

PHYTIUM 飞腾

FT-1500A/16 高性能通用微处理器
数据手册
(V2.8)

天津飞腾信息技术有限公司

二零一九年五月

版本历史

以下为此文档释放过的更新版本

日期	版本号	作者	更新说明
2014.10.20	1.0	技术支持部	初版
2014.12.16	2.0	技术支持部	调整文档结构
2015.04.16	2.1	技术支持部	更新封装、上电时序
2015.07.07	2.2	技术支持部	调整 LPC
2015.09.02	2.3	技术支持部	调整上电时序
2016.03.21	2.4	技术支持部	增加 GPIO 复用列表, 装焊温度曲线等内容; 调整了文档结构及部分内容
2017.04.05	2.5	技术支持部	调整文档结构
2018.2.07	2.6	技术支持部	调整文档结构; 增加 FCR 寄存器说明
2018.06.05	2.7	技术支持部	完善 spi 寄存器配置寄存器列表; 完善电气特性
2019.05.09	2.8	技术支持部	增加地址空间描述, 调整文档结构, 完善 LPC 说明

技术支持邮箱: support@phytium.com.cn

版权所有© 天津飞腾信息技术有限公司 2019。

此文档用于指导用户的相关应用和开发工作。天津飞腾信息技术有限公司对此文档内容拥有版权, 并受法律保护。

目 录

1	芯片介绍	1
2	技术指标	1
3	功能描述	1
4	CPU 接口	2
4.1	地址空间分配	2
4.1.1	PCIE 配置、IO 和 MEM32 地址空间划分	3
4.1.2	MIO 地址空间	3
4.1.3	GMAC 控制器地址空间	3
4.2	DDR 接口	3
4.3	PCIE 接口	4
4.3.1	寄存器说明	4
4.4	GMAC 接口	5
4.5	LPC 接口	7
4.5.1	寄存器说明	7
4.6	SPI 接口	8
4.6.1	寄存器说明	8
4.7	UART 接口	11
4.7.1	寄存器说明	12
4.8	I2C 接口	24
4.8.1	寄存器说明	24
4.9	GPIO 接口	37
4.9.1	GPIO 复用说明	38
4.9.2	GPIO 寄存器说明	39
4.10	上电时序	42
5	电气特性	44
5.1	极限工作条件	44
5.2	典型工作参数	44
6	封装数据	45
6.1	封装尺寸	45
7	装焊温度曲线	46
7.1	无铅焊接温度曲线中各温区的作用	46
7.2	有铅焊接温度曲线中各温区的作用	47
8	引脚描述	48
8.1	通用 IO 类引脚 (118 PIN)	48
8.2	GMAC 引脚 (55 PIN)	53

8.3 PCIE 引脚（164 PIN）	55
8.4 DDR3 引脚（624 PIN）	60
8.5 电源引脚（983 PIN）	78

PHYTILUM

1 芯片介绍

FT-1500A 系列 16 核处理器芯片(FT-1500A/16)集成 16 个自主开发的 ARMv8 指令集兼容微处理器内核 FTC660, 采用片上并行系统(PSoC)体系结构。主要应用于较高计算能力和较高吞吐率的服务器(如各种办公业务系统应用/事务处理器、数据库服务器、存储服务器、物联网/云计算服务器)等领域, 可满足网络服务、事务处理、数据中心存储、数据库、网络交换等业务。

2 技术指标

主要技术指标如下:

- 兼容 ARM V8 架构, 支持 ARM64 指令集
- 支持 32 位指令模式
- 集成 16 个 FTC660 处理器核
- 核心时钟频率 1.5GHz (标配)
- L1 数据 Cache 32KB
- L2 Cache 8MB
- L3 Cache 8MB
- 峰值性能 96GFlops@1.5GHz
- 典型功耗 35W
- 核电压 0.9V
- 集成 4 个 64 位总线 DDR3 存储控制器, 速率可达 1600Mbps, 访存带宽可达 51.2GB/s
- 集成 32 Lane PCI Express v3.0 接口, 最多支持 4 个 PCIE root complex, 不能作为 endpoint 使用
- BGA1944 封装, 1.0mm 球间距, 封装尺寸 42.5mmX60mm
- IO 电压 1.8V, 包括 UART、I2C、GMII、GPIOA、GPIOB、GPIOC、GPIOD、LPC
- 支持电源关断、时钟关断、DVFS
- 支持商业、工业等分级, 温度范围为标准的商业温度范围(0°C~70°C)和工业级温度范围(-40°C~+85°C)

3 功能描述

表 3-1 功能描述

硬件特性		说明
Core	兼容 ARM V8 指令集的 FTC660 核 (FTC660)	16 个, 典型工作频率 1.5GHz, 支持电源关断

存储控制器	DDR3 SDRAM 控制器	4 个控制器，支持带 ECC 的 DDR3 DIMM，支持 RDIMM、UDIMM、SODIMM、表贴颗粒
外设	PCIE3.0 Root Complex	2 个 x16，每个 x16 可分拆为 2 个 x8，独立控制
	千兆以太网控制器	2 个控制器，均支持 1000M/100M/10M 自适应，物理接口支持 MII、GMII 规范，电压 1.8V
	Flash 控制器	1 个 SPI 接口的 Flash 控制器，支持 4 个片选，单片最大支持容量为 512MB，电压 1.8V
	UART	2 个 UART，其中 UART0 为 9 线全功能串口，UART1 为 3 线调试串口，电压 1.8V，调试串口默认使用 UART1
	I2C	2 个 I2C master 控制器，电压 1.8V
	GPIO	4 个 8 位 GPIO 接口，GPIOA[0:7]，GPIOB[0:7]，GPIOC[0:7]，GPIOD[0:7]，电压 1.8V
	LPC	1 个 LPC 接口，兼容 Intel Low Pin Count 协议，可接 SuperIO 等外围芯片，电压 1.8V
电源	内核电源	0.9V
	存控电源	1.5V
	IO 电源	1.8V

4 CPU 接口

4.1 地址空间分配

全局寄存器地址空间分配如下：

表 4-1 系统地址空间

地址空间	设备
0x000_00000000~0x000_1FFFFFFF	SPI，预留 512MB
0x000_20000000~0x000_27FFFFFFF	LPC，预留 128MB
0x000_28000000~0x000_2FFFFFFF	预留 128MB，包括 MIO(uart/i2c/wdt/gpio)，cru，Error，icu，MCU0、MCU1、pcie 控制寄存器，PMA 配置寄存器，gmac，pmc，mmu，gpv 等
0x000_30000000~0x000_39FFFFFFF	DTI，预留 160MB(现在 32MB+4MB)包括 generic Timer、0x000_30000000 是 ROMTable 基址
0x000_3A000000~0x000_3AFFFFFFF	MN,16MB
0x000_40000000~0x000_7FFFFFFF	PCIE 的配置、IO 和 MEM32 空间，1GB

4.1.1 PCIE 配置、IO 和 MEM32 地址空间划分

表 4-2 PCIE 配置、IO、MEM32 地址空间划分

地址范围	大小	用途
0x000_40000000~0x000_4FFFFFFF	256MB	配置空间
0x000_50000000~0x000_5FFFFFFF	256MB	IO 空间
0x000_60000000~0x000_7FFFFFFF	512MB	MEM32 空间

4.1.2 MIO 地址空间

表 4-3 MIO 内部空间划分

起始地址空间	设备名	说明
0x28000000	UART0	9 针全线制串口
0x28001000	UART1	3 线制串口
0x28002000	I2C0	既可做 Master 也可做 Slave
0x28003000	I2C1	既可做 Master 也可做 Slave
0x28004000	WDT0	看门狗
0x28005000	WDT1	看门狗
0x28006000	GPIO	4 个 8 位接口, 最多 32 位
其它	保留	

4.1.3 GMAC 控制器地址空间

表 4-4 GMAC 控制器地址空间划分

GMAC 控制器 0 基地址: 0x28c00000

GMAC 控制器 1 基地址: 0x28c02000

地址偏移	大小	描述
0x0000~0x0FFF	4KB	GMAC 寄存器空间
0x1000~0x1FFF	4KB	GMAC DMA 寄存器空间

4.2 DDR 接口

DDR 接口支持 X4、X8 和 X16 一共三种 DDR 颗粒类型, 如表 4-5 所示。DDR 接口支持 1.5V DDR3 的 RDIMM、UDIMM、SODIMM 以及表贴颗粒等内存条类型。

表 4-5 DDR 接口支持的内存形式

颗粒芯片位宽	颗粒芯片数目
X16	4
X8	8
X4	16

注：该表芯片数目中，不包含 ECC 内存颗粒

4.3 PCIE 接口

PCIE 接口支持 PCIE3.0 规范，兼容 PCIE2.0、PCIE1.0 规范，其特点如下：

- 包含 2 个 x16 root complex 控制器，分为 PCIE 0 和 PCIE 1；
- 每个 X16 最多拆分为 2 个 root，也就是说每个 X16 的拆分模式可以为 1 个 X16 或者 2 个 X8；
- 可支持 X1，X2，X4，X8 支持翻转（逆序）。
- PCIE 翻转兼容说明：

若是 PCIE0[0:7]、PCIE1[0:7]，翻转（逆序）后连接 PCIE 插槽，则只支持 X8 外设。若是 PCIE0[8:15]、PCIE1[8:15] 则不受此约束限制，用户可根据需要进行配置。

4.3.1 寄存器说明

4.3.1.1 基地址

表 4-6 PCIE 基地址

名称	基地址
PCIE	0x2810_0000

4.3.1.2 寄存器列表

表 4-7 寄存器描述

偏移地址	名称	读写方式	说明
0x0F00	REG_PEU_BIF_MODE	R/W	拆分模式寄存器
0x0F08	REG_PEU_TX_LANE_FLIP_EN	R/W	链路反转设置寄存器

4.3.1.2.1 拆分模式寄存器(0x0E00)

表 4-8 拆分模式寄存器

位	名称	读写方式	默认值	说明
[3:2]	PCIE1_bif_mode	R/W	0x1	PCIE 1 的分拆模式设置：

				<ul style="list-style-type: none"> •00–全部关闭 •01–一个 16x •10–两个 8x
[1:0]	PCIE0_bif_mode	R/W	0x1	PCIE 0 的分拆模式设置： <ul style="list-style-type: none"> •00–全部关闭 •01–一个 16x •10–两个 8x

4.3.1.2.2 链路反转设置寄存器(0x0E08)

表 4-9 链路反转设置寄存器

位	名称	读写方式	默认值	说明
[15]	pcie1_c3_tx_lane_flip_en	R/W	0x0	为 1 表示手动设置 lane 翻转
[14]	pcie1_c2_tx_lane_flip_en	R/W	0x0	
[13]	pcie1_c1_tx_lane_flip_en	R/W	0x0	
[12]	pcie1_c0_tx_lane_flip_en	R/W	0x0	
[11]	pcie1_c3_rx_lane_flip_en	R/W	0x0	
[10]	pcie1_c2_rx_lane_flip_en	R/W	0x0	
[9]	pcie1_c1_rx_lane_flip_en	R/W	0x0	
[8]	pcie1_c0_rx_lane_flip_en	R/W	0x0	
[7]	pcie0_c3_tx_lane_flip_en	R/W	0x0	
[6]	pcie0_c2_tx_lane_flip_en	R/W	0x0	
[5]	pcie0_c1_tx_lane_flip_en	R/W	0x0	
[4]	pcie0_c0_tx_lane_flip_en	R/W	0x0	
[3]	pcie0_c3_rx_lane_flip_en	R/W	0x0	
[2]	pcie0_c2_rx_lane_flip_en	R/W	0x0	
[1]	pcie0_c1_rx_lane_flip_en	R/W	0x0	
[0]	pcie0_c0_rx_lane_flip_en	R/W	0x0	

4.4 GMAC 接口

以太网控制器（GMAC）的主要功能是在兼容 IEEE802.3-2005 标准的以太网中发送和接收数据，支持多种工业标准 PHY 接口。

GMAC 接口特点：

- 支持 1000Mbps/100Mbps/10Mbps
- 支持 IEEE 802.3-2005 Ethernet MAC, Gigabit Media Independent Interface (GMII) / Media Independent Interface (MII)

GMAC 包括 GMAC 0 和 GMAC 1 两个相同的以太网控制器。接口信号主要分三类：时钟接口、数据接口和管理接口。

外围接口的需求：

- CPU 的 OSC_CLK 125MHz 时钟，需外部提供。
- GMU_CLK_OSC、G1_CLK_TX 和 G1_CLK_RX 在不使用情况下需要接地。
- 如果没有使用 GMAC，125Mhz 外部时钟输入引脚 GMAC_OSC 建议接地。
- 外围接口电路与下列引脚连接的信号线，有等长需求，如表 4-10 所示，每个分组内各信号线要求等长。数据信号比时钟信号长 0~0.2ns，下表中每组的首个信号为该组的时钟信号，其它为数据信号。

表 4-10 GMAC

组	I/O PAD
1	G0_CLK_GTX G0_TX[0:7] G0_TXEN G0_TXER
2	G1_CLK_GTX G1_TX[0:7] G1_TXEN G1_TXER
3	G0_CLK_RX G0_COL G0_CRS G0_RX[0:7] G0_RXDV G0_RXER
4	G1_CLK_RX G1_COL G1_CRS G1_RX[0:7] G1_RXDV G1_RXER
5	G0_MDC G0_MDIO
6	G1_MDC G1_MDIO

4.5 LPC 接口

LPC (Low Pin Count) 主要用来连接鼠标、键盘、串口、低速 Flash 等设备。LPC 接口采用 1.8V 的 LVCMOS, 因而在连接 LPC 设备时需要进行电平转换。芯片集成了一个 LPC 控制器, LPC 接口建议通过 CPLD/FPGA, 进行电平的转换。

LPC 的时钟信号为 GPIOD2_EXT_LPC_LCK, 时钟频率为 33MHz, 必须由外部提供。

CPU 中的 LPC 控制器只支持 4 字节对齐地址的访问, 如果使用单字节访问, 就会造成部分地址无法使用, 因此建议使用 4 字节访问, 详情请参考《FT-1500A 平台 LPC 兼容规范》。

4.5.1 寄存器说明

4.5.1.1 基地址

表 4-11 LPC 基地址

名称	基地址
LPC	0x27FF_0000

4.5.1.2 寄存器列表

表 4-12 LPC 寄存器说明

偏移地址	名称	读写方式	默认值	含义
0xFFE8	串行中断配置寄存器[31:0]	R/W	32'h8000_0000	设置 LPC 接口数据读取使能、起始周期、串行中断模式和支持设备数量。

注: 操作 LPC 时, 请确保 LPC 的外部电路连接正常, 否则读写 LPC 寄存器会失败。

4.5.1.2.1 串行中断配置寄存器(0xFFE8)

