI-C	C		✔ (ON)	-	-	-	-	-	-
SW31	QSPI-D	C		✔ (ON)	-	-	-	-	-	-
SW13	QSPI-E	C	✔	-		-	-	-	-	-
(SW42)	QSPI-F	C	---	---	-	-	-	-	-	-
SW4	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW10	MODESW-A	C	ON	OFF	OFF	ON	ON	ON	OFF	ON
SW11	MODESW-B	C	OFF	ON	OFF	ON*1	ON	ON	ON	ON
SW12	MODESW-C	C	ON	ON	ON	ON	ON	-	-	-
SW14	SSI34-M/S	C	✔	-		-	-	-	-	-
SW15	USB-SW	C	✔			-	-	-	-	-
SW16	SDHI0/JTAG2-A	C	✔	-		-	-	-	-	-
JP2	SDHI0/JTAG2-B	C	✔	-		-	-	-	-	-
SW28	VDDQ_SD0	C	OFF	-	-	-	-	-	-	-
SW17	LVDS0-DIR	C		✔	-	-	-	-	-	-
SW48	LVDS1-DIR	C		✔	-	-	-	-	-	-
SW44	LVDS0-SEL	C	ON	ON	-	-	-	-	-	-
SW47	LVDS1-SEL	C	✔ (OFF)		-	-	-	-	-	-
SW45	LVDS-LNK	C		✔ (ON)	-	-	-	-	-	-
SW50	LVDS-DK	C			✔	-	-	-	-	-
SW29	MIPI-SW	C		✔	-	-	-	-	-	-
SW49	PCIe/EXIO_A	C	✔ (OFF)		-	-	-	-	-	-
SW23	ACCSW	C		-	✔ (OFF)	-	-	-	-	-
SW36	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW20	PUSHSW0	C	Push SW	-	-	-	-	-	-	-
SW21	PUSHSW1	C	Push SW	-	-	-	-	-	-	-
SW22	PUSHSW2	C	Push SW	-	-	-	-	-	-	-
JP1	V_TEST	C	Open	-	-	-	-	-	-	-
JP6	VDDQ25_AVB0	C	✔	-	-	-	-	-	-	-
JP7	VDDQ_DDR	C	✔	-	-	-	-	-	-	-
(JP8)	PWR-A	C	-	-	---	-	---	-	-	-
(JP9)	PWR-B	C	---	-	-	-	-	-	-	-
(JP10)	PWR-C	C	---	-	-	-	-	-	-	-

C: Component side of the board, S: Solder side of the board

*1: If debugger is connected, please set SW11-4 pin off.

CONFIDENTIAL

Table 4 setting Dip switches (R-Car D3 System Evaluation Board Draak)

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW56	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW10	MODESW-A	C	ON	ON	ON	ON	OFF	ON	-	-
SW11	MODESW-B	C	ON	ON	ON	OFF	ON	OFF	-	-
SW12	MODESW-C	C	OFF	ON	OFF	OFF	-	-	-	-
SW1	QSPI-A	S	All OFF						-	-
SW2	QSPI-B	S	All OFF						-	-
SW3	QSPI-C	C		✓(ON)	-	-	-	-	-	-
SW13	QSPI-D	C	✓(.)	-		-	-	-	-	-
SW31	QSPI-E	C		✓(ON)	-	-	-	-	-	-
SW42	Not mounted	C	-	-	-	-	-	-	-	-
SW30	PHYAD	C	OFF	OFF	-	-	-	-	-	-
SW44	LVDS-A	C	ON	ON	-	-	-	-	-	-
SW47	LVDS-B	C		✓(ON)	-	-	-	-	-	-
SW17	LVDS-C	C		✓(ON)	-	-	-	-	-	-
SW48	LVDS-D	C		✓(ON)	-	-	-	-	-	-
SW45	LVDS-E	C		✓(ON)	-	-	-	-	-	-
SW49	VI4_HDMIIN-A	C	All ON				-	-	-	-
SW50	VI4_HDMIIN-B	C	All ON							
SW51	VI4_HDMIIN-C	C	All ON							
SW52	VI4_HDMIIN-D	C	All ON							
SW60	VI4_HDMIIN-E	C	✓	-		-	-	-	-	-
SW53	VI4_CVBSIN-A	C	All OFF							
SW54	VI4_CVBSIN-B	C	All OFF			-	-	-	-
SW57	TACTSW0	C	Tactile SW	-	-	-	-	-	-	-
SW58	TACTSW1	C	Tactile SW	-	-	-	-	-	-	-
SW59	TACTSW2	C	Tactile SW	-	-	-	-	-	-	-
SW23	ACCSW	C	-	-	-	-	-	-	-	-
SW55	PRESET#	C		-	-	-	-	-	-	-
JP1	VDDQ_MMC	C	✓	-		-	-	-	-	-
JP2	VDDQ25_AVB	C	✓	-		-	-	-	-	-
JP3	Not mounted	C	-	-	-	-	-	-	-	-
JP4	D1.8V_EN	C		✓	-	-	-	-	-	-
JP7	VDDQ_DDR	C		✓	-	-	-	-	-	-
JP8	VDD_EN	C		✓	-	-	-	-	-	-
JP6	D2.5V_EN	C		✓	-	-	-	-	-	-
JP5	D3.3V_EN	C		✓	-	-	-	-	-	-
JP9	PRESETCONT	C		✓	-	-	-	-	-	-
JP4	D1.8V_EN	C		✓	-	-	-	-	-	-

3. Building Instructions

You can build BSP by using Yocto Project. Please execute following steps in ${WORK} directory on Linux Host PC. Filesystem by making following instruction is the one for testing current BSP package in Renesas. Please note that Renesas has not been verified with any other build configuration or modified recipes except "core-image-weston" configuration which is based on upstream Yocto Project deliverables and some additional packages correspond to gstreamer.

Note) Renesas executed following instructions with clean ${WORK}/build directory. You may use wipe-sysroot and/or bitbake -c cleansstate to reflect modifications of configuration files for Recipe as in open source Yocto Project's standards, however Renesas strongly recommends to use recipe with clean ${WORK}/build directory for each configurations because there are some implicit dependency for header files exist to keep compatibility between application build scheme with/without proprietary software.

Step 1 installation of required commands

Ubuntu is used as Linux Host PC since Yocto Project Quick Start specifies Ubuntu as one of the distribution. In case of that you can install the required commands as follows.
Please refer to http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html for detail.

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat cpio python3 python3-pip python3-pexpect \
xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa \
libsdl1.2-dev pylint3 xterm
```

Note) There is a bitbake command in ${WORK}/poky/scripts/. Command path is available after step 6.

Note) When you use terminal interactions to build such as menuconfig under non-X terminal (ssh, etc.), please install "screen" command package to Host PC.

Note) Please set up user name and e-mail in Git. You can set up with 'git config --global'. Please refer to online manual for git command.

Note) In Renesas environment, Ubuntu version is 20.04 LTS and git version is 2.7.4.

Step 2 download of required files

Required files (poky) are downloaded by git clone.

```
$ cd ${WORK}
$ git clone git://git.yoctoproject.org/poky
$ git clone git://git.openembedded.org/meta-openembedded
$ git clone git://github.com/renesas-rcar/meta-renesas.git
```

Step 3 checkout

Please checkout available version of each git clone.

```
$ cd ${WORK}/poky
$ git checkout -b tmp 74b22db6879b388d700f61e08cb3f239cf940d18

$ cd ${WORK}/meta-openembedded
$ git checkout -b tmp 814eec96c2a29172da57a425a3609f8b6fcc6afe

$ cd ${WORK}/meta-renesas
$ git checkout -b tmp 8d64d3516353eefbd15015de23f64506527022c3
```

Note) tmp is a temporary name of a local branch. We can use checkout command without branch. Please note that HEAD refers directly to commit (detached HEAD).

3.1　In case of BSP + 3D Graphics + Multimedia package

When you use proprietary Multimedia and 3D Graphics software from Renesas, please execute as following steps. Otherwise please skip to section 3.2 or 3.3.

Step 4 copy proprietary software into recipe directory structure

To use licensed 3D graphics software and Multimedia package from Renesas, please copy deliverables of those software into recipe directory structure. Renesas provide shell script to copy those software.

```
Copy All Proprietary Software Packages to ${PKGS_DIR}:
$ mkdir ${PKGS_DIR}
$ cp <zip of Proprietary Software Package> ${PKGS_DIR}
Install them into recipe directory structure by shell script:
$ cd ${WORK}/meta-renesas
$ sh \
  meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh \
  ${PKGS_DIR}
```

Note) Subdirectory is not supporting in ${PKGS_DIR}. Please store all packages on the root of ${PKGS_DIR}.
Note) Please use regular alphanumeric file name ([A-Za-z0-9_] e.g.) for ${PKGS_DIR} due to restrictions of current copy script.

Step 5 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd ${WORK}
$ source poky/oe-init-build-env
```

Step 6 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp ${WORK}/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/poky-gcc/mmp/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x, ebisu.

Step 7 enable 3D Graphics and Multimedia package to install GSX modules

To enable 3D Graphics packages which use GSX, please overwrite local.conf as following instructions or modify local.conf as descriptions in ${WORK}/meta-renesas/meta-rcar-gen3/README.proprietary.md.

```
To install GSX kernel module:
Overwrite local.conf
$ cp conf/local-wayland.conf conf/local.conf (For Wayland)
```

Step 8 enable Multimedia package

Please modify configurations in ${WORK}/build/conf/local.conf by following instructions.

Note) Please refer to ${WORK}/meta-renesas/meta-rcar-gen3/README.proprietary.md.

The following standard multimedia packages are enabled

No.	Functions	Explanation
1	MMNGR	Memory manager driver & shared libraries
2	VSPM	VSP driver & FDP driver & shared libraries
3	VSP2	VSP2 driver
4	OMX	OMX common parts

To enable optional multimedia functions, please add DISTRO_FEATURES_append to ${WORK}/build/conf/local.conf as DISTRO_FEATURES_append = " <function name>".

Note) These configurations exist near the end of local.conf.

Note) DISTRO_FEATURES_append are commented out by the default. To enable functions, please uncomment it.

For example
[Disable]
#DISTRO_FEATURES_append = " h263dec_lib"

[Enable (default)]
DISTRO_FEATURES_append = " h263dec_lib"

The following list is package name to enable/disable as optional multimedia functions

No.	Function name	Default value	Explanation
1	h264dec_lib	Enable	H264 decoder library RTM8RC0000ZMD4LQ00JPL3E
2	h264enc_lib	Disable	H264 encoder library RTM8RC0000ZME1LQ00JPL3E
3	h265dec_lib	Disable	H265 decoder library RTM8RC0000ZMDALQ00JPL3E
4	mpeg2dec_lib	Disable	MPEG2 decoder library RTM8RC0000ZMD5LQ00JPL3E
5	mpeg4dec_lib	Disable	MPEG4 decoder library RTM8RC0000ZMD2LQ00JPL3E
6	vc1dec_lib	Disable	VC-1 decoder library RTM8RC0000ZMD3LQ00JPL3E
7	h263dec_lib	Disable	H263 decoder library RTM8RC0000ZMD4LQ00JPL3E
8	divxdec_lib	Disable	DivX decoder library RTM8RC0000ZMD7LQ00JPL3E
9	rvdec_lib	Disable	RealVideo decoder library RTM8RC0000ZMD6LQ00JPL3E
10	aaclcdec_lib	Enable	AAC-LC decoder library RTM8RC0000ZND1LQ00JPL3E
11	aaclcdec_mdw	Enable	AAC-LC 2ch decoder middleware library RTM8RC0000ZAD1LQ00JPL3E
12	aacpv2dec_lib	Disable	aacPlus V2 Decoder Library RTM8RC0000ZND2LQ00JPL3E
13	aacpv2dec_mdw	Disable	aacPlus V2 Decoder Middleware Library RTM8RC0000ZAD2LQ00JPL3E
14	mp3dec_lib	Disable	MP3 decoder library RTM8RC0000ZND3LQ00JPL3E
15	mp3dec_mdw	Disable	MP3 decoder middleware library RTM8RC0000ZAD3LQ00JPL3E
16	wmadec_lib	Disable	WMA decoder library RTM8RC0000ZND4LQ00JPL3E
17	wmadec_mdw	Disable	WMA decoder middleware library RTM8RC0000ZAD4LQ00JPL3E
18	aaclcenc_lib	Disable	AAC-LC encoder library RTM8RC0000ZNE1LQ00JPL3E
19	aaclcenc_mdw	Disable	AAC-LC encoder middleware library RTM8RC0000ZAE1LQ00JPL3E
20	flacdec_lib	Disable	FLAC decoder library RTM8RC0000ZND6LQ00JPL3E
21	alacdec_lib	Disable	ALAC decoder library RTM8RC0000ZND7LQ00JPL3E
22	dddec_lib	Disable	Dolby(R) Digital decoder library RTM0AC0000XADD5D30SL41C

CONFIDENTIAL

No.	Function name	Default value	Explanation
23	dddec_mdw	Disable	Dolby(R) Digital decoder middleware library RTM0AC0000ADDD5MZ1SL41C
24	vp8dec_lib	Disable	Media Component VP8 Decoder Library for Linux RTM8RC0000ZMD8LQ00JPL3E
25	vp8enc_lib	Disable	Media Component VP8 Encoder Library for Linux RTM8RC0000ZME8LQ00JPL3E
26	vp9dec_lib	Disable	Media Component VP9 Decoder Library for Linux RTM8RC0000ZMD9LQ00JPL3E
27	cmsbcm	Disable	Basic Color Management Middleware for Linux RTM8RC0000ZVC1LQ00JPL3E
28	cmsblc	Disable	CMM3 Backlight Control Middleware for Linux RTM8RC0000ZVC3LQ00JPL3E
29	cmsdgc	Disable	VSP2 Dynamic Gamma Correction Middleware for Linux RTM8RC0000ZVC2LQ00JPL3E
30	adsp	Disable	ADSP Interface for Linux RTM8RC0000ZNA1SS00JFL3E
31	avb	Disable	Ether-AVB
32	osal	Disable	Operating System Abstract Layer RTM8RC0000ZSAL2S00JPL3E
33	imr	Disable	Image Rendering Driver RTM8RC0000ZRRDSS00JPL3E

