Fastboot Table

User Guide

Issue 02

Date 2015-05-20

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes how to use and configure the fastboot table. It applies to the HiSilicon Linux software development kit (SDK) and Android SDK.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3796C	V1XX
Hi3798C	V1XX
Hi3798M	V1XX
Hi3796M	V1XX
Hi3798C	V2XX

Intended Audience

This document is intended for:

- Technical support personnel
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 02 (2015-05-20)

This issue is the second official release, which incorporates the following changes:

Hi3798C V2*XX* is supported.

Chapter 6 Modification Instances

Section 6.1.3 is added, and section 6.2.1 is modified.

Issue 01 (2014-10-30)

This issue is the first official release, which incorporates the following change:

Hi3796M V100 is supported.

Issue 00B02 (2014-08-19)

This issue is the second draft release.

Issue 00B01 (2014-06-20)

This issue is the first draft release.

Contents

Figures

Tables

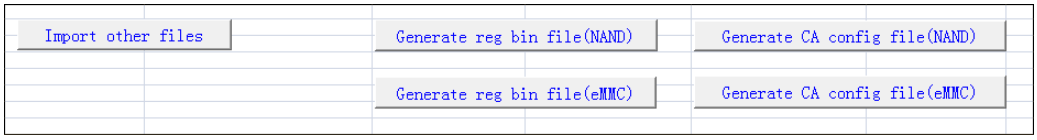
1 Usage

CAUTION

For details about the naming conventions, applicable boards, and intended customers of the fastboot table, see **readme_*** in the document contents of the fastboot table contained in the SDK.

The fastboot table describes register configurations. Several buttons are available in the **main** sheet, as shown in Figure 1-1.

Figure 1-1 main sheet



The button functions are described as follows:

- **Import other files**

  Imports contents from other tables. (This button is applicable only to R&D personnel.)
- **Generate reg bin file(NAND)**

  Generates the register configuration file ***_nand.reg** in the SDK for the non-CA board with the NAND flash. This file is compiled to the boot during compilation.
- **Generate reg bin file(eMMC)**

  Generates the board register configuration file ***_emmc.reg** in the SDK for the non-CA board with the eMMC. This file is compiled to the boot during compilation.
- **Generate CA config file(NAND)**

  Generates the configuration file used by CA boot for the CA board with the NAND flash.
- **Generate CA config file(eMMC)**

  Generates the configuration file used by CA boot for the CA board with the eMMC.

2 Content

The fastboot table contains register configurations, including key parameters of board hardware, phase-locked loop (PLL), double data rate (DDR), peripheral clock, pin multiplexing, and Ethernet port configurations. The fastboot table is divided into two parts: ITEM 1 and ITEM 2. Basic parameters are set in ITEM 1. ITEM 2 are reserved at present, but cannot be deleted.

The fastboot table consists of multiple sheets. Generally, one module or one mode is configured in one sheet. However, the DDR is configured in multiple sheets. The registers of each module are configured based on the system startup sequence, and the sheets are arranged by priority. You are advised not to change the sheet configuration sequence. The sheets are described as follows:

- **main**

  Provides command buttons and version upgrade records. You can fill in version information based on product requirements.

- **hardware_key_info**

  Describes key hardware information, including the dominant frequency of the CPU and DDR, basic DDR information, dynamic voltage and frequency scaling (DVFS) channel selection, basic Ethernet port information, and system driver configurations. Information in this sheet is only for reference, and you do not need to modify the information.

- **sys_clk**

  Sets the system clock ratio, other clock parameters, and CPU mode, and displays the clock frequency of the CPU and DDR. You are prohibited from modifying the information in this sheet.

- **sysctrl_noCA**

  Sets system control for the non-CA boards. Do not modify the information (except pin multiplexing) in this sheet. Otherwise, the system may fail.

- **sysctrl_CA** (optional)

  Sets system control for the CA boards. Do not modify the information (except pin multiplexing of the front panel) in this sheet. Otherwise, the system may fail.

  Note that **sysctrl_noCA** or **sysctrl_CA** is compiled to generate the image based on whether the board is a non-CA board or a CA board. If pin multiplexing of the front panel needs to be modified, you are advised to modify both sheets.

- **mddrc**, **ddrphy**, **ddr_poweron**, and **ddr_wakeup**

Sets DDR initialization, including DDR controller and DDR PHY settings. Do not modify the information in this sheet. Otherwise, the system may fail.

- **qos_ctrl**

  Sets the priorities of system modules and peripheral modules. Do not modify the information in this sheet. Otherwise, the system may fail.

