NXP Semiconductors
Application Note

Document Number: AN3996
Rev. 2, 06/2017

i.MX35 Boot Options

1. Introduction

This application note describes the different booting options available for the i.MX35 processor. The i.MX35 offers several interfaces to boot the processors, and thus it becomes flexible to implement multimedia applications.

This application note discusses the boot ROM, which is a memory chip that allows workstations to be booted from a server or other remote station. The boot ROM supports features such as booting from various external memory devices, support for downloading code through a ROM bootloader, device configuration, and secure boot. The boot ROM is responsible for reading inputs, such as the boot pins and eFuses that determine the behavior of the boot flow.

Refer to the System Boot chapter in *i.MX35 (MCIMX35) Multimedia Applications Processor Reference Manual* (document IMX35RM), for more information about booting modes.

Contents

2. Boot options

The i.MX35 processor begins booting from power-on reset (POR) or warm reset. The reset boot sequence uses the high-assurance boot (HAB) library, to provide a secure booting environment. The HAB is a combination of hardware and software, with public key infrastructure (PKI) protocol to protect the system—from the executing unauthorized images or programs. In order for the HAB to allow the user code to run, the code must be signed by the private key holder, that matches the public key in the device. The HAB library in the processor's ROM also provides a number of API functions, that allows the user to authenticate any defined region and signature, during run-time. The HAB also operates in bypass mode or direct external boot, in which the processor boots directly from the external memory, like traditional microprocessors.

Depending on the HAB type security configuration, the boot capabilities are different on the i.MX35 package. Full flexibility is supported in the development (or engineering) configuration. But significant limitations are imposed on the production (or secure) configuration. A third option in which the security features are disabled is also supported.

NOTE

Customers should contact their NXP sales representative, to use the HAB/secure.

The ROM also provides a mechanism to download, and flash new code, through a serial connection. The flash programming is facilitated by downloading a downloader application into the RAM. The downloader uses either a high-speed USB (on UTMI or ULPI PHYs) or full speed USB (on serial PHY) or a UART connection.

2.1. i.MX35 input/output muxing

The i.MX35 provides several combinations to enable different modules, with a reduce pin count in its package. This is accomplished by the I/O muxing that allows pin configuration. But, with I/O muxing, configuration restrictions becomes critical that necessitates care in setting the mux.

2.2. i.MX35 boot modes

The boot type is determined by the values in the boot mode contacts, BMOD[1:0], sampled at end of the reset signal, and stored in the Reset Control and Source Register (RCSR) of the system reset controller Clock Controller Module (CCM).

Table 1 gives the different boot modes.

Table 1. Boot mode summary

BMOD[1:0]	Boot mode	Boot details
00	Internal boot	Executes ROM code that handles booting, as described in Table 2.
01	—	Used internally by NXP, so this bit combination cannot be used.
10	External (direct) boot	Hardware only (direct boot through the interface, independent of boot ROM code) boot from the WEIM (wireless external interface module) interface and NAND boot. With BMOD[1:0] = 10, two different booting modes are obtained, based on the value of BT_MEM_CTL[1:0].
10	Startup mode	BT_MEM_CTL[1:0] = b11. This mode is useful for software development, using tools, capable of developing JTAGs. With reset signal, the address stored in the Program Counter (PC) becomes 0x00000000. This means that ROM code has not run, and the state of all the internal SOC registers are untouched. The system is now ready to run any application that is loaded to it through a JTAG debugging tool. Though devices can be connected to the SOC, through JTAG in other boot modes, in this mode it is guaranteed that all internal SOC registers are at its default (or reset) values.
11	Serial boot loader	Load and execute code, through serial devices (high speed USB, through integrated PHY or external and UART).

The differences between internal and external mode are as follows:

- Internal ROM code cannot be executed in external mode, and also the security features cannot be used.
- All peripherals are not available in external mode. Only those related to WEIM interface and NAND booting are available.

NOTE

For typical application board usage, the internal PHY option is recommended. The use of an external transceiver is not recommended, as it comes at the expense of the availability of pins.

The flash header is a data structure that provides information about the application. The flash header must be located at a fixed address offset, depending on the type of the external flash device connected to the i.MX35. Refer to *i.MX35 (MCIMX35) Multimedia Applications Processor Reference Manual* (document IMX35RM), for more information about headers.

2.3. Boot options and eFuse settings

Table 2 shows the eFuse settings used by the ROM, in the boot process. By default, these fuses are unburned.

NOTE

The bits tagged fuse only are available as fuse only.

Table 2. Fuse descriptions

Fuse/GPIO	Definitions	Settings for each combination of bits
DIR_BT_DIS (fuse only)	Direct external memory boot disable	0 Direct boot to external memory is allowed 1 Direct boot to external memory is not allowed
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	00 WEIM 01 NAND Flash 10 ATA HDD 11 Expansion device (SD, eSD, MMC, eMMC, serial ROM)
BT_PAGE_SIZE[1:0]	Represents NAND Flash page size, and is used in conjunction with the BT_MEM_CTL[1:0] setting.	If BT_MEM_CTL = NAND Flash, then: 00 512 bytes 01 2 Kbyte 10 4 Kbyte 11 Reserved
BT_SPARE_SIZE[1:0] (fuse only)	Specifies the size of spare bytes for the NAND Flash devices. Here, BT_MEM_CTL[1:0] = 01. BT_SPARE_SIZE[0] is used as a fast boot mode indication for the eSD 2.10 protocol.	00 16-bytes spare (for 0.5-Kbyte page size device) 01 64-bytes spare (for 2-Kbyte page size device) 10 128-bytes spare (for 4-Kbyte page size device) 11 218-bytes spare (for 4-Kbyte page size device) If the bootable device is SD, then: n0FAST_BOOT bit 29 in ACMD41 argument is 0 n1FAST_BOOT bit 29 in ACMD41 argument is 1
BT_BUS_WIDTH	It represents the bus width, and is used in conjunction with the BT_MEM_CTL[1:0] setting. For Serial Peripheral Interface (SPI), it determines the device type (EEPROM or serial Flash).	If BT_MEM_CTL[1:0] = NAND Flash, then: 0 8 bit 1 16 bit If BT_MEM_CTL[1:0] = WEIM (NOR), then: 0 16 bit 1 Reserved If BT_MEM_CTL[1:0] = expansion device (SPI), then: 0 2-bytes address SPI device (EEPROM) 1 3-bytes address SPI device (serial Flash)
BT_MEM_TYPE[1:0]	Represents boot memory type, as is interpreted by the boot ROM software, according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter delays and timing, in support of direct boot.	If BT_MEM_CTL = WEIM, then: 00 NOR 01 Reserved 10 Samsung OneNAND™ 11 Reserved If BT_MEM_CTL = NAND Flash, then: 00 3 address cycles