The following list is dependent package name

No.	Function name	Type Name	Dependent Packages
1	h264dec_lib	RTM8RC0000ZMD4LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
2	h264enc_lib	RTM8RC0000ZME1LQ00JPL3E	RTM8RC0000ZME0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
3	h265dec_lib	RTM8RC0000ZMDALQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
4	mpeg2dec_lib	RTM8RC0000ZMD5LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
5	mpeg4dec_lib	RTM8RC0000ZMD2LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
6	vc1dec_lib	RTM8RC0000ZMD3LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
7	h263dec_lib	RTM8RC0000ZMD4LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
8	divxdec_lib	RTM8RC0000ZMD7LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
9	rvdec_lib	RTM8RC0000ZMD6LQ00JPL3E	RTM8RC0000ZMD0LQ00JPL3E RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZMX0DQ00JFL3E
10	aaclcdec_lib	RTM8RC0000ZND1LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM8RC0000ZAD1LQ00JPL3E
11	aaclcdec_mdw	RTM8RC0000ZAD1LQ00JPL3E	-
12	aacpv2dec_lib	RTM8RC0000ZND2LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM8RC0000ZAD2LQ00JPL3E
13	aacpv2dec_mdw	RTM8RC0000ZAD2LQ00JPL3E	-
14	mp3dec_lib	RTM8RC0000ZND3LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM8RC0000ZAD3LQ00JPL3E
15	mp3dec_mdw	RTM8RC0000ZAD3LQ00JPL3E	-
16	wmadec_lib	RTM8RC0000ZND4LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM8RC0000ZAD4LQ00JPL3E
17	wmadec_mdw	RTM8RC0000ZAD4LQ00JPL3E	-
18	aaclcenc_lib	RTM8RC0000ZNE1LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM8RC0000ZAE1LQ00JPL3E

No.	Function name	Type Name	Dependent Packages
19	aaclcenc_mdw	RTM8RC0000ZAE1LQ00JPL3E	-
20	flacdec_lib	RTM8RC0000ZND6LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RCG3AFLDL4101ZDO
21	alacdec_lib	RTM8RC0000ZND7LQ00JPL3E	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RCG3AFLDL4101ZDO
22	dddec_lib	RTM0AC0000XADD5D30SL41C	RTM8RC0000ZMX0LQ00JPL3E RTM8RC0000ZNX0LQ00JPL3E RTM0AC0000ADDD5MZ1SL41C
23	dddec_mdw	RTM0AC0000ADDD5MZ1SL41C	-
24	cmsbcm	RTM8RC0000ZVC1LQ00JPL3E	-
25	cmsblc	RTM8RC0000ZVC3LQ00JPL3E	-
26	cmsdgc	RTM8RC0000ZVC2LQ00JPL3E	-
27	adsp	RTM8RC0000ZNA1SS00JFL3E	RTM8RC0000ZNA2DS00JFL3E RTM8RC0000ZNA3SS00JFL3E
28	avb	-	-
29	osal	RTM8RC0000ZSAL2S00JPL3E	
30	lmr	RTM8RC0000ZRRDSS00JPL3E	RTM8RC0000ZSAL2S00JPL3E

RENESAS

Step 9 select SoC

In case of Salvator X/XS board, please set SOC_FAMILY in ${WORK}/build/conf/local.conf as SOC_FAMILY = "<soc type name>".

```
[H3]
SOC_FAMILY = "r8a7795"
[M3]
SOC_FAMILY = "r8a7796"
[M3N]
SOC_FAMILY = "r8a77965"
```

In case of Ebisu board, there is no need to set SOC_FAMILY in ${WORK}/build/conf/local.conf because it was already added in machine config: ebisu.conf.

Step 10 building with bitbake

Please build as follows. The file system (core-image-weston-<supported board name>.tar.bz2) is created in ${WORK}/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x, ebisu.

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in ${WORK}/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in ${WORK}/poky/meta/conf/bitbake.conf.

```
$ cd ${WORK}/build
$ bitbake core-image-weston (for Wayland)
```

3.2   In case of BSP + 3D Graphics (without Multimedia package)

When you use licensed 3D Graphics software but will not use proprietary Multimedia software from Renesas, please execute as following steps. Otherwise please skip to section 3.3.

Step 4 copy proprietary software into recipe directory structure

To use licensed 3D graphics software and Multimedia package from Renesas, please copy deliverables of those software into recipe directory structure. Renesas provide shell script to copy those software.

```
Copy All Proprietary Software Packages to ${PKGS_DIR}:
$ mkdir ${PKGS_DIR}
$ cp <zip of Proprietary Software Package> ${PKGS_DIR}
Install them into recipe directory structure by shell script:
$ cd ${WORK}/meta-renesas
$ sh \
  meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh \
  ${PKGS_DIR}
```

Note) Subdirectory is not supporting in ${PKGS_DIR}. Please store all packages on the root of ${PKGS_DIR}.
Note) Please use regular alphanumeric file name ([A-Za-z0-9_] e.g.) for ${PKGS_DIR} due to restrictions of current copy script.

Step 5 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd ${WORK}
$ source poky/oe-init-build-env
```

Step 6 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp ${WORK}/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/poky-gcc/gfx-only/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x, ebisu.

Step 7 enable 3D Graphics package to install GSX modules

To enable 3D Graphics packages which use GSX, please overwrite local.conf as following instructions or modify local.conf as descriptions in ${WORK}/meta-renesas/meta-rcar-gen3/README.proprietary.md.

```
To install GSX kernel module:
Overwrite local.conf
$ cp conf/local-wayland.conf conf/local.conf (For Wayland)
```

Step 8 select SoC

In case of Salvator-X/XS board, please set SOC_FAMILY in ${WORK}/build/conf/local.conf as SOC_FAMILY = "<soc type name>".

```
[H3]
SOC_FAMILY = "r8a7795"
[M3]
SOC_FAMILY = "r8a7796"
[M3N]
SOC_FAMILY = "r8a77965"
```

In case of Ebisu, board, there is no need to set SOC_FAMILY in ${WORK}/build/conf/local.conf because it was already added in machine config: ebisu.conf.

Step 9 building with bitbake

Please build as follows. The file system (core-image-weston-<supported board name>.tar.bz2) is created in ${WORK}/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x, ebisu.

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in ${WORK}/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in ${WORK}/poky/meta/conf/bitbake.conf.

```
$ cd ${WORK}/build
$ bitbake core-image-weston (for Wayland)
```

3.3   In case of BSP Only

When you will not use neither proprietary Multimedia software nor licensed 3D Graphics software from Renesas, please execute as following steps.

Step 4 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd ${WORK}
$ source poky/oe-init-build-env
```

Step 5 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp ${WORK}/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/poky-gcc/bsp/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x, ebisu, draak.

Note) core-image-weston is not supported even though local-wayland.conf is provided.

Step 6 select SoC

In case of Salvator-X/XS board, please set SOC_FAMILY in ${WORK}/build/conf/local.conf as SOC_FAMILY = "<soc type name>".

```
[H3]
SOC_FAMILY = "r8a7795"
[M3]
SOC_FAMILY = "r8a7796"
[M3N]
SOC_FAMILY = "r8a77965"
```

In case of Ebisu, Draak board, there is no need to set SOC_FAMILY in ${WORK}/build/conf/local.conf because it was already added in machine config: ebisu.conf, draak.conf.

Step 7 building with bitbake

Please build as follows. The file system (core-image-minimal-<supported board name>.tar.bz2) is created in ${WORK}/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x, ebisu, draak.

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in ${WORK}/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in ${WORK}/poky/meta/conf/bitbake.conf.

```
$ cd ${WORK}/build
$ bitbake core-image-minimal
```

4.   Writing of IPL/Secure

4.1   Writing data

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0*	0xE6320000	0x000000	Loader(Boot parameter)
bl2-*	0xE6304000	0x040000	Loader
cert_header_sa6*	0xE6320000	0x180000	Loader(Certification)
bl31-*	0x44000000	0x1C0000	ARM Trusted Firmware
tee-*	0x44100000	0x200000	OP-Tee
u-boot-elf-*	0x50000000	0x640000	U-Boot

Note) U-Boot/IPL for Salvator-XS, Salvator-X, Ebisu, Ebisu-4D and Draak are as follows:

Board	SoC	DDR	U-Boot/IPL
Salvator-XS	H3	8GB (2GB x 4ch)	bl2-salvator-x-4x2g.srec bl31-salvator-x-4x2g.srec bootparam_sa0-4x2g.srec cert_header_sa6-4x2g.srec tee-salvator-x.srec u-boot-elf-salvator-x.srec
		4GB (1GB x 4ch)	bl2-salvator-x.srec bl31-salvator-x.srec bootparam_sa0.srec cert_header_sa6.srec tee-salvator-x.srec u-boot-elf-salvator-x.srec
	M3	4GB	
		8GB (4GB x 2ch)	bl2-salvator-x.srec bl31-salvator-x.srec bootparam_sa0.srec cert_header_sa6.srec tee-salvator-x.srec u-boot-elf-salvator-x.srec
	M3N	2GB	bl2-salvator-x.srec bl31-salvator-x.srec bootparam_sa0.srec cert_header_sa6.srec tee-salvator-x.srec u-boot-elf-salvator-x.srec
Salvator-X	H3	4GB (1GB x 4ch)	bl2-salvator-x.srec bl31-salvator-x.srec bootparam_sa0.srec cert_header_sa6.srec tee-salvator-x.srec u-boot-elf-salvator-x.srec
	M3	4GB	
	M3N	2GB	

Ebisu	E3	1GB	bl2-ebisu.srec bl31-ebisu.srec bootparam_sa0.srec cert_header_sa6.srec tee-ebisu.srec u-boot-elf-ebisu.srec
Ebisu-4D	E3	2GB	bl2-ebisu-4d.srec bl31-ebisu-4d.srec bootparam_sa0-4d.srec cert_header_sa6-4d.srec tee-ebisu.srec u-boot-elf-ebisu.srec
Draak	D3	512MB	bootparam_sa0.srec bl2-draak.srec cert_header_sa6.srec bl31-draak.srec tee-draak.srec u-boot-elf-draak.srec

Note) Please refer to Chapter 5 for U-Boot.

4.2   Dip-Switch

4.2.1 For H3/M3/M3N/E3

a)   QSPI Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	ON	ON	ON	ON	ON	ON	-	-
SW2	QSPI-B	ON	ON	ON	ON	ON	ON	-	-
SW3	QSPI-C	set(OFF)		-	-	-	-	-	-
SW10	MODESW-A	Refer to 2.2				ON	OFF	ON	ON
SW31*1	QSPI-D	OFF	-	-	-	-	-	-	-

*1: Only E3 System Evaluation Board needs to set SW31.

b)   Hyper Flash Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW2	QSPI-B	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW3	QSPI-C		Set(ON)	-	-	-	-	-	-
SW10	MODESW-A	Refer to 2.2							
SW31*1	QSPI-D	ON	-	-	-	-	-	-	-

*1: Only E3 System Evaluation Board needs to set SW31.

4.2.2 For D3

a)   QSPI Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6
SW1	QSPI-A	ON	ON	ON	ON	ON	ON
SW2	QSPI-B	ON	ON	ON	ON	ON	ON
SW3	QSPI-C	OFF	-	-	-	-	-
SW13	QSPI-D		✓●	-	-	-	-
SW31	QSPI-E	OFF	-	-	-	-	-
SW10	MODESW- A	ON	ON	ON	OFF	ON	ON

b)   Hyper Flash Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6
SW1	QSPI-A	OFF	OFF	OFF	OFF	OFF	OFF
SW2	QSPI-B	OFF	OFF	OFF	OFF	OFF	OFF
SW3	QSPI-C	ON	-	-	-	-	-
SW13	QSPI-D		✓●	-	-	-	-
SW31	QSPI-E	ON	-	-	-	-	-
SW10	MODESW- A	ON	ON	ON	ON	OFF	ON

4.3   How to write

Please connect R-Car H3/M3/E3/D3 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC.

In case of R-Car H3/M3/M3N/E3/D3, Ethernet connector is CN22, serial connector is CN25.

Step 1 connect cable

Connect USB Host connector of Windows Host PC that is virtual COM port to CN25 of R-Car H3/M3/E3/D3 System Evaluation Board with USB cable for displaying console.

Step 2 setting the terminal software

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port.

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none.

Step 3 write data file to Hyper Flash

A file is written in Hyper Flash in the following procedures.
• Set dip switch "QSPI Mode".
• Reset board then start mini monitor.

- Set dip switch (Select Hyper Flash)

  SW1: ALL OFF

  SW2: ALL OFF

  SW3: ON

  SW31: ON (R-Car M3 Kriek, R-Car E3 and R-Car D3)

  Note) SW10 does not need to change.