- **crg_ctrl**

  Controls clock gating and resetting for each module. The clock of each module is disabled by default. Do not modify the information in this sheet. Otherwise, the system may fail.

- **clk_ssmod**

  Sets the spread spectrum for each PPL. The spread spectrums of all PLLs are disabled by default. Do not modify the information in this sheet. Otherwise, the system may fail.

- **peri_cfg**

  Manages the power consumption of peripheral modules. The USB, VDAC, and ADAC modules are in the Power down state by default. Do not modify the information in this sheet. Otherwise, the system may fail.

- **pin_mux_drv_emmc**

  Configures pin multiplexing and drive capability for boards with the eMMC. Pin multiplexing is determined by specific application scenarios, and drive capability is determined by PCB layers. You can modify the information in this sheet based on the *Hi379X VXXX Hardware User Guide*.

- **pin_mux_drv_nand**

  Configures pin multiplexing and drive capability for boards with the NAND flash. Pin multiplexing is determined by specific application scenarios, and drive capability is determined by PCB layers. You can modify the information in this sheet based on the *Hi379X VXXX Hardware User Guide.*

  Note that **pin_mux_drv_emmc** or **pin_mux_drv_nand** is compiled to generate the image based on whether the flash memory is the NAND flash or eMMC. If pin multiplexing needs to be modified, you are advised to modify both sheets (the difference between the two sheets lies only in the eMMC or NAND flash configuration part).

- **eth_phy_cfg**

  Sets Ethernet ports, including settings of the MAC channels, MAC port types, PHY addresses, MDC/MDIO control (not for Hi3798M V100), and external PHY reset signals (not for Hi3798M V100). You can modify the information in this sheet based on product requirements.

- **others**

  Describes ITEM 2 (reserved at present).

3 Function Description

3.1 Overview

This section describes a sheet of the fastboot table, as shown in Figure 3-1.

Figure 3-1 Configuring peripherals

Module Name	mddrc	MDDRC Initialize					
Base Address	0xf8a31000	Add module				Add register	
ITEM1/2	1						
Priority	3						
Execution Required for Standby Wakeup	Y						
Execution Required for Normal Boot	Y						
Register	Offset Address	Value Written to or Read from Register	delay	Read or Write	Bits to Be Read or Written	Start Bit to Be Read or Written	Register Attribute
DDRC_CFG_INIT	0x24	0x8	0	write	31	0	0x0000000FF
DDRC_CFG_AREF	0x2c	0x0	0	write	31	0	0x0000000FF
DDRC_CFG_WORKMODE	0x40	0x11002001	0	write	31	0	0x0000000FF
DDRC_CFG_DDRMODE	0x50	0x26	0	write	31	0	0x0000000FF
DDRC_CFG_RNKVOL	0x60	0x152	0	write	31	0	0x0000000FF
DDRC_CFG_EMRS01	0x70	0x61f70	0	write	31	0	0x0000000FF
DDRC_CFG_EMRS23	0x74	0x18	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING0	0x80	0x4639d610	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING1	0x84	0x3f38b080	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING2	0x88	0x44016000	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING3	0x8c	0xffd1f784	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING4	0x90	0x820f18	0	write	31	0	0x0000000FF
DDRC_CFG_TIMING5	0x94	0x2707	0	write	31	0	0x0000000FF
DDRC_CFG_ODT	0xc0	0x1	0	write	31	0	0x0000000FF
DDRC_CFG_DMCLVL	0xc4	0xc	0	write	31	0	0x0000000FF
DDRC_CFG_DDRPHY	0x200	0x1000	0	write	31	0	0x0000000FF
DDRC_CFG_SREF	0x20	0x8101	0	write	31	0	0x0000000FF

Items in the sheet shown in Figure 3-1 are described as follows:

- Two buttons
  - **Add module**: Clicking this button adds a sheet on the right of the current sheet.
  - **Add register**: Clicking this button adds a row under the current row.
- **Module Name**

  Indicates the usage of this sheet and the module to be configured. Generally, **Module Name** is the same as the name of the current sheet.
- **Base Address**

Indicates the base address for registers. In this sheet, only the offset address for each register is provided. The actual address for each register is the value specified in **Base Address** plus the corresponding offset address.

- **Priority**

  Specifies the configuration sequence. The priority of each sheet is unique and is numbered from 0.

- **Execution Required for Standby Wakeup**

  Indicates whether the configurations in the current sheet are required when the system wakes up from the standby mode.

  - **Y**: required
  - **N**: not required

- **Execution Required for Normal Boot**

  Indicates whether the configurations in the current sheet are required when the system is powered on.