Table 2. Fuse descriptions

Fuse/GPIO	Definitions	Settings for each combination of bits
		01 4 address cycles 10 5 address cycles 11 Reserved If BT_MEM_CTL = ATA HDD, then: 00 Reserved 01 P-ATA HDD 10 Reserved 11 Reserved If BT_MEM_CTL = expansion card device, then: 00 SD, MMC, eMMC or eSD 01 Reserved 10 Serial ROM through I^2C 11 Serial ROM through SPI
BT_SDMMC_SRC (fuse only)	Chooses the specific Enhanced Secure Digital Host Controller (eSDHC) controller used for booting.	00 eSDHC-1 01 eSDHC-2 10 eSDHC-3
BT_ECC_SEL	Defines 4 or 8-bit ECC. Also used as a fast boot mode indication for eMMC 4.3 protocol.	0 4-bit ECC 1 8-bit ECC If the bootable device is MMC, then: 0 Do not use eMMC fast boot mode 1 Use eMMC fast boot mode
BT_USB_SRC[1:0]	USB PHY selection	00 UTMI PHY 01 ULPI PHY 10 Serial PHY: ATLAS™ 11 Serial PHY: PHILIPS™ ISP1301
BT_EEPROM_CFG (fuse only)	Selects whether Device Configuration Data (DCD) is loaded from the EEPROM (electrically erasable programmable ROM) prior to boot from other devices. Note that BT_EEPROM_CFG is not applicable when EEPROM is used as a boot device.	0 Use EEPROM DCD 1 Do not use EEPROM DCD
BT_WEIM_MUXED (fuse only)	Puts WEIM in muxed mode and back.	For BT_MEM_CTL[1:0] = WEIM (NOR): 0 Not muxed 1 WEIM in address muxed mode
GPIO_BT_SEL (fuse only)	Represents General Purpose Input/Output (GPIO) boot select, and determines whether certain boot fuse values are obtained from eFuses or GPIO.	0 Fuse values are determined by the GPIO contacts 1 Fuse values are determined by eFuses

2.3.1. Boot related pins

Additional boot configuration settings are obtained, either from programmable eFuses or by contacts sampled at POR negation. Selection between these two options is based on the value of the GPIO_BT_SEL fuse (where by default, it is un-blown):

- If the GPIO_BT_SEL is blown, then all boot options are configured by the eFuses, as explained in Table 2 Boot ROM code software reads the values, either from the RCSR register, or from the eFuses, through the IC identification module (IIM).
- If the GPIO_BT_SEL is left un-blown, then the various boot options are determined by sampling the dedicated contacts at reset. Every eFuse option is overridden with the dedicated contact(s),

such that the same functionality is available under both boot options. Table 3 lists the boot option contacts. In this case, regardless of the fuse values, the boot ROM code reads the values of the options, from the RCSR register of the SRC module. Refer to *i.MX35 IC Identification Module (IIM) Fusebox* (document AN3652), for the detailed information about blown fuses in the i.MX35.

Table 3. Boot related pins

BGA	Pin name	eFuse name	Function at boot
W10	BOOT_MODE0	N/A	Boot mode select pins
U9	BOOT_MODE1	N/A	Boot mode select pins
U15	CSI_D8	BT_MEM_CTRL[0]	Boot memory device
W17	CSI_D9	BT_MEM_CTRL[1]	Boot memory device
V16	CSI_D10	BT_MEM_TYPE[0]	Boot memory type
T15	CSI_D11	BT_MEM_TYPE[1]	Boot memory type
W16	CSI_D12	BT_PAGE_SIZE[0]	NAND Flash page size
V15	CSI_D13	BT_PAGE_SIZE[1]	NAND Flash page size
U14	CSI_D14	BT_ECC_SEL	Define 4/8-bit ECC
Y16	CSI_D15	BT_USB_SRC[0]	USB PHY selection
V14	CSI_HSYNC	BT_USB_SRC[1]	USB PHY selection
T14	CSI_VSYNC	BT_BUS_WIDTH	NAND bus width

2.3.2. Level selection of the boot related pins

To set the logic values for the boot related pins, two options are available:

- Fixed pull-up (pull-up resistors are tied to 3.3V = NVCC_CSI = NVCC_CRM = CPU_3V15) or pull-down (pull-down resistors are tied to the system ground), as seen in Figure 1 in the CPU board.

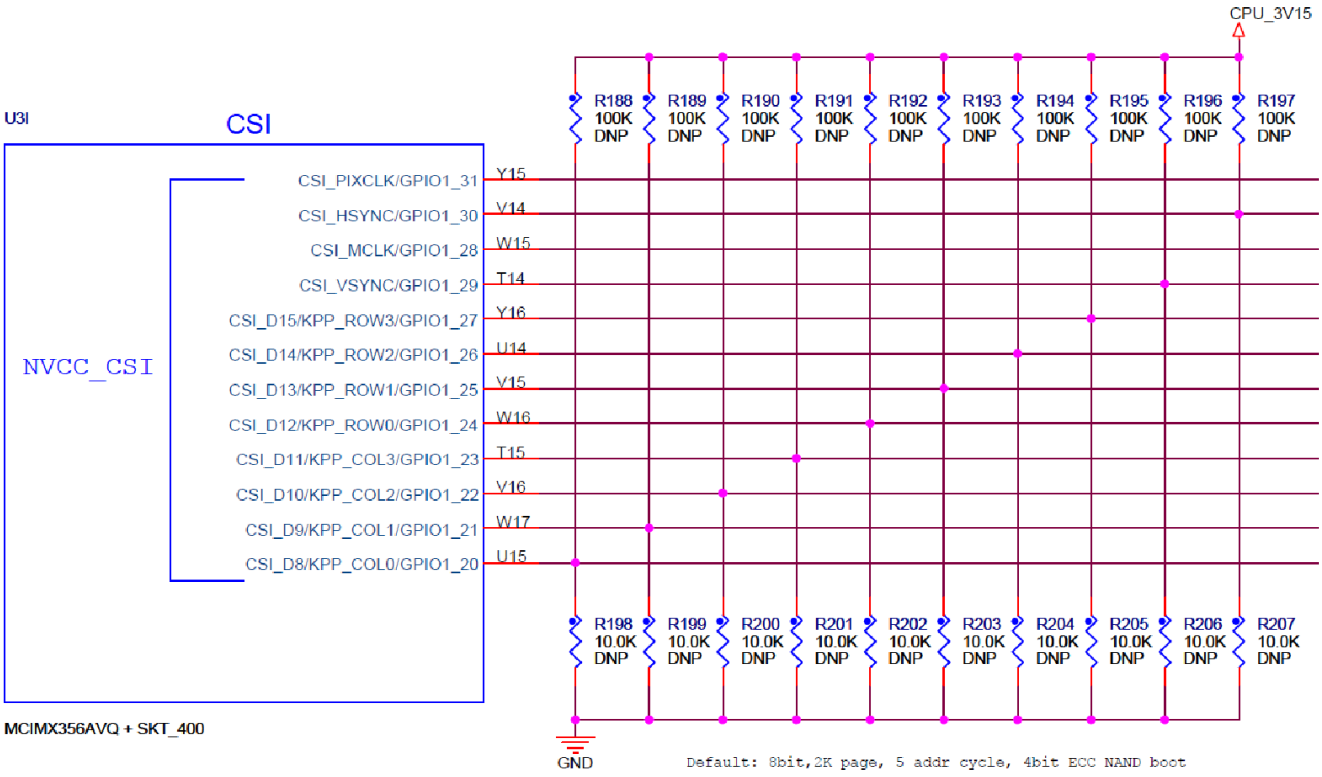

Figure 1. CPU board with fixed pull-up and pull-down

- Dip switch option on the personality board, as shown in Figure 2.