- Execute xls2 command (load program to flash).

```
R-Car Gen3 Sample Loader V3.03 2017.08.25
 For Salvator , Kriek , and StarterKit.
 Board Judge       : Used Board-ID
 DDR_Init          : boardcnf[7] Salvator / Starter Kit (H3SIP_VER2.0)
 INITIAL SETTING : Salvator-XS / R-Car H3 ES3.0
 CPU               : AArch64 CA57
 DRAM              : LPDDR4 DDR3200
 DEVICE            : QSPI Flash(S25FS128) at 40MHz DMA
 BOOT              : Normal Boot
 BACKUP            : DDR Cold Boot
 jump to 0xE6330000

 R-Car Gen3 MiniMonitor V3.03 2017.08.25
  Work Memory      : SystemRAM
  Board Name       : Salvator-XS
  Product Code     : R-Car H3 ES3.0


 >xls2
 ===== Qspi/HyperFlash writing of Gen3 Board Command =============
 Load Program to Spiflash
 Writes to any of SPI address.
 Please select,FlashMemory.
    1 : QspiFlash         (U5 : S25FS128S)
    2 : QspiFlash Board (CN3: S25FL512S)
    3 : HyperFlash       (SiP internal)
  Select (1-3)>
```

- Select HyperFlash. Input "3".
- After "SW1 SW2 All OFF! Setting OK? (Push Y key)" is displayed, input "y".
- After "SW3 ON! Setting OK? (Push Y key)" is displayed, input "y".
- After "SW31 ON!   Setting OK? (Push Y key) is displayed, input "y". (R-Car E3 only)
- After "SW31 ON!   Setting OK? (Push Y key) [In the case of Kriek Board]" is displayed, input "y". (R-Car M3 Kriek only)
- After "Please Input Program Top Address" is displayed, input Program Top Address in 4.1 and "Enter".
- After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address in 4.1 and "Enter".
- After "Please send ! ('.' & CR stop load)" is displayed, In case of Tera Term, transmit srec file in 4.1 by "File -> Send file…".
- If there are some data in writing area, "SPI Data Clear(H'FF) Check :H'00000000-0003FFFF Clear OK?(y/n)" is displayed. Then input "y".
- After "SAVE SPI-FLASH .... complete!" is displayed, the prompt returns. It means finish.
- Please repeat the xls2 command, if other files are written.
- Power OFF.
- Set dip switch to "Hyper Flash Mode".
  Note) SW1, SW2, and SW3 are already Hyper Flash mode. Please set SW10.

4.4   IPL/Secure write

Please write the file described in Chapter 4.1 to Hyper Flash.

The data file is stored in the ${WORK}/build/tmp/deploy/images/<board_name> directory.

5. Confirm starting of U-Boot and Linux

Please connect R-Car H3/M3/E3/D3 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC with TFTP and NFS server as Figure 1. Then please confirm normal starting of U-Boot and Linux with following step. Please refer to 2.2 for dip switch setting.

In case of R-Car H3/M3/M3N/E3/D3 , Ethernet connector is CN22, serial connector is CN25, D-Sub connector is CN15.

Step 1 setting Linux Host PC

Please install TFTP server and NFS server in Linux Host PC with apt-get command and so on. Please set /etc/xinetd.d/tftp of TFTP server and /etc/exports of NFS server according to your environment.

Step 2 connect cable

Connect USB Host connector of Windows Host PC that is virtual COM port to CN25 of R-Car H3/M3/E3/D3 System Evaluation Board with USB cable for displaying console.

Step 3 setting the terminal software

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port.

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none.

Step 4 write U-Boot to Hyper Flash

Filename	Program Top Address	Flash Save Address	Description
u-boot-elf-*	0x50000000	0x640000	U-Boot

Note) *: salvator-x.srec, ebisu.srec, and draak.srec.

The data file is stored in the ${WORK}/build/tmp/deploy/images/<board_name> directory.
Refer to Chapter 4.3 Step3 for writing procedure.

Step 5 set U-Boot environment variables

Please refer to 2.2 for dip switch setting.

Please start U-Boot by board reset. Please set and save environment variable as follows.

```
=> setenv ethaddr xx:xx:xx:xx:xx:xx
=> setenv ipaddr 192.168.0.20
=> setenv serverip 192.168.0.1
=> setenv bootcmd 'tftp 0x48080000 Image;tftp 0x48000000
<Device_Tree>;booti 0x48080000 - 0x48000000'
```

Note) For R-Car H3 (SOC_FAMILY r8a7795), the <Device_Tree> are as follow:

- r8a77950-salvator-x.dtb: for R-Car H3 Ver.1.0 on Salvator-X board.

- r8a77951-salvator-x.dtb: for R-Car H3 Ver.2.0 on Salvator-X board.

- r8a77951-salvator-xs.dtb: for R-Car H3 Ver.2.0 and Ver.3.0 on Salvator-XS board.

- r8a779m1-salvator-xs.dtb: for R-Car H3e 2GHz on Salvator-XS board.

- RCar H3 Ver.3.0 supports Salvator-XS board only.

For R-Car M3 (SOC_FAMILY r8a7796), the <Device_Tree> are as follow:

- r8a77960-salvator-x.dtb: for R-Car M3 Ver.1.x on Salvator-X board.

- r8a77960-salvator-xs.dtb: for R-Car M3 Ver.1.x on Salvator-XS board.

- r8a77961-salvator-xs.dtb: for R-Car M3 Ver.3.0 on Salvator-XS board.

- r8a779m3-salvator-xs.dtb: for R-Car M3e 2GHz on Salvator-XS board.

For R-Car M3N (SOC_FAMILY r8a77965), the <Device_Tree> are as follow:

- r8a77965-salvator-x.dtb: for R-Car M3N Ver.1.1 on Salvator-X board.

- r8a77965-salvator-xs.dtb: for R-Car M3N Ver.1.1 on Salvator-XS board.

- r8a779m5-salvator-xs.dtb: for R-Car M3Ne 2GHz on Salvator-XS board.

For R-Car E3 (SOC_FAMILY r8a77990), the <Device_Tree> are as follow:

- r8a77990-es10-ebisu.dtb: for R-Car E3 Ver.1.0 on Ebisu board.

- r8a77990-es10-ebisu-4d.dtb: for R-Car E3 Ver.1.0 on Ebisu-4D board.

- r8a77990-ebisu.dtb: for R-Car E3 Ver.1.1 on Ebisu board.

- r8a77990-ebisu-4d.dtb: for R-Car E3 Ver.1.1 on Ebisu-4D board.

 For R-Car D3 (SOC_FAMILY r8a77995), the <Device_Tree> are as follow:

- r8a77995-draak.dtb: for R-Car D3 Ver.1.x on Draak board.

Step 6 change the bootargs by U-Boot

To change bootargs which passed to the kernel in boot sequence, please modify it by "setenv bootargs" command of U-Boot.

```
=> setenv bootargs 'rw root=/dev/nfs nfsroot=192.168.0.1:/export/rfs
ip=192.168.0.20'
```

Note) When you disable big.LITTLE archtecture in R-Car H3/M3, please add "maxcpus" in bootargs as follow:

- R-Car H3 : "maxcpus=4"

-     R-Car M3 : "maxcpus=2"

It means that Cortex-A53 does not boot in R-Car H3/M3. For additional information, please refer to User's manual of Kernel.

Step 7 save environment variables

```
=> saveenv
```

Step 8 set file system

Please extract file system (core-image- weston-<supported board name>.tar.bz2). Please export /export directory of NFS server.

```
$ mkdir /export/rfs
$ cd /export/rfs
$ sudo tar xvf core-image-weston(minimal)-<supported board name>.tar.bz2
```

Note) <supported board name> is the following: salvator-x, ebisu, draak.

Step 9 start Linux

After board reset, U-Boot is started. After countdown, Linux boot messages are displayed. Please confirm login prompt after Linux boot messages.

Note) When MAC Address is rewritten, it is necessary to reset.

Note) In core-image-weston, please connect input device (keyboard, mouse) to USB port (salvator-x: CN10, ebisu, draak: CN9). In R-Car M3/M3N, only CN10 of USB2_1 port can be used.

6. Exporting Toolchains

Please refer Documents from Yocto Project to export Toolchains such as
https://docs.yoctoproject.org/3.1.11/sdk-manual/sdk-manual.html.
And please use build target of bitbake as "core-image-weston-sdk -c populate_sdk" to generate package.

Note) When you use "ld" directly but not via gcc (in case of building Kernel, Driver or U-Boot), please disable LDFLAGS with 'unset LDFLAGS'. Furthermore, in kernel build, 'make menuconfig' occurs error by ncurses. In this case, please set PKG_CONFIG_PATH and disable PKG_CONFIG_SYSROOT_DIR.

```
$ export PKG_CONFIG_PATH=$OECORE_NATIVE_SYSROOT/usr/lib/pkgconfig
$ unset PKG_CONFIG_SYSROOT_DIR
```

Note) Please do not use same shell environment to other compilation/debugging purpose (also make menuconfig of linux kernel, e.g.) but cross compilation for R-Car H3/M3/M3N/E3/D3 which shell environment with "source" command to setup environment variables for the SDK. Because some environment variables for cross compilation interferes execution of other tools on the same shell environment.

Example of instruction:

In following examples, it's assumed that it's already extracted and prepared recipe environment such as in the instructions of Section 3 (must done just before execution of bitbake, at least). You may reuse ${WORK}/build while you reuse same configuration after executing bitbake as in Section 3 for this purpose.

Step 1 configure architectures of Host PC which are installed this toolchain

Please modify SDKMACHINE description on ${WORK}/build/conf/local.conf.

```
On ${WORK}/build/conf/local.conf

# This variable specified the architecture to build SDK/ADT items for and means
# you can build the SDK packages for architectures other than the machine you are
# running the build on (i.e. building i686 packages on an x86_64 host.
# Supported values are i686 and x86_64
#SDKMACHINE ?= "i686"
SDKMACHINE ?= "x86_64"
```

Note) 32bit Ubuntu 14.04 is not supported.

CONFIDENTIAL

Step 2 building toolchain package with bitbake

```
$ cd ${WORK}/build
$ bitbake core-image-weston-sdk -c populate_sdk
$ cp tmp/deploy/sdk/ poky-glibc-x86_64-core-image-weston-sdk-aarch64-
salvator-x-toolchain-3.1.11.sh (shared dir. where able to access from each
Host PCs)
```

Note) Please perform "bitbake core-image-minimal -c populate_sdk" in BSP Only.

Step 3 Install toolchain on each Host PCs

```
$ sudo (shared dir. where able to access from each Host PCs)/ poky-glibc-
x86_64-core-image-weston-sdk-aarch64-salvator-x-toolchain-3.1.11.sh

 [sudo] password for (INSTALL person): (password of your account)
Enter target directory for SDK (default: /opt/poky/3.1.11): (just a return)
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

Step 4 setup environment variables for each compilation on each Host PCs

Please setup environment variables as follows or integrate set-up sequence into your build script or Makefile.