- **Register**

  Indicates the register name, for facilitating maintenance.

- **Offset Address**

  Indicates the offset address for each register relative to the base address.

- **Value Written to or Read from Register**

  Indicates the value to be written or read for comparison.

- **delay**

  Indicates the delay time after a register is configured.

- **Read or Write**

  Specifies the write or read operation.

- **Bits to Be Read or Written**

  Indicates the number of bits to be read or written. The value is the number of actual bits to be read or written minus 1. For example, if 32 bits need to be written, the value is **31**; if one bit needs to be written, the value is **0**.

- **Start Bit to Be Read or Written**

  Indicates the start bit to be read or written. For example, if you want to write data from bit 0, the value is **0**, and so on.

- **Register Attribute**

  Indicates the register attribute. The contents in this column are automatically generated and can be ignored.

3.2 Pin Multiplexing and Drive Capability

For the Hi3798C/Hi3796C, the valid contents of pin multiplexing registers include pin multiplexing (bit[3:0]) and drive capabilities (bit[7:4]), which are configured in different registers. Registers in the upper part of the **pin_mux_drv_nand** and **pin_mux_drv_emmc** sheets are used for configuring pin multiplexing, and those in the lower part are used for configuring drive capabilities. See Figure 3-2.

Figure 3-2 pin_mux_drv (for the Hi3798C/Hi3796C)

demo_gpio0	0x0000019C	0x00000002	0	write	7	0	0x0000003F	GPIO	GPIO14_0	AT41	
demo_gpio1	0x000001A0	0x00000002	0	write	7	0	0x0000003F		GPIO14_1	AR39	
demo_gpio3	0x000001A8	0x00000002	0	write	7	0	0x0000003F		GPIO14_3	AW39	
demo_gpio4	0x000001AC	0x00000002	0	write	7	0	0x0000003F		GPIO14_4	AY40	
demo_gpio5	0x000001B0	0x00000002	0	write	7	0	0x0000003F		GPIO14_5	AR40	
demo_gpio6	0x000001B4	0x00000002	0	write	7	0	0x0000003F		GPIO14_6	AR38	
i2s0_gpio47	0x000001B8	0x00000002	0	write	7	0	0x0000003F		GPIO14_7	AW2	
i2s0_gpio97	0x000001BC	0x00000002	0	write	7	0	0x0000003F		GPIO9_7	AY3	
i2s0_gpio100	0x000001C0	0x00000002	0	write	7	0	0x0000003F		GPIO10_0	AY4	
i2s0_bootsel1	0x000001C4	0x00000003	0	write	7	0	0x0000003F		GPIO10_1	BA4	
i2s0_gpio35	0x000001CC	0x00000002	0	write	7	0	0x0000003F		GPIO13_5	AY5	
i2s1_gpio27	0x000001D8	0x00000002	0	write	7	0	0x0000003F		GPIO12_7	C28	
i2s1_gpio30	0x000001DC	0x00000002	0	write	7	0	0x0000003F		GPIO13_0	A29	
i2s1_bootsel0	0x000001E0	0x00000002	0	write	7	0	0x0000003F		GPIO13_1	A28	
i2c0_gpio32	0x000001E4	0x00000002	0	write	7	0	0x0000003F		GPIO13_2	D28	
i2c0_gpio33	0x000001E8	0x00000002	0	write	7	0	0x0000003F		GPIO13_3	B28	
sfc_dio_ctrl	0x00000800	0x000000a0	0	write	7	0	0x0000003F	SPI drive cfg	SFC_DIO	FAST	10
sfc_wpn_ctrl	0x00000804	0x000000a0	0	write	7	0	0x0000003F		SFC_WPN_IO2	FAST	10
sfc_clk_ctrl	0x00000808	0x000000a0	0	write	7	0	0x0000003F		SFC_CLK	FAST	10
sfc_doi_ctrl	0x0000080C	0x000000a0	0	write	7	0	0x0000003F		SFC_DOI	FAST	10
sfc_holdn_ctrl	0x00000810	0x000000a0	0	write	7	0	0x0000003F		SFC_HOLDN_IO3	FAST	10
sfc_csln_ctrl	0x00000814	0x000000b0	0	write	7	0	0x0000003F		SFC_CS1N	FAST	11

For the Hi3798M, the valid contents of pin multiplexing registers include pin multiplexing (bit[7:0]) and drive capabilities (bit[11:8]), which are configured in different registers. **pin_mux_drv_nand** and **pin_mux_drv_emmc** list the pin multiplexing and drive capability configuration registers. See Figure 3-3.