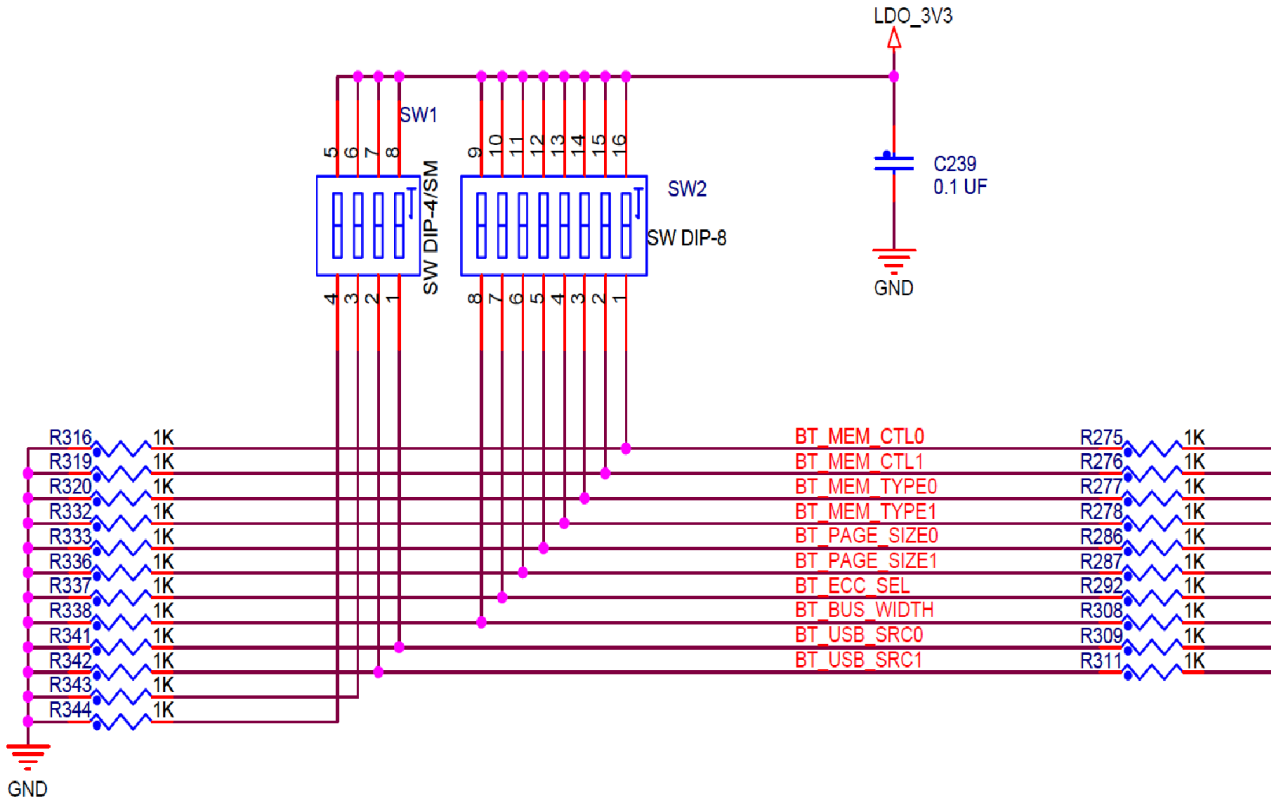

Figure 2. Personality board with the DIP switch option

2.3.3. Boot sequence

Figure 3 shows the i.MX35 boot sequence flow. Color coding (see legend) indicates the controlling sources (hardware only, boot contacts or eFuses).

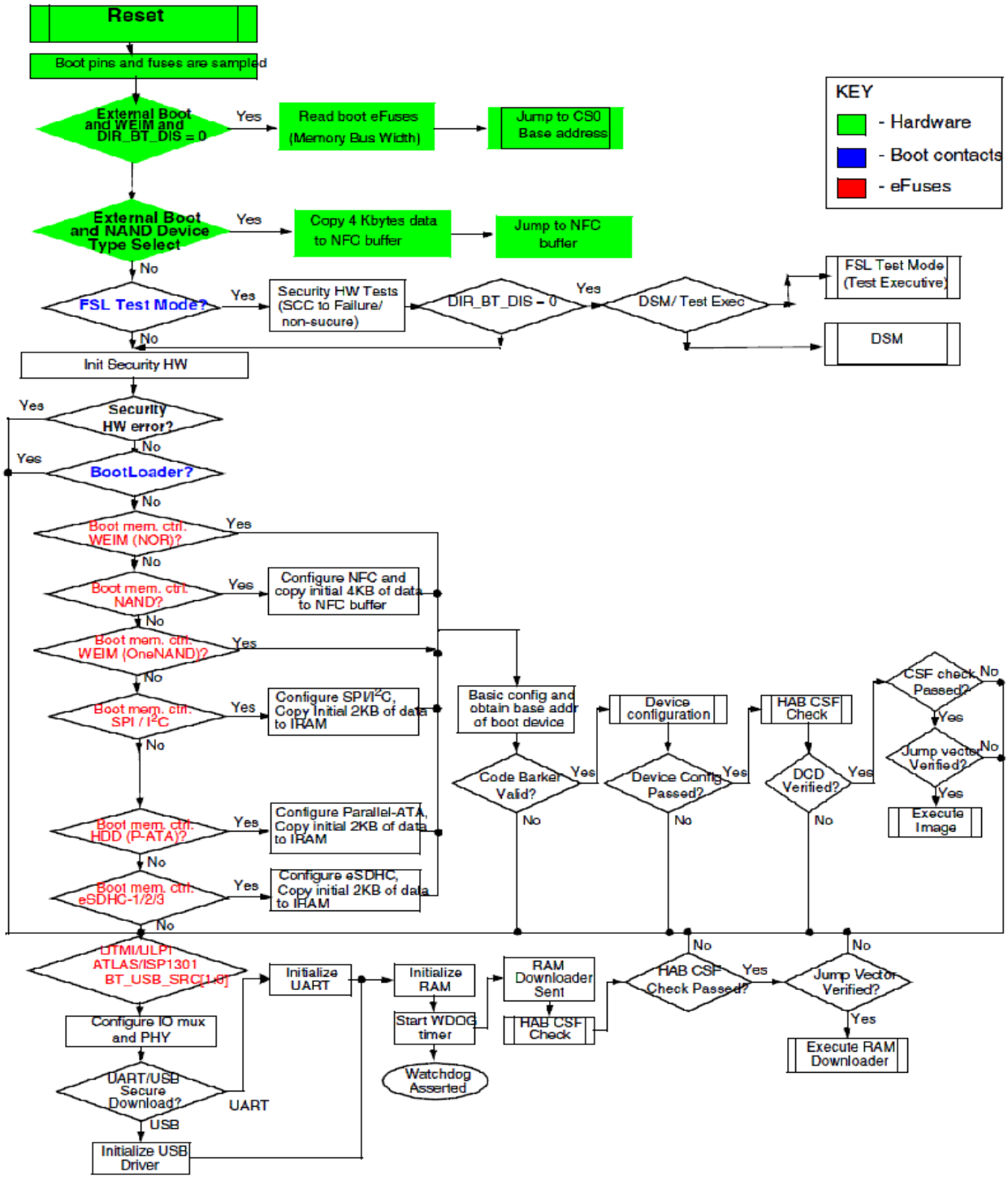

Figure 3. Boot sequence diagram

NOTE

For external boot, NAND device—DIR_BT_DIS must be set to 0, to allow the direct boot to external memory.

3. Supported devices for internal boot

Internal boot is selected by driving a value of 0b00, on the BMOD contacts at device power-up. The boot code in the internal ROM performs the following steps:

1. Initializes the hardware
2. Validates application image using the HAB library
3. Jumps to an address derived from the application image

If any error occurs during internal boot, the boot code jumps to the UART/USB secure download. Internal boot mode is the only mode in which secure boot and secure serial download of the device are possible.