```
$ cd (Your working directory)
$ source /opt/poky/3.1.11/environment-setup-aarch64-poky-linux
$ export LDFLAGS=""
$ ${CC} (Your source code).c .....
```

RENESAS

7.　Memory map

Following Figure 2 to Figure show memory map of this R-Car H3/M3/M3N/E3/D3 Linux BSP package.

Note)

- The volume of SDRAM is total 8GB (R-Car H3 Ver.3.0 8GB, R-Car M3 Ver.3.0), 4GB (R-Car H3 Ver.3.0 4GB/Ver.2.0, M3 Ver.1.x), 2GB (R-Car M3N/R-Car E3 System Evaluation Board Ebisu-4D), 1GB (R-Car E3 System Evaluation Board Ebisu), 512MB (R-Car D3 Evaluation Board Draak).

- 2GB from 0x00_4000_0000 to 0x00_BFFF_FFFF is a shadow area from 0x04_0000_0000 to 0x04_7FFF_FFFF (R-Car H3/M3/M3N/E3)

- 512MB from 0x00_4000_0000 to 0x00_5FFF_FFFF is a shadow area from 0x04_0000_0000 to 0x04_1FFF_FFFF (R-Car D3)

- The following regions are used as a secure region. It doesn't allow U-Boot and kernel to access those regions.

  ➢ 63MB from 0x00_43F0_0000 to 0x00_47DF_FFFF in SDRAM

  ➢ 16KB from 0x00_E630_0000 to 0x00_E630_3FFF in System RAM

- 512MB from 0x00_4000_0000 to 0x00_5FFF_FFFF is a shadow area from 0x04_0000_0000 to 0x04_1FFF_FFFF (R-Car D3)

- **In case the configuration of BSP + 3D Graphics + Multimedia package, it doesn't allow to store any data in "CMA for Lossy comp" (default: 0x00_5400_0000 - 0x00_56FF_FFFF) region which is for media playback before kernel boots up. Any data stored in this region are read through the decompression module in AXI-Bus, so a normal data (not a decoded frame) will be corrupted.**