表 4-13 串行中断配置寄存器

地址	名称	读写方式	含义	默认值
0x27FF_FFFC	lpc addr cfg reg[7:0]	R/W	配置 APB 接口地址的设备类型	8'b11100100
0x27FF_FFF8	lpc long wait ctrl req[31:0]	R/W	长等待超时控制寄存器	32'h0
0x27FF_FFF4	lpc int status reg[31:0]	RO	中断状态 (串行中断) bit29~0: 串行中断, bit30 DMA 请求中断	32'h0
0x27FF_FFF0	lpc int clear reg[31:0]	R/W	清除中断寄存器	32'h0
0x27FF_FFEC	lpc msg length cfg reg[3:0]	R/W	firmware memory 类型的报文长度	

			配置寄存器（暂未使用）	
0x27FF_FFE8	nu_serirq_config[31:0]	R/W	配置寄存器（bit31: 针对读数据每次读4 bytes数据使能标志（1'b1: 读1byte）； bit1~0:起始周期配置（2'b11: 8； 2'b10:6； 否则4，默认4 clk）， bit2: 串行中断模式配置默认连续模式 默认为连续模式）， bit3~4: 支持的串行中断设备数量（2'b01 代表32 否则16默认16）	32'h8000_0000
0x27FF_FFE4	lpc err status reg[2:0]	RO	错误状态寄存器，记录最近一次发生错误的状态。 3'b000: 无效 3'b010: 长等待超时 3'b001: 短等待超时 3'b100: SYNC error 3'b110: 非法的读写操作	3'h0
0x27FF_FFE0	lpc firmware id sel reg[2:0]	R/W	firmware memory设备ID选择配置寄存器 3'b001 22: 19作为ID 3'b010 23: 20作为ID 3'b011 24: 21作为ID 默认30: 27作为ID	4'h0
0x27FF_FFDC	lpc DMA ID cfg reg[2:0]	R/W	DMA 设备ID配置寄存器	3'h6

4.6 SPI 接口

SPI 接口仅支持连接用于存储 BIOS 用的 FLASH 芯片。

4.6.1 寄存器说明

4.6.1.1 基地址

表 4-14 SPI 基地址

名称	基地址
SPI	0x1FFF_FF00

4.6.1.2 寄存器列表

表 4-15 寄存器列表

地址	寄存器名	说明
0x00	配置寄存器	配置访问数据空间时产生的读写命令、SCK 分频等信息
0x14	Flash 容量设置寄存器	设置所连接的 FLASH 容量，每个片选所接容量必须相同
0x18	写缓冲 flush 寄存器	写 1 将把写缓冲中的数据 flush 到 FLASH
0x20	命令端口寄存器	通过端口方式访问 SPI 时设置的命令
0x24	地址端口寄存器	通过端口方式访问 SPI 时设置的地址
0x28	高位数据寄存器	通过端口方式访问 SPI 时的高 4 字节数据
0x2C	低位数据寄存器	通过端口方式访问 SPI 时的低 4 字节数据，读写将触发 SPI 总线操作

4.6.1.2.1 配置寄存器 (0x00)

表 4-16 配置寄存器

位	名称	读写方式	默认值	说明
[7:6]	cmd_mode	R/W	0x0	设置读写数据空间时所发出的 SPI 命令 <ul style="list-style-type: none"> •00-发 read(03)和 pp(02); •01-发 4read(13)和 4pp(12); •10-发 fast_read(0b)和 pp(02); •11-发 4fast_read(0c)和 4pp(12);
[5]	addr_mode	R/W	0x0	设置发出 read、fast_read 和 pp 命令时的地址模式 <ul style="list-style-type: none"> •0-3 字节地址 •1-4 字节地址
[4]	dummy	R/W	0x0	设置发出 fast_read 和 4fast_read 时是否发 dummy 字节 <ul style="list-style-type: none"> •0-不发 •1-发
[3]	wr_mode	R/W	0x0	<ul style="list-style-type: none"> •0-每次写请求直接发编程命令; •1-写数据先放入缓冲，多次写合并编程;
[2:0]	sck_sel	R/W	0	<ul style="list-style-type: none"> •000: 如果定义了 SPEEDSIM_FLASH, sck 为输入时钟 clk 的 2 分频, 否则为 16 分频; •001: sck 为 pclk 的 2 分频; •010: sck 为 pclk 的 4 分频;

				<ul style="list-style-type: none"> •011: sck 为 pclk 的 8 分频; •100: sck 为 pclk 的 16 分频; •101: sck 为 pclk 的 32 分频; •110: sck 为 pclk 的 64 分频; •111: sck 为 pclk 的 128 分频。
--	--	--	--	---

4.6.1.2.2 容量寄存器 (0x14)

表 4-17 容量寄存器

位	名称	读写方式	默认值	说明
[2:0]	flash_capacity	R/W	0x0	<ul style="list-style-type: none"> •0-16MB •1-32MB •2-64MB •3-128MB •4-256MB •5-512MB •6-1GB •7-非法值

4.6.1.2.3 写缓冲寄存器 (0x18)

表 4-18 写缓冲寄存器

位	名称	读写方式	默认值	说明
[0]	flush	R/W	0x0	写 1 将产生 flush 操作

4.6.1.2.4 命令端口寄存器 (0x20)

表 4-19 命令端口寄存器

位	名称	读写方式	默认值	说明
[17:16]	flash_sel	R/W	0x0	选择要操作的 FLASH
[15:8]	command	R/W	0x0	要发出的 SPI 命令
[7]	reserved	R/W	0x0	保留
[6]	addr_mode0	R/W	0x0	是否带地址 •0-不带地址

				•1-带地址
[5]	addr_mode1	R/W	0x0	地址模式 •0-3 字节地址 •1-4 字节地址
[4]	dummy	R/W	0x0	是否发 dummy 字节 •0-不发 •1-发
[3:0]	R/W_num	R/W	0x0	读写字节数目, 有效值为 0, 1, 2, 4, 8

4.6.1.2.5 地址端口寄存器 (0x24)

表 4-20 地址端口寄存器

位	名称	读写方式	默认值	说明
[31:0]	addr	R/W	0x0	地址

4.6.1.2.6 高位寄存器 (0x28)

表 4-21 高位寄存器

位	名称	读写方式	默认值	说明
[31:0]	data	R/W	0x0	高 4 字节数据

4.6.1.2.7 低位寄存器 (0x2C)

表 4-22 低位寄存器

位	名称	读写方式	默认值	说明
[31:0]	data	R/W	0x0	低 4 字节数据

4.7 UART 接口

UART (Universal Asynchronous Receiver/Transmitter), 即通用异步接收/发送装置, 定义了一种并行数据与串行数据进行转换的协议。芯片中包含 1 个 9 线制接口 UART0 和 1 个 3 线制接口 UART1, 兼容 16550 标准。

4.7.1 寄存器说明

4.7.1.1 基地址

表 4-23 UART 基地址

名称	基地址
UART0	0x2800_0000
UART1	0x2800_1000

4.7.1.2 寄存器列表

表 4-24 UART 寄存器说明

偏移地址	名字	读写方式	默认值	说明
0x00	RBR	RO	0x0	接收缓存寄存器 LCR[7] bit = 0
	THR	WO	0x0	发送保持寄存器 LCR[7] bit = 0
	DLL	R/W	0x0	波特率除数低位字节 LCR[7] bit = 1
0x04	DLH	R/W	0x0	波特率除数高位字节 LCR[7] bit = 1
	IER	R/W	0x0	中断使能寄存器 LCR[7] bit = 0
0x08	IIR	RO	0x01	中断识别寄存器
	FCR	WO	0x0	先进先出控制寄存器
0x0C	LCR	R/W	0x0	行控制寄存器
0x14	LSR	RO	0x60	行状态寄存器
0x7C	USR	RO	0x6	串口状态寄存器

4.7.1.2.1 RBR(0x00)

表 4-25 RBR

位	名称	读写方式	默认值	说明
[31:8]	保留	RO	0x0	保留

[7]	Receive Buffer Register	RO	0x0	<p>UART 模式下串口接收端的数据 buffer。当 Line Status Register (LSR) 数据就绪位 (DR) 被设置之后该寄存器上的数据才有效。</p> <p>在 non-FIFO (FIFO_MODE=NONE) 模式或者 FIFOs 被禁用 (FCR[0]设置为 0) 的情况下, RBR 中的数据必须在下一个数据到来之前被读取, 否则将被覆盖, 从而导致 over-run 错误。</p> <p>在 FIFO (FIFO_MODE!=NONE) 模式或者 FIFOs 可用 (FCR[0]设置为 1) 的情况下, 该寄存器会从头开始访问接收 FIFO, 如果接收 FIFO 满并且下一个数据到达之前该寄存器没有被读取, FIFO 里的数据会被保留, 但是任何准备进来的数据将丢失, 并导致 over-run 错误。</p>
-----	-------------------------	----	-----	--

4.7.1.2.2 THR(0x00)

表 4-26 THR

位	名称	读写方式	默认值	说明
[31:8]	保留	R/W	0x0	保留
[7]	Transmit Holding Register	WO	0x0	<p>UART 模式串口输出端的数据发送 (sir_out_n)。只有当 THR Empty (THRE) 位 (LSR[5]) 被设置之后数据才可以被写入。</p> <p>在 non-FIFO 模式或者 FIFOs 禁用 (FCR[0]=0) 并且 THRE 已被设置的情况下, 向 THR 写入一个字符会清除 THRE, 在 THRE 被重新设置之前任何向 THR 的写入都会导致 THR 数据被覆盖。</p> <p>在 FIFO 模式或者 FIFOs 可用 (FCR[0]=1) 并且 THRE 已被设置的情况下, 在 FIFO 满之前可以向 THR 写入 X 个字符。X (default=16) 取决于所配置的 FIFO Depth 的值。当 FIFO 满之后, 任何写入都无效。</p>

4.7.1.2.3 DLH(0x04)

表 4-27 DLH

位	名称	读写方式	默认值	说明
[31:8]	保留	R/W	0x0	保留
[7]	Divisor Latch (High)	R/W	0x0	<p>16 位数据的高 8 位，用于存储 UART 波特率除数。如果 UART_16550_COMPATIBLE == No，只有在 DLAB(LCR[7])位设置和 UART 没有占用(USR[0]为 0)的时候该寄存器才能被访问；其他情况，只有在 DLAB (LCR[7])被设置，才能访问。输出波特率等于串口时钟 (pclk 为一个时钟，sclk 为两个时钟 (CLOCK_MODE=ENABLE)) 频率除以 16 倍的波特率除数，如下：波特率=系统时钟/16*波特率除数。</p> <p>注意：除数寄存器 (DLL 和 DHL) 设置为 0，会禁用波特率时钟，没有串行通信。此外，一旦设置了 DLH，在传输和接收数据之前应该至少等待 8 个最慢的 DW_apb_uart 时钟周期。</p>

4.7.1.2.4 DLL(0x00)

表 4-28 DLL

位	名称	读写方式	默认值	说明
[31:8]	保留	R/W	0x0	保留
[7]	Divisor Latch (Low)	R/W	0x0	<p>16 位数据的低 8 位，用于 UART 波特率除数。</p> <p>如果 UART_16550_COMPATIBLE == No，只有在 DLAB(LCR[7])位设置和 UART 没有占用 (USR[0]为 0)的时候该寄存器才能被访问；其他情况，只有在 DLAB (LCR[7])被设置，才能访问。输出波特率等于串口时钟 (pclk 为一个时钟，sclk 为两个时钟 (CLOCK_MODE=ENABLE)) 频率除以 16</p>

				<p>倍的波特率除数, 如下: 波特率=系统时钟/16*波特率除数。</p> <p>注意: 除数暂存器 (DLL 和 DHL) 设置为 0, 会禁用波特率时钟没有串行通信。此外, 一旦设置了 DLL, 在传输和接收数据之前应该至少等待 8 个最慢的 DW_apb_uart 时钟周期</p>
--	--	--	--	---

4.7.1.2.5 IER(0x04)

表 4-29 IER

位	名称	读写方式	默认值	说明
[31:8]	保留	RO	0x0	保留
[7]	PTIME	R/W	0x0	<p>只有当 THRE_MODE_USER == Enabled 时, 可编程 THRE 中断模式才能开启并被写入。该位总是可读。用来启用/禁用中断的产生。</p> <ul style="list-style-type: none"> •0-禁用 •1-启用
[6:4]	Reserved	RO	0x0	
[3]	EDSSI	R/W	0x0	<p>Modem 状态中断标识位。该位用来启用/禁用 Modem 状态中断的产生。这个中断优先级为 4。</p> <ul style="list-style-type: none"> •0-禁用 •1-启用
[2]	ELSI	R/W	0x0	<p>Receiver Line 状态中断标识位。该位用来启用/禁用 Receiver Line 状态中断的产生。该中断优先级最高。</p> <ul style="list-style-type: none"> •0-禁用 •1-启用
[1]	ETBEI	R/W	0x0	<p>Transmit Holding Register Empty 中断标识位。该位用来启用/禁用 Transmitter Holding Register Empty Interrupt 的产生。该中断优先级为 3。</p> <ul style="list-style-type: none"> •0-禁用 •1-启用
[0]	ERBFI	R/W	0x0	<p>接收数据可用中断标识位。该位用来启用/禁用接受数据可用中断和数据超时中断 (在 FIFO 模式和 FIFO 打开) 的产生。该中断优先级为 2。</p> <ul style="list-style-type: none"> •0-禁用

				•1-启用
--	--	--	--	-------

4.7.1.2.6 IIR(0x08)

表 4-30 IIR

位	名称	读写方式	默认值	说明
[31:8]	保留	RO	0x0	保留
[7:6]	FIFOs Enabled (or FIFOSE)	RO	0x0	FIFOs 的启用。这个用来显示 FIFOs 为启用还是禁用。 •0-禁用 •1-启用
[5:4]	保留	N/A	0x0	保留
[3:0]	Interrupt ID (or IID)	RO	0x1	中断 ID。该位用来显示等待执行的最高优先级中断，可以是以下类型中的一个： •0000-modem status •0001-no interrupt pending •0010-THR empty •0100-received data available •0110-receiver line status •0111-busy detect •1100-character timeout bit[3]表示只有 FIFOs 可用并且用于辨别一个 Character Timeout condition interrupt 时该中断才起作用。

4.7.1.2.7 LCR(0x0C)

表 4-31 LCR

位	名称	读写方式	默认值	说明
[31:8]	保留	RO	0x0	保留
[7]	DLAB	R/W	0x0	除数锁存访问位。如果 UART_16550_COMPATIBLE==NO，则只有当 UART 空闲（USR[0]等于 0）的时候可写；否则总是可写、可读。

