



**Fastboot 表格配置**

## **使用指南**

文档版本 05

发布日期 2018-10-10

版权所有 © 上海海思技术有限公司2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



**HISILICON**、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 上海海思技术有限公司

地址：            深圳市龙岗区坂田华为总部办公楼    邮编：518129

网址：            <http://www.hisilicon.com/cn/>

客户服务邮箱：  [support@hisilicon.com](mailto:support@hisilicon.com)



## 前言

## 概述

本文档主要介绍Fastboot表格的使用方法，如何配置，方便客户使用。此文档适用于海思Linux SDK平台和Android SDK平台。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3796C芯片	V1XX
Hi3798C芯片	V1XX
Hi3798M芯片	V1XX
Hi3796M芯片	V1XX
Hi3798C芯片	V2XX
Hi3798M芯片	V2XX(H)
Hi3796M芯片	V2XX
Hi3716M芯片	V43X
Hi3798M芯片	V3XX(H)

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师



## 作者信息

章节号	章节名称	作者信息
全文	全文	Z00182267/T00171014

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2014-06-30	00B01	第1次临时版本发布。
2014-07-23	00B02	全面更新评审修改后的版本。
2014-10-30	01	新增支持Hi3796MV100芯片。
2015-05-11	02	新增6.1.3章节；修改6.2.1章节。
2016-11-04	03	新增支持Hi3798MV200芯片。
2017-04-05	04	新增支持Hi3796MV200芯片。
2018-10-10	05	新增支持Hi3798MV300、Hi3798MV300H、Hi3798MV200H、Hi3716MV430芯片。



## 目 录

前言.....	i
1 用途.....	1
2 内容.....	2
3 详细说明.....	4
3.1 表格概述.....	4
3.2 管脚复用和驱动配置说明.....	5
4 修改及使用注意事项.....	8
5 使用方法.....	10
6 修改样例.....	11
6.1 网口配置.....	11
6.1.1 Hi3798C/Hi3796C V100.....	11
6.1.2 Hi3798M V100.....	12
6.1.3 Hi3798C V200_A.....	13
6.2 DDR 容量查看.....	14
6.2.1 Hi3798C/Hi3796C V100/Hi3798CV200_A.....	14
6.2.2 Hi3798M V100.....	15



# 1 用途

## 注意

Fastboot配置表格的命名规则、适用单板和客户选择，请参考SDK发布包Fastboot配置表格文档目录中的《readme\_\*》。

Fastboot配置表格中的内容是有关寄存器的配置，在表格的“main”页有下面几个按钮，如图1-1所示。

图 1-1 表格 main 页

Import other files	Generate reg bin file(NAND)	Generate CA config file(NAND)
	Generate reg bin file(eMMC)	Generate CA config file(eMMC)

上图中，各个按钮的功能如下：

- Import other files  
用来从其它表格导入增加的内容（研发内部使用，客户不用关注）。
- Generate reg bin file(NAND)  
针对焊接NAND Flash的非高安单板，用来生成寄存器配置文件，即SDK发布包中的“\*\_nand.reg”文件，编译时会将该文件编译进boot中。
- Generate reg bin file(eMMC)  
针对焊接eMMC的非高安单板，用来生成板寄存器配置文件，即SDK发布包中的“\*\_emmc.reg”文件，编译时会将该文件编译进boot中。
- Generate CA config file(NAND)  
针对焊接NAND Flash的高安单板，用来生成CA boot使用的配置文件。
- Generate CA config file(eMMC)  
针对焊接eMMC的高安单板，用来生成CA boot使用的配置文件。



# 2 内容

表格中的内容全是寄存器的配置，主要涉及单板硬件关键参数说明、PLL、DDR、外设时钟、管脚复用、网口配置等。总体分为ITEM1、ITEM2两部分，基本参数都在ITEM1中设置，ITEM2暂时保留，但不能删除。

我们称一个工作表为一页(sheet)。一般一页配置一个模块或模式，DDR的配置会有多个页（下边会有详细说明），各模块寄存器的配置是按照系统启动工作的先后顺序来的，所以各页也是按照优先级进行划分的，顺序不能轻易改动。以下是各页的详细说明：

- main  
主界面，包含操作按钮和版本升级记录。用户可以根据自己产品的信息标注版本。
- hardware\_key\_info

硬件关键信息描述，包含boot下CPU和DDR的主频、DDR基本信息、DVFS控制通道选择、网口基本信息、小系统驱动配置信息等。仅仅是信息说明，作为信息参考，用户无需修改。