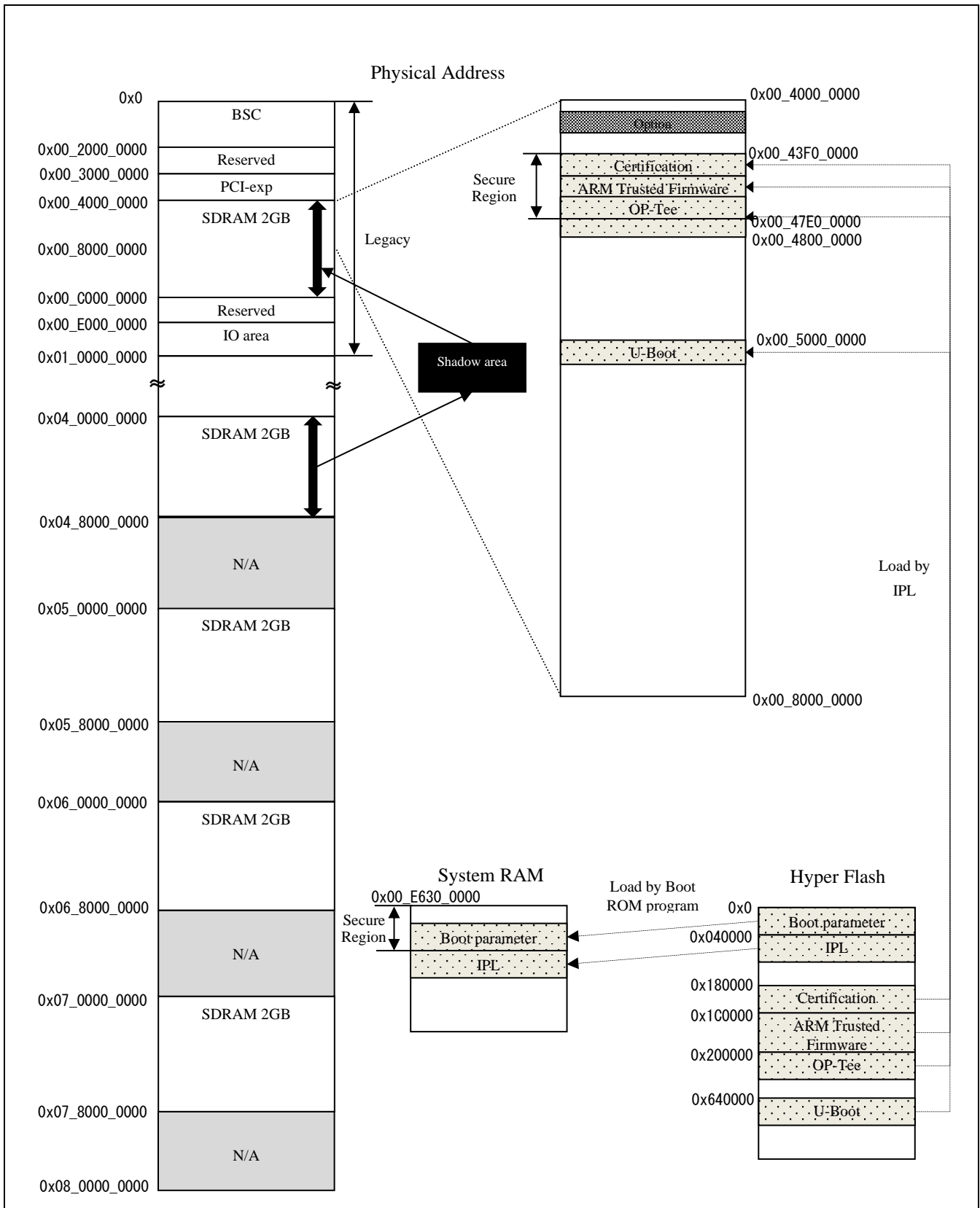
Figure 2. H3 Ver.3.0 8GB memory map (Boot)
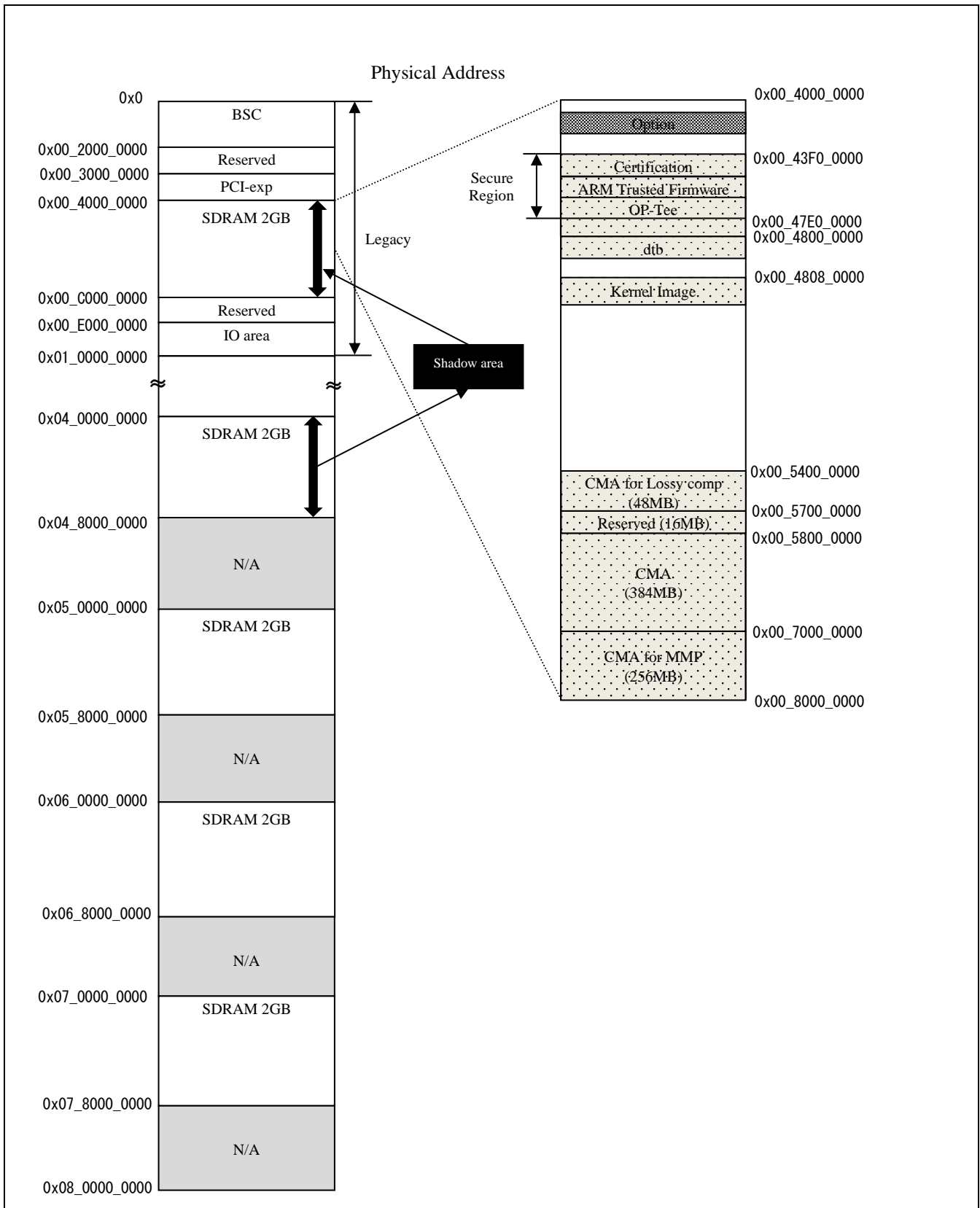
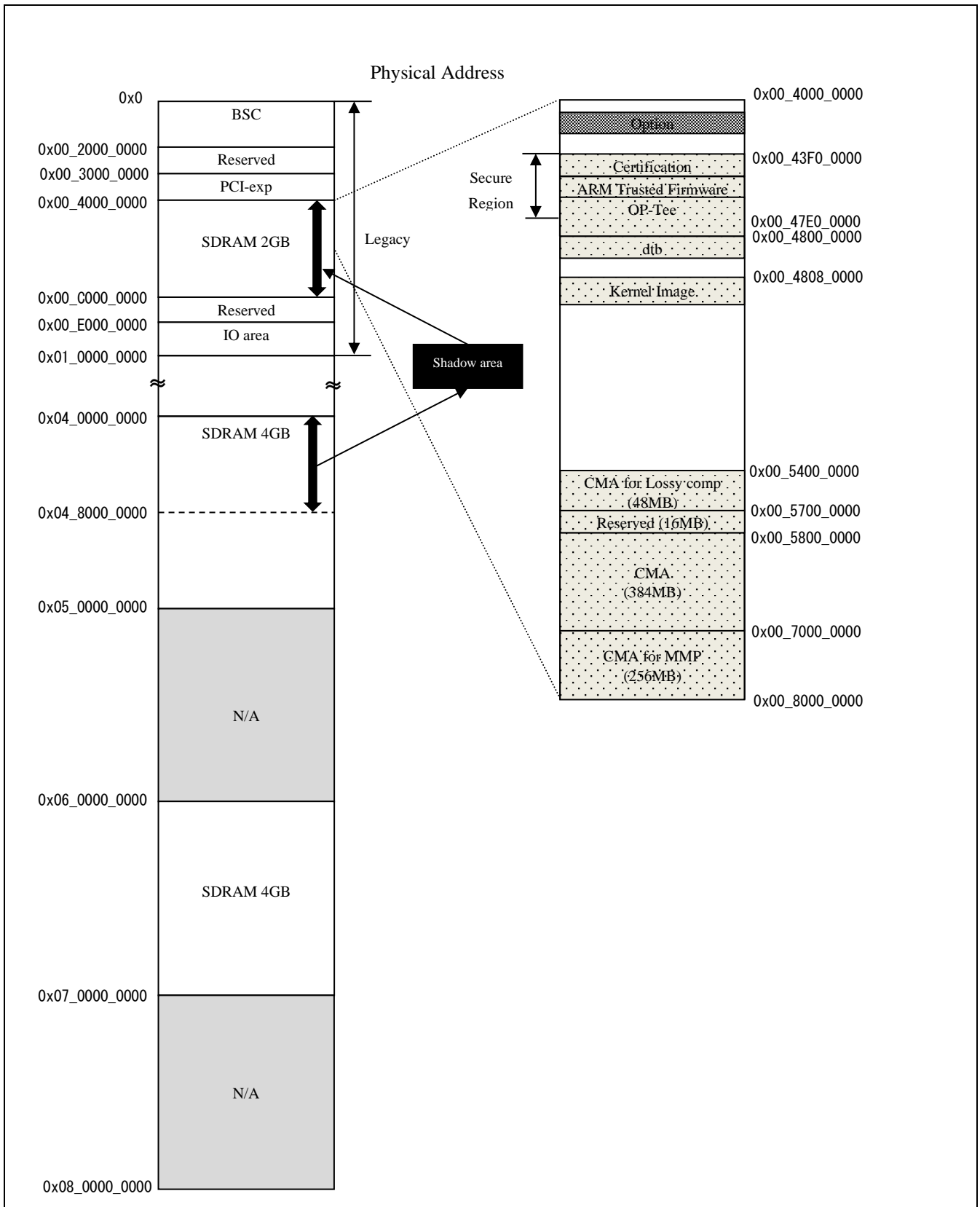**Figure 3. H3 Ver.3.0 8GB memory map (Linux)**
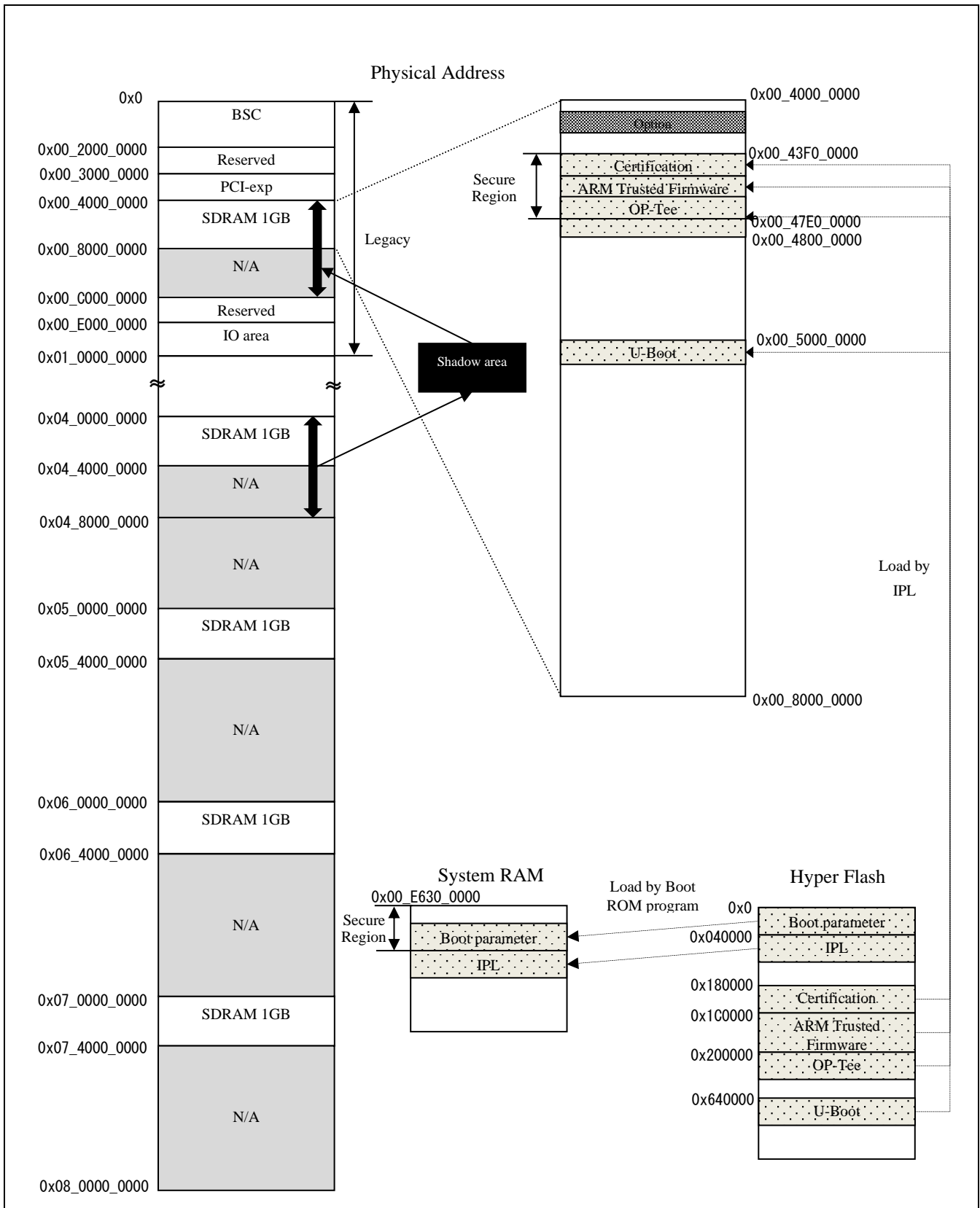
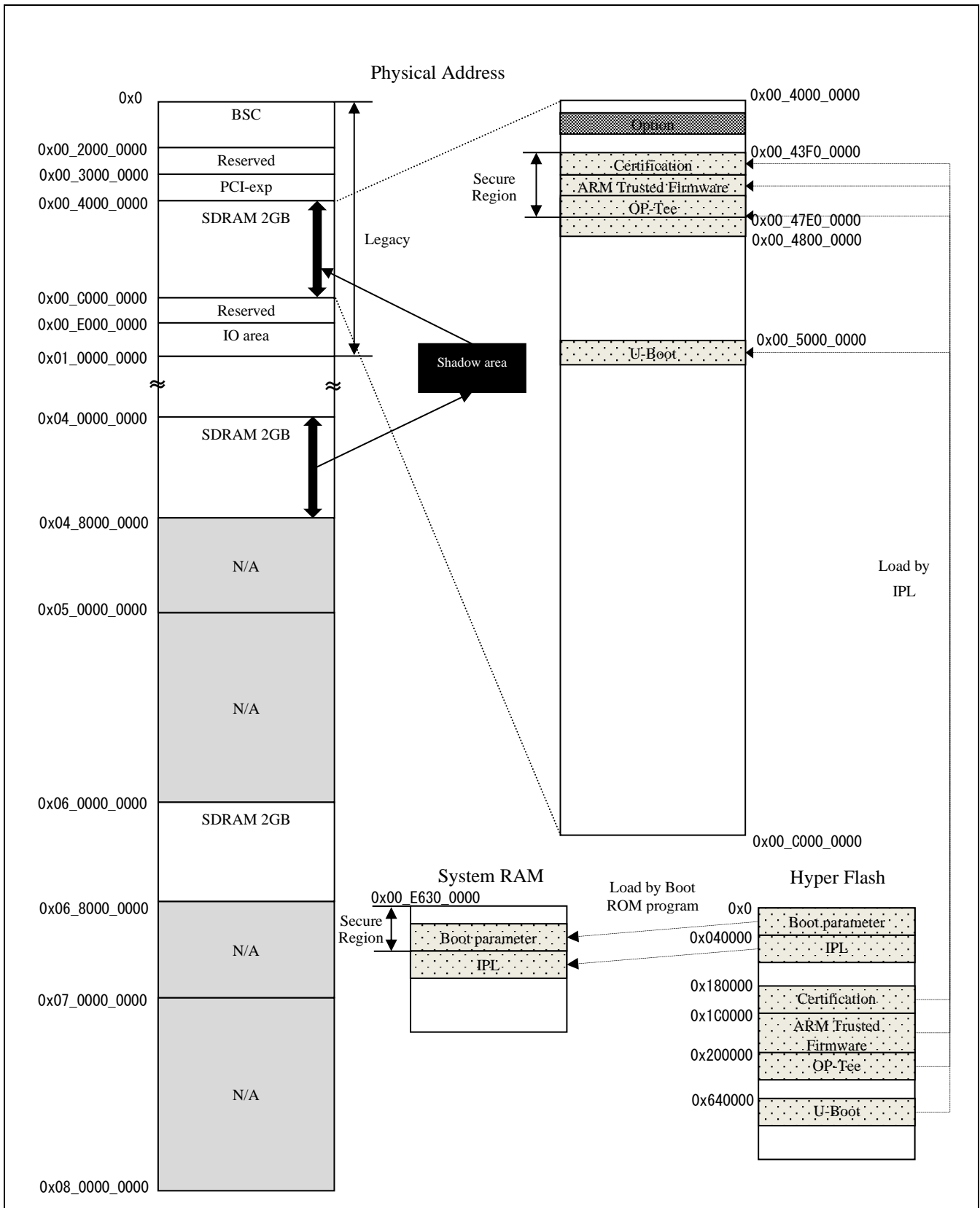**Figure 4. H3 Ver.2.0/Ver.3.0 4GB memory map (Boot)**
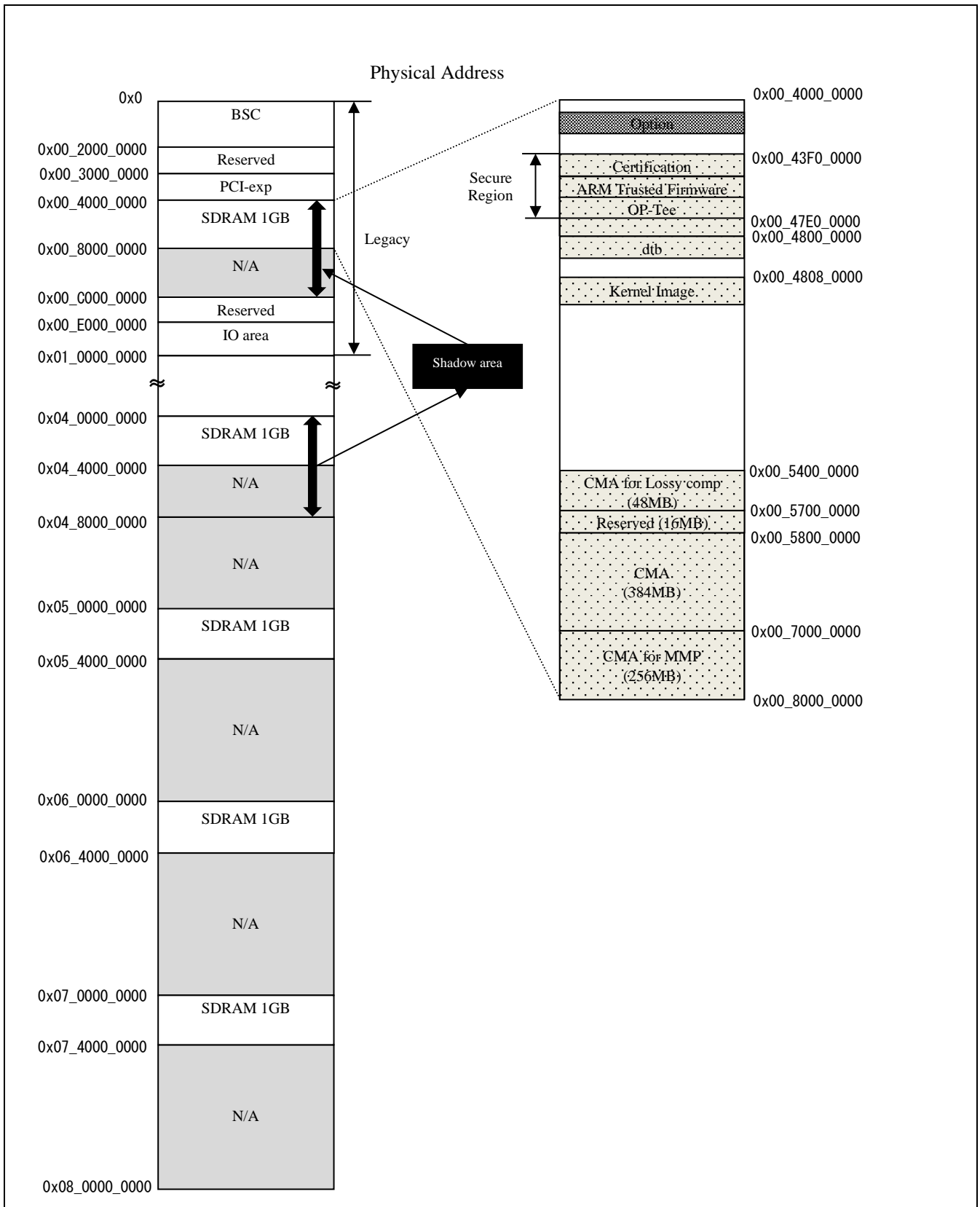
Figure 5. H3 Ver.2.0/Ver.3.0 4GB memory map (Linux)
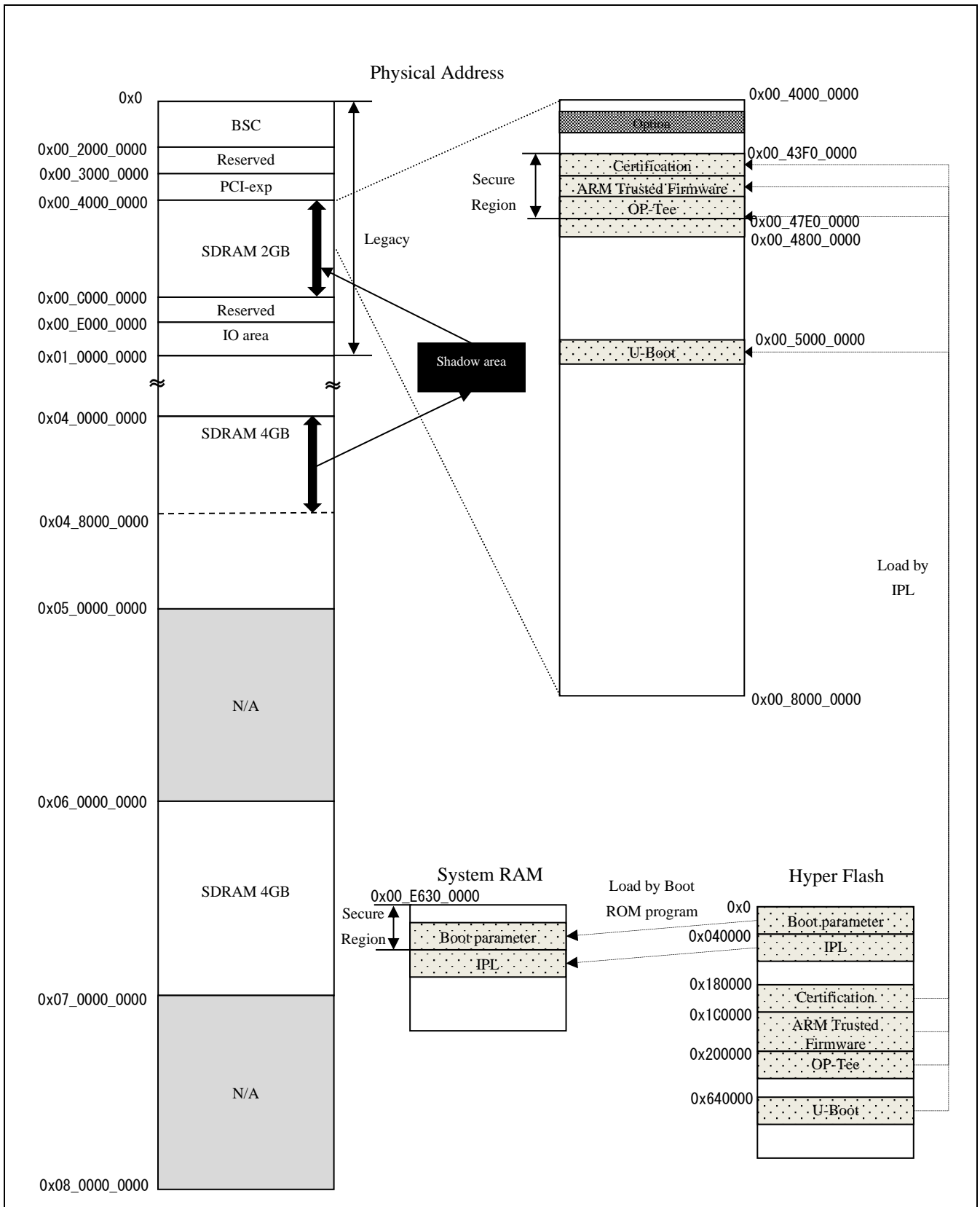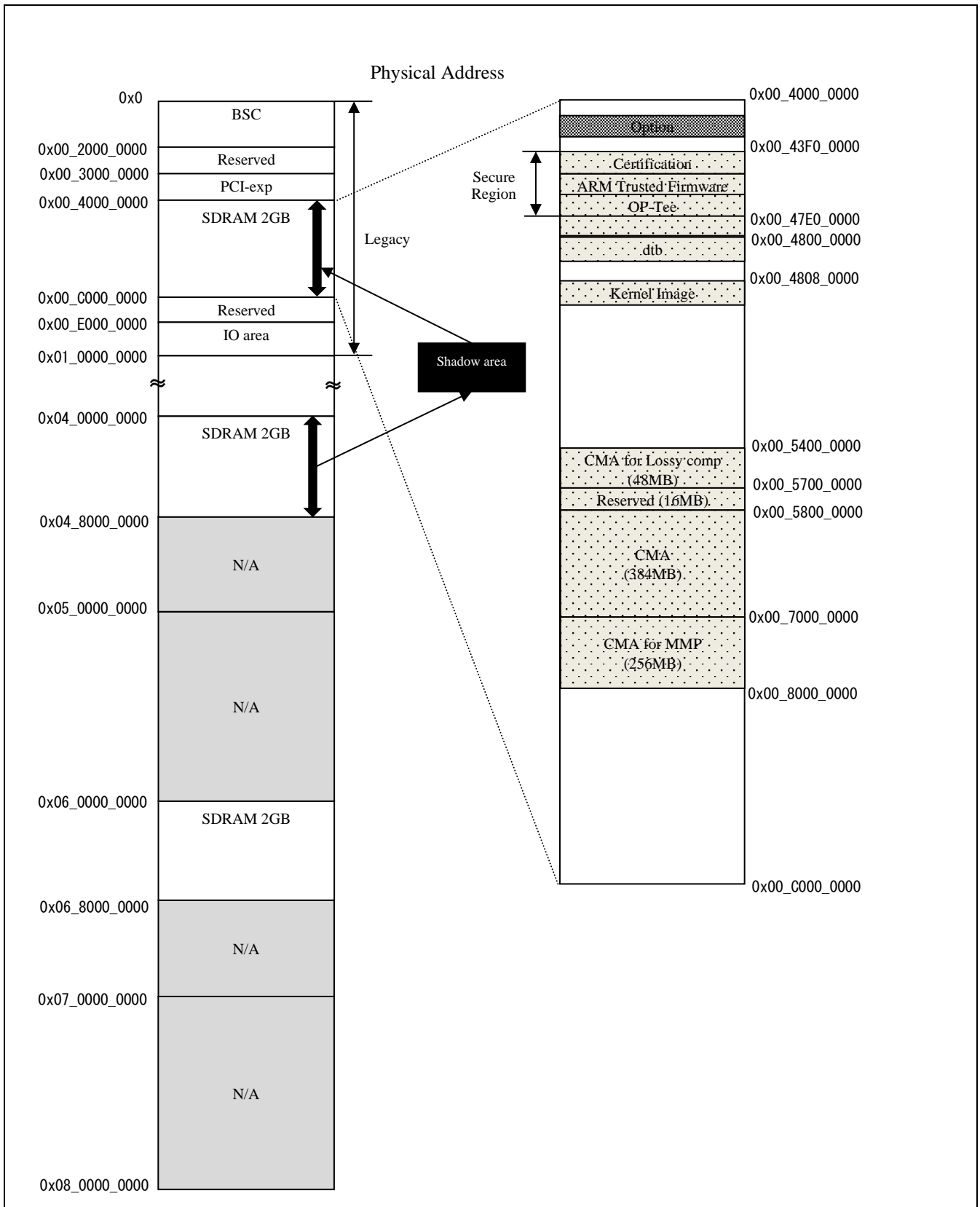
Figure 6. M3 Ver.3.0 8GB memory map (Boot)

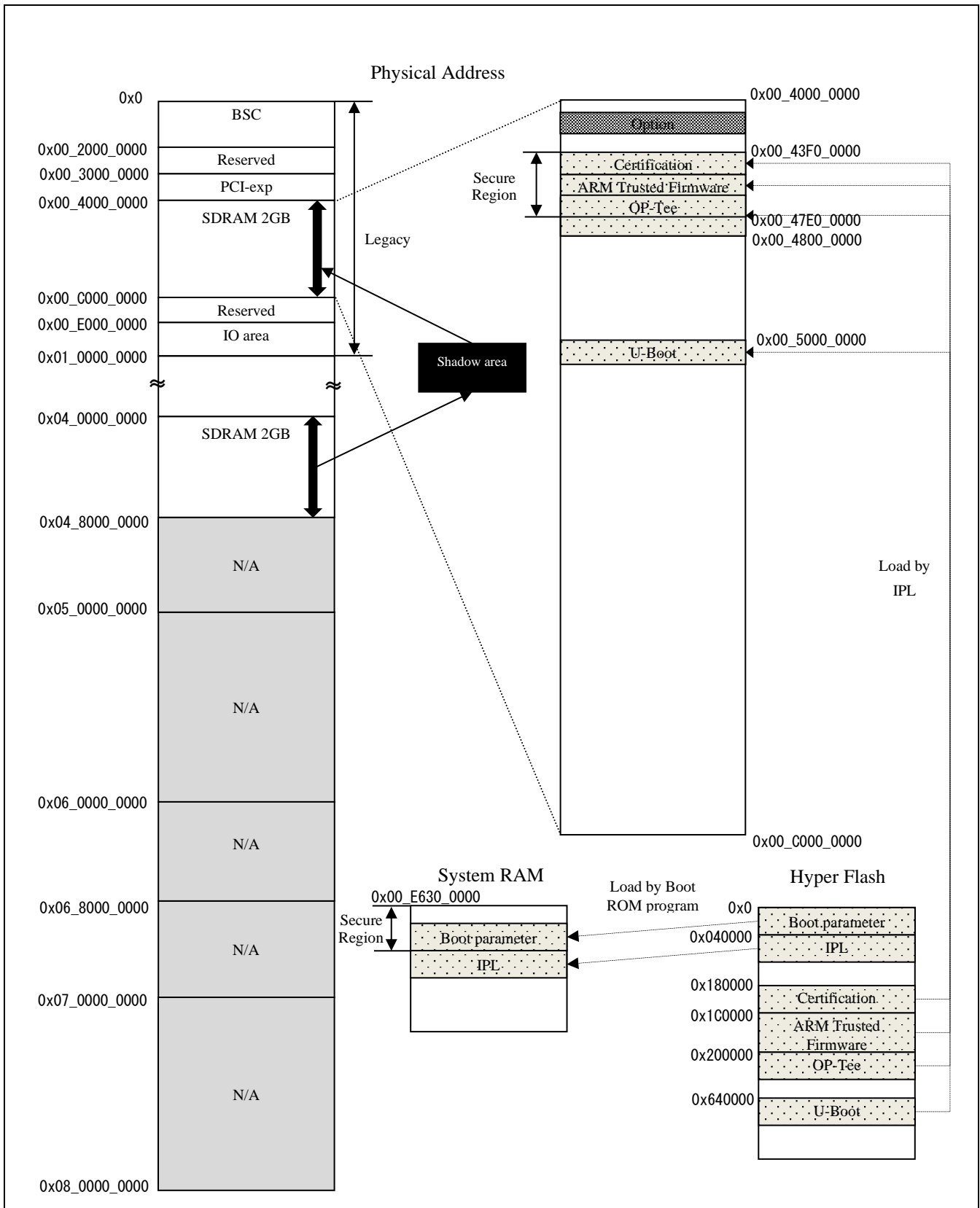**Figure 7. M3 Ver.3.0 8GB memory map (Linux)**

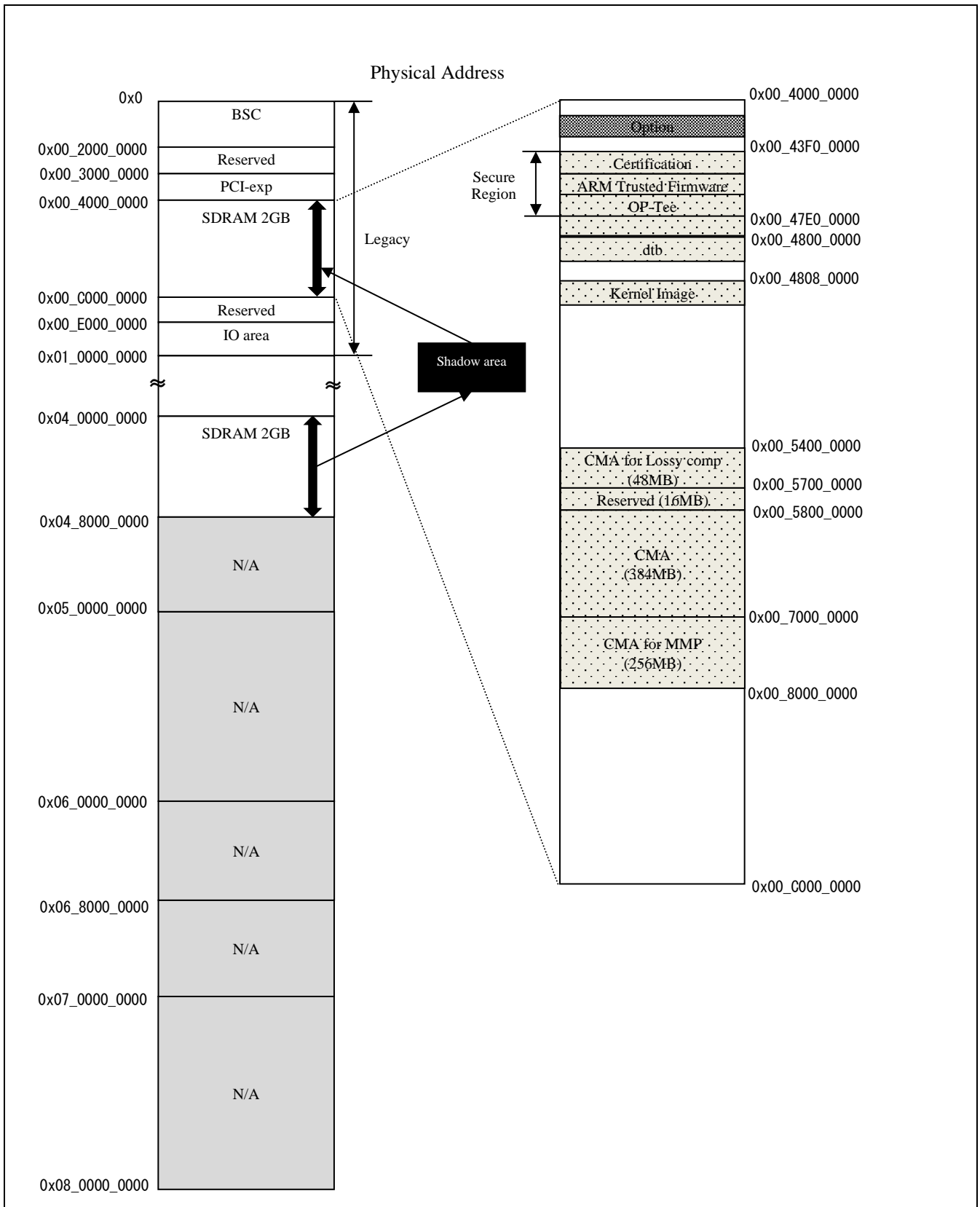**Figure 8. M3 Ver.1.x (4GB) memory map (Boot)**

Physical Address

Left column (Physical Address)	Right column (detail)
0x0 — BSC	0x00_4000_0000
0x00_2000_0000 — Reserved	Option — 0x00_43F0_0000
0x00_3000_0000 — PCI-exp	Certification
0x00_4000_0000 — SDRAM 2GB	ARM Trusted Firmware
Legacy	OP-Tee — 0x00_47E0_0000
	.dtb — 0x00_4800_0000
0x00_C000_0000 — Reserved	— 0x00_4808_0000
0x00_E000_0000 — IO area	Kernel Image
0x01_0000_0000	Shadow area
0x04_0000_0000 — SDRAM 2GB	CMA for Lossy comp (48MB) — 0x00_5400_0000
	— 0x00_5700_0000