				该位被用来使能除数锁存器（DLL 和 DLH）的读和写，从而设置 UART 的波特率。该位在初始化波特率之后必须被清除以用来访问其他寄存器。
[6]	Break (or BC)	R/W	0x0	Break 控制位。该位用来产生一个发送到接收设备的 break 状态。如果设置成 1，串行输出被强制成 spacing (logic 0) 状态。如果不在 Loopback 模式（取决于 MCR[4]），sout 信号将被强制成低电平直到 Break 位被清除。如果 SIR_MODE==Enable 并且为活动状态（MCR[6]设置成 1），sir_out_n 信号持续发送脉冲。如果在 Loopback 模式下，break 状态对于接收端来说是内部 loop back 并且 sir_out_n 线强制成低。
[5]	Stick Parity (reserved)	RO	0x0	保留
[4]	EPS	R/W	0x0	偶校验选择位。如果 UART_16550_COMPATIBLE==NO，则只有当 UART 空闲的时候可写，否则一直可写，可读。当校验使能的时候（PEN 设置成 1），此位被用来选择奇还是偶校验。如果该位被设置成 1，一个 logic 1s 偶数被发送并检查，如果设置成 0，一个 logic 1s 奇数被发送并检查
[3]	PEN	R/W	0x0	校验使能位。如果 UART_16550_COMPATIBLE==NO，则只有当 UART 空闲的时候可写，否则一直可写、可读。 在发送串行字符时该位被用来启用或者禁用产生奇偶校验，在接收串行字符时该位被用来启用或者禁用奇偶检查
[2]	STOP	R/W	0x0	停止位个数选择位。如果 UART_16550_COMPATIBLE==NO，则只有当 UART 空闲的时候可写，否则一直可写、可读。 该位用来在外设发送和接收数据的时候选择每个字符的停止位个数。如果设置成 0，在串行数据中发送一个停止位。如果设置成 1，并

				<p>且数据位设置成 5 (LCR[1: 0]设置成 0), 发送 1.5 个的停止位。否则发送两个停止位。注意不管选择了多少个停止位, 接收端都只会检测第一个停止位。</p> <ul style="list-style-type: none"> •0-1 stop bit •1-1.5 停止位当 DLS(LCR[1: 0]等于 0), 否则 2 停止位 <p>注意: 由于在传输一些配置和波特时钟除数值相关的字符中可能被插入一些空闲时间, 所以 DW_apd_uart 停止位间隔可能会变长。</p>
[1:0]	DLS (or CLS, as used in legacy)	R/W	0x0	<p>数据长度选择位。如果 UART_16550_COMPATIBLE==NO, 则只有当 UART 空闲的时候可写, 否则一直可写、可读。该位被用来选择外设发送和接收时每个字符的数据位个数。可供选择的个数范围如下</p> <ul style="list-style-type: none"> •00-5 bits •01-6 bits •10-7 bits •11-8 bits

4.7.1.2.8 FCR (0x08)

表 4-32 FCR

位	名称	读写方式	默认值	说明
[31:8]	保留	R/W	0x0	保留
[7:6]	RCVR Trigger (or RT)	WO	0x0	<p>RCVR 触发位。被用于选择数据有效的中断产生时接收 FIFO 触发标准。在使用自动流控模式, 当 rts_n 信号被失效时, 它是确定的。当 dma_rx_req_n 信号被声明在确定的模式操作时 FIFO 通常是确定的。支持的触发方式如下:</p> <ul style="list-style-type: none"> •00-1 character in the FIFO •01-FIFO ¼ full •10-FIFO ½ full •11-FIFO 2 less than full
[5:4]	TX Empty	WO	0x0	TX 空触发, 当 THRE_MODE_USER ==

	Trigger (or TET)			Disabled, 写操作无效。当该模式有效时, THRE 中断产生时被用于选通 TX 的临界值。当 dma_rx_req_n 信号被声明在确定的模式操作时它通常是确定的 支持的触发方式如下: •00–FIFO empty •01–2 characters in the FIFO •10–FIFO ¼ full •11–FIFO ½ full
[3]	DMA Mode (or DMAM)	WO	0x0	DMA 模式。在附加的 DMA handshaking 信号没有选择情况下, DMA 信号发送模式用于决定 dma_tx_req_n 和 dma_rx_req_n 输出信号。 •0–模式 0 •1–模式 1
[2]	XMIT FIFO Reset (or XFIFOR)	WO	0x0	XMIT FIFO 复位。复位这个控制器, FIFO 的发送和接收为空。当附加 DMA handshaking 信号被选通(DMA_EXTRA == YES), DMA TX 请求和单独的信号会失效。注意: 该位会自动清零。不需要进行清零操作。
[1]	RCVR FIFO Reset (or RFIFOR)	WO	0x0	RCVR FIFO 复位。复位这个控制器, FIFO 的发送和接收为空。当附加 DMA handshaking 信号被选通(DMA_EXTRA == YES), DMA RX 请求和单独的信号会失效。注意: 该位会自动清零。不需要进行清零操作。
[0]	FIFO Enable (or FIFOE)	WO	0x0	FIFO 使能位。该位使能或禁用发送(XMIT)和接收(RCVR) FIFO。当该位被改变, XMIT 和 RCVR FIFO 复位。

4.7.1.2.9 LSR(0X14)

表 4-33 LSR

位	名称	读写方式	默认值	说明
[31:8]	保留	RO	0x0	保留

[7]	RFE	RO	0x0	<p>接收 FIFO 错误位。</p> <p>仅当 FFIFO_MODE!=NONE 并且 FIFO 使能时有效 (FCR [0]设置为 1)。</p> <p>该位用来指示在 FIFO 中至少有一个奇偶错误或帧错误或数据终止发送。</p> <ul style="list-style-type: none"> •0-RX FIFO 没有错误 •1-RX FIFO 错误 <p>当 LSR 被读取, 并且错误在接收 FIFO 顶端, 而且 FIFO 里没有其他错误时被清除。</p>
[6]	TEMT	RO	0x1	<p>发送器空位。</p> <p>仅当 FFIFO_MODE!=NONE 并且 FIFO 使能时有效 (FCR [0]=1)。</p> <p>只要发送移位寄存器和 FIFO 都是空, 该位就置位。如果在非 FIFO 模式或 FIFO 禁用, 只要发送器保持寄存器和发送移位寄存器都为空, 该位被设置。</p>
[5]	THRE	RO	0x1	<p>发送保持寄存器空标志位。</p> <p>如果 THRE_MODE_USER == Disabled 或禁用 THRE 模式 (不论 FIFO 的是否正在执行), 该位表明 THR 或发送 FIFO 为空。不论数据从 THR 或 TX FIFO 写到发送移位寄存器还是没有数据写到 THR 或 TX FIFO, 该位都会被置位。如果启用了 THRE 中断, 这也会突然引发 THRE 中断。如果 THRE_MODE_USER == Enabled 并且 FIFO_MODE!=NONE, 而且所有模式都处于激活状态 (IER[7]=1 并且 FCR[0]=1), 此时该位将指示发送 FIFO 是否满 (阈值由 FCR [4] 阈值设置), 而不是控制 THRE 中断。</p>
[4]	BI	RO	0x0	<p>发送中断位。该位用于指示串口输入数据时发生了中断, 如果在 UART 模式 (SIR_MODE == Disabled), 每当输入的逻辑 0 的保持时间超过开始时间 + 数据位 + 停止位和奇偶校验的总和时, 则该位置位。如果在红外模式 (SIR_MODE == Enabled), 每当输入的逻辑 0 的保持时间超过开始时间 + 数据位 + 停止位和奇偶校验的总和时, 则该位置位。</p> <p>当串口仅接收到一个字符, 并且该字符内所有位全部为 0 时, 该位置位。被中断的数据到达 FIFO</p>

				<p>的顶端时，该位被置位。</p> <p>读 LSR 将清除 BI 位。</p> <p>在 non-FIFO 模式下，BI 位一直置位，直到 LSD 被读取。</p>
[3]	FE	RO	0x0	<p>帧错误位。</p> <p>该位用来表示在接收器中发生一个帧错误。当接收的数据没有检测到一个有效的 STOP 位时将会发生帧错误。在 FIFO 模式中，由于帧错误与接收字符有关，当帧错误的字符到达 FIFO 的顶部时才设置该位，当一个帧错误发生时，DW_apb_uart 尝试重新同步。他会假设错误是由于下一个字符的开始位错误造成的，然后继续接收其他位。</p> <p>特别注意：如果 break interrupt(LSR[4])发生该位将被置位。</p> <ul style="list-style-type: none"> •0-无帧错误 •1-帧错误 <p>读 LSR 将清除 FE 位。</p>
[2]	PE	RO	0x0	<p>当 LCR[3]=1 时，此时指示在接收器中的奇偶校验错误的发生。在 FIFO 模式中，由于奇偶错误与接收字符有关，当奇偶性错误的字符到达 FIFO 的顶部时才设置该位，</p> <p>特别注意：如果 break interrupt(LSR[4])发生该位将被置位。</p> <ul style="list-style-type: none"> •0-无奇偶校验错误 •1-奇偶校验错误 <p>读 LSR 将清除 PE 位。</p>

[1]	OE	RO	0x0	<p>溢出错误位。</p> <p>该是用来指示溢出错误的发生。如果在读取旧的数据前接收到一个新数据字符将发生这种情况。</p> <p>在 non-FIFO 模式下，当之前的字符没有从 RBR 中读出，此时一个新的字符到达接收器时 OE 位被置位。发生这种情况时，RBR 的数据被覆盖。在 FIFO 模式中，当 FIFO 已满且新字符到达接收器时溢出错误也会发生。此时在 FIFO 中的数据被保留，而在接收移位寄存器中的数据将会丢失。</p> <ul style="list-style-type: none"> •0-无溢出错误。 •1-溢出错误 <p>读 LSR 将清除 OE 位。</p>
[0]	DR	RO	0x0	<p>数据就绪位。该位用来指示在 RBR 或者接收 FIFO 中至少已接收一个字符</p> <ul style="list-style-type: none"> •0-无数据就绪 •1-数据就绪 <p>在 non-FIFO 模式下，读取 RBR 时该位被清除，或者在 FIFO 模式下，接收 FIFO 为空时该位被清除。</p>

4. 7. 1. 2. 10 USR(0x7C)

表 4-34 USR

位	名称	读写方式	默认值	说明
[31:5]	保留	RO	0x0	保留
[4]	RFF	RO	0x0	<p>指示接收 FIFO 全满。该位仅当 FIFO_STAT== YES 时有效。</p> <ul style="list-style-type: none"> •0-接收 FIFO 未满足 •1-接收 FIFO 满 <p>当 RX FIFO 未满足该位被清零。</p>
[3]	RFNE	RO	0x0	<p>指示接收 FIFO 不为空。该位仅当 FIFO_STAT== YES 时有效。</p>

				<ul style="list-style-type: none"> •0-接收 FIFO 为空 •1-接收 FIFO 不为空 <p>当 RX FIFO 为空时该位被清零。</p>
[2]	TFE	RO	0x1	<p>指示发送 FIFO 空。该位只有当 FIFO_STAT== YES 有效。</p> <ul style="list-style-type: none"> •0-发送 FIFO 不为空 •1-发送 FIFO 空 <p>当 TX FIFO 不再是空，该位被清零。</p>
[1]	TFNF	RO	0x1	<p>指示发送 FIFO 未满。该位只有当 FIFO_STAT== YES 有效。</p> <ul style="list-style-type: none"> •0-发送 FIFO 满 •1-发送 FIFO 未满 <p>当 TX FIFO 满时该位被清零。</p>
[0]	BUSY	RO	0x0	<p>指示 UART 忙。</p> <p>该位只有当 UART_16550_COMPATIBLE == NO 时有效，该位清 0 时表明 DW_apb_uart 空闲或处于 idle 状态。</p> <ul style="list-style-type: none"> •0-DW_apb_uart 空闲或处于 idle 状态 •1-DW_apb_uart 忙（主动传输数据） <p>该位将在以下任意条件发生时设置为 1（忙）：</p> <ol style="list-style-type: none"> 1.正在进行数据传输 2.当不使用 FIFO 的访问模式和波特率除数为非零时，并且当前传输数据在 THR 中。 3.接口正在进行数据接收 4.当 RBR 正在接收数据时，FIFO 访问模式不允许使用 <p>注意：尽管其他设备发送了一个新的字符，串口的 busy 位却处于清除状态，也就是说，虽然一个新字</p>

				符的 start 位刚到达 DW_apb_uart，但是在 THR 和 RBR 却没有数据，此时串口的 busy 位也处于清除状态。因为判定 start 位有效是在该位的中间，而不是起始位置，具体何时有效要根据波特率的分频设置。如果第二个系统时钟已经执行（CLOCK_MODE ==Enabled），对该位的判定也会被较慢时钟推迟的几个周期。
--	--	--	--	---

4.8 I2C 接口

I2C（Inter-Integrated Circuit）总线分别包含一条串行数据线 SDA 与一条串行时钟线 SCL。I2C 用于连接微控制器及其外围设备，是微电子通信控制领域广泛采用的一种总线标准。它是同步通信的一种特殊形式，具有接口线少、控制方式简单、器件封装形式小、通信速率较高等优点。芯片包含 2 个 I2C 接口，分别为 I2C 0 和 I2C 1。

4.8.1 寄存器说明

4.8.1.1 基地址

表 4-35 I2C 基地址

名称	基地址
I2C 0	0x2800_2000
I2C 1	0x2800_3000

4.8.1.2 寄存器列表

表 4-36 I2C 寄存器列表

偏移地址	名称	读写类型	默认值	说明
0x00	IC_CON	R/W	0x7F	I2C 控制寄存器
0x04	IC_TAR	R/W	0x1055	I2C 目的地址寄存器
0x08	IC_SAR	R/W	0x55	I2C 从设备地址寄存器
0x10	IC_DATA_CMD	R/W	0x0	I2C 接收/发送数据 buff 和命令寄存器
0x14	IC_SS_SCL_HCNT	R/W	0x190	I2C 标准速率时钟计数器，高 32 位
0x18	IC_SS_SCL_LCNT	R/W	0x1D	I2C 标准速率时钟计数器，低 32 位

	NT		6	
0x1C	IC_FS_SCL_HC NT	R/W	0x3C	I2C 快速速率时钟计数器, 高 32 位
0x20	IC_FS_SCL_LC NT	R/W	0x82	I2C 快速速率时钟计数器, 低 32 位
0x24	IC_HS_SCL_H CNT	R/W	0x6	I2C 高速速率时钟计数器, 高 32 位
0x28	IC_HS_SCL_LC NT	R/W	0x10	I2C 高速速率时钟计数器, 低 32 位
0x30	IC_INTR_MAS K	R/W	0x8F F	I2C 中断 mask 寄存器
0x34	IC_RAW_INTR _STAT	RO	0x0	I2C 原始中断状态寄存器
0x38	IC_RX_TL	R/W	0x0	I2C 接收 FIFO 门限寄存器
0x3C	IC_TX_TL	R/W	0x0	I2C 发送 FIFO 门限寄存器
0x6C	IC_ENABLE	R/W	0x0	I2C 使能寄存器
0x70	IC_STATUS	RO	0x6	I2C 状态寄存器

4.8.1.2.1 IC_CON(0x00)

表 4-37 IC_CON(0x00)寄存器说明

位	名称	读写方式	默认值	说明
[15:7]	-	-	-	-
[6]	IC_SLAVE_DISABLE	R/W	IC_SLAVE_DISABLE 配置参数	<p>此位是 I2C Slave 功能是否关闭的控制位。即在使用 I2C 功能时通过配置此参数控制 I2C Slave 功能是打开还是关闭。软件驱动可以在系统复位后配置此参数, 即通过软件配置 Slave 的使能或关闭并不是必需的。在默认状态下和复位状态下 I2C 的 Slave 功能均是使能的。如果此位设置为 1, 则 I2C 控制器只能作为 Master 使用, 不能响应反向 Slave 的请求。</p> <ul style="list-style-type: none"> •0-使能 I2C Slave 功能 •1-关闭 I2C Slave 功能
[5]	IC_RESTART_EN	R/W	IC_RESTART_EN 配置参数	<p>此位设置作为 I2C Master 使用时是否支持 restart 功能。某些 I2C Slave 设备不能处理 Restart 信号, 但多数 I2C Slave 设备均能处理 Restart 信号。</p> <ul style="list-style-type: none"> •0-不支持 Restart