- sys\_clk  
用于设置系统的时钟比例关系、时钟配置、CPU模式配置，可以查看boot下的CPU和DDR的时钟频率。禁止用户修改。
- sysctrl\_noCA  
非高安单板系统控制设置，除了前面板管脚复用，其他用户无需修改，修改可能导致系统无法正常工作。
- sysctrl\_CA  
如果表格里面有高安这一个页面，高安单板系统控制设置，除了前面板管脚复用，其他用户无需修改，修改可能导致系统无法正常工作。  
**注意：**实际生成的镜像文件会根据noCA还是CA选择编译sysctrl\_noCA或者sysctrl\_CA，所以修改前面板管脚复用时请注意对应关系，推荐两个文件一起修改，防止修改遗漏。
- mddrc、ddrphy、ddr\_poweron、ddr\_wakeup  
ddr初始化配置，包含DDR控制器和DDR PHY的设置，用户无需修改，修改可能导致系统无法正常工作。
- qos\_ctrl  
系统模块和外设模块的优先级设置，用户无需修改，修改可能导致系统无法正常工作。



- **crg\_ctrl**  
各模块的时钟门控及复位控制，默认各模块时钟关闭，用户无需修改，修改可能导致系统无法正常工作。
- **clk\_ssmod**  
芯片各个PLL展频设置，默认所有PLL展频关闭，用户无需修改，修改可能导致系统无法正常工作。
- **peri\_cfg**  
外设模块低功耗管理，默认USB、VDAC、ADAC模块处于Power down状态，用户无需修改，修改可能导致系统无法正常工作。
- **pin\_mux\_drv\_emmc**  
用于焊接eMMC的单板，配置管脚复用和驱动能力，管脚复用和具体使用有关，驱动能力和板层有关，用户可根据《Hi379X VXXX XXX 硬件 用户指南.pdf》，针对性修改。
- **pin\_mux\_drv\_nand**  
用于焊接NAND Flash的单板，配置管脚复用和驱动能力，管脚复用和具体使用有关，驱动能力和板层有关，用户可根据《Hi379X VXXX XXX 硬件 用户指南.pdf》，针对性修改。  
**注意：**实际生成的镜像文件会根据NAND还是eMMC选择编译pin\_mux\_drv\_emmc或者pin\_mux\_drv\_nand，所以修改管脚复用时请注意对应关系，推荐两个文件一起修改，防止修改遗漏，两者仅在eMMC和NAND部分配置有差异。
- **eth\_phy\_cfg**  
以太网接口配置，包括MAC选择、MAC接口类型选择、PHY地址选择、MDC/MDIO控制选择（Hi3798M V100没有）、外置PHY复位信号选择（Hi3798M V100没有），用户根据实际产品形态修改。
- **others**  
ITEM2项，保留。





# 3 详细说明

## 3.1 表格概述

下面以表格中的一页表为例，作简单介绍。如图3-1所示。

图 3-1 配置外设控制

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	0x000000FF
DDRC_CFG_AREF	0x2c	0x0	0	write	31	0	0x000000FF
DDRC_CFG_WORKMODE	0x40	0x1002001	0	write	31	0	0x000000FF
DDRC_CFG_DDRMODE	0x50	0x26	0	write	31	0	0x000000FF
DDRC_CFG_RNKVOL	0x60	0x152	0	write	31	0	0x000000FF
DDRC_CFG_EMRS01	0x70	0x61f70	0	write	31	0	0x000000FF
DDRC_CFG_EMRS23	0x74	0x18	0	write	31	0	0x000000FF
DDRC_CFG_TIMING0	0x80	0x4639d610	0	write	31	0	0x000000FF
DDRC_CFG_TIMING1	0x84	0x3f38b080	0	write	31	0	0x000000FF
DDRC_CFG_TIMING2	0x88	0x44016000	0	write	31	0	0x000000FF
DDRC_CFG_TIMING3	0x8c	0xffd1f784	0	write	31	0	0x000000FF
DDRC_CFG_TIMING4	0x90	0x820f18	0	write	31	0	0x000000FF
DDRC_CFG_TIMING5	0x94	0x2707	0	write	31	0	0x000000FF
DDRC_CFG_ODT	0xc0	0x1	0	write	31	0	0x000000FF
DDRC_CFG_DMCLVL	0xc4	0xc	0	write	31	0	0x000000FF
DDRC_CFG_DDRPHY	0x200	0x1000	0	write	31	0	0x000000FF
DDRC_CFG_SREF	0x20	0x8101	0	write	31	0	0x000000FF