0x04_8000_0000 — N/A	Reserved (16MB) — 0x00_5800_0000
0x05_0000_0000 — N/A	CMA (384MB)
0x06_0000_0000 — SDRAM 2GB	— 0x00_7000_0000
	CMA for MMP (256MB)
0x06_8000_0000 — N/A	— 0x00_8000_0000
0x07_0000_0000 — N/A	
0x08_0000_0000	— 0x00_C000_0000

Secure Region

Figure 9. M3 Ver.1.x (4GB) memory map (Linux)

Figure 10. M3N memory map (Boot)
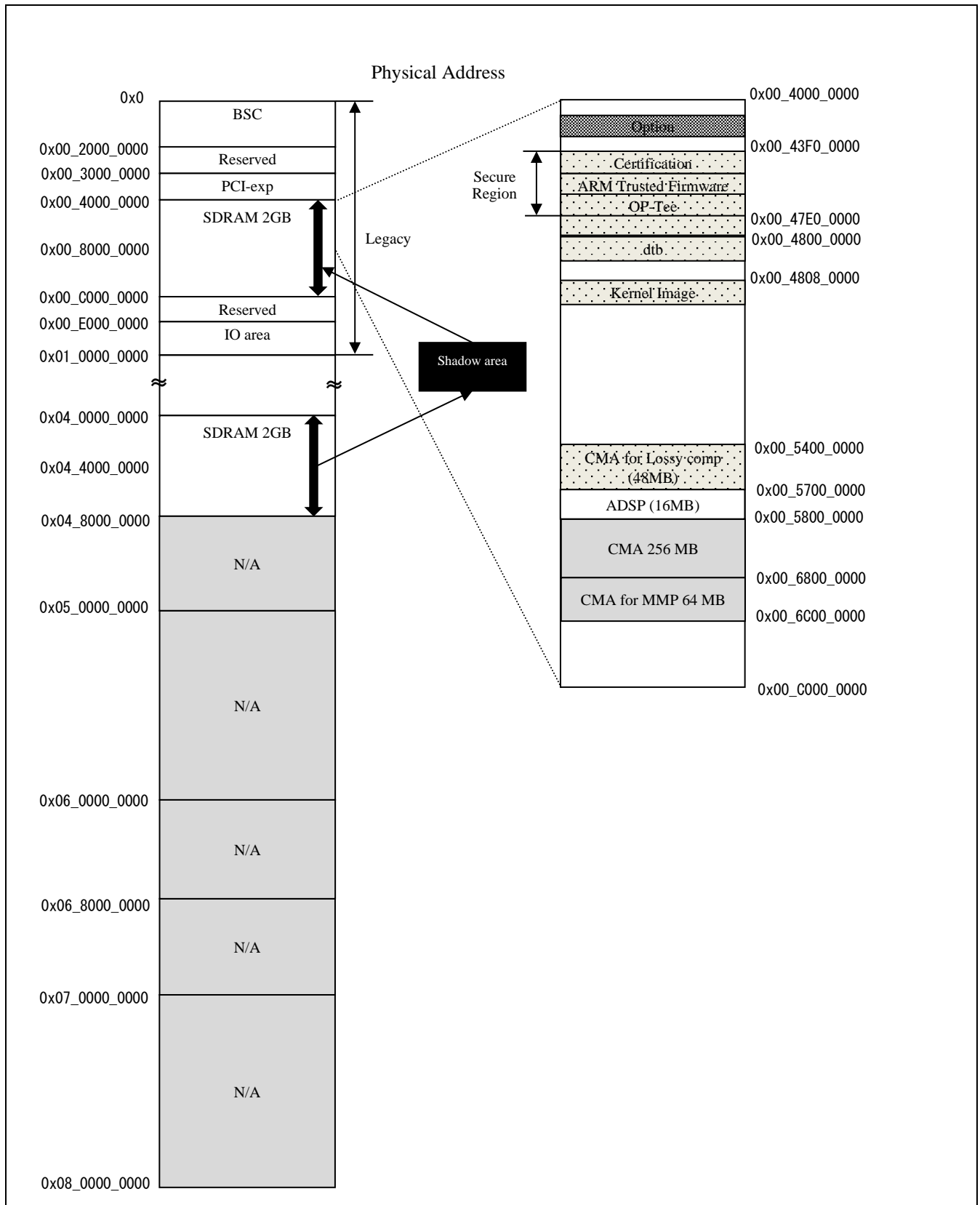
Figure 11. M3N memory map (Linux)
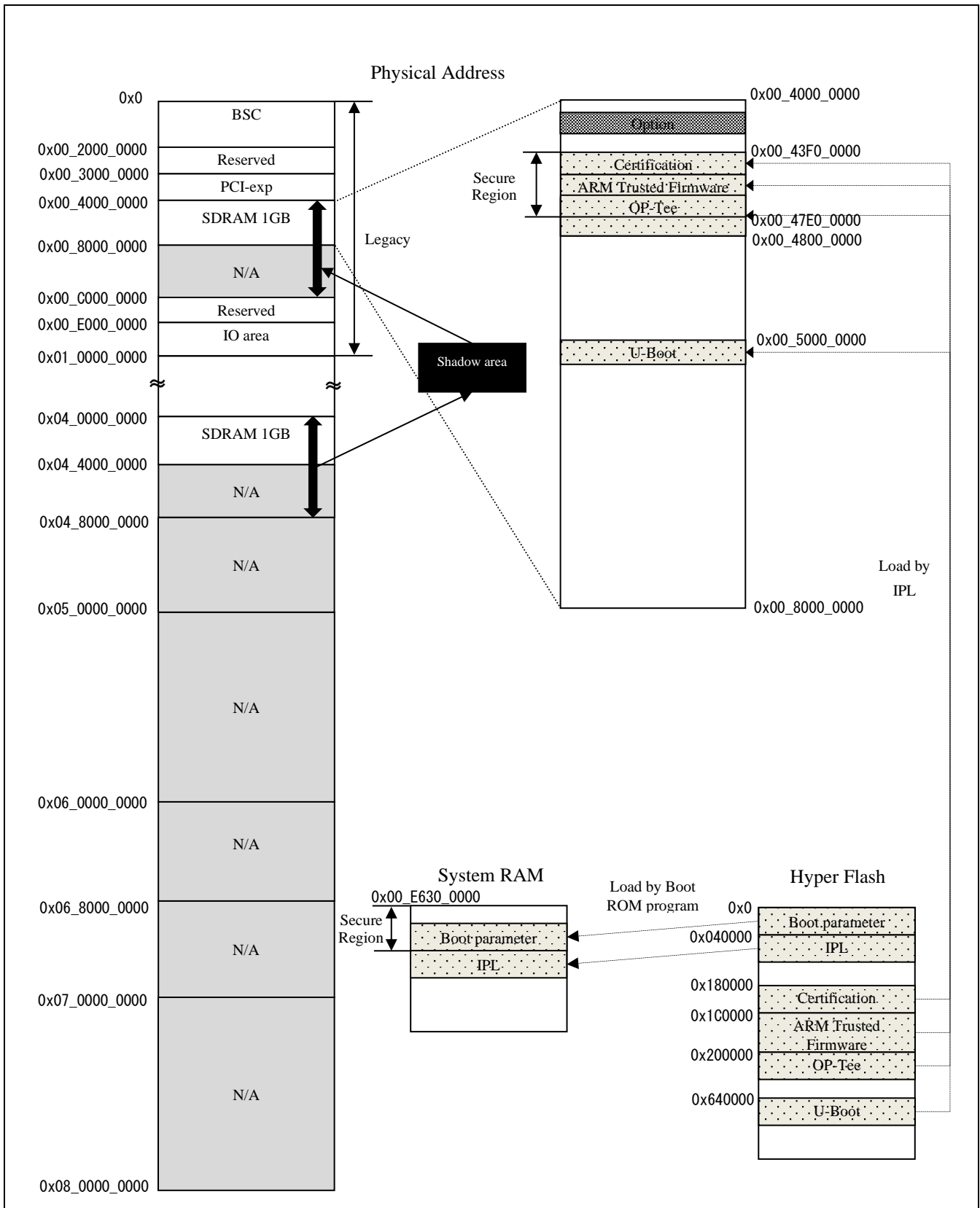
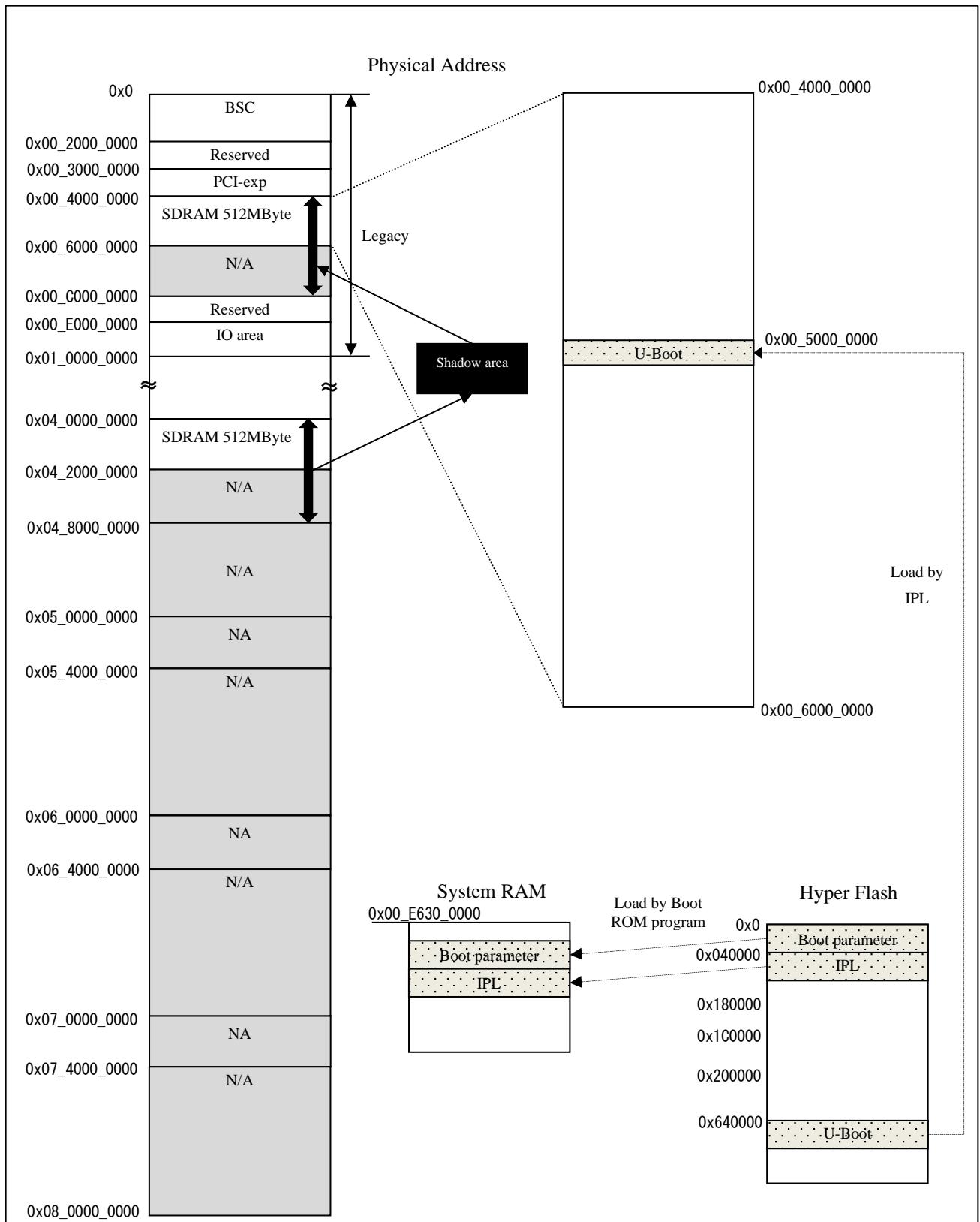**Figure 12. E3 System Evaluation Board Ebisu memory map (Boot)**
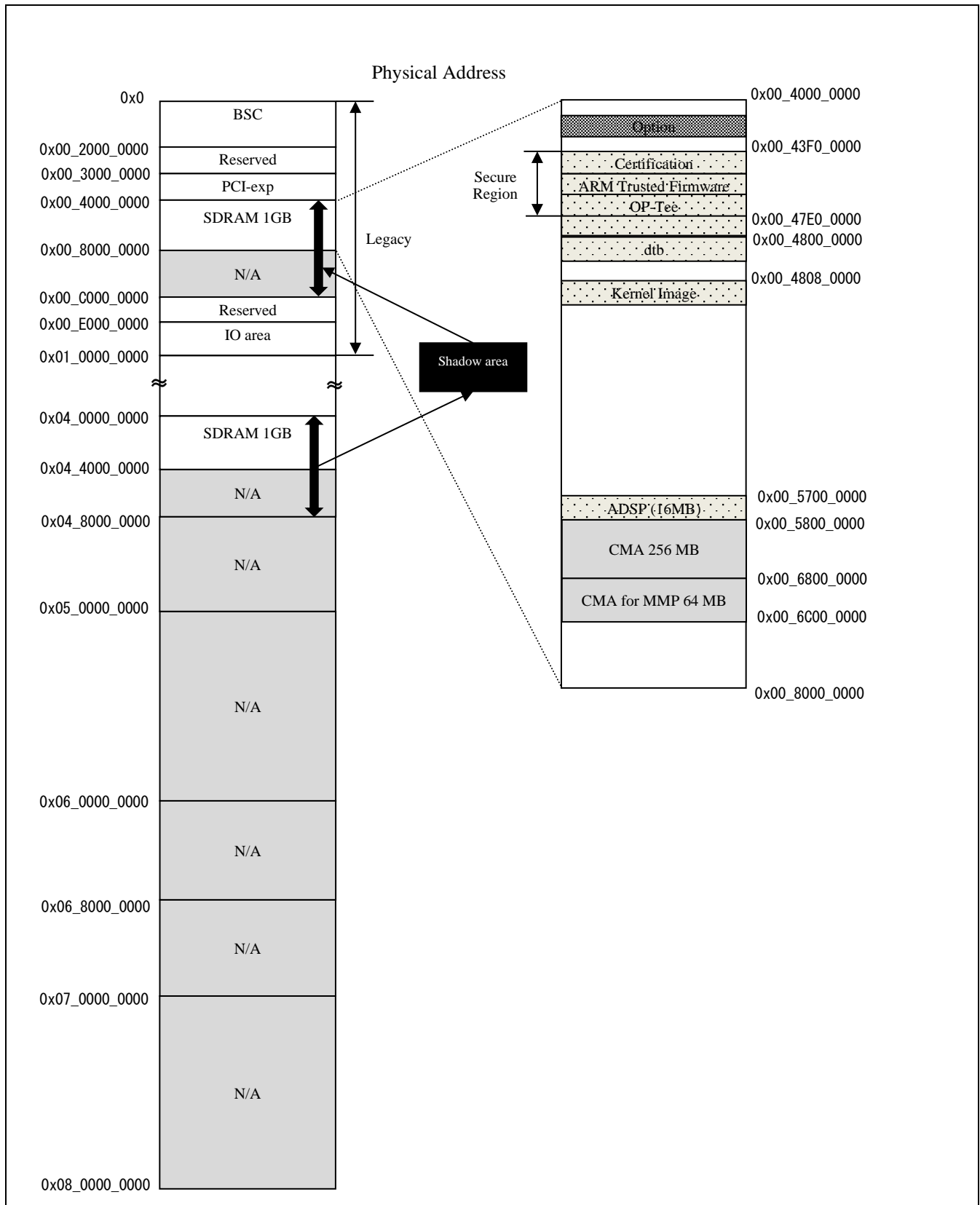
Figure 13. E3 System Evaluation Board Ebisu memory map (Linux)
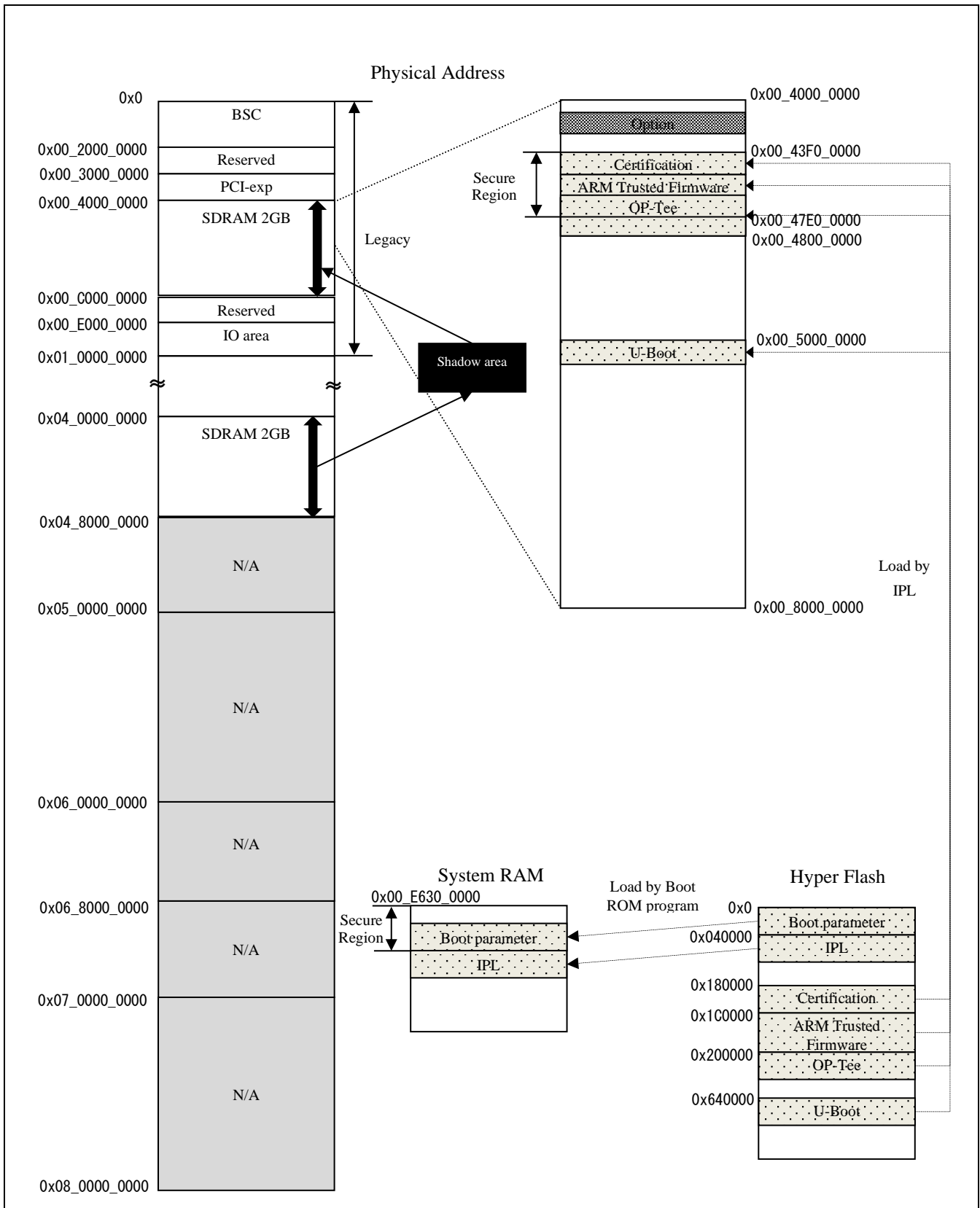
Figure 14. E3 System Evaluation Board Ebisu-4D memory map (Boot)
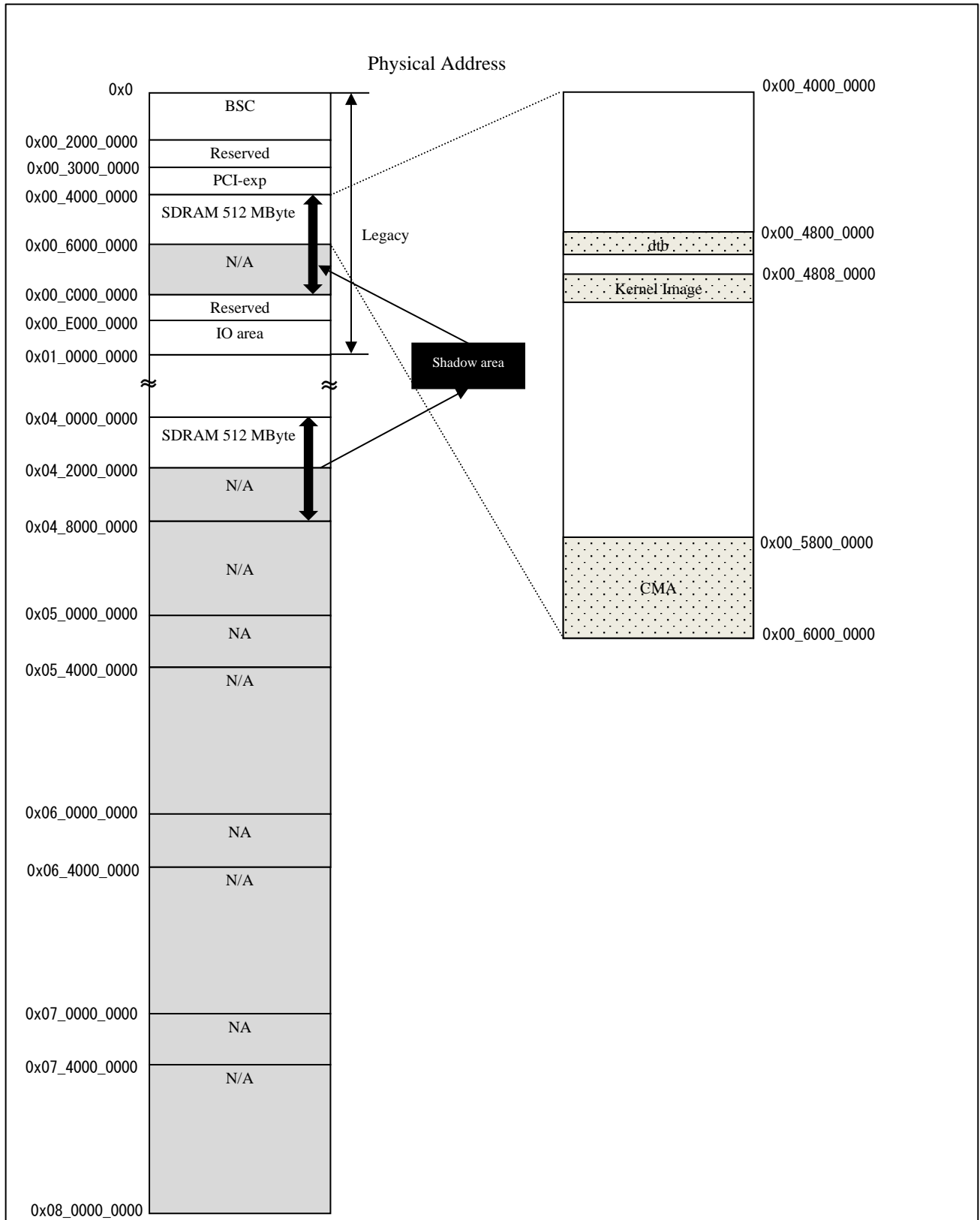
Figure 15. E3 System Evaluation Board Ebisu-4D memory map (Linux)

Figure 16. D3 memory map (Boot)

Figure 17. D3 memory map (Linux)

Note)

- Kernel region is assigned by Kernel device tree arch/arm64/boot/dts/renesas/xxx.dts and totally mapped to 8GB (R-Car H3 Ver.3.0 8GB, R-Car M3 Ver.3.0), 4GB (R-Car H3 Ver.3.0 4GB/Ver.2.0/M3 Ver.1.x), 2GB (R-Car M3N), 1GB (R-Car E3 System Evaluation Board Ebisu) and 2GB (R-Car E3 System Evaluation Board Ebisu-4D), 512MB (R-Car D3 System Evaluation Board Draak).

- Kernel region consists of 4 parts: (R-Car H3 Ver.3.0 8GB)