The i.MX35 supports the following boot devices:

- 32 or 16-bit NOR Flash, through the WEIM interface (located on CS0)
- 16-bit OneNAND Flash memory, through the WEIM interface (located on CS0)
- Multi-Level Cell (MLC) and Single-Level Cell (SLC) NAND Flash, through NAND Flash Controller (NFC) interface, with the following features:
  - Page size of 512 bytes, 2 or 4 Kbyte
  - 8 or 16-bit bus width
  - Error Checking and Correction (ECC) of 4 or 8 bits
- Parallel Advanced Technology Attachment (P-ATA) HDD with P-ATA interface
- All types of high capacity, standard capacity, dual voltage, and high voltage cards:
  - Secure Digital card (SD)
  - MultiMedia card (MMC)
  - Embedded Secure Digital card (eSD)
  - Embedded MultiMedia Card (eMMC)
  - It also supports eSD fast boot and eMMC boot mode, through all eSDHC ports.
- Serial ROM and serial Flash, which are accessed with:
  - Serial Peripheral Interface (SPI) protocol through CSPI module
  - I^2C protocol through I^2C module

NOTE

NDA is required to get the i.MX35 Security Manual.

3.1. Basic initialization

Table 4 shows the clock settings (configured by ROM code), required for the internal boot. On reset, the core has access to all the shared peripherals.

Table 4. Clock settings for internal boot

Clock	Frequency (MHz)
ap_core_clk	266
ap_ahb_clk	133
ap_ipg_clk	66.5
usb_clk	60
nfc_clk	21.6

3.2. NOR Flash boot operation

Boot from the NOR Flash, through the WEIM interface, is supported for debug purposes. It is used only in special cases. As the WEIM port does not have dedicated contacts for booting, the muxing of the bus is set only at boot, by the boot ROM code. The NOR Flash interface works in an asynchronous mode, and based on the eFuse settings, supports either muxed address/data or non-muxed schemes.

Table 5 shows the settings used by the ROM in NOR Flash boot operation.

Table 5. NOR boot configuration

Fuse/GPIO	Definition	Settings for each Combination of Bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	00 WEIM
BT_BUS_WIDTH	Represents bus width, and is used in conjunction with the BT_MEM_CTL[1:0] setting	If BT_MEM_CTL[1:0] = WEIM (NOR), then: 0 16 bit 1 Reserved
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware, to alter delays and timing, in support of the direct boot.	If BT_MEM_CTL = WEIM, then: 00 NOR 01 Reserved 10 Samsung OneNAND 11 Reserved
BT_WEIM_MUXED (fuse only)	Puts WEIM in muxed mode and back.	For BT_MEM_CTL[1:0] = WEIM (NOR): 0 Not muxed 1 WEIM in address muxed mode

In addition to setting the boot mode pins, the pins connecting the external NOR WEIM device must be selected. Since WEIM is the default option of the i.MX35, the ROM code does not need to modify any IOMUX settings, and the boot is carried out in the EMI port.

3.3. OneNAND Flash boot operation

OneNAND Flash devices are available only with a 16-bit interface. The OneNAND Flash driver gets the device page size by issuing a software command to the device, and collecting the response from the device, after system power-up. OneNAND Flash boot proceeds as follows:

1. At system power-up, OneNAND automatically copies 1 Kbyte of data from the start of the flash array (sector 0 and sector 1, page 0, block 0), to its boot RAM. This data includes the flash header.
2. Boot ROM copies the 1-Kbyte boot RAM contents to the destination address (defined in the

application header), and decrements the length of the image to be read from the OneNAND by 1 Kbyte. The copying operation is a simple memory copy (the boot RAM area is memory mapped), and the length of the image to be read from the OneNAND is specified in the flash header structure.

Any failure in the data load from OneNAND Flash causes the boot ROM to switch to serial download. Table 6 shows the settings used by the ROM in OneNAND Flash boot operation.

Table 6. OneNAND boot configuration

Fuse/GPIO	Definition	Settings for each Combination of Bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	
BT_BUS_WIDTH	Represents bus width, and is used in conjunction with the BT_MEM_CTL[1:0] setting. For SPI, it determines the device type (EEPROM or serial Flash).	If BT_MEM_CTL[1:0] = WEIM (NOR), then: 0 16 bit 1 Reserved
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter delays and timing, in support of direct boot.	If BT_MEM_CTL = WEIM, then: 00 NOR 01 Reserved 10 Samsung OneNAND 11 Reserved

3.4. NAND Flash boot operation

Most of the MLC/SLC NAND Flash devices from different vendors are supported by the boot ROM. In particular, NAND Flash boot requires BT_MEM_CTL to be set to 0b01. These parameters are either provided by the eFuses or sampled in the GPIO pins, during boot.

Table 7 shows the parameters used to configure the external NAND Flash.

Table 7. NAND boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	Must be set to 0b01 for NAND Flash boot
BT_PAGE_SIZE[1:0]	Represents NAND Flash page size, and is used in conjunction with the BT_MEM_CTL[1:0] setting.	If BT_MEM_CTL = NAND Flash, then: 00 512 bytes 01 2 Kbyte 10 4 Kbyte 11 Reserved
BT_SPARE_SIZE[1:0] (fuse only)	Specifies the size of the spare bytes for the NAND Flash devices (BT_MEM_CTL[1:0] = NAND Flash). BT_SPARE_SIZE[0] is used as a fast boot mode indication for the eSD 2.10 protocol.	00 6-bytes spare (for 0.5-Kbyte page size device) 01 64-bytes spare (for 2-Kbyte page size device) 10 128-bytes spare (for 4-Kbyte page size device) 11 218-bytes spare (for 4-Kbyte page size device) If the bootable device is SD, then:

Table 7. NAND boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
		n0FAST_BOOT bit 29 in ACMD41 argument is 0 n1FAST_BOOT bit 29 in ACMD41 argument is 1
BT_BUS_WIDTH	Represents bus width, and is used in conjunction with the BT_MEM_CTL[1:0] setting. For SPI, it determines the device type (EEPROM or serial Flash).	If BT_MEM_CTL[1:0] = NAND Flash, then: 0 8 bit 1 16 bit
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter the delays and timing, in support of direct boot.	If BT_MEM_CTL = NAND Flash, then: 00 3 address cycles 01 4 address cycles 10 5 address cycles 11 Reserved
BT_ECC_SEL	Defines 4 or 8-bit ECC. Also used as a fast boot mode indication for the eMMC 4.3 protocol.	0 4-bit ECC 1 8-bit ECC

Because MLC NAND Flash devices do not guarantee error-free boot blocks, the i.MX35 boot code requires the first 4 Kbyte of the boot code to be duplicated in the subsequent block, to serve as a second copy option.

The boot ROM code makes use of the duplicate boot code as follows:

1. On device power-on, the boot ROM copies the first 4 Kbyte of boot code from the NAND Flash to the NFC buffer.
2. ROM code checks whether the first 4 Kbyte of boot data is copied as explained the step1. The procedure for the same is explained as follows:

- If there is no ECC (Error Checking and Correction) error, then DCD is verified.
  - If DCD verification is successful, then the rest of the boot code image is copied to the destination RAM (internal RAM or SDRAM), and secure boot is performed.
- If ECC detects an error, then the boot ROM code copies the duplicate 4 Kbyte of boot data from NAND Flash.
- If there is no error in then data copy, then the boot code image is copied to destination RAM (internal RAM or SDRAM), and secure boot is performed.
  - If an error is detected in the duplicate boot code, then the boot ROM code logs an error, and jumps to the USB/UART bootloader. The logged error can be queried through the serial protocol.

3.4.1. NAND boot devices address cycle values

In the case of NAND boot, software look-up table is not available in the boot ROM, to determine the address cycle value. This value is determined by the eFuses. The various NAND Flash configuration parameters that are obtained from the eFuses, are described in Table 2.

3.4.2. NAND boot error detection and correction

The NFC automatically generates an ECC code for both main and spare data, during the NFC data load/read to/ from the NAND Flash, and NFC updates ECC in the ECC status register. The NFC performs error detection and error correction. If the number of ECC errors does not exceed the allowable limit (four for 4-bit ECC and eight for 8-bit ECC), then the boot code corrects those errors.