表格标题说明如下：

- 两个按钮：
  - “Add module” 用来在当前页的右边增加一个页表；
  - “Add register” 用来在当前行的下方增加一行。
- Module Name
 

表示该页的用途，是针对哪部分配置的。一般填写和当前页相同的名字。



- **Base Address**  
在表格中，每一行的寄存器都只标示了偏移地址，而寄存器的实际地址就是这个“基地址”加上“偏移地址”。
- **Priority**  
用来标明配置的顺序，每张表的优先级都不同，从0开始递增，客户无需修改。
- **Execution Required for Standby Wakeup**  
标明该页的配置在待机唤醒时是否需要执行。
  - Y: 需要;
  - N: 不需要。
- **Execution Required for Normal Boot**  
在上电起来时是否需要执行。
- **Register**  
标识该寄存器的名称，以便于维护。
- **Offset Address**  
表示寄存器相对于基地址的偏移地址。
- **Value Written to or Read from Register**  
标识该值是需要写入的值，或者是读出比较的值。
- **delay**  
表示在配置完该寄存器后需要延时的值。
- **Read or Write**  
读写控制选择。
- **Bits to Be Read or Written**  
读写的bit数，该值为实际读写的位数减1，比如32bit全写，则这一栏填31，如果写1个bit，这一栏写0。
- **Start Bit to Be Read or Written**  
如果是从0bit开始写，此处就填0，其他以此类推。
- **Register Attribute**  
这一栏由表格自动生成，不用理会。

## 3.2 管脚复用和驱动配置说明

对于Hi3798C/Hi3796C，管脚复用寄存器有效内容分为管脚复用（bit[3:0]）和驱动能力（bit[7:4]）两部分，且在不同的寄存器里面配置，表格的pin\_mux\_drv\_nand和pin\_mux\_drv\_emmc页面的上面部分为管脚复用的配置寄存器，下面明确备注有驱动能力配置的为驱动能力配置的寄存器，如图3-2所示。



图 3-2 pin\_mux\_drv 配置值

deno_gpio0	0x0000019C	0x00000002	0	write	7	0	0x0000003F	GPIO	GPIO14_0	AT41		
deno_gpio1	0x000001A0	0x00000002	0	write	7	0	0x0000003F		GPIO14_1	AR39		
deno_gpio3	0x000001A8	0x00000002	0	write	7	0	0x0000003F		GPIO14_3	AW39		
deno_gpio4	0x000001AC	0x00000002	0	write	7	0	0x0000003F		GPIO14_4	AY40		
deno_gpio5	0x000001B0	0x00000002	0	write	7	0	0x0000003F		GPIO14_5	AR40		
deno_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	0x000001C8	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		
i2s0_gpio32	0x000001E4	0x00000002	0	write	7	0	0x0000003F		GPIO13_2	D28		
i2s0_gpio33	0x000001E8	0x00000002	0	write	7	0	0x0000003F		GPIO13_3	B28		
sfc_dio_ctrl	0x00000800	0x00000040	0	write	7	0	0x0000003F	SPI drive cfg	SFC_DIO	FAST	10	
sfc_vpn_ctrl	0x00000804	0x00000040	0	write	7	0	0x0000003F		SFC_VPN_I02	FAST	10	
sfc_clk_ctrl	0x00000808	0x00000040	0	write	7	0	0x0000003F		SFC_CLK	FAST	10	
sfc_doi_ctrl	0x0000080C	0x00000040	0	write	7	0	0x0000003F		SFC_D01	FAST	10	
sfc_holdn_ctrl	0x00000810	0x00000040	0	write	7	0	0x0000003F		SFC_HOLDN_I03	FAST	10	
sfc_csln_ctrl	0x00000814	0x00000040	0	write	7	0	0x0000003F		SFC_CS1N	FAST	11	

对于Hi3798M，管脚复用寄存器有效内容分为管脚复用（bit[7:0]）和驱动能力（bit[11:8]）两部分，且在不同的寄存器里面配置，表格pin\_mux\_drv\_nand和pin\_mux\_drv\_emmc页面为管脚复用和驱动能力配置寄存器，如图3-2所示。