Figure 3-3 pin_mux_drv (for the Hi3798M)

Register	Offset Address	Value Written to or Read from Register	delay	Read or Write	Bits to Be Read or Written	Start Bit to Be Read or Written	Register Attribute
ioshare_0	0x0	0x1f01	0	write	11	0	0x0000005F
ioshare_1	0x4	0xf01	0	write	11	0	0x0000005F
ioshare_2	0x8	0xf01	0	write	11	0	0x0000005F
ioshare_3	0xc	0xf01	0	write	11	0	0x0000005F
ioshare_4	0x10	0xf01	0	write	11	0	0x0000005F
ioshare_5	0x14	0xf01	0	write	11	0	0x0000005F
ioshare_6	0x18	0xf01	0	write	11	0	0x0000005F
ioshare_7	0x1c	0xf01	0	write	11	0	0x0000005F
ioshare_8	0x20	0x701	0	write	11	0	0x0000005F
ioshare_9	0x24	0x701	0	write	11	0	0x0000005F
ioshare_10	0x28	0x701	0	write	11	0	0x0000005F
ioshare_11	0x2c	0xf01	0	write	11	0	0x0000005F
ioshare_12	0x30	0xd01	0	write	11	0	0x0000005F
ioshare_13	0x34	0x701	0	write	11	0	0x0000005F
ioshare_21	0x54	0x700	0	write	11	0	0x0000005F
ioshare_22	0x58	0x700	0	write	11	0	0x0000005F
ioshare_23	0x5c	0xf00	0	write	11	0	0x0000005F
ioshare_24	0x60	0x500	0	write	11	0	0x0000005F
ioshare_25	0x64	0x300	0	write	11	0	0x0000005F
ioshare_31	0x7c	0xf00	0	write	11	0	0x0000005F
ioshare_32	0x80	0xf00	0	write	11	0	0x0000005F
ioshare_43	0xac	0xf01	0	write	11	0	0x0000005F
ioshare_44	0xb0	0xf00	0	write	11	0	0x0000005F
ioshare_45	0xb4	0x1	0	write	7	0	0x0000003F
ioshare_46	0xb8	0x1	0	write	7	0	0x0000003F
ioshare_47	0xbc	0x1	0	write	7	0	0x0000003F
ioshare_48	0xc0	0x1	0	write	7	0	0x0000003F
ioshare_54	0xd8	0xf01	0	write	11	0	0x0000005F
ioshare_55	0xdc	0xf01	0	write	11	0	0x0000005F
ioshare_56	0xe0	0x701	0	write	11	0	0x0000005F
ioshare_57	0xe4	0x701	0	write	11	0	0x0000005F
ioshare_58	0xe8	0x701	0	write	11	0	0x0000005F
ioshare_59	0xec	0x501	0	write	11	0	0x0000005F
ioshare_60	0xf0	0x701	0	write	11	0	0x0000005F
ioshare_61	0xf4	0x701	0	write	11	0	0x0000005F
ioshare_62	0xf8	0x701	0	write	11	0	0x0000005F
ioshare_63	0xfc	0x701	0	write	11	0	0x0000005F
ioshare_64	0x100	0x701	0	write	11	0	0x0000005F

- For I/O with fixed drive capability, only pin multiplexing is available (determined by the specified application scenarios). In the fastboot table, set **Start Bit to Be Read or Written** to **0** and **Bits to Be Read or Written** to **7** for bit[7:0]. A total of 8 bits are set.
- For I/O with adjustable drive capability, pin multiplexing and drive capability are available. The drive capability must be set based on the PCB layers. In the fastboot table, set **Start Bit to Be Read or Written** to **0** and **Bits to Be Read or Written** to **11** for bit[11:0]. A total of 12 bits are set.