On device power-on, the boot ROM copies the first 4 Kbyte of the boot code from the NAND Flash device to the NFC buffer. To accommodate possible read errors, the user is required to duplicate the first 4 Kbyte of boot code to the subsequent block, to serve as a second copy option.

The boot ROM code makes use of the duplicate boot code as follows:

- If no ECC errors are detected in first boot block, the boot execution performs the secure internal boot.
- If ECC error is detected in the first 4-Kbyte boot data in the first block, then the boot ROM code copies the 4-Kbyte boot data from the subsequent block:
  - If no error is detected in the subsequent boot block, boot flow continues performing the secure internal boot.
  - If an error is detected in the subsequent boot block, the boot ROM code logs an error, and jumps to USB/UART bootloader. The logged error can be queried through the serial protocol.

In addition to setting the boot mode pins, the pins connecting the external NAND device must be selected. Table 8 lists the IOMUX options and pins that are used for the NFC interface.

Table 8. NAND boot device IOMUX

Interface	Function	Pad name	BGA	ALT
NFC	NANDF_CE0	NF_CE0	G3	ALT0
	NANDF_CLE	NFCLE	E1	ALT0
	NANDF_ALE	NFALE	F2	ALT0
	NANDF_WE_B	NFWE_B	G2	ALT0
	NANDF_RE_B	NFRE_B	F1	ALT0
	NANDF_WP_B	NFWP_B	F4	ALT0
	NANDF_RB	NFRB	F3	ALT0
	EIM_D[15:0]	D[15:0]	—	ALT0

3.5. ATA-HDD (P-ATA) boot operation

Boot is enabled from ATA-6 compatible devices. The first 512 bytes of the offset zero in the device is reserved for this. The image should be flashed at this 512 bytes of the offset zero (corresponding to the logical block address 1 or LBA 1).

Table 9 shows the settings used by the ROM in ATA-HDD boot operation.

Table 9. ATA boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	Must be set to 0b10 ATA HDD
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter delays and timing, in support of direct boot.	If BT_MEM_CTL = ATA HDD, then: 00 Reserved 01 P-ATA HDD 10 Reserved 11 Reserved

In addition to setting the boot mode pins, the pins connecting the ATA device must be selected. Table 10 lists the IOMUX options and pins that are used for the ATA interface.

Table 10. ATA boot device IOMUX

Interface	Function	Pad Name	BGA	ALT
ATA	HDD_RESET_B	ATA_RESET_B	T6	ALT0
	HDD_DIR	ATA_BUFF_EN	T5	ALT0
	HDD_DIOW	ATA_DIOW	W6	ALT0
	HDD_DIOR	ATA_DIOR	Y6	ALT0
	HDD_CS0	ATA_CS0	V7	ALT0
	HDD_CS1	ATA_CS1	T7	ALT0
	HDD_DMARQ	ATA_DMARQ	T3	ALT0
	HDD_INTRQ	ATA_INTRQ	V2	ALT0
	HDD_DA0	ATA_DA0	R4	ALT0
	HDD_DA1	ATA_DA1	V1	ALT0
	HDD_DA2	ATA_DA2	R5	ALT0
	HDD_IORDY	ATA_IORDY	U6	ALT0
	HDD_DMACK	ATA_DMACK	V6	ALT0
	HDD_D[0:15]	ATA_DATA[0:15]	—	ALT0

3.6. Boot operation on MMC, eMMC, SD, and eSD devices

Internal boot is supported from the MMC, eMMC, SD, and eSD devices. Here the boot ROM uses a bus width of only 1 bit.

Table 11 shows the eFuse settings, corresponding input contact, and their effects in the boot configuration. The boot partition used by the device is determined by the Ext_CSD[179] internal register bit— BOOT_PARTITION_ENABLE field.

Table 11. Cards boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	Must be set to 0b11, and is required to be boot from the xMMC or xSD devices
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter the delays and timing, in support of the direct boot.	If BT_MEM_CTL = expansion card device, then: 00 SD, MMC, eMMC, or eSD 01 Reserved 10 Not used for expansion card 11 Not used for expansion card
BT_SPARE_SIZE[0] (fuse only)	Specifies the size of the spare bytes of the NAND Flash devices (BT_MEM_CTL[1:0] = NAND Flash). BT_SPARE_SIZE[0] is used as a fast boot mode indication for the eSD 2.10	If the bootable device is SD, then: n0FAST_BOOT bit 29 in ACMD41 argument is 0 n1FAST_BOOT bit 29 in ACMD41 argument is 1

Table 11. Cards boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
	protocol.	
BT_SDMMC_SRC (fuse only)	Chooses the specific eSDHC controller, used for boot.	00 eSDHC-1 01 eSDHC-2 10 eSDHC-3
HAB_CUS[1:0] (fuse only)	Represents the HAB customer code, and selects the customer code, which is the input to the HAB.	Used by the HAB library.

3.6.1. MMC and eMMC card identification and initialization

During the identification and initialization phase, both the MMC and eMMC follow the same sequence:

1. The card bus frequency is set to 312.5 kHz.
2. The card voltage validation is performed. High voltage settings are checked.
3. The card capacity is checked (both high and low-capacity MMC/eMMC cards are supported).

After the identification and initialization are completed, the boot code switches its frequency to the card bus frequency of 16.666 MHz.

3.6.2. eMMC card boot partition and special boot mode

For eMMC cards, the boot partition can be selected, after the card initialization. The ROM reads the BOOT_PARTITION_ENABLE field in the internal register—Ext_CSD[179], to determine the boot partition. If no boot partition is specified, then, by default, the ROM boots from the user partition.

The eMMC devices support a special feature that identifies the eMMC device during the boot. This feature is selected by the BT_ECC_SEL fuse, and is supported by the eMMC devices in eSDHC-1, eSDHC-2, and eSDHC-3. This feature is initiated by pulling the CMD line low, when the BOOT ACK is enabled. The steps are as follows:

1. The eMMC device sends the BOOT ACK to the ROM, through DATA0.
2. The ROM waits for 50 ms for the BOOT ACK [S010E].
   - If the BOOT ACK is received, then the eMMC is booted in the boot mode.
   - If BOOT ACK is not received, then the device boots as a normal MMC card, from the selected boot partition.

3.6.3. SD and eSD boot operation

During the identification and initialization phase, the SD and eSD follow the same sequence as follows:

1. The card bus frequency is set to 312.5 kHz.
2. The card voltage validation is performed, and the high voltage settings are checked. If the check fails, then the low voltage settings are checked.
3. The card capacity is checked (both the high and low-capacity SD/eSD cards are supported).

During the card initialization, the boot code tries to set the boot partition for all the SD or eSD devices. If this fails, the boot code assumes the card is a normal SD card. For a normal SD card of an eSD device, without a boot partition, this step returns as fail. After the identification and initialization are completed, the boot code switches its frequency to the card bus frequency of 16.666 MHz.

The ROM also supports FAST_BOOT boot mode for the eSD cards. This mode can be selected by the BT_SPARE_SIZE[0] fuse. But, unlike the MMC cards, the boot ROM cannot switch from 1-bit access mode to 4 or 8-bit access mode, in the case of SD cards.

In addition to setting the boot mode pins, the pins connecting the card device must be selected. Table 12 and Cross Refsable list the IOMUX options and pins that are used for the card interface.