图 3-3 pin\_mux\_drv 配置值

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



- 对于驱动能力不可调的IO，只有管脚复用部分有效（与实际接口应用场景相关），在Fastboot配置表格中，只操作bit[7:0]，Start Bit to Be Read or Written值为0，Bits to Be Read or Written值为7（总共操作8bit）。
- 对于驱动能力可调的IO，除了管脚复用部分外，同时需要根据PCB的板层，配置驱动能力部分，因此需要操作bit[11:0]，Start Bit to Be Read or Written值为0，Bits to Be Read or Written值为11（总共操作12bit）。

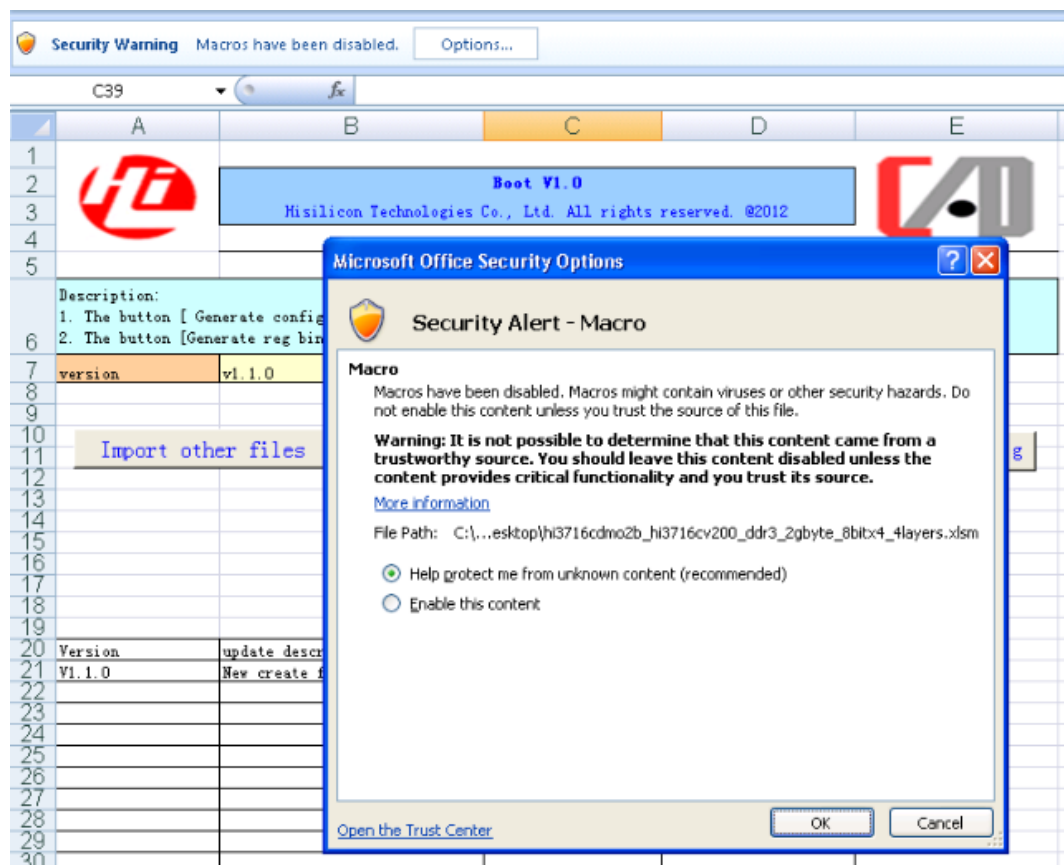


# 4 修改及使用注意事项

修改及使用注意事项如下：

- 表格的使用  
如图4-1，在点按钮之前，需要打开宏。

图 4-1 表格启动宏



- 表格的修改  
一般表格中需要修改的地方只有pin\_mux\_drv\_nand/pin\_mux\_drv\_emmc和eth\_phy\_cfg，不同的产品形态硬件连接不一样，PCB板层不一样，管脚复用、驱



动能力和网口配置就会不一样，需要根据产品形态修改。**其他部分客户无需修改，误改可能导致系统无法正常启动，客户有需求，请与海思FAE提需求。**

- 表格修改后注意各页优先级选项的连贯一致、工作表与模块名的一致性以及工作表的内容中间不能有空行。
- 表格内容的增加不能手工生成并修改，需要用excel表格中的宏。



# 5 使用方法

使用步骤如下：

## 步骤1 生成x.reg。

当需要编译某个boot时，打开单板对应的表格文件。

在main页中，单击“Generate reg bin file(NAND)”或“Generate reg bin file(eMMC)”按钮，在当前目录下生成x.reg。