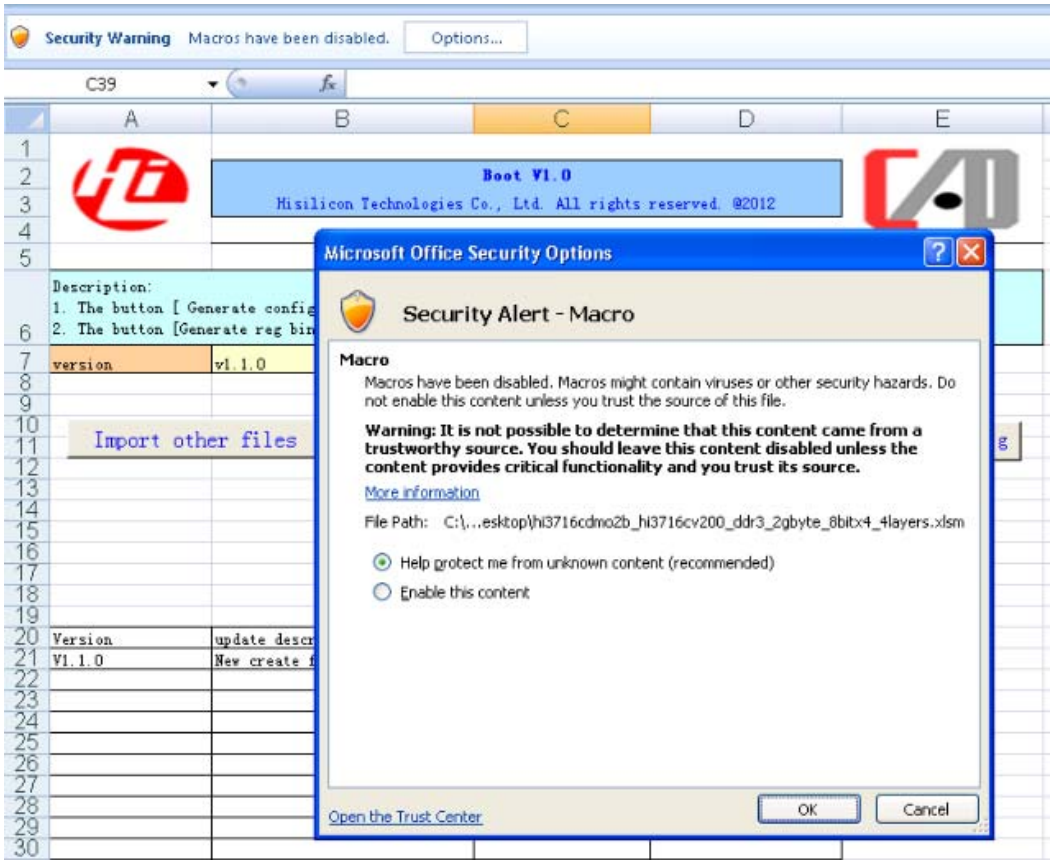
4 Precautions

Take the following precautions before you use or modify the fastboot table:

- Table usage

  Enable macros before you click any button in the fastboot table, as shown in Figure 4-1.

Figure 4-1 Enabling macros



- Table modification

  Generally, you need to modify only pin multiplexing, drive capability, and Ethernet port configurations in **pin_mux_drv_nand/pin_mux_drv_emmc** and **eth_phy_cfg** of the

fastboot table based on the hardware connections and PCB layers of products. Do not modify other information. Otherwise, the system may fail. If a customer requests to modify other information, contact HiSilicon FAEs.

- Ensure that the priorities of sheets are consecutive, each sheet name is the same as the value specified in **Module Name**, and no blank row is left in each sheet.
- Add or modify the information in the fastboot table by using macros.

5 Operation Procedure

To compile a boot, perform the following steps:

Step 1 Generate the **x**.**reg** file.

Open the boot table corresponding to the used board.

In the **main** sheet, click **Generate reg bin file(NAND)** or **Generate reg bin file(eMMC)**. The **x.reg** file is generated in the current directory.

Step 2 Compile the file.

- Linux
  - Store the generated **x.reg** file to **${LINUX_SDK}/source/boot/sysreg/**. (Skip this step if the file is generated in this directory.)
  - Run **make hiboot_clean;make hiboot_install** in the **${LINUX_SDK}** directory to compile the file.
  - After compilation, the **fastboot-burn.bin** file is generated in **${LINUX_SDK}/pub/image**. Burn the file directly.
- Android
  - Store the generated **x.reg** file to **${ANDROID_SDK}/device/hisilicon/bigfish/sdk/source/boot/sysreg/**. (Skip this step if the file is generated in this directory.)
  - Compile the file in the **${ANDROID_SDK}** directory by running the following commands:

    ```
    source build/envsetup.sh
    lunch configuration
    make hiboot
    ```

  - After compilation, the **fastboot.bin** file is generated in **${ ANDROID_SDK }/ out/target/product/${*configured product*}/Emmc** (or **Nand**)/. Burn the file directly.