Table 12. Card 8-bit boot device IOMUX

Interface	Function	Pad name	BGA	ALT
eSDHC1	SD1_CMD	SD1_CMD	Y19	ALT0
	SD1_CLK	SD1_CLK	V18	ALT0
	SD1_DATA0	SD1_DATA0	R14	ALT0
	SD1_DATA1	SD1_DATA1	U16	ALT0
	SD1_DATA2	SD1_DATA2	W18	ALT0
	SD1_DATA3	SD1_DATA3	V17	ALT0
	SD1_DATA4	SD2_CMD	U13	ALT2
	SD1_DATA5	SD2_CLK	W14	ALT2
	SD1_DATA6	SD2_DATA0	V13	ALT2
	SD1_DATA7	SD2_DATA1	T13	ALT2

Table 13. Card 8-bit boot device IOMUX

Interface	Function	Pad name	BGA	ALT
eSDHC1	SD1_CMD	SD1_CMD	Y19	ALT0
	SD1_CLK	SD1_CLK	V18	ALT0
	SD1_DATA0	SD1_DATA0	R14	ALT0
	SD1_DATA1	SD1_DATA1	U16	ALT0
	SD1_DATA2	SD1_DATA2	W18	ALT0
	SD1_DATA3	SD1_DATA3	V17	ALT0
eSDHC2	SD2_CMD	SD2_CMD	U13	ALT0
	SD2_CLK	SD2_CLK	U14	ALT0
	SD2_DATA0	SD2_DATA0	V13	ALT0
	SD2_DATA1	SD2_DATA1	T13	ALT0
	SD2_DATA2	SD2_DATA2	Y14	ALT0
	SD2_DATA3	SD2_DATA3	U12	ALT0
eSDHC3	SD3_CMD	LD18	K18	ALT3
	SD3_CLK	LD19	K20	ALT3
	SD3_DATA0	LD20	K16	ALT3
	SD3_DATA1	LD21	K17	ALT3
	SD3_DATA2	LD22	K15	ALT3
	SD3_DATA3	LD23	L19	ALT3

3.7. Serial ROM boot operation

The i.MX35 supports boot from serial memory devices, such as EEPROM and serial Flash, through the SPI (CSPI-1, at chip select #1) and I^2C (I^2C-1) interfaces. The boot ROM determines the device type and interface, according to the eFuse (or equivalent input contact) settings, as shown in Table 14.

Table 14. Serial boot configuration

Fuse/GPIO	Definition	Settings for each combination of bits
BT_MEM_CTL[1:0]	Boot memory control type (memory device)	Must be 0b11, for serial ROM
BT_BUS_WIDTH	Represents bus width, and is used in conjunction with the BT_MEM_CTL[1:0] setting. For SPI, it determines the device type (EEPROM or serial Flash).	If BT_MEM_CTL[1:0] = expansion device (SPI), then: 0 2-bytes address SPI device (EEPROM) 1 3-bytes address SPI device (serial Flash)
BT_MEM_TYPE[1:0]	Represents boot memory type, and is interpreted by the boot ROM software according to the BT_MEM_CTL setting. Signals are interpreted by the hardware to alter the delays and timing, in support of the direct boot.	If BT_MEM_CTL = serial, then: 00 Not used for serial ROM 01 Reserved 10 Serial ROM through I^2C 11 Serial ROM through SPI

3.7.1. Serial ROM boot through SPI

For SPI boot, the serial ROM must reside in the chip select #1 of the CSPI-1 module. Serial ROM type (EEPROM or serial Flash) is determined by the BT_BUS_WIDTH setting, as shown in Table 14.

In addition to setting the boot mode pins, the pins connecting the CSPI-1 device must be selected. Table 15 lists the IOMUX options and pins that are used for the CSPI-1 interface.

Table 15. CSPI-1 boot device IOMUX

Interface	Function	Pad name	BGA	ALT
CSPI-1	CSPI1_MOSI	CSPI1_MOSI	W9	ALT0
	CSPI1_MISO	CSPI1_MISO	V9	ALT0
	CSPI1_SCLK	CSPI1_SCLK	W8	ALT0
	CSPI1_SPI_RDY	CSPI1_SPI_RDY	T8	ALT0
	CSPI1_SS0	CSPI1_SS0	Y8	ALT0
	CSPI1_SS1	CSPI1_SS1	U8	ALT0

3.7.2. Serial ROM boot through I^2C

For I^2C boot, the serial ROM must be connected to the I^2C-1 module. Table 16 lists the IOMUX options and pins that are used for the I^2C-1 interface.

Table 16. I^2C-1 boot device IOMUX

Interface	Function	Pad name	BGA	ALT
I^2C-1	I2C1_CLK	I2C1_CLK	M20	ALT0
	I2C1_DAT	I2C1_DAT	N17	ALT0

Table 17 shows the device select codes used by the device to boot from EEPROM.

Table 17. EEPROM through I^2C device select code

	Device type identifier				Chip enable address			RW
Bits	7	6	5	4	3	2	1	0
Device select code	1	0	1	0	0	0	0	RW

NOTE

During booting, the frequency should not exceed 400 kHz, for an input clock source of 32 MHz.

4. External boot mode

The external boot is selected by driving a value of 0b10 on the BMOD contacts, during the device power-up, and keeping the eFuse DIR_BT_DIS un-blown. In this mode, the core boots directly from the external memory (in the case of the WEIM) or by running the NAND Flash boot.

External boot modes are considered as non-secure by its definition—the software in flash is executed regardless of its authenticity, and the Security Controller (SCC) is automatically put into a non-secure state, so that it cannot be used to decrypt information with the device unique secret key.

The supported direct external flash types are as follows:

- NOR Flash through WEIM. The i.MX35 supports muxed or non-muxed address data boot, from the WEIM interface.
- NAND Flash boot.

For external NAND Flash boot, NAND Flash with a long reset time (tPOR) is not supported. For a successful external NAND boot, it's safe to use NAND Flash with tPOR < 0.3ms @ 24MHz crystal.

5. Serial download boot mode

The serial downloader is invoked under the following conditions:

- Serial downloader is explicitly specified (BMOD = 0b11)
- ROM in internal boot (BMOD = 0b00), and the eFuse values are not valid for any external device
- Security hardware fails
- Run-time exception occurs
- Error is returned by the HAB functions, in the production mode

To determine the active serial port (either UART or USB), MCU ROM polls the UART and USB status register for 32 seconds. If there is no activity on either of the ports, within the predefined polling loop time, then the ROM powers down the device, using the watchdog (WDOG). In the USB/UART bootloader valid case, the WDOG is serviced periodically. If the communication between the host and Baseboard IC (BB IC) hangs for more than 32 seconds or the processor goes into an endless loop, then the WDOG expires, and powers down the device.

NOTE

If a code is downloaded into the system memory using a tool such as the Advanced Tool Kit (ATK), then the WDOG has to be serviced in the code. The WDOG cannot be turned OFF, if the user has jumped out of the iROM to another piece of code.

If the UART is selected as the communication path, then the UART is configured for:

- Baud rate of 115.2 Kbps
- Parity disable
- One stop bit
- 8-bit TX-RX character length no flow control

The data is then downloaded using the serial protocol. If the USB is selected, then the USB core and external transceiver are configured.

Figure 4 shows the USB/UART boot flow.