## 步骤2 编译。

### ● Linux

- 将生成的x.reg文件放至\${LINUX\_SDK}/source/boot/sysreg/目录下，如果直接在该目录下生成，则略去此步；
- \${LINUX\_SDK}目录下编译：make hiboot\_clean;make hiboot
- 编译完成后在\${LINUX\_SDK}/pub/image目录下生成fastboot-burn.bin文件，使用该文件即可直接烧写。

### ● Android

- 将生成的x.reg文件放至\${ANDROID\_SDK}/device/hisilicon/bigfish/sdk/source/boot/sysreg/目录下，如果直接在该目录下生成，则略去此步；
- \${ANDROID\_SDK}目录下编译：  
source build/envsetup.sh  
lunch 选择配置  
make hiboot
- 编译完成后在\${ ANDROID\_SDK }/ out/target/product/\${配置产品}/Emmc(或Nand)/ 目录下生成fastboot.bin文件，使用该文件即可直接烧写。

----结束



# 6 修改样例

## 6.1 网口配置

### 6.1.1 Hi3798C/Hi3796C V100

以Hi3798C V100为例，Hi3798C V100内部自带两个MAC，可以同时支持两路以太网接口，包括：

- 1路外置GE PHY；
- 1路外置FE PHY+1路外置GE PHY；

在Fastboot配置表格中，以太网接口相关的配置主要有以下几点：

- MAC接口类型选择，MII0接口可以选择MII/RMII/RGMII三种类型，MII1只能选择RMII类型，Hi3798C V100默认配置MII接口模式；
- MAC对应的PHY ADDR选择，外置PHY地址必须和硬件一致；
- MAC对应PHY的MDCK/MDIO选择，MDCK0/MDIO0或者MDCK1/MDIO1，Hi3798C V100，默认配置MDCK0/MDIO0；
- 外置PHY复位管脚GPIO选择，使用GPIO作为复位信号，需要选择GPIO的组号和组内位号；
- 接口管脚复用和驱动能力配置，和具体产品的接口选择和板层有关。

例如：

- MAC0配置选择内置GE PHY；
- 选择MDCK0/MDIO0控制；
- 外置GE PHY复位管脚选择GPIO13\_0；
- 板层选择四层板。
- 接口管脚复用在表格里面没有配置，也就是默认为RGMII功能。

则网口配置如[图6-1](#)所示。





图 6-1 eth\_phy\_cfg 配置值

Module Name	eth_phy_cfg	eth_phy_cfg						
Base Address	0xf8000000			Add module			Add register	
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/RMII 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	0x0000003F	MAC1_PHY_Select MDIO1/MDCK1; 0 or 1

eth\_phy\_cfg配置值说明：

- 第8行  
MAC0接口类型选择RGMII接口。
- 第9行  
内置GE PHY的PHY地址配置为1。
- 第10行  
GE PHY选择MDCK0/MDIO0为控制信号。
- 第11-12行  
MAC0对应的PHY的复位GPIO选择不用就要求配置为保留值0xFF。
- 第13行  
MAC1选择MDCK1/MDIO1为控制信号。

## 6.1.2 Hi3798M V100

Hi3798M V100内部自带一个内置FE PHY，支持一路以太网接口。

在Fastboot配置表格中，以太网接口相关的配置主要有以下几点：

- MAC对应的PHY ADDR选择，外置PHY地址必须和硬件一致，对于Hi3798M设置0x1；
- MAC对应PHY的MDCK/MDIO选择，对于Hi3798M V100设置0x0；
- 外置PHY复位管脚GPIO选择，使用GPIO作为复位信号，需要选择GPIO的组号和组内位号，对于Hi3798M V100设置0xFF；
- 接口管脚复用和驱动能力配置，和具体产品的接口选择和板层有关，对于Hi3798M V100不需要配置。

例如：

- MAC0配置选择内置GE PHY；
- 选择MDCK0/MDIO0控制；
- 外置GE PHY复位管脚选择GPIO13\_0；
- 板层选择四层板。
- 接口管脚复用在表格里面没有配置，也就是默认为RGMII功能。