  ➢ 1920MB from 0x00_4800_0000 to 0x00_BFFF_FFFF

  ➢ 2GB from 0x05_0000_0000 to 0x05_7FFF_FFFF

  ➢ 2GB from 0x06_0000_0000 to 0x06_7FFF_FFFF

  ➢ 2GB from 0x07_0000_0000 to 0x07_7FFF_FFFF

- Kernel region consists of 4 parts: (R-Car H3 Ver.3.0 4GB/Ver.2.0)

  ➢ 896MB from 0x00_4800_0000 to 0x00_7FFF_FFFF

  ➢ 1GB from 0x05_0000_0000 to 0x05_3FFF_FFFF

  ➢ 1GB from 0x06_0000_0000 to 0x06_3FFF_FFFF

  ➢ 1GB from 0x07_0000_0000 to 0x07_3FFF_FFFF

- Kernel region consists of 2 parts: (R-Car M3 Ver.1.x)

  ➢ 1920MB from 0x00_4800_0000 to 0x00_BFFF_FFFF

  ➢ 2GB from 0x06_0000_0000 to 0x06_7FFF_FFFF

- Kernel region consists of 3 parts: (R-Car M3 Ver.3.0)

  ➢ 1920MB from 0x00_4800_0000 to 0x00_BFFF_FFFF

  ➢ 2GB from 0x04_80000_0000 to 0x04_FFFF_FFFF

  ➢ 4GB from 0x06_0000_0000 to 0x06_FFFF_FFFF

- Kernel region consists of 1 part: (R-Car M3N)

  ➢ 1920MB from 0x00_4800_0000 to 0x00_BFFF_FFFF

- Kernel region consists of 1 part: (R-Car E3 System Evaluation Board Ebisu)

  ➢ 896MB from 0x00_4800_0000 to 0x00_7FFF_FFFF

- Kernel region consists of 1 part: (R-Car E3 System Evaluation Board Ebisu-4D)

  ➢ 1920MB from 0x00_4800_0000 to 0x00_BFFF_FFFF

There are three types of CMA regions.

They are defined in device tree (arch/arm64/boot/dts/renesas/xxxx.dts).

➢ Default CMA region: It is for kernel, general drivers and multimedia package.