----End

6 Modification Instances

6.1 Configuring Ethernet Ports

6.1.1 Hi3798C V100/Hi3796C V100

This section describes how to configure Ethernet ports for the Hi3798C V100. The Hi3798C V100 provides two internal MAC modules and supports two Ethernet ports at a time, for example:

- One external GE PHY port
- One external FE PHY port and one external GE PHY port

In the fastboot configuration table, configurations related to the Ethernet ports include the following:

- MAC port type. For MII0, MII, RMII, and RGMII can be selected; for MII1, only RMII can be selected. Hi3798C V100 uses the MII mode by default.
- PHY address corresponding to the MAC. The external PHY addresses must be consistent with the hardware.
- MDCK/MDIO of the PHY corresponding to the MAC (MDCK0/MDIO0 or MDCK1/MDIO1). Hi3798C V100 uses MDCK0/MDIO0 by default.
- External PHY reset pin GPIO. If the GPIO signal is used as the reset signal, the GPIO group ID and GPIO ID must be specified.
- Pin multiplexing and drive capability configuration, which are related to the chip interfaces and number of PCB layers

For example:

- Select the embedded GE PHY for MAC0.
- Select MDCK0/MDIO0 as the control signal.
- Select GPIO13_0 as the external GE PHY reset pin.
- Select the 4-layer PCB.
- Pin multiplexing is not configured, which is RGMII by default.

Figure 6-1 shows the corresponding Ethernet port configurations.

Figure 6-1 eth_phy_cfg

Module Name	eth_phy_cfg	eth_phy_cfg		Add module			Add register	
Base Address	0xf8000000							
ITEM1/2	1							
Priority	14							
Execution Required for Standby Wakeup	Y							
Execution Required for Normal Boot	Y							
Register	Offset Address	Value Written to or Read from Register	delay	Read or Write	Bits to Be Read or Written	Start Bit to Be Read or Written	Register Attribute	
MAC0_IF	0x184300c	0x1	0	write	3	5	0x0000281F	MAC0 Select MII Interface 5:7(000: GMII/MII mode; 001: RGMII mode; 100: RMII mode。
MAC0_PHY_ADDR	0xA8	0x1	0	write	7	0	0x0000003F	MAC0_PHY_ADDR = 0x1, range[1--31]
MAC0_PHY_CTRL	0xA8	0x0	0	write	7	8	0x0000403F	MAC0_PHY_Select MDIO0/MDCK0; 0 or 1
MAC0_PHY_RST_GPIO	0xAC	0xD	0	write	7	0	0x0000003F	MAC0 PHY RST GPIO Group Num, fill 0xFF if not used。
MAC0_PHY_RST_GPIO_BIT	0xAC	0x0	0	write	7	8	0x0000403F	MAC0 PHY RST GPIO Bit Num, fill 0xFF if not used
MAC1_PHY_CTRL	0xA8	0x1	0	write	7	24	0x00000C03F	MAC1_PHY_Select MDIO1/MDCK1; 0 or 1

The configuration items on the **eth_phy_cfg** sheet are described as follows:

- Row 8

  The MAC0 interface type is set to RGMII.
- Row 9

  The address for the embedded GE PHY is set to **1**.
- Row 10

  The GE PHY uses MDCK0/MDIO0 as the control signal.
- Rows 11 to 12

  The reset GPIO pin of the PHY corresponding to MAC0 is not required, which is set to the reserved value **0xFF**.
- Row 13

  MAC1 uses MDCK1/MDIO1 as the control signal.

6.1.2 Hi3798M V100

Hi3798M V100 has an embedded FE PHY and supports one Ethernet port.

In the fastboot configuration table, configurations related to the Ethernet ports include the following:

- PHY address corresponding to the MAC. The external PHY addresses must be consistent with the hardware (0x1 for Hi3798M V100).
- MDCK/MDIO of the PHY corresponding to the MAC (0x0 for Hi3798M V100)
- External PHY reset pin GPIO. If the GPIO signal is used as the reset signal, the GPIO group ID and GPIO ID must be specified (0xFF for Hi3798M V100).
- Pin multiplexing and drive capability configuration, which are related to the chip interfaces and number of PCB layers (not required for Hi3798M V100)

For example:

- Select the embedded GE PHY for MAC0.
- Select MDCK0/MDIO0 as the control signal.
- Select GPIO13_0 as the external GE PHY reset pin.
- Select the 4-layer PCB.

- Pin multiplexing is not configured, which is RGMII by default.

Figure 6-2 shows the corresponding Ethernet port configurations.

Figure 6-2 eth_phy_cfg

1	Module Name	int_fephy	when use internal fephy					
2	Base Address	0xf0000000		Add module			Add register	
3	ITEM1/2	2						
4	Priority	13						
5	Execution Required for Standby Wakeup	Y						
6	Execution Required for Normal Boot	Y						
7	Register	Offset Address	Value Written to or Read from Register	delay	Read or Write	Bits to Be Read or Written	Start Bit to Be Read or Written	Register Attribute
8	PERI_CTRL	0x8a20008	0x0	0	write	0	8	0x00004007
9	MAC0_IF	0x984300c	0x0	0	write	2	5	0x00002817
10	PHY_CTRL_ADDR	0x80000A8	0x2	0	write	7	0	0x0000003F
11	PHY_CTRL_ADDR	0x80000A8	0x0	0	write	7	8	0x0000403F
12	MAC0_PHY_RST_GPIO	0x80000AC	0xff	0	write	7	0	0x0000003F
13	MAC0_PHY_RST_GPIO_BIT	0x80000AC	0xff	0	write	7	8	0x0000403F
14	PERI_CTRL	0x8a20008	0x0	0	write	0	9	0x00004807
15	MAC1_IF	0x9843010	0x1	0	write	2	5	0x00002817
16	PHY_CTRL_ADDR	0x80000A8	0x1	0	write	7	16	0x00000803F
17	PHY_CTRL_ADDR	0x80000A8	0x1	0	write	7	24	0x00000C03F
18	MAC1_PHY_RST_GPIO	0x80000AC	0x5	0	write	7	16	0x00000803F
19	MAC1_PHY_RST_GPIO_BIT	0x80000AC	0x1	0	write	7	24	0x00000C03F

The configuration items on the **eth_phy_cfg** sheet are described as follows:

- Row 8

  The address for the embedded FE PHY is set to **1**.
- Row 9

  The embedded FE PHY uses MDCK0/MDIO0 as the control signal.
- Rows 10 to 11

  The reset GPIO pin of the PHY corresponding to the MAC is not required, which is set to the reserved value **0xFF**.

6.1.3 Hi3798C V200_A

Hi3798C V200_A has three MACs and can support three Ethernet ports at the same time:

- Three external GE PHYs/FE PHYs
- One external FE PHY+Two external GE PHYs/FE PHYs

In the fastboot configuration table, configurations related to the Ethernet ports include the following:

- MAC port type. For MAC0, MII, RMII, and RGMII can be selected; for MAC1 or MAC2, only RMII or RGMII can be selected. Hi3798C V200_A uses the RGMII mode by default.
- PHY address corresponding to the MAC. The external PHY addresses must be consistent with the hardware.
- MDCK/MDIO of the PHY corresponding to the MAC (MDCK0/MDIO0 or MDCK1/MDIO1)

- External PHY reset pin GPIO. If the GPIO signal is used as the reset signal, the GPIO group ID and GPIO ID must be specified. If the default reset signal is used or the reset signal is not used, it is set to **0xFF**.
- Pin multiplexing and drive capability configurations, which are related to the chip interfaces and number of PCB layers

For example:

- Select the embedded GE PHY for MAC0.
- Select MDCK1/MDIO1 as the control signal.
- Select the dedicated RESET pin as the external GE PHY reset pin.
- Select the 4-layer PCB.
- Pin multiplexing is not configured, which is RGMII by default.

Figure 6-3 shows the corresponding Ethernet port configurations.

Figure 6-3 eth_phy_cfg

	A	B	C	D	E	F	G	H	I
1	Module Name	eth_phy_cfg	eth_phy_cfg						
2	Base Address	0xf8000000		Add module				Add register	
3	ITEM1/2	1							
4	Priority	14							
5	Execution Required for Standby Wakeup	Y							
6	Execution Required for Normal Boot	Y							
7	Register	Offset Address	Value Written to or Read from Register	delay	Read or Write	Bits to Be Read or Written	Start Bit to Be Read or Written	Register Attribute	
8	PERI_FEPHY_LDO_CTRL	0xa20844	0x18	0	write	31	0	0x0000000FF	PERI_FEPHY_LDO_CTRL: enable & vset = 4'h8
9	PERI_CTRL	0xa20008	0x1	0	write	0	8	0x00004007	MAC0 Select GRMII0 (0:FE PHY/1:RGMII0 IO)
10	MAC0_IF	0x184300c	0x1	0	write	2	5	0x00002817	MAC0 Select RGMII Interface (0-mii, 1-rgmii, 4-rmii)
11	MAC0_PHY_ADDR	0xA8	0x1	0	write	7	0	0x0000003F	MAC0_PHY_ADDR = 0x1, range[1–31] (FEPHY Addr = 0x2)
12	MAC0_PHY_CTRL	0xA8	0x1	0	write	7	8	0x0000403F	MAC0_PHY_Select MDIO1/MDCK1; 0 or 1
13	MAC0_PHY_RST_GPIO	0xAC	0xff	0	write	7	0	0x0000003F	MAC0 PHY RST GPIO Group Num, fill 0xFF if not used
14	MAC0_PHY_RST_GPIO_BIT	0xAC	0xff	0	write	7	8	0x0000403F	MAC0 PHY RST GPIO Bit Num, fill 0xFF if not used
15	MAC1_IF	0x1843010	0x1	0	write	2	5	0x00002817	MAC1 Select RGMII Interface (0-mii, 1-rgmii, 4-rmii)
16	MAC1_PHY_ADDR	0xA8	0x3	0	write	7	16	0x00000803F	MAC1_PHY_ADDR =0x3
17	MAC1_PHY_CTRL	0xA8	0x1	0	write	7	24	0x00000C03F	MAC1_PHY_Select MDIO1/MDCK1; 0 or 1
18	MAC1_PHY_RST_GPIO	0xAC	0xff	0	write	7	16	0x00000803F	MAC1 PHY RST GPIO Group Num, fill 0xFF if not used
19	MAC1_PHY_RST_GPIO_BIT	0xAC	0xff	0	write	7	24	0x00000C03F	MAC1 PHY RST GPIO Bit Num, fill 0xFF if not used
20	MAC2_IF	0xa222dc	0x1	0	write	2	5	0x00002817	MAC2 select RGMII Interface
21	MAC2_PHY_ADDR	0xA0	0x7	0	write	7	0	0x0000003F	MAC2_PHY_ADDR =0x7
22	MAC2_PHY_RST_GPIO	0xA0	0xff	0	write	7	8	0x0000403F	MAC2 PHY RST GPIO Group Num, fill 0xFF if not used
23	MAC2_PHY_RST_GPIO_BIT	0xA0	0xff	0	write	7	16	0x00000803F	MAC2 PHY RST GPIO Bit Num, fill 0xFF if not used

The configuration items on the **eth_phy_cfg** sheet are described as follows:

- Row 9

  The MAC0 interface type is set to the one that connects to the external PHY.
- Row 10

  The MAC0 interface type is set to RGMII.
- Row 11

  The address for the embedded GE PHY is set to **1**.
- Row 12

  The GE PHY uses MDCK1/MDIO1 as the control signal.
- Rows 13 to 14

  The reserved value 0xFF is used if the reset GPIO of the PHY corresponding to MAC0 is not used. The dedicated external PHY reset pin is used by default.

6.2 Viewing the DDR Capacity

6.2.1 Hi3798C V100/Hi3796C V100/Hi3798C V200_A

Take Hi3798C V100 as an example. In **hi3798cdmo1a_hi3798cv100_ddr3_16bit x4_4layers.xlsm**, four 256 Mbits x 16 (totally 2 GB) DDR SDRAMs are used. Therefore, the DDRC registers with the addresses of 0x8060 and 0x9060 are set to **0x142**. See Figure 6-4.

Figure 6-4 0x8060 and 0x9060 registers

DDRC_CFG_RNKVOL	0x8060	0x142	0	write	31	0	0x000000FF
DDRC_CFG_RNKVOL	0x9060	0x142	0	write	31	0	0x000000FF

Table 6-1 describes the mapping between the configured value and the capacity of a DDR.

Table 6-1 Mapping between the configured value and the capacity of a DDR

Capacity of a DDR	0x8060/0x9060 Register Configurations
128 Mbits x 16	0x132
256 Mbits x 16	0x142
512 Mbits x 16	0x152

In the HiSilicon reference design, the DDR capacity in the table is the maximum value supported. If the actual DDR capacity is less than the configured value, it is supported by default and you do not need to modify the table.

6.2.2 Hi3798M V100

In **hi3798mdmo1b_hi3798mv100_ddr3_16bit x 2_2layers.xlsm**, two 256 Mbits x 16 (totally 1 GB) DDR SDRAMs are used. Therefore, the DDRC register with the address of 0x60 is set to **0x152**. See Figure 6-5.

Figure 6-5 0x60 register

12	DDRC_CFG_RNKVOL	0x60	**0x152**	0	write	31	0	0x000000FF

Table 6-2 describes the mapping between the 0x60 register configurations and the DDR capacity.

Table 6-2 Mapping between the 0x60 register configurations and the DDR capacity

Capacity of a DDR	0x60 Register Configurations
128 Mbits x 16	0x142
256 Mbits x 8	0x152

In the HiSilicon reference design, the DDR capacity in the table is the maximum value supported. If the actual DDR capacity is less than the configured value, it is supported by default and you do not need to modify the table.