则网口配置如图6-1所示。

图 6-2 eth\_phy\_cfg 配置值

Module Name	eth_phy_cfg	eth_phy_cfg					
Base Address	0xf8000000		Add module			Add register	
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
MAC_PHY_ADDR	0xA8	0x1	0	write	7	0	0x0000003F
MAC_PHY_CTRL	0xA8	0x0	0	write	7	8	0x0000403F
MAC_PHY_RST_GPIO	0xAC	0xff	0	write	7	0	0x0000003F
MAC_PHY_RST_GPIO_BIT	0xAC	0xff	0	write	7	8	0x0000403F

eth\_phy\_cfg配置值说明：

- 第8行  
内置FE PHY的PHY地址配置为1。
- 第9行  
MDCK0/MDIO0控制信号，内置FE PHY设置为0。
- 第10-11行  
MAC对应的PHY的复位GPIO选择不用就要求配置为保留值0xFF。

6.1.3 Hi3798C V200\_A

Hi3798C V200\_A内部自带三个MAC，可以同时支持三路以太网接口，包括：

- 3路外置GE PHY/FE PHY；
- 1路外置FE PHY+2路外置GE PHY/FE PHY；

在Fastboot配置表格中，以太网接口相关的配置主要有以下几点：

- MAC接口类型选择，MAC0接口可以选择MII/RMII/RGMII三种类型，MAC1/MAC2只能选择RGMII/RMII类型，Hi3798C V200\_A默认配置RGMII接口模式；
- MAC对应的PHY ADDR选择，外置PHY地址必须和硬件一致；
- MAC对应PHY的MDCK/MDIO选择，MDCK0/MDIO0或者MDCK1/MDIO1；
- 外置PHY复位管脚GPIO选择，使用GPIO作为复位信号，需要选择GPIO的组号和组内位号，使用默认RESET信号，或者不使用时，配置为0xFF；
- 接口管脚复用和驱动能力配置，和具体产品的接口选择和板层有关。

例如：

- MAC0配置选择内置GE PHY；
- 选择MDCK1/MDIO1控制；
- 外置GE PHY复位管脚选择专用RESET管脚；
- 板层选择四层板；



- 接口管脚复用在表格里没有配置，也就是默认为RGMII功能。

则网口配置如图6-3所示。

图 6-3 eth\_phy\_cfg 配置值

	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	0x000000FF	PERI_FEPHY_LDO_CTRL, enable & wset = 4'h8
9	PERI_CTRL	0xa20006	0x1	0	write	0	8	0x00004007	MAC0 Select GRMII (0xFE PHY/1:RGMII 10)
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

eth\_phy\_cfg配置值说明：

- 第9行  
MAC0接口类型选择对接外置PHY。
- 第10行  
MAC0接口类型选择RGMII接口。
- 第11行  
内置GE PHY的PHY地址配置为1。
- 第12行  
GE PHY选择MDCK1/MDIO1为控制信号。
- 第13-14行  
MAC0对应的PHY的复位GPIO选择不用就要求配置为保留值0xFF，默认选择专用外置PHY复位管脚。

## 6.2 DDR 容量查看

### 6.2.1 Hi3798C/Hi3796C V100/Hi3798CV200\_A

以Hi3798C V100为例，hi3798cdmola\_hi3798cv100\_ddr3\_16bit x4\_4layers.xlsm配置的DDR为4颗256Mb x16，总容量为2GB。则DDRC配置0x8060、0x9060寄存器为0x142。如图6-4所示。

图 6-4 DDR 配置 0x8060、0x9060 寄存器修改

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



不同容量对应的值如下，配置值只与单片DDR颗粒容量和位宽大小有关，如表6-1所示。

表 6-1 DDR 不同容量对应的配置值

单颗DDR颗粒容量	128Mb x 16	256Mb x 16	512Mb x 16
0x8060、0x9060配置值	0x132	0x142	0x152

海思参考设计中，表格的DDR容量配置默认是大容量的配置兼容小容量的配置。

6.2.2 Hi3798M V100

hi3798mdm01b\_hi3798mv100\_ddr3\_16bit x 2\_2layers.xlsm配置的DDR为2颗256Mb x 16，总容量为1GB。则DDRC配置0x60寄存器为0x152。如图6-4所示。

图 6-5 DDR 配置 0x60 寄存器修改

DDRC_CFG_RNKVOL	0x60	0x152	0	write	31	0	0x0000000FF
-----------------	------	-------	---	-------	----	---	-------------

不同容量对应的值如下，配置值只与单片DDR颗粒容量大小有关，如表6-1所示。

表 6-2 DDR 不同容量对应的配置值

单颗DDR颗粒容量	128Mb x 16	256Mb x 16
0x60配置值	0x142	0x152

海思参考设计中，表格的DDR容量配置默认是大容量的配置兼容小容量的配置。