				<ul style="list-style-type: none"> •1-支持 Restart <p>当设备不支持 RESTART 功能时，I2C 的 Master 控制器支持以下功能：</p> <ul style="list-style-type: none"> •不发送起始字节 •不支持 Hs 工作模式 •不能进行 10 位地址读操作。 <p>在不支持 Restart 功能时进行以上操作，IC_RAW_INTR_STAT 寄存器中的 TX_BART 标志会被置起。</p>
[4]	C_10BIT_ADDR_MASTER or C_10BIT_ADDR_MASTER_rd_only	R/W	IC_10BITADDR_MASTER 配置参数	<p>当 I2C_DYNAMIC_TAR_UPDATE 参数为 0 (“No”) 时，此位为 IC_10BITADDR_MASTER，控制其作为 I2C Master 时使用 7 位地址模式还是 10 位地址模式进行通信。</p> <p>当 I2C_DYNAMIC_TAR_UPDATE 参数为 1 (“Yes”) 时，此位为 IC_10BITADDR_MASTER_rd_only，读写方式为只读状态，从此处读取的值为 IC_TAR 的第 12 位所设置的值，其含义为：</p> <ul style="list-style-type: none"> •0-7 位地址模式 •1-10 位地址模式
[3]	IC_10BIT_ADDR_SLAVE	R/W	IC_10BITADDR_SLAVE 配置参数	<p>当工作在 slave 模式时，此位用来选择 I2C 控制器响应 7 位地址访问模式还是响应 10 位地址访问请求模式</p> <ul style="list-style-type: none"> •0-7 位地址模式。 <p>此模式下，对于 10 位地址访问请求，I2C 控制器忽略请求，不响应；对于 7 位地址访问请求，I2C 控制器将请求中的 7 位地址与 IC_SAR 寄存器中的 7 位地址值进行比对，若两者一致则响应，若不一致则不响应。</p> <ul style="list-style-type: none"> •1-10 位地址模式。 <p>此模式下，I2C 控制器只响应与 IC_SAR 寄存器中的 10 位地址相匹配的 10 位地址访问请求。</p>
[2:1]	SPEED	R/W	IC_MAX_SPEED_MODE 配置参数	<p>这个参数用来设定 I2C 控制器工作在 Master 模式时的速率。此参数值的范围为 1~IC_MAX_SPEED_MODE。如果软件设定的值不在 1~IC_MAX_SPEED_MODE 范围内，硬件会将其更改为 IC_MAX_SPEED_MODE，以起到保护作用。</p> <ul style="list-style-type: none"> •1: 标准模式 (0 to 100 Kbit/s) •2: 快速模式 (≤ 400 Kbit/s) •3: 高速模式 (≤ 3.4 Mbit/s)
[0]	MASTER	R/W	IC_MASTER	此位是 I2C Master 的使能位。

	_MODE		R_MODE 配置参数	<ul style="list-style-type: none"> •0-关闭 master 功能 •1-使能 master 功能
--	-------	--	----------------	--

4.8.1.2.2 IC_TAR(0x04)

表 4-38 IC_TAR(0x04)寄存器说明

位	名称	读取类型	默认值	说明
[15:13]	-	-	-	-
[12]	IC_10BIT ADDR_M ASTER	R/W	IC_10BITAD DR_MASTE R 配置参数	<p>此位用来选择工作在 I2C Master 时使用 7 位地址模式还是 10 位地址模式进行通信。</p> <ul style="list-style-type: none"> •0-7 位地址模式 •1-10 位地址模式 <p>声明：此位只有在 I2C_DYNAMIC_TAR_UPDATE 为 “Yes”时才有效。</p>
[11]	SPECIAL	R/W	0x0	<p>此位用来选择 I2C 通信使用广播呼叫地址格式还是使用 START BYTE 格式</p> <ul style="list-style-type: none"> •0-使用 IC_TAR 地址格式，忽略 GC_OR_START 设置 •1-使用 GC_OR_START 设定的格式
[10]	GC_OR_S TART	R/W	0x0	<p>如果位 11 (SPECIAL)为 1， 则此位设定 DW_apb_i2c 使用广播呼叫地址格式或是 START BYTE 格式。</p> <ul style="list-style-type: none"> •0-使用广播呼叫地址格式。 <p>此模式下只能进行写操作。如果尝试在此模式下进行读操作， 则 IC_RAW_INTR_STAT 寄存器中的第 6 位 (TX_ABRT) 将会被置位。 如果 SPECIAL 位一直为 1， I2C 控制器则会一直工作在这种模式下。</p> <ul style="list-style-type: none"> •1-START BYTE 格式
[9:0]	IC_TAR	R/W	IC_DEFAULT_TAR_SLAVE_ADDR 配置参数	<p>此处用来存放 Master 通信的目的地址。使用广播呼叫地址格式时此参数可以忽略，使用 START BYTE 格式时只需 CPU 向此处进行一次写操作。</p>

4.8.1.2.3 IC_SAR(0x08)

表 4-39 IC_SAR(0x08)寄存器说明

位	名称	读写方式	默认值	说明
[15:10]	-	-	-	-

[9:0]	IC_SAR	R/W	IC_DEFAULT_SLAVE_ADDRESS 配置参数	IC_SAR 存放 I2C 工作在 Slave 模式下的 Slave 地址。7 位地址模式下只使用 IC_SAR[6:0]。只有在关闭 I2C 接口功能时 (IC_ENABLE=0) 才能更新 IC_SAR 的值，在 I2C 接口处于使能状态时不能改变 IC_SAR 的值。
-------	--------	-----	-------------------------------	---

4.8.1.2.4 IC_DATA_CMD(0x10)

表 4-40 IC_DATA_CMD(0x10)寄存器说明

位	名称	读取类型	默认值	说明
[15:11]	-	-	-	-
[10]	RESTART	WO	-	<p>此位设置是否在发送或接收一个字节数据前发起 RESTART，且只有在 IC_EMPTYFIFO_HOLD_MASTER_EN 为 1 时有效。</p> <ul style="list-style-type: none"> •1-如果 IC_RESTART_EN=1，不管传输方向与上次传输一致还是相反，在发送或接收数据前会发起一个 RESTART；如果 IC_RESTART_EN=0，则使用 START/Stop 配对模式，每次以 START 作为一次传输的开始，以 Stop 结束一次传输。 •0-如果 IC_RESTART_EN=1，则只有在传输方向与上次发生改变时发起一个 RESTART；如果 IC_RESTART_EN=0，则使用 START/Stop 配对模式，每次以 START 作为一次传输的开始，以 Stop 结束一次传输。
[9]	STOP	WO	-	<p>此位设置是否在发送或接收到一个字节数据后发起 STOP，且只有在 IC_EMPTYFIFO_HOLD_MASTER_EN 为 1 时有效。</p> <ul style="list-style-type: none"> •1-不管 Tx FIFO 是否为空，在发送或接收数据后都会发起一个 STOP。如果 Tx FIFO 不为空，则在发送或接收数据后，总线的 Master 端会立即通过产生 START 和申请总线仲裁的方式开始一次新的通信。 •0-不管 Tx FIFO 是否为空，在发送或接收数据后都不发起 STOP。如果 Tx FIFO 不为空，则继续发送或接收当前通信的其他数据字节(由 CMD 位决定是发送还是接收)；如果 Tx FIFO 为空，总线的 Master 端会持续拉低 SCL 信号线并将总线挂起，直到 Tx FIFO 中有新的有效值。
[8]	CMD	WO	0x0	<p>此位是 I2C 控制器工作在 Master 模式时进行读写操作的控制位。控制器工作在 Slave 模式时，此位值无效。</p> <ul style="list-style-type: none"> •1-读 •0-写 <p>工作在 Slave 接收模式时不需要考虑 CMD 位的设定。工作在</p>

				Slave 发送模式时，CMD=0 表示 IC_DATA_CMD 中的数据将被发送。在对 CMD 位进行操作时需要考虑以下情况：无论 IC_RAW_INTR_STAT 中的 SPECIAL 位(第 11 位)是否被清 0，在发送广播呼叫地址格式后进行读操作都会导致 TX_ABRT 中断被置位 (IC_RAW_INTR_STAT 寄存器中的第 6 位)；如果在收到 RD_REQ 中断后软件置 CMD 位为 1 也同样会导致 TX_ABRT 中断事件的发生，即 TX_ABRT 位被置 1。
[7:0]	DAT	WO	0x0	DAT 中存放用来发送的数据或从 I2C 总线上接收到的数据。在开始一次读操作时向 DAT 中写入数据将被 DW_apb_i2c 忽略，但此时从 DAT 读取的数据则是从 I2C 总线接口接收到的数据。

4.8.1.2.5 IC_SS_SCL_HCNT(0x14)

表 4-41 IC_SS_SCL_HCNT(0X14)寄存器

位	名称	读取类型	默认值	说明
[15:0]	IC_SS_SCL_HCNT	R/W	IC_SS_SCL_HGH_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设置，用于指示正确的 I/O 时序以及设置标准速率下 SCL 高电平持续时间的计数值。</p> <p>该寄存器仅当 I2C 接口在不使能情况下 (当 IC_ENABLE=0 时) 可写。其他情况下的写操作无效。寄存器最小取值为 6，比 6 小的值无法设置，若设置值小于 6，则硬件将寄存器值设置为 6。当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低 32 位数据，之后再配置高 32 位。</p> <p>当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。</p>

4.8.1.2.6 IC_SS_SCL_LCNT(0x18)

表 4-42 IC_SS_SCL_LCNT(0X18)寄存器

位	名称	读取类型	默认值	说明
[15:0]	IC_SS_SCL_LCNT	R/W	IC_SS_SCL_LOW_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设置，用于明确正确的 I/O 时序。该寄存器用于设置标准速率下 SCL 低电平持续时间的计数值。</p> <p>该寄存器仅当 I2C 接口在不使能情况下</p>

				<p>(当 IC_ENABLE=0 时) 可写。其他情况下的写操作无效。</p> <p>寄存器最小取值为 8，比 8 小的值无法设置，若设置值小于 8，则硬件将强制把寄存器的值设置为 8。</p> <p>当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低 32 位数据，之后再配置高 32 位。当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。</p>
--	--	--	--	---

4.8.1.2.7 IC_FS_SCL_HCNT(0x1C)

表 4-43 IC_FS_SCL_HCNT(0X1C)寄存器

位	名称	读写方式	默认值	说明
[15:0]	IC_FS_SCL_HCNT	R/W	IC_FS_SCL_HIGH_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设计，用于明确 I/O 时序。该寄存器用于设置快速模式下 SCL 高电平持续时间的计数值。用于发送高速模式下的 Mater Code 和 START BYTE 或 General Call。</p> <p>当 IC_MAX_SPEED_MODE= standard 时，此寄存器的读写方式为只读且返回值为全 0。该寄存器仅当 I2C 接口在不使能情况下（当 IC_ENABLE=0 时）可写。其他情况下的写操作无效。</p> <p>寄存器最小取值为 6，比 6 小的值无法设置，若设置值小于 6，则硬件将寄存器的值强制设置为 6。当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低字节（8 位）数据，之后再配置高字节（8 位）。</p> <p>当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。</p>

4.8.1.2.8 IC_FS_SCL_LCNT(0x20)

表 4-44 IC_FS_SCL_LCNT(0X20)寄存器

位	名称	读写方式	默认值	说明
[15:0]	IC_FS_SCL_LCNT	R/W	IC_FS_SCL_LOW_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设计，用于明确 I/O 时序。该寄存器用于设置快速模式下 SCL 低电平持续时间的计数值。用于发送高速模式下的 Mater Code 和 START BYTE 或 General Call。</p> <p>当 IC_MAX_SPEED_MODE= standard 时，此寄存器的读写方式为只读且返回值为全 0。该寄存器仅当 I2C 接口在不使能情况下（当 IC_ENABLE=0 时）可写。其他情况下的写操作无效。</p> <p>寄存器最小取值为 8，比 8 小的值无法设置，若设置值小于 8，则硬件将寄存器值设置为 8。</p> <p>当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低字节（8 位）数据，之后再配置高 32 位字节（8 位）。当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。</p>

4.8.1.2.9 IC_HS_SCL_HCNT(0x24)

表 4-45 IC_HS_SCL_HCNT(0X24)寄存器

位	名称	读写方式	默认值	说明
[15:0]	IC_HS_SCL_HCNT	R/W	IC_HS_SCL_HIGH_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设计，用于明确 I/O 时序。该寄存器用于设置高速模式下 SCL 高电平持续时间的计数值。</p> <p>SCL 高电平时间依赖于总线的负载情况。接 100pF 的负载时，高电平时间为 60ns;接 400pF 的负载时，高电平时间为 120ns。IC_MAX_SPEED_MODE!= high</p>

				<p>时，此寄存器为只读且返回值为全 0。</p> <p>该寄存器仅当 I2C 接口在不使能情况下（当 IC_ENABLE=0 时）可写。其他情况下的写操作无效。</p> <p>寄存器最小取值为 6，比 6 小的值无法设置，若设置值小于 6，则硬件将寄存器值设置为 6。当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低字节（8 位）数据，之后再配置高字节（8 位）。</p> <p>当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。</p>
--	--	--	--	--

4.8.1.2.10 IC_HS_SCL_LCNT(0x28)

表 4-46 IC_HS_SCL_LCNT(0X28)寄存器

位	名称	读写方式	默认值	说明
[15:0]	IC_HS_SCL_LCNT	R/W	IC_HS_SCL_LOW_COUNT 配置参数	<p>该寄存器必须在 I2C 总线传输之前进行设计，用于明确正确的 I/O 时序。该寄存器用于设置高速模式下 SCL 低电平持续时间的计数值。</p> <p>SCL 低电平时间依赖于总线的负载情况。接 100pF 的负载时，低电平时间为 160ns；接 400pF 的负载时，低电平时间为 320ns。</p> <p>IC_MAX_SPEED_MODE!= high 时，此寄存器为只读且返回值为全 0。</p> <p>该寄存器仅当 I2C 接口在不使能情况下（当 IC_ENABLE=0 时）可写。其他情况下的写操作无效。</p> <p>当 APB_DATA_WIDTH=8 时，寄存器设置的顺序尤为关键，此时，首先应配置计数器的低字节（8 位）数据，之后再配置高字节（8 位）。</p> <p>寄存器最小取值为 8，比 8 小的值无法设置，若设置值小于 8，则硬件将寄存器值设置为 8。</p>

				当 IC_HC_COUNT_VALUES 为 1 时，该寄存器只读。
--	--	--	--	------------------------------------