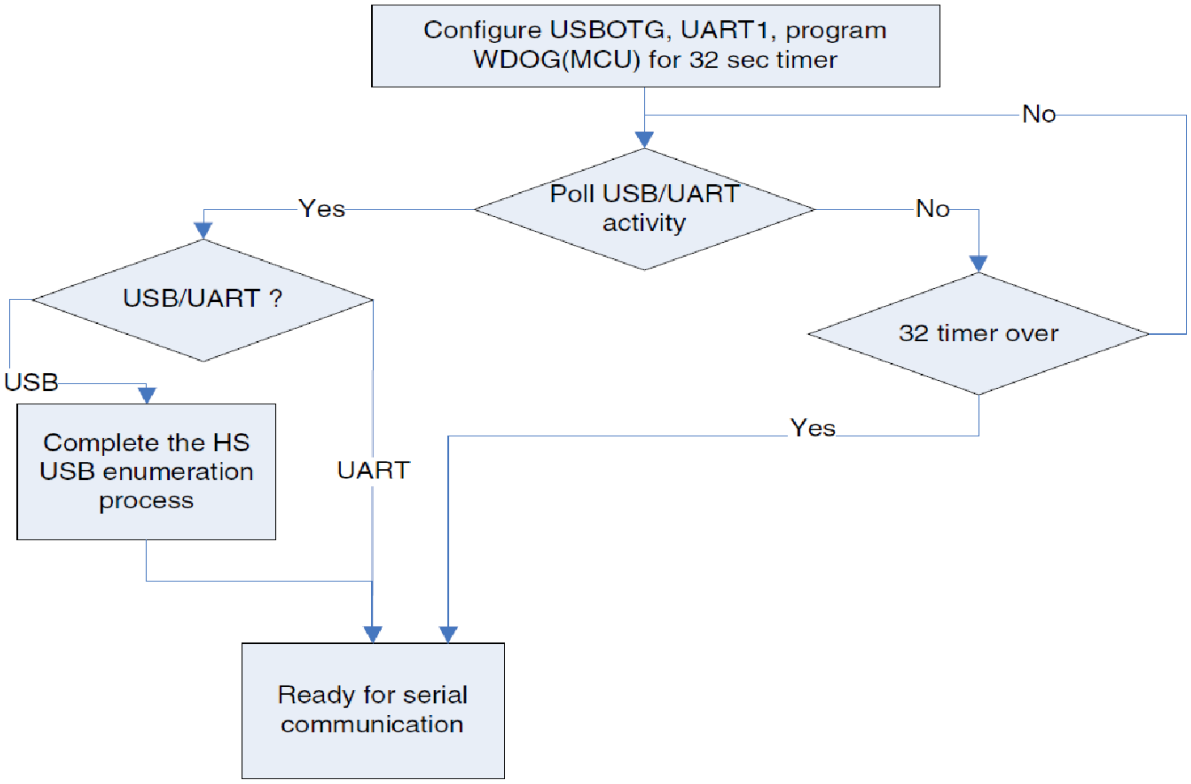

Figure 4. USB/UART boot flow

In addition to setting the boot mode pins, the pins connecting the UART device must be selected. Table 18 lists the IOMUX options and pins that are used for the UART interface.

Table 18. UART boot device IOMUX

Interface	Function	Pad name	BGA	ALT
UART-1	UART1_RX	RXD1	U2	ALT0
	UART1_TX	TXD1	R6	ALT0
	UART1_RTS	RTS1	U1	ALT0
	UART1_CTS	CTS1	R3	ALT0

5.1. USB configuration details

The i.MX35 supports high speed (HS for UTMI, ULPI) and full speed (FS for ATLAS and ISP1301), low-level USBOTG function device drivers. The USB interface is determined by the eFuse BT_USB_SRC[1:0].

Table 19 shows the supported boot modes, and the corresponding eFuse settings.

Table 19. Supported boot modes

Boot mode	BT_USB_SRC eFuse Value
UTMI PHY	BT_USB_SRC[1:0] = 00

Table 19. Supported boot modes

Boot mode	BT_USB_SRC eFuse Value
ULPI PHY	BT_USB_SRC[1:0] = 01
Serial PHY: Atlas	BT_USB_SRC[1:0] = 10
Serial PHY: Philips 1301	BT_USB_SRC[1:0] = 11

Table 20 shows the Vendor ID (VID), Product ID (PID), and strings used for the USB device driver.

Table 20. USB device driver information

Descriptor	Value	Remarks
VID	0x15A2	NXP vendor number
PID	0x0030	Allocated based on Before Part Number
String descriptor 1	NXP Semiconductor, Inc	Manufacturer
String descriptor 2	SP blank RINGO SE blank RINGO NS blank RINGO	Product
String descriptor 4	NXP flash	—
String descriptor 4	NXP flash	—

Figure 5 shows a typical USB boot flow.

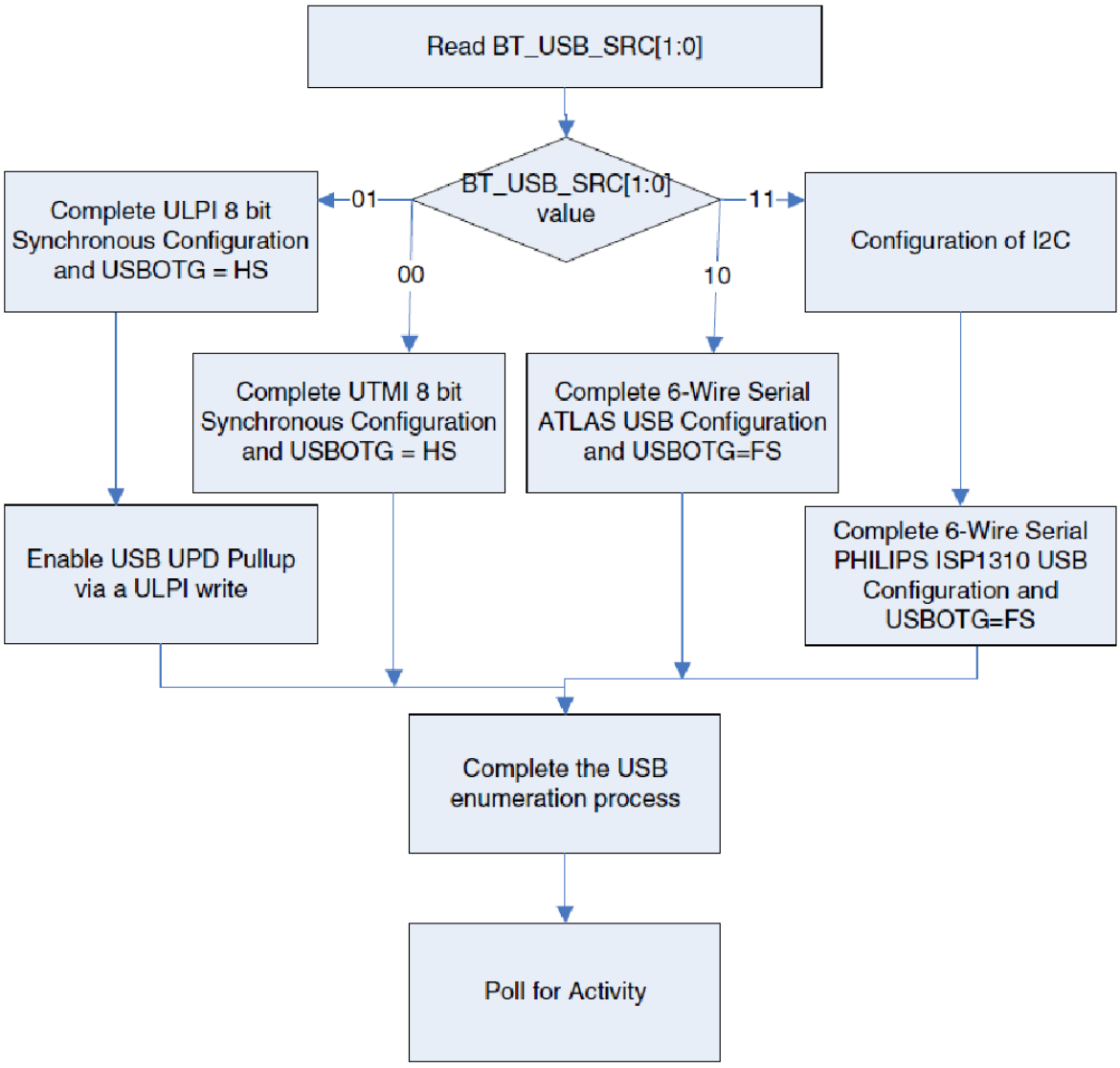

Figure 5. USB boot flow

In addition to setting the boot mode pins, the pins connecting the USB device must be selected. Table 21 lists the IOMUX options and pins that are used for the USB interface.