```
linux,cma {
   compatible = "shared-dma-pool";
   reusable;
   reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
   linux,cma-default;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```

Note)

● 128 MB in this CMA (H3/M3/M3N 384MB, E3 256MB) is reserved for kernel and general drivers, and the remaining H3/M3/M3N 256 MB, E3 128MB is reserved for multimedia package.

● The CMA region can be adjusted by changing the start address and the size.

● Should take care of the lack of memory allocated by kernel and general drivers when reducing the region size.

➢ CMA region for MMP: It is for multimedia package (specific H/Ws).

```
mmp_reserved: linux,multimedia {
   compatible = "shared-dma-pool";
   reusable;
   reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```
Note)

● Refer to User's manual of Memory Manager in order to change CMA region for MMP.

➢ H3/M3/M3N CMA region for Lossy comp: It is for storing data compressed by Frame Compression Near Lossless (FCNL) for media playback.

```
lossy_decompress: linux,lossy_decompress {
   no-map;
   reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```
Note)

● Refer to User's manual of IPL in order to change CMA region for Lossy comp.
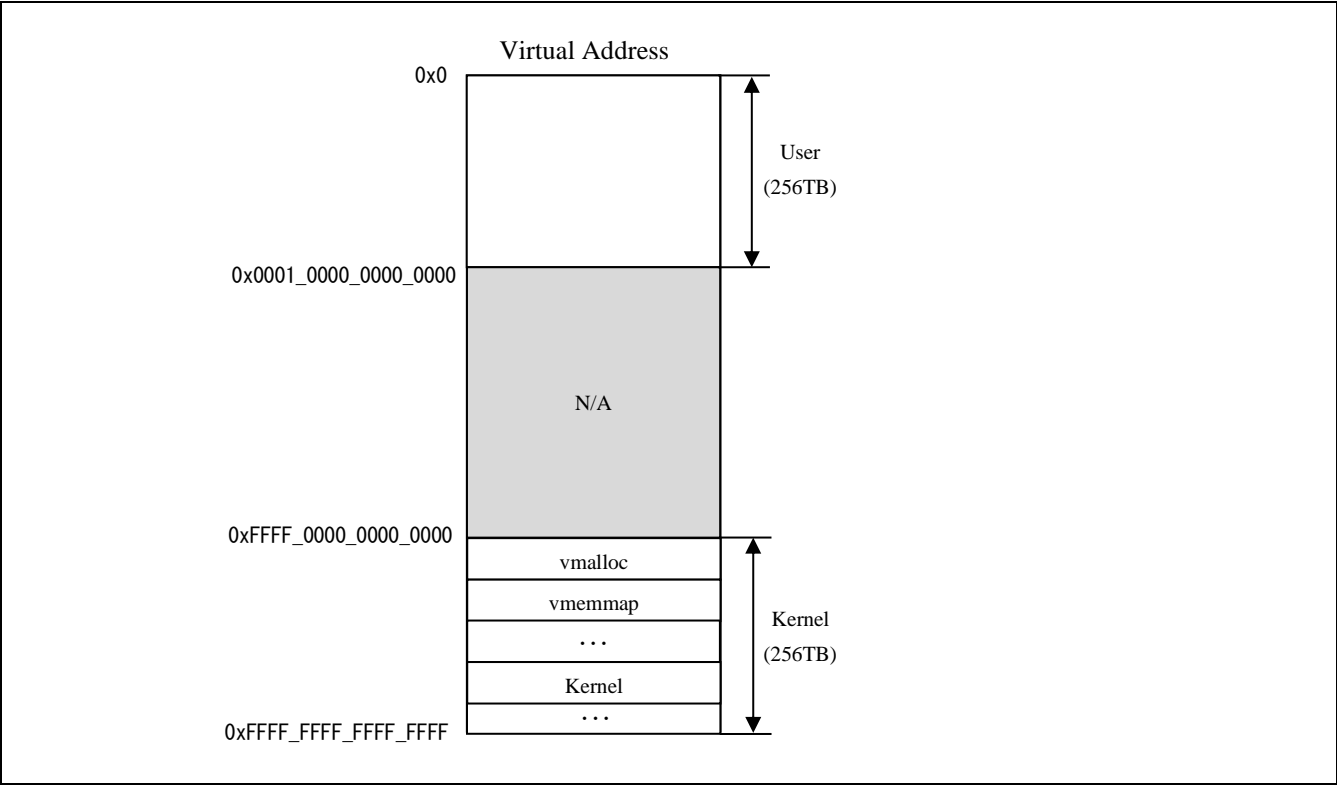
Figure 18. H3/M3/M3N/E3/D3 memory map (Virtual)

Note)
- Kernel uses 4KB page size (VA_BITS=48) and 4 levels of translation tables. Both regions of User and Kernel are 256TB. Refer to Documentation/arm64/memory.txt.

- Detail information about kernel memory map in virtual address space, refer to User's manual of Kernel.

8. U-Boot command

Please refer to U-Boot user's manual about available U-Boot command for R-Car H3/M3/M3N/E3/D3 Linux BSP. The help or "?" command shows U-Boot command list, but be careful that it includes some unsupported command.