4.8.1.2.11 IC_INTR_MASK(0x30)

表 4-47 IC_INTR_MASK(0X30)寄存器

位	名称	读写方式	默认值	说明
[15:12]	-	-	-	-
[11]	M_GEN_CALL	R/W	0x1	Bit[11] 表示中断事件标志屏蔽控制位。置位时，如果对应的中断事件发生，则不会置位 IC_INTR_STAT 寄存器中对应的中断标志位
[10]	M_START_DET	R/W	0x0	
[9]	M_STOP_DET	R/W	0x0	
[8]	M_ACTIVITY	R/W	0x0	
[7]	M_RX_DONE	R/W	0x1	
[6]	M_TX_ABRT	R/W	0x1	
[5]	M_RD_REQ	R/W	0x1	
[4]	M_TX_EMPTY	R/W	0x1	
[3]	M_TX_OVER	R/W	0x1	
[2]	M_RX_FULL	R/W	0x1	
[1]	M_RX_OVER	R/W	0x1	
[0]	M_RX_UNDER	R/W	0x1	

4.8.1.2.12 IC_RAW_INTR_STAT(0x34)

表 4-48 IC_RAW_INTR_STAT(0x34)寄存器说明

位	名称	读写方式	默认值	说明
[15:12]	-	-	-	-
[11]	GEN_CALL	RO	0x0	只有接收并识别到广播呼叫地址格式时才会被置位。一旦 GEN_CALL 置位，则只有通过关闭 I2C 控制器或 CPU 读取 IC_CLR_GEN_CALL 寄存器中的第 0 位，GEN_CALL 位才能被清 0。I2C 控制器会把接收到的数据存放在 Rx 缓冲区中。
[10]	START_DET	RO	0x0	此位状态表示在 I2C 总线接口上是否产生了 START 或 RESTART，与控制器工作在 Master 模式还是 Slave 模式无关。
[9]	STOP_DET	RO	0x0	此位状态表示在 I2C 总线接口上是否产生了 STOP。与控制器工作在 Master 模式还是 Slave 模式无关。

[8]	ACTIVITY	RO	0x0	<p>此位标志 I2C 控制器的活动状态。</p> <p>有 4 种方法可以清除 ACTIVITY 标志：</p> <ul style="list-style-type: none"> •关闭 DW_apb_i2c •读取 IC_CLR_ACTIVITY 寄存器 •读取 IC_CLR_INTR 寄存器 •系统复位 <p>一旦被置位则会一直保持置位，直到通过以上四种方式中的一种将其标志清 0。即使在 Idle 状态下如果采取清 0 动作的话也会一直保持置位。</p>
[7]	RX_DONE	RO	0x0	<p>I2C 控制器工作在 Slave 发送模式下，发送完数据的最后一个字节后，在规定时间内没有收到 Master 端的回应 (ACK)，RX_DONE 将会被置位表示结束。</p>
[6]	TX_ABRT	RO	0x0	<p>该数据位表示 I2C 无法完成传输 FIFO 中存储的计划动作。这个情况在 I2C master 和 slave 中都有可能发生，被认为是传输终止标志。</p> <p>当该数据位为 1 时，IC_TX_ABRT_SOURCE 寄存器将记录传输终止的原因。</p>
[5]	RD_REQ	RO	0x0	<p>读请求标志。当 I2C 控制器工作在 Slave 模式下，且有 Master 尝试从 DW_apb_i2c 中读取数据时，RD_REQ 被置位。I2C 控制器在处理 RD_REQ 请求期间会将 SCL 保持低电平。RD_REQ 是处理器必须响应的中断请求，并在请求处理完成时把 Master 所要的数据放到 IC_DATA_CMD 寄存器中。读取 IC_CLR_RD_REQ 寄存器的值可以将 RD_REQ 标志清 0。</p>
[4]	TX_EMPTY	RO	0x0	<p>当发送缓冲区小于等于 IC_TX_TL 寄存器中设定的门限值时将置位 TX_EMPTY。当缓冲区大于门限值时，硬件会自动把 TX_EMPTY 清 0。IC_ENABLE bit0=0 时，TXFIFO 被刷新复位，TXFIFO 可以认为为空，此时 TX_EMPTY 被置为 1。当总线处于非活动状态时 ic_en=0，TX_EMPTY=0。</p>
[3]	TX_OVER	RO	0x0	<p>在发送过程中，如果发送缓冲区大小达到 IC_TX_BUFFER_DEPTH 且处理器还在尝试通过向 IC_DATA_CMD 中写数据来发起另一个 I2C 命令时，TX_OVER 被置位。即使在控制器功能被关闭的情况下 (IC_ENABLE[0]=0) RX_OVER 状态也会一直保持置位，直到总线进入空闲状态。ic_en=0</p>

				时, TX_OVER 被清 0。
[2]	RX_FULL	RO	0x0	当接收缓冲区大于等于 IC_RX_TL 中设定的门限值 (RX_TL) 时, RX_FULL 置位。当缓冲区小于门限值时, 硬件会自动把 RX_FULL 清 0。IC_ENABLE bit0=0 时, RXFIFO 被刷新复位, RXFIFO 为空, 此时 RX_FULL 被清 0。
[1]	RX_OVER	RO	0x0	当接收缓冲区大小达到 IC_RX_BUFFER_DEPTH, 且还继续从外部接收数据时, RX_OVER 置位。TX_OVER 事件会被 I2C 控制器响应, 且在缓冲区满后接收到的所有数据均被丢弃。即使在控制器功能被关闭的情况下 (IC_ENABLE[0]=0) RX_OVER 状态也会一直保持置位, 直到总线进入空闲状态。ic_en=0 时, RX_OVER 被清 0。
[0]	RX_UNDER	RO	0x0	处理器通过访问 IC_DATA_CMD 寄存器获取接收缓冲区的数据时, 若接收缓冲区为空, RX_UNDER 被置位。即使在控制器功能被关闭的情况下 (IC_ENABLE[0]=0) RX_UNDER 状态也会一直保持置位, 直到总线进入空闲状态。ic_en =0 时, RX_UNDER 被清 0。

4.8.1.2.13 IC_RX_TL(0x38)

表 4-49 IC_RX_TL(0x38)寄存器

位	名称	读写方式	默认值	说明
[15:8]	-	-	-	-
[7:0]	RX_TL	R/W	IC_RX_TL 配置参数	接收缓冲区满中断 (RX_FULL) 触发门限控制。有效范围 0~255, 但最大值不能超出缓冲区的深度。如果设定值超出缓冲区的最大深度, 其实际设置的有效大小为缓冲区的最大深度值。0 表示接收缓冲区大于等于 1 时触发中断, 255 表示接收缓冲区大于等于 256 时触发中断。

4.8.1.2.14 IC_TX_TL(0x3C)

表 4-50 IC_TX_TL(0x3C)寄存器

位	名称	读写方式	默认值	说明
---	----	------	-----	----

[15:8]	-	-	-	-
[7:0]	TX_TL	R/W	IC_TX_TL 配置参数	发送缓冲区满中断 (TX_EMPTY) 触发门限控制。有效范围 0~255，但最大值不能超出缓冲区的深度。如果设定值超出缓冲区的最大深度，其实际设置的有效大小为缓冲区的最大深度值。0 表示发送缓冲区小于等于 0 时触发中断，255 表示发送缓冲区小于等于 255 时触发中断。

4.8.1.2.15 IC_ENABLE(0x6C)

表 4-51 IC_ENABLE(0X6C)寄存器

位	名称	读写方式	默认值	说明
[15:1]	-	-	-	-
[0]	ENABLE	R/W	0x0	<p>I2C 控制器使能或关闭控制位。</p> <ul style="list-style-type: none"> •0-关闭 I2C 控制器功能 •1-使能 I2C 控制器功能 <p>以下现象会在 I2C 控制器功能关闭时出现：</p> <ul style="list-style-type: none"> •TXFIFO 和 RXFIFO 被刷新 •IC_INTR_STAT 寄存器中的状态保持不变。 <p>在控制器发送数据过程中关闭 I2C 控制器功能，则在当前发送操作完成后，清空发送缓冲区中的内容。</p> <p>在控制器接收数据过程中关闭 I2C 控制器功能，通信将在接收完当前字节后停止，且不响应使用 asynchronous pclk and ic_clk 的系统 (IC_CLK_TYPE=1)。在使能或关闭控制器时有 2 个 ic_clk 的延迟。</p>

4.8.1.2.16 IC_STATUS(0x70)

表 4-52 IC_STATUS(0X70)寄存器

位	名称	读写方式	默认值	说明
[31:7]	-	-	-	-
[6]	SLV_ACTI	RO	0x0	Slave FSM 活动状态标志。Slave FSM(Slave

	VITY			Finite State Machine 不在 Idle 状态时被置位 <ul style="list-style-type: none"> •0–Slave FSM 处于 Idle 状态, 此时 I2C 控制器的 Slave 功能处于非活动状态。 •1–Slave FSM 处于非 Idle 状态, 此时 I2C 控制器的 Slave 功能处于活动状态。
[5]	MST_ACTIVITY	RO	0x0	Master FSM 活动状态标志。Master FSM(Master Finite State Machine) 处于非 Idle 状态时被置位。 <ul style="list-style-type: none"> •0–Master FSM 处于 Idle 状态, 此时 I2C 控制器的 Master 功能处于非活动状态 •1–Master FSM 处于非 Idle 状态, 此时 I2C 控制器的 Master 功能处于活动状态。
[4]	RFF	RO	0x0	接收 FIFO 全满标志。当接收 FIFO 全满时置位; FIFO 中有一个或一个以上为空时 0。 <ul style="list-style-type: none"> •0–接收 FIFO 未滿 •1–接收 FIFO 全滿
[3]	RFNE	RO	0x0	接收 FIFO 不为空标志。当接收 FIFO 不为空时置位, 为空时清 0。 <ul style="list-style-type: none"> •0–接收 FIFO 为空 •1–接收 FIFO 不为空
[2]	TFE	RO	0x1	发送 FIFO 全空标志。发送 FIFO 全空时置位; 发送 FIFO 有一个或一个以上不为空的值时清 0。此标志的产生不伴随有中断发生。 <ul style="list-style-type: none"> •0–发送 FIFO 不为空 •1–发送 FIFO 为空
[1]	TFNF	RO	0x1	发送 FIFO 未滿标志。发送 FIFO 中有一个或一个以上位置为空时置位; 发送 FIFO 满时清 0。 <ul style="list-style-type: none"> •0–发送 FIFO 已滿 •1–发送 FIFO 未滿
[0]	ACTIVITY	RO	0x0	I2C 控制器活动状态标志

4.9 GPIO 接口

芯片包含了 32 个 GPIO 端口, 分成 4 组, 分别是 GPIOA[0:7], GPIOB[0:7], GPIOC[0:7], GPIOD[0:7]。其中有部分 GPIO 端口是复用的, 详细复用情况见表 4-53。如果要选择 GPIO 功能, 设置 REG_CRU_SEL_GPIO(0x28100C00)寄存器的值为 0x0。GPIO 端口可通过软件分别配置成输入或输出。

4.9.1 GPIO 复用说明

4.9.1.1 GPIO 复用寄存器地址

表 4-53 GPIO 复用寄存器地址

名称	基地址
GPIO 复用控制寄存器	0x2810_0C00

4.9.1.2 GPIO 复用寄存器描述

表 4-54 GPIO 复用寄存器描述

位	名称	读写方式	默认值	说明
[5:0]	GPIO 复用控制位	R/W	0x10	每一位控制一组 GPIO 端口的复用： [0]位控制 <code>cru_sel_uart_func</code> , 1 选择此功能, 0 选择 GPIO 功能。 [1]位控制 <code>cru_sel_lpc_func0</code> , 1 选择此功能, 0 选择 GPIO 功能。 [2]位控制 <code>cru_sel_lpc_func1</code> , 1 选择此功能, 0 选择 GPIO 功能。 [3]位控制 <code>cru_sel_spics</code> , 1 选择此功能, 0 选择 GPIO 功能。 [4]位控制 <code>cru_sel_rst_state</code> , 1 选择此功能, 0 选择 GPIO 功能。 [5]位控制 GMAC 配置信号 <code>cru_gmu_cfg_cttw</code> , 1 选择此功能, 0 选择 GPIO 功能。

当选择某种复用功能后, 对应的 GPIO 端口说明如表 4-55 所示。例如选择了 `cru_sel_lpc_func0`, 和 `cru_sel_lpc_func1`, 那么对应的 `GPIOB[7]`, `GPIO[6:7]`, `GPIOD[0:7]`被复用成 LPC 端口。

表 4-55 GPIO 复用寄存器说明

控制线	GPIO	复用功能
	<code>cru_sel_xxx=0</code>	<code>cru_sel_xxx=1</code>
CRU_SEL_UART_FUNC NC	PortC[0]	UART0.CD
	PortC[1]	UART0.DTR
	PortC[2]	UART0.DSR

	PortC[3]	UART0.RTS
	PortC[4]	UART0.CTS
	PortC[5]	UART0.RI
CRU_SEL_LPC_FUNC 0	PortC[6]	LPC.irq_outen
	PortC[7]	LPC.irq_n
	PortD[0]	LPC.lframe_n
	PortD[1]	LPC.lreset_n
	PortD[2]	LPC.lck
	PortD[3]	LPC.lad_outen
	PortD[4]	LPC.lad[0]
	PortD[5]	LPC.lad[1]
	PortD[6]	LPC.lad[2]
	PortD[7]	LPC.lad[3]
CRU_SEL_LPC_FUNC 1	PortB[7]	LPC.ldrq_n
CRU_SEL_RST_STAT E	PortA[0]	clk_obv_sel[0]
	PortA[1]	clk_obv_sel[1]
	PortA[2]	clk_obv_sel[2]
	PortA[3]	clk_obv_sel[3]
	PortA[4]	rst_fsm[0]
	PortA[5]	rst_fsm[1]
	PortA[6]	rst_fsm[2]
	PortA[7]	rst_fsm[3]
	PortA_Dir_0	rst_fsm[4]
	PortB_Dir_0	cru_clk_obv
CRU_SEL_SPICS	PortB[0]	spi_csn[2]
	PortB[1]	spi_csn[3]
CRU_SEL_LINKUP1	PortB[2]	peu_linkup[1]
CRU_SEL_LINKUP2	PortB[3]	peu_linkup[2]
CRU_SEL_LINKUP3	PortB[4]	peu_linkup[3]

4.9.2 GPIO 寄存器说明

4.9.2.1 基地址

表 4-56 GPIO 基地址

名称	基地址
GPIO	0x2800_6000

4.9.2.2 寄存器列表

表 4-57 GPIO 内部寄存器描述

偏移地址	名称	读写类型	说明
0x00	GPIOA_DR	R/W	端口 A 数据寄存器 r 位宽: GPIO_PWIDTH_A 默认值: GPIO_SWPORTA_RESET
0x04	GPIOA_DDR	R/W	端口 A 数据方向寄存器 位宽: GPIO_PWIDTH_A 默认值: GPIO_DFLT_DIR_A (for all bits)
0x0c	GPIOB_DR	R/W	端口 B 数据寄存器 位宽: GPIO_PWIDTH_B 默认值: GPIO_SWPORTB_RESET
0x10	GPIOB_DDR	R/W	端口 B 数据方向寄存器 数据位宽: GPIO_PWIDTH_B 默认值: GPIO_DFLT_DIR_B (for all bits)
0x18	GPIOC_DR	R/W	端口 C 数据寄存器 位宽: GPIO_PWIDTH_C 默认值: GPIO_SWPORTC_RESET
0x1c	GPIOC_DDR	R/W	端口 C 数据方向寄存器 位宽: GPIO_PWIDTH_C 默认值: GPIO_DFLT_DIR_C (for all bits)
0x24	GPIOD_DR	R/W	端口 D 数据寄存器 位宽: GPIO_PWIDTH_D 默认值: GPIO_SWPORTD_RESET
0x28	GPIOD_DDR	R/W	端口 D 数据方向寄存器 位宽: GPIO_PWIDTH_D 默认值: GPIO_DFLT_DIR_D (for all bits)
0x50	GPIO_EXT_P ORTA	RO	端口 A 外部端口寄存器 位宽: GPIO_PWIDTH_A 默认值: 0x0
0x54	GPIO_EXT_P ORTB	RO	端口 B 外部端口寄存器 位宽: GPIO_PWIDTH_B

			默认值: 0x0
0x58	GPIO_EXT_P ORTC	RO	端口 C 外部端口寄存器 位宽: GPIO_PWIDTH_C 默认值: 0x0
0x5c	GPIO_EXT_P ORTD	RO	端口 D 外部端口寄存器 位宽: GPIO_PWIDTH_D 默认值: 0x0

注：当 GPIO 的端口被设置为输入时，需要从 GPIO_EXT_PORTX 中读取输入的数据 GPIO_PWIDTH_X=8；其中 X 表示 A/B/C/D。本章内名称以此类推。

4.9.2.2.1 数据寄存器 GPIOX_DR (0x00)

表 4-58 数据寄存器(GPIOX_DR)

位	名称	读写方式	默认值	说明
[31:GPIO_PWIDTH_X]	保留	R/W	-	
[GPIO_PWIDTH_X-1:0]	端口 X 数据寄存器	R/W	GPIO_SWPORTX_RESET	如果端口 X 数据方向寄存器中对应位设置为输出模式，并且端口 X 的对应控制位设置为软件控制，则向这个寄存器写的值是将要通过端口 X 对应的 I/O 引脚输出的信号值

4.9.2.2.2 方向寄存器 GPIOX_DDR (0x04)

表 4-59 方向寄存器(GPIOX_DDR)

位	名称	读写方式	默认值	说明
[31:GPIO_PWIDTH_X]	-		-	
[GPIO_PWIDTH_X-1:0]	端口 X 方向寄存器	R/W	GPIO_DFLT_DIR_X	向这个寄存器写的值每一位分别控制端口 X 对应位的输入输出模式。可以通过 GPIO_DFLT_DIR_X 参数设置系统复位后的默认方式为输入或输出。

				<ul style="list-style-type: none"> •0-输入(默认); •1-输出;
--	--	--	--	--

4.9.2.2.3 外部数据寄存器 GPIO_EXT_PORTX (0x50)

表 4-60 外部数据寄存器(GPIO_EXT_PORTX)

位	名称	读写方式	默认值	说明
[31:GPIO_PWIDTH_H_X]	-	-	-	-
[GPIO_PWIDTH_X-1:0]	外部端口 X	RO	0x0	当端口 X 设置为输入时, 从该位置读取的值为端口 X 上输入的信号状态。当端口 X 设置为输出时, 从这个位置读取的值为通过端口 X 输出的信号值。

4.10 上电时序

表 4-61 上电复位信号列表

信号名	说明
PWR_BTN_EN	上电按钮, 低电平有效
ATX_EN	ATX 电源加电信号, 低电平有效
ATX_GD	ATX 电源加电完成信号
VDDIO_EN	1.8V IO 电源加电信号, 低电平有效
VDDIO_GD	1.8V IO 电源加电信号加电完成信号
VDD_CORE_EN	CPU 核心电源加电信号
VDD_CORE_GD	CPU 核心电源加电完成信号
VDDA_PCIE_EN	PCIE 0.95V 核心电源加电信号, 低电平有效
VDDA_PCIE_GD	PCIE 0.95V 核心电源加电完成信号
VDD_MCU_EN	1.5V DDR 电源加电信号, 低电平有效
VDD_MCU_GD	1.5V DDR 电源加电完成信号
RESET_N	热复位信号, 低有效
POR_N	上电复位信号, 低有效
PCIE_SLOT_RST_N	所有 PCIE 插槽端的复位信号, 信号数目根据 PCIE 设备而定。

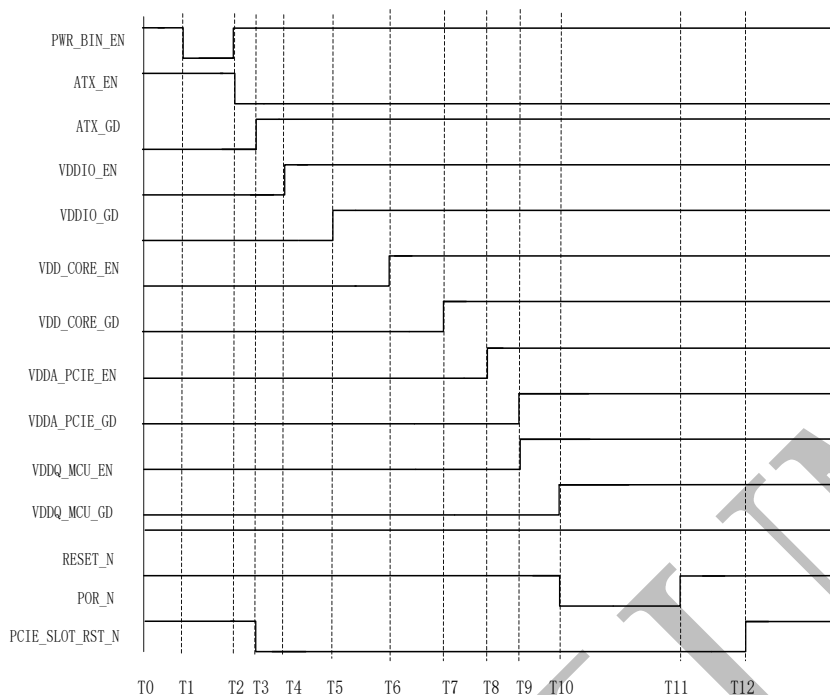


图 4-1 上电复位参考时序图

时序说明:

表 4-62 上电复位参考时序说明

时刻	说明
T0	作为上电复位控制器的 CPLD 或 FPGA 加电并完成复位
T1	加电按钮按下
T2	加电按钮弹起, ATX 电源加电
T3	收到 ATX 加电完成信号, 拉低 PCIE_SLOT_RST
T4	相对 T3 时刻约延时约 20ms, 给 1.8V IO 电源加电
T5	收到加电完成信号
T6	相对 T5 时刻延时 20ms, VDD_CORE 电源上电
T7	收到上电完成信号
T8	相对 T7 时刻延时 10ms, 给 PCIE 0.95V 核心电源加电
T9	收到上电完成信号, 给 VDDQ_MCU 1.5V 电源加电
T10	收到上电完成信号, 保持 REST_N 信号高电平, 拉低 POR_N 开始上电复位
T11	相对 T9 时刻延时 40ms, 拉高 POR_N, 完成上电复位
T12	相对 T9 时刻延时 100ms, 拉高 PCIE_SLOT_RST_N, 完成 PCIE 设备复位

5 电气特性

5.1 极限工作条件

表 5-1 极限工作条件

电压范围	值
内核电压范围 (VDD_090)	-0.3V~1.10V
0.75V 参考电压范围 (VDD_075)	-0.3V~1.5V
0.95V IO 电压范围 (VDD_095)	-0.3V~1.5V
1.5V IO 电压范围 (VDD_15)	-0.3V~1.8V
1.8V IO 电压范围 (VDD_18)	-0.3V~2.0V

5.2 典型工作参数

表 5-2 典型工作参数

参数	符号	条件	最小值	典型值	最大值	单位
内核电源	VDD_CORE			0.9		V
MCU 电源	VDDQ_MCU _x			1.5		V
PCIE 模拟部分电源	VDDA_PCIE			0.95		V
PCIE 数字部分电源	VDDHV_PCIE			1.5		V
IO 电源	VDDIO			1.8		V
锁相环电源	VDDHV_PLL _x			1.8		V
高电平输入电压	V _{IH}		1.26			V
低电平输入电压	V _{IL}				0.54	V
内核电源电流	IVDD_CORE	1.5GHz, 0.9V		40		A
MCU 电源电流	IVDDQ_MCU _{xx}	800MHz, 1.5V		5		A
PCIE 模拟部分电源电流	IVDDA_PCIE	8Gbps, 0.95V		3		A
PCIE 数字部分电源电流	IVDDHV_PCI _E	8Gbps, 1.5V		2		A
IO 电源电流	IVDDIO			1		A
锁相环电源电流	IVDDHV_PLL _x			0.1		A
高电平输出电流	I _{OH}		4.4	9.6	33	mA
低电平输出电流	I _{OL}		-5.5	-9.1	-25	mA

表 5-3 典型工作参数

电压范围	值
内核电压 (VDD_090)	0.9×(1±5%)V
0.75V 参考电压 (VDD_075)	0.75×(1±5%)V
0.95V IO 电压 (VDD_095)	0.95×(1±5%)V
1.5V IO 电压 (VDD_15)	1.5×(1±5%)V
1.8V IO 电压 (VDD_18)	1.8×(1±5%)V

6 封装数据

6.1 封装尺寸

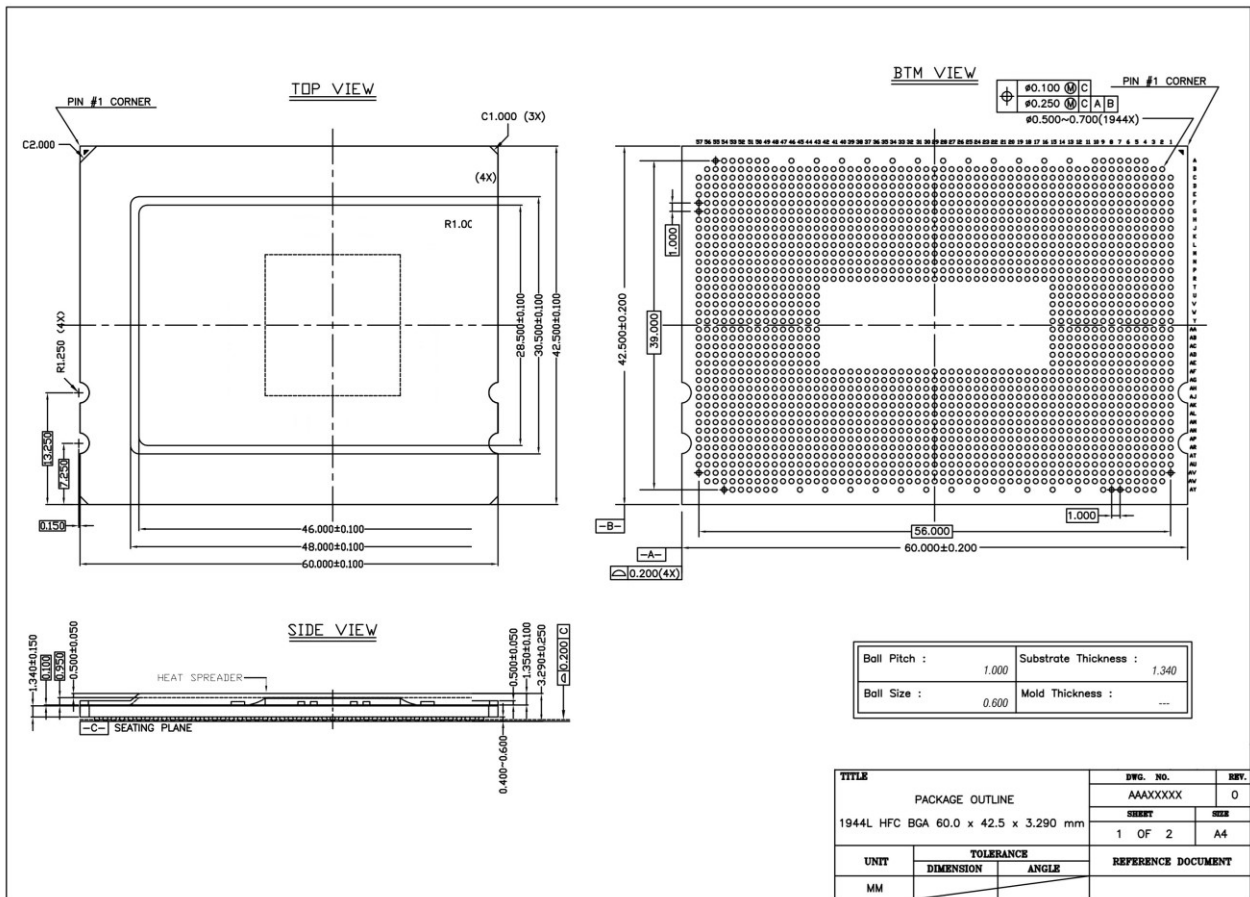


图 6-1 CPU 封装图

表 6-1 CPU 封装尺寸

符号名称	最小/mm	公称/mm	最大/mm
总厚度	3.0	3.29	3.6
球高度	0.4	0.5	0.6
外壳尺寸	42.3	42.5	42.7

	59.8	60	60.2
球间距		1.00	

7 装焊温度曲线

如果采用 Sn10Pb90 作为 BGA 焊球, 则焊接时即可采用无铅焊膏, 也可采用含铅焊膏 (Sn63Pb37)。如果采用无铅 BGA 焊球, 则焊接时必须采用无铅焊膏。如果同一块板上既有含铅器件又有无铅器件, 则焊接时必须采用无铅焊膏。

7.1 无铅焊接温度曲线中各温区的作用

采用德国 Ersa 公司制造的 Hotflow11 回流焊炉和 Sensor Shuttle 温度传感器进行测试点温度采集, 最终得到一条如图 7-1 所示的适应于无铅焊料的温度曲线。

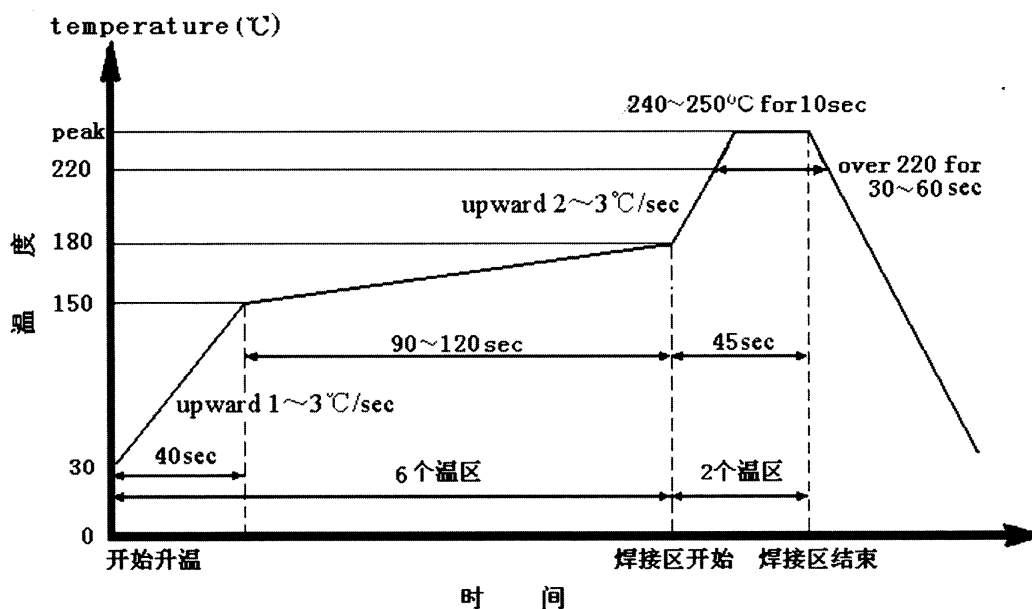


图 7-1 无铅回流焊接曲线

该温度曲线分为五个部分：

(1) 第一升温区：将 PCB 及元器件从室温加热到 150°C。在这个区，由于受基板材料与元件的限制，PCB 及元器件应以 1~3°C/Sec 的速率连续上升，最理想为接近 2°C/Sec，温度升得太快，会对元件造成热冲击或导致 PCB 变形。此时焊膏中的溶剂、气体开始蒸发，同时，焊膏中的助焊剂润湿焊盘、元器件端子和引脚，焊膏软化、塌落、覆盖焊盘、元器件端子和引脚并与氧气隔离。整个升温过程持续 40Sec 左右。

(2) 预热区，又称保温区：温度从 150°C 上升到 180°C，PCB 和元器件得到充分的预热，以防突然进入焊接高温区而损坏 PCB 和元器件，保温区热风温度不变，PCB 和元器件依靠传热温度自然升高 30°C 左右，它的主要功能是提供足够的热能，令焊膏中的助焊剂开始活化，将金属氧化物和某

些污染从焊盘、元件引脚和焊膏金属颗粒上清除，与此同时，挥发性的溶剂和水汽从焊膏中挥发。整个过程持续 90~120Sec（因不同种类焊膏而异）。预热时间不足或过长皆都会导致后期焊锡球的产生。

(3) 第二升温区：温度从 180°C 上升到无铅焊料的熔点 217°C 以上，这个区是助焊剂活动的高峰期，于焊接前做最后的氧化分解，一般时间 20~30Sec，尽量靠近 20Sec。时间过长会使助焊剂中的松香过早耗尽引起再氧化，令焊接不良或产生焊锡球。

(4) 焊接区：温度从 220°C 至峰值温度再回到 220°C，升温速率 2~3°C/Sec。在这个区焊膏中的金属颗粒首先单独熔化，并覆盖在金属表面上。当单个的金属颗粒全部熔化后，液态焊锡对 PCB 的焊盘、元器件端头和引脚润湿、扩散、漫流或回流混合形成焊锡接点。峰值温度的设定一般为焊膏熔点加上 30°C。这个区域的时间为一般为 30~60s（实际焊接时最好 60s~90s），视元件大小不同而不同。假如这个区的温度设得太高，会使温升速率超过 2~3°C/Sec，或达到的峰值温度比理想的高，会引起 PCB 的过分变形，并损坏元器件。

(5) 冷却区：焊料凝固，形成平滑光亮的焊点。冷却速率 4~5°C/Sec，较快的冷却速率可得到较细的颗粒结构和较强的焊接强度与较亮的焊点，但太快会引起元件内部的热应力。

7.2 有铅焊接温度曲线中各温区的作用

以下是日本千住（SENJU）公司有铅焊膏推荐使用的温度曲线。

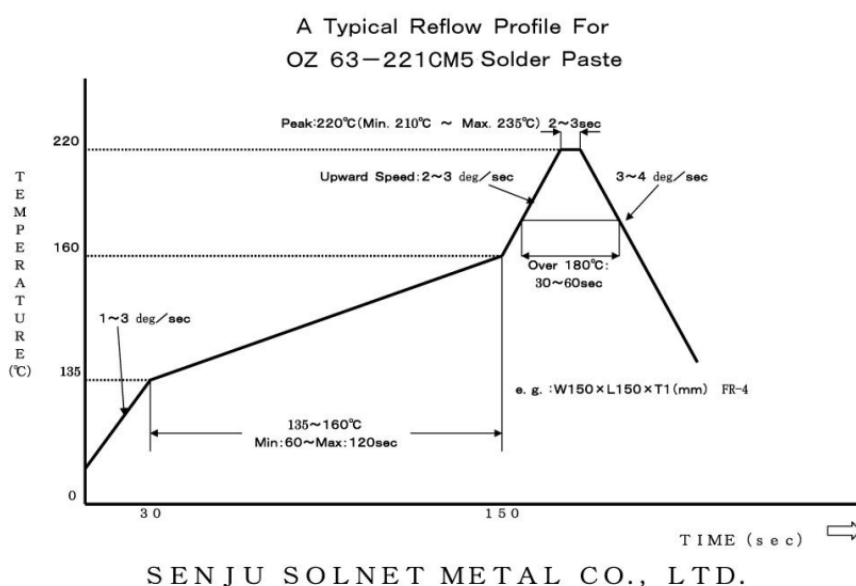


图 7-2 温度曲线

各温区的作用与有铅焊接类似，焊接区的时间一般为 30~60s（实际焊接时最好控制在 60s~90s）。

8 引脚描述

8.1 通用 IO 类引脚 (118 PIN)

表 8-1 通用 IO 类引脚

信号名	引脚编号	信号类型	片内上下拉	功能说明	正常运行连接方式
AA64NAA32	AY53	I	up	1'b0:启动直接进入 AArch32 模式 1'b1:启动直接进入 AArch64 模式	
SHUTDOWN	AY52	O		1'b1:通知片外 CPLD, CPU 可以关电	
CLK_REF	AC46	I	down	参考时钟, 默认 50MHz	
RESET_N	AV52	I	up	热复位信号, 低有效	
POR_N	AW52	I	up	上电复位信号, 低有效	
RSV1	AD46	I		保留引脚	接地
RSV2	AC47	I		保留引脚	接地
RSV3	AB47	I		保留引脚	接地
RSV4	AF57	I		保留引脚	接地
RSV5	AE53	I		保留引脚	接地
RSV6	AE55	O		保留引脚	浮空
CRU_DTI_RST_OK	AE46	O		复位完成信号	
CRU_I2C_SDA	AF56	B	Up	带外诊断专用 I2C 接口	
CRU_I2C_SCL	AF55	I	Up	带外诊断专用 I2C 接口	
ERROR_INT	AV51	O		错误中断输出, 用于带外诊断	
RSV7	AE52	I		保留引脚	接地
RSV8	AW51	I		保留引脚	接地
RSV9	AY51	I		保留引脚	接地
RSV10	AF52	I		保留引脚	接地
RSV11	AG55	I		保留引脚	接地
RSV12	AG56	I		保留引脚	接地
RSV13	AG57	I		保留引脚	接地
RSV14	AH57	O		保留引脚	浮空
RSV15	AH49	I		保留引脚	接地
RSV16	AJ52	I		保留引脚	接地
RSV17	AH50	I		保留引脚	接地
RSV18	AG53	I		保留引脚	接地

RSV19	AK51	I		保留引脚	接地
RSV20	AH53	I		保留引脚	接地
RSV21	AB49	I		保留引脚	接地
RSV22	AF49	I		保留引脚	接地
RSV23	AG49	I		保留引脚	接地
RSV24	AH52	I		保留引脚	接地
RSV25	AG52	I		保留引脚	接地
RSV26	AE47	I		保留引脚	接地
RSV27	AG46	I		保留引脚	接地
RSV28	AJ57	I		保留引脚	接地
RSV29	AH55	I		保留引脚	接地
RSV30	AH56	O		保留引脚	浮空
RSV31	AF46	O		保留引脚	浮空
RSV32	AJ56	I		保留引脚	接地
RSV33	AJ49	O		保留引脚	浮空
RSV34	AF53	O		保留引脚	浮空
MCU0_I2C_SCL	AY50			MCU0 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU0_I2C_SDA	AW50			MCU0 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU1_I2C_SCL	A52			MCU1 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU1_I2C_SDA	B52			MCU1 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
UART0_SIN	AV55	I	up	串口 0 数据输入	
UART0_SOUT	AV56	O		串口 0 数据输出	
UART1_SIN	AW55	I	up	串口 1 数据输入	
UART1_SOUT	AW56	O		串口 1 数据输出	
GPIOC0_UART0_CD	AT55	B	up	GPIO C 第 0 位输入输出; 串口 0 Data Carrier 检测 Modem 状态输入信号	
GPIOC1_UART0_DTR	AU57	B	up	GPIO C 第 1 位输入输出; 串口 0 Modem Control 数据终端 ready 输出信号	
GPIOC2_UART0_DSR	AU55	B	up	GPIO C 第 2 位输入输出; 串口 0 Data Set Ready Modem 状态输入信号	
GPIOC3_UART0_RTS	AT56	B	up	GPIO C 第 3 位输入输出; 串口 0 Modem Control 发送请求输出信号	

GPIOC4_UART0_CTS	AU56	B	up	GPIO C 第 4 位输入输出；清除发送 Modem 状态输入信号
GPIOC5_UART0_RI	AV57	B	up	GPIOC5 输入输出；Ring Indicator Modem 状态输入信号
I2C0_SCL	AJ55	B	up	I2C 0 接口的时钟信号
I2C0_SDA	AK57	B	up	I2C 0 接口的数据信号
I2C1_SCL	AK56	B	up	I2C 1 接口的时钟信号
I2C1_SDA	AK55	B	up	I2C 1 接口的数据信号
SPI_EXT_CSN0	AL55	O		SPI 接口 0 号片选
SPI_EXT_CSN1	AL56	O		SPI 接口 1 号片选
SPI_EXT_SCK	AL57	O		SPI 接口时钟信号
SPI_EXT_SO	AM56	O		SPI 接口数据输出信号
EXT_SPI_SI	AM57	I	down	SPI 接口数据输入信号
GPIOC6_LPC_EXT_IRQ_OUTEN	AR56	B	up	GPIO C 第 6 位输入输出；
GPIOC7_LPC_IRQ	AR55	B	up	GPIO C 第 7 位输入输出；
GPIOD0_LPC_EXT_LFRAME	AN56	B	up	GPIO D 第 0 位输入输出；
GPIOD1_EXT_LPC_LRESET	AM55	B	up	GPIO D 第 1 位输入输出；
GPIOD2_EXT_LPC_LCK	AN57	B	up	GPIO D 第 2 位输入输出；
GPIOD3_LPC_EXT_LAD_OUTEN	AN55	B	up	GPIO D 第 3 位输入输出；
GPIOD4_LPC_LAD0	AP57	B	up	GPIO D 第 4 位输入输出；
GPIOD5_LPC_LAD1	AP56	B	up	GPIO D 第 5 位输入输出；
GPIOD6_LPC_LAD2	AP55	B	up	GPIO D 第 6 位输入输出；
GPIOD7_LPC_LAD3	AR57	B	up	GPIO D 第 7 位输入输出；
GPIOB7_EXT_LPC_LDRQ	AT57	B	up	GPIO B 第 7 位输入输出；
GPIOA0_CLK_OBV_SEL0	AE57	B	up	GPIO A 第 0 位输入输出；时钟观测选择信号的输入信号 0
GPIOA1_CLK_OBV_SEL1	AD52	B	up	GPIO A 第 1 位输入输出；时钟观测选择信号的输入信号 1
GPIOA2_CLK_OBV_SEL2	AD56	B	up	GPIO A 第 2 位输入输出；时钟观测选择信号的输入信号 2

GPIOA3_CLK_OBV_SEL3	AC53	B	up	GPIO A 第 3 位输入输出; 时钟观测选择信号的输入信号 3	
GPIOA4_RST_FSM0	AC55	B	up	GPIO A 第 4 位输入输出; 复位状态机输出信号 0	
GPIOA5_RST_FSM1	AC57	B	up	GPIO A 第 5 位输入输出; 复位状态机输出信号 1	
GPIOA6_RST_FSM2	AB52	B	up	GPIO A 第 6 位输入输出; 复位状态机输出信号 2	
GPIOA7_RST_FSM3	AB56	B	up	GPIO A 第 7 位输入输出; 复位状态机输出信号 3	
GPIOB0_SPI_EXT_CS_N2	AE56	B	up	GPIO B 第 0 位输入输出; SPI 接口 2 号片选	
GPIOB1_SPI_EXT_CS_N3	AD53	B	up	GPIO B 第 1 位输入输出; SPI 接口 3 号片选	
GPIOB2_PEU_LINKUP1	AD55	B	up	GPIO B 第 2 位输入输出; PCIE linkup 输出信号, PCIE 连接上后, 输出低电平	
GPIOB3_PEU_LINKUP2	AD57	B	up	GPIO B 第 3 位输入输出; PCIE linkup 输出信号, PCIE 连接上后, 输出低电平	
GPIOB4_PEU_LINKUP3	AC52	B	up	GPIO B 第 4 位输入输出; PCIE linkup 输出信号, PCIE 连接上后, 输出低电平	
GPIOB5	AC56	B	up	GPIO B 第 5 位输入输出;	
GPIOB6	AB53	B	up	GPIO B 第 6 位输入输出;	
GPIOA_DDR_RST_FSM4	AB57	O		GPIO A 数据方向; 复位状态机输出信号 4	
GPIOB_DDR_CRU_LINK_OBV	AB55	O		GPIO B 数据方向; CRU 观测时钟输出信号	
PCIE_LINKUP	AU51			PCIE 控制器 0 的 Linkup 信号, PCIE 连接上后, 输出低电平	
RSV35	AC49			保留引脚	浮空
RSV36	AC50			保留引脚	浮空
RSV37	AD49			保留引脚	浮空
RSV38	AD50			保留引脚	浮空
RSV39	AE50			保留引脚	浮空
RSV40	AB48	I		保留引脚	接地
RSV41	AB50	I		保留引脚	接 VDD_IO
RSV42	AB51	I		保留引脚	接地

RSV43	AD47	I		保留引脚	接地
RSV44	AG50	I		保留引脚	接地
RSV45	AF50	I		保留引脚	接地
RSV46	AJ53	I		保留引脚	接地
RSV47	AJ50	I		保留引脚	接地
RSV48	AF47	I		保留引脚	接地
RSV49	AG47	I		保留引脚	接地
RSV50	AH48	I		保留引脚	接地
RSV51	AE49	I		保留引脚	接地
RSV52	AU54	I		保留引脚	接地
RSV53	AY54	O		保留引脚	浮空
RSV54	AV53	O		保留引脚	浮空
RSV55	AW53	O		保留引脚	浮空
RSV56	AW54	O		保留引脚	浮空
RSV57	AV54	O		保留引脚	浮空

注：

(1)信号类型：I 输入引脚，O 输出引脚，B 双向引脚，Z 三态或高阻引脚，A 模拟信号脚，GND 地，P 电源引脚，RSV 保留引脚，NC 空引脚；

(2)片内上下拉：Up 片内上拉，down 片内下拉。

上述引脚中存在功能复用情况，具体的复用及控制信号如表 8-2。

表 8-2 通用 IO 类引脚的功能复用情况

控制线	功能 0	功能 1
	cru_sel_xxx=0	cru_sel_xxx=1
cru_sel_uart_func	GPIOC0	UART0.CD
	GPIOC1	UART0.DTR
	GPIOC2	UART0.DSR
	GPIOC3	UART0.RTS
	GPIOC4	UART0.CTS
	GPIOC5	UART0.RI
cru_sel_lpc_func0	GPIOC6	LPC.irq_outen
	GPIOC7	LPC.irq_n
	GPIOD0	LPC.lframe_n
	GPIOD2	LPC.lck
	GPIOD3	LPC.lad_outen, 指示 LAD 的方向, 用于控制电平

		转换芯片的转换方向
	GPIOD4	LPC.lad[0]
	GPIOD5	LPC.lad[1]
	GPIOD6	LPC.lad[2]
	GPIOD7	LPC.lad[3]
cru_sel_lpc_func1	GPIOB7	LPC.ldrq_n
cru_sel_rst_state	GPIOA0	clk_obv_sel[0], 选择被观察时钟的选择信号
	GPIOA1	clk_obv_sel[1], 选择被观察时钟的选择信号
	GPIOA2	clk_obv_sel[2], 选择被观察时钟的选择信号
	GPIOA3	clk_obv_sel[3], 选择被观察时钟的选择信号
	GPIOA4	rst_fsm[0], 复位状态机输出, 用于观察复位状态
	GPIOA5	rst_fsm[1], 复位状态机输出, 用于观察复位状态
	GPIOA6	rst_fsm[2], 复位状态机输出, 用于观察复位状态
	GPIOA7	rst_fsm[3], 复位状态机输出, 用于观察复位状态
	GPIOA_DDR, GPIOA 的 bit0 的方向控制脚的输出, 一般用于控制电平转换芯片的 Dir 接口	rst_fsm[4], 复位状态机输出, 用于观察复位状态
	GPIOB_DDR, GPIOB 的 bit0 的方向控制脚的输出, 一般用于控制电平转换芯片的 Dir 接口	cru_clk_obv, 被选择输出的观察时钟
cru_sel_spics	GPIOB0	spi_csn[2], SPI Flash 接口的片选 2
	GPIOB1	spi_csn[3], SPI Flash 接口的片选 3
cru_sel_linkup_1	GPIOB2	peu_linkup[1], PCIE 1 号控制器的 linkup 指示
cru_sel_linkup_2	GPIOB3	peu_linkup[2], PCIE 2 号控制器的 linkup 指示
cru_sel_linkup_3	GPIOB4	peu_linkup[3], PCIE 3 号控制器的 linkup 指示

注: 复位完成后, cru_sel_rst_state 的值为 1, 即选择的是功能 1 而不是功能 0, 如果要使用功能 0, 则需要通过指令将 CRU 部件中的 cru_sel_rst_state 控制位置位为 0。其他复用控制信号 cru_sel_xxx 的复位后默认值则是 0, 即默认选择功能 0。

8.2 GMAC 引脚 (55 PIN)

表 8-3 GMAC 引脚

信号名	引脚编号	信号类型	片内上下拉	功能说明
GMU_CLK_OSC	P54	I		输入 osc 时钟
G0_CLK_GTX	K56	O		GMAC0 向 PHY 输出 TX 时钟

G0_CLK_RX	P56	I		GMAC0 输入 RX 时钟
G0_CLK_TX	P57	I		GMAC0 输入 TX 时钟
G0_COL	N53	I		GMAC0 PHY 冲突检测信号
G0_CRS	P53	I		GMAC0 PHY 发送接收非 IDLE 状态
G0_MDC	P52	O		GMAC0 管理接口时钟
G0_MDIO	N54	B	down	GMAC0 管理接口输入输出数据
G0_RX0	N56	I		GMAC0 数据输入 bit0
G0_RX1	N55	I		GMAC0 数据输入 bit1
G0_RX2	M57	I		GMAC0 数据输入 bit2
G0_RX3	M56	I		GMAC0 数据输入 bit3
G0_RX4	M55	I		GMAC0 数据输入 bit4
G0_RX5	L57	I		GMAC0 数据输入 bit5
G0_RX6	L56	I		GMAC0 数据输入 bit6
G0_RX7	L55	I		GMAC0 数据输入 bit7
G0_RXDV	N57	I		GMAC0 数据输入有效
G0_RXER	P55	I		GMAC0 接收 Error
G0_TX0	G56	O		GMAC0 数据输出 bit
G0_TX1	G57	O		GMAC0 数据输出 bit
G0_TX2	H55	O		GMAC0 数据输出 bit
G0_TX3	H56	O		GMAC0 数据输出 bit
G0_TX4	H57	O		GMAC0 数据输出 bit
G0_TX5	J55	O		GMAC0 数据输出 bit
G0_TX6	J56	O		GMAC0 数据输出 bit
G0_TX7	J57	O		GMAC0 数据输出 bit
G0_TXEN	K57	O		GMAC0 数据输出有效
G0_TXER	K55	O		GMAC0 发送 Error
G1_CLK_GTX	B56	O		GMAC0 向 PHY 输出 TX 时钟
G1_CLK_RX	F57	I		GMAC0 输入 RX 时钟
G1_CLK_TX	G55	I		GMAC0 输入 TX 时钟
G1_COL	N52	I		GMAC0 PHY 冲突检测信号
G1_CRS	M52	I		GMAC0 PHY 发送接收非 IDLE 状态
G1_MDC	M53	O		GMAC0 管理接口时钟
G1_MDIO	M54	B	down	GMAC0 管理接口输入输出数据
G1_RX0	E57	I		GMAC0 数据输入 bit0
G1_RX1	E56	I		GMAC0 数据输入 bit1
G1_RX2	E55	I		GMAC0 数据输入 bit2

G1_RX3	D57	I		GMAC0 数据输入 bit3
G1_RX4	D56	I		GMAC0 数据输入 bit4
G1_RX5	D55	I		GMAC0 数据输入 bit5
G1_RX6	C57	I		GMAC0 数据输入 bit6
G1_RX7	C56	I		GMAC0 数据输入 bit7
G1_RXDV	F55	I		GMAC0 数据输入有效
G1_RXER	F56	I		GMAC0 接收 Error
G1_TX0	C52	O		GMAC0 数据输出 bit
G1_TX1	A53	O		GMAC0 数据输出 bit
G1_TX2	B53	O		GMAC0 数据输出 bit
G1_TX3	C53	O		GMAC0 数据输出 bit
G1_TX4	A54	O		GMAC0 数据输出 bit
G1_TX5	B54	O		GMAC0 数据输出 bit
G1_TX6	C54	O		GMAC0 数据输出 bit
G1_TX7	A55	O		GMAC0 数据输出 bit
G1_TXEN	C55	O		GMAC0 数据输出有效
G1_TXER	B55	O		GMAC0 发送 Error

注：

(1)信号类型：I 输入引脚，O 输出引脚，Z 三态或高阻引脚，A 模拟信号脚，GND 地，P 电源引脚，RSV 保留引脚，NC 空引脚；

(2)片内上下拉：Up 片内上拉，Down 片内下拉，空白为既无上拉也无下拉。

8.3 PCIE 引脚（164 PIN）

表 8-4 PCIE 引脚

信号名	引脚编号	信号类型	片内上下拉	功能说明
PCIE0_CLKN	D2	I		100MHz 差分时钟输入
PCIE0_CLKP	D1	I		
PCIE0_AMON0	K9	A		调试观察接口，正常模式浮空处理
PCIE0_AMON1	K10	A		
PCIE0_AMON2	E8	A		
PCIE0_AMON3	D10	A		
PCIE0_DMON0	Y9	A		
PCIE0_DMON1	V10	A		
PCIE0_DMON2	T10	A		
PCIE0_DMON3	P10	A		

PCIE0_DMON B0	Y10	A		
PCIE0_DMON B1	V9	A		
PCIE0_DMON B2	T9	A		
PCIE0_DMON B3	P9	A		
PCIE0_RBIAS0	M9	A		参考电阻偏置, 1K 欧姆 1%精度电阻与地连接
PCIE0_RBIAS1	G10	A		
PCIE0_RBIAS2	E7	A		
PCIE0_RBIAS3	D9	A		
PCIE0_RX00N	AA3	I		输入链路差分信号
PCIE0_RX00P	AA4	I		
PCIE0_RX01N	Y1	I		
PCIE0_RX01P	Y2	I		
PCIE0_RX02N	W3	I		
PCIE0_RX02P	W4	I		
PCIE0_RX03N	V1	I		
PCIE0_RX03P	V2	I		
PCIE0_RX04N	U3	I		
PCIE0_RX04P	U4	I		
PCIE0_RX05N	T1	I		
PCIE0_RX05P	T2	I		
PCIE0_RX06N	R3	I		
PCIE0_RX06P	R4	I		
PCIE0_RX07N	P1	I		
PCIE0_RX07P	P2	I		
PCIE0_RX08N	N3	I		
PCIE0_RX08P	N4	I		
PCIE0_RX09N	M1	I		
PCIE0_RX09P	M2	I		
PCIE0_RX10N	L3	I		
PCIE0_RX10P	L4	I		
PCIE0_RX11N	K1	I		
PCIE0_RX11P	K2	I		

PCIE0_RX12N	J3	I	
PCIE0_RX12P	J4	I	
PCIE0_RX13N	H1	I	
PCIE0_RX13P	H2	I	
PCIE0_RX14N	G3	I	
PCIE0_RX14P	G4	I	
PCIE0_RX15N	F1	I	
PCIE0_RX15P	F2	I	
PCIE0_TX00N	AA7	O	
PCIE0_TX00P	AA8	O	
PCIE0_TX01N	Y5	O	
PCIE0_TX01P	Y6	O	
PCIE0_TX02N	W7	O	
PCIE0_TX02P	W8	O	
PCIE0_TX03N	V5	O	
PCIE0_TX03P	V6	O	
PCIE0_TX04N	U7	O	
PCIE0_TX04P	U8	O	
PCIE0_TX05N	T5	O	
PCIE0_TX05P	T6	O	
PCIE0_TX06N	R7	O	
PCIE0_TX06P	R8	O	
PCIE0_TX07N	P5	O	
PCIE0_TX07P	P6	O	
PCIE0_TX08N	N7	O	
PCIE0_TX08P	N8	O	
PCIE0_TX09N	M5	O	
PCIE0_TX09P	M6	O	
PCIE0_TX10N	L7	O	
PCIE0_TX10P	L8	O	
PCIE0_TX11N	K5	O	
PCIE0_TX11P	K6	O	
PCIE0_TX12N	J7	O	
PCIE0_TX12P	J8	O	
PCIE0_TX13N	H5	O	
PCIE0_TX13P	H6	O	

输出链路差分信号

PCIE0_TX14N	G7	O		
PCIE0_TX14P	G8	O		
PCIE0_TX15N	F5	O		
PCIE0_TX15P	F6	O		
PCIE1_CLKN	E4	I		100MHz 差分时钟输入
PCIE1_CLKP	E3	I		
PCIE1_AMON0	H9	A		调试观察接口,正常运行模式下浮空处理即可
PCIE1_AMON1	D6	A		
PCIE1_AMON2	AL10	A		
PCIE1_AMON3	AP9	A		
PCIE1_DMON0	AB10	A		
PCIE1_DMON1	AD10	A		
PCIE1_DMON2	AF10	A		
PCIE1_DMON3	AH10	A		
PCIE1_DMON B0	AB9	A		
PCIE1_DMON B1	AD9	A		
PCIE1_DMON B2	AF9	A		
PCIE1_DMON B3	AH9	A		
PCIE1_RBIA0	F9	A		参考电阻偏置,1K 欧姆 1%精度电阻与地连接
PCIE1_RBIA1	D5	A		
PCIE1_RBIA2	AK9	A		
PCIE1_RBIA3	AM9	A		
PCIE1_RX00N	AB1	I		输入链路差分信号
PCIE1_RX00P	AB2	I		
PCIE1_RX01N	AC3	I		
PCIE1_RX01P	AC4	I		
PCIE1_RX02N	AD1	I		
PCIE1_RX02P	AD2	I		
PCIE1_RX03N	AE3	I		
PCIE1_RX03P	AE4	I		
PCIE1_RX04N	AF1	I		
PCIE1_RX04P	AF2	I		

PCIE1_RX05N	AG3	I	
PCIE1_RX05P	AG4	I	
PCIE1_RX06N	AH1	I	
PCIE1_RX06P	AH2	I	
PCIE1_RX07N	AJ3	I	
PCIE1_RX07P	AJ4	I	
PCIE1_RX08N	AK1	I	
PCIE1_RX08P	AK2	I	
PCIE1_RX09N	AL3	I	
PCIE1_RX09P	AL4	I	
PCIE1_RX10N	AM1	I	
PCIE1_RX10P	AM2	I	
PCIE1_RX11N	AN3	I	
PCIE1_RX11P	AN4	I	
PCIE1_RX12N	AP1	I	
PCIE1_RX12P	AP2	I	
PCIE1_RX13N	AR3	I	
PCIE1_RX13P	AR4	I	
PCIE1_RX14N	AT1	I	
PCIE1_RX14P	AT2	I	
PCIE1_RX15N	AU3	I	
PCIE1_RX15P	AU4	I	
PCIE1_TX00N	AB5	O	
PCIE1_TX00P	AB6	O	
PCIE1_TX01N	AC7	O	
PCIE1_TX01P	AC8	O	
PCIE1_TX02N	AD5	O	
PCIE1_TX02P	AD6	O	
PCIE1_TX03N	AE7	O	
PCIE1_TX03P	AE8	O	
PCIE1_TX04N	AF5	O	
PCIE1_TX04P	AF6	O	
PCIE1_TX05N	AG7	O	
PCIE1_TX05P	AG8	O	
PCIE1_TX06N	AH5	O	
PCIE1_TX06P	AH6	O	

输出链路差分信号

PCIE1_TX07N	AJ7	O	
PCIE1_TX07P	AJ8	O	
PCIE1_TX08N	AK5	O	
PCIE1_TX08P	AK6	O	
PCIE1_TX09N	AL7	O	
PCIE1_TX09P	AL8	O	
PCIE1_TX10N	AM5	O	
PCIE1_TX10P	AM6	O	
PCIE1_TX11N	AN7	O	
PCIE1_TX11P	AN8	O	
PCIE1_TX12N	AP5	O	
PCIE1_TX12P	AP6	O	
PCIE1_TX13N	AR7	O	
PCIE1_TX13P	AR8	O	
PCIE1_TX14N	AT5	O	
PCIE1_TX14P	AT6	O	
PCIE1