Table 21. USB boot device IOMUX

Interface	Function	Pad name	BGA	ALT
USB	USBOTG_CLK	GPIO2_0	U11	ALT1
	USBOTG_DIR	ATA_DIOR	Y6	ALT2
	USBOTG_STP	ATA_DIOW	W6	ALT2
	USBOTG_NXT	ATA_DMACK	V6	ALT2
	USBOTG_DATA[7]	ATA_DATA5	W4	ALT2
	USBOTG_DATA[6]	ATA_DATA4	Y4	ALT2
	USBOTG_DATA[5]	ATA_DATA3	U5	ALT2
	USBOTG_DATA[4]	ATA_DATA2	V5	ALT2
	USBOTG_DATA[3]	ATA_DATA1	W5	ALT2

Table 21. USB boot device IOMUX

Interface	Function	Pad name	BGA	ALT
USB	USBOTG_DATA[2]	ATA_DATA0	Y5	ALT2
	USBOTG_DATA[1]	ATA_IORDY	U6	ALT2
	USBOTG_DATA[0]	ATA_RESET_B	T6	ALT2

6. Serial download protocol

This section describes the serial download protocol used for all the boot devices in the serial bootloader, which is in the i.MX35. Each stage in the protocol begins with a command, issued by the host to the device, followed by a response from the device to the host. For most of the commands, this procedure completes the protocol stage. The exception is the Write File command, which has an additional data stream, sent from the host to the device, after the response.

The protocol is terminated by the host, by issuing a Write File command, with the application—file type. After processing the Write File commands, the device interprets the next command as a Completed command, and resumes the execution of the boot flow (if required to authenticate the downloaded application), and then execute it.

6.1. Get Status command

The Get Status command retrieves the error log stored, during a failed boot (or the success code when the serial bootloader is deliberately selected).

Table 22 shows the Get Status commands and responses.

Table 22. Get Status command

Command	0x05	0x05	—	—	—	—	—	—	—	—	—
Response	SC	SC	SC	SC	SC	N/A	N/A	N/A	N/A	N/A	N/A

The fields in Table 22 have the following interpretation:

- SC—Status code, and four copies of the status code are sent
- —Don't care
- N/A—No response

6.2. Read Memory command

The Read Memory command reads stream of bytes, half-words or words of data, starting from a given address in the memory. At production level security, this command is ignored.

Table 23 shows the Read Memory command.

Table 23. Read Memory command

Command	0x01	0x01	A[3:0]	DS	C[3:0]	—	—	—	—	—
Response (success)	ACK[3:0]		Data stream starting from lowest address							
Response (failure)	ACK[3:0]		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

The fields in Table 23 have the following interpretation:

- A[3:0]—Target address, starting with the most significant byte.
- DS—Data size with values 0x08—byte, 0x10—half-word, and 0x20—word. Unsupported DS values results in response failure.
- C[3:0]—Number of data elements (bytes, half-words or words) in the data stream, starting with the most significant byte.
- ACK[3:0]—Takes the values, 0x56, 0x78, 0x78, 0x56.
- —Don't care.
- N/A—No response.

6.3. Write Memory command

The Write Memory command, shown in Table 24, writes a byte, half-word or word of the data, to a given address in the memory.

Table 24. Write Memory command

Command	0x02	0x02	A[3:0]	DS	—	—	—	—	D[3:0]	—
Response (success)	ACK[3:0]		0x2	0x8A	0x8A	0x12	N/A	N/A	N/A	N/A
Response (failure)	ACK[3:0]		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

The fields in Table 24 have the following interpretation:

- A[3:0]—Target address, starting with the most significant byte.
- DS—Data size with values 0x08—byte, 0x10—half-word, and 0x20—word. Unsupported DS values results in response failure.
- D[3:0]—Data to be written, starting with the most significant byte. For DS = 0x08, only D[0] is used, and for DS = 0x10, only D[1:0] is used.
- ACK[3:0]—Takes the values, 0x12, 0x34, 0x34, 0x12 for production-level security, and the values, 0x56, 0x78, 0x78, 0x56 otherwise.
- —Don't care.
- N/A—No response.

At production level security, the address ranges that are written are restricted to those listed for DCD. Attempts to write outside the valid ranges are ignored, and results in response failure. For other security levels, no restrictions apply to the target address.

6.4. Re-Enumerate command

The Re-Enumerate command resets the USB connection with an updated descriptor.

Table 25 shows the Re-Enumerate commands and responses.

Table 25. Re-Enumerate command

Command	0x05	0x05	—	—	—	—	—	—	—	—	—
Response	SC	SC	SC	SC	SC	N/A	N/A	N/A	N/A	N/A	N/A

The fields in Table 25 have the following interpretation:

- SN[3:0]—Serial number for enumeration descriptor
- —Don't care
- N/A—No response

6.5. Write File command

The Write File command writes stream of bytes to a given address in the memory. The byte stream may be assigned a file type, to distinguish between Command Sequence File (CSF), DCD, and application files. Application file type leads to the termination of the serial download protocol with the next command (whatever the command byte values) being treated as the Completed command.

Table 26 shows the Write File command.

Table 26. Write File command

Command	0x05	0x05	A[3:0]	—	C[3:0]	—	—	—	—	FT
Response	ACK[3:0]		N/A							
Data	Byte stream with data for lowest address first.									

The fields in Table 26 have the following interpretation:

- A[3:0]—Target address, starting with the most significant byte.
- C[3:0]—Number of bytes in the data stream, starting with the most significant byte.
- FT—File type with the values 0xAA—application (terminates protocol), 0xCC—CSF, and 0xEE—DCD. With the unrecognized FT values, the file is downloaded, but the pointers to the three essential files are not modified.
- ACK[3:0]—Takes the values, 0x12, 0x34, 0x34, 0x12 for production-level security, and the values 0x56, 0x78, 0x78, 0x56, otherwise.
- —Don't care.
- N/A—No response.

At production level security, the address ranges written are restricted to those listed for DCD. Attempts to write outside the valid ranges are ignored, and results in response failure. For the other security levels, no restrictions apply to the target address.

6.6. Completed command

The Completed command is required after the Write File command with the application file type, to terminate the protocol. The contents of the command are irrelevant, but a command must be sent. This command triggers authentication and DCD processing (if required), followed by the execution of the application. At the production-level security, if the application authentication fails, the serial download

protocol resumes with the status code for the authentication failure available through the Get Status command.

The Completed command is shown in Table 27.

Table 27. Completed command

Command	0x05	0x05	A[3:0]	—	C[3:0]	—	—	—	—	FT
Response	ACK[3:0]		N/A							
Data	Byte stream with data for lowest address first.									

The fields in Table 27 have the following interpretation:

- —Don't care
- N/A—No response

7. Revision history

Table 28. Revision history

Revision number	Date	Substantive changes
0	04/2010	Initial release
1	07/2014	Section 3.4, Update to note in “NAND Flash Boot Operation” regarding a reset command. Section 4 Additional information added to, “External Boot Mode” regarding external NAND flash boot.
2	06/2017	Section 3 “through Near Field Communication (NFC) interface” changed to “through NAND Flash Controller (NFC) interface”. Section 3.4 Note removed.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN3996
Rev. 2
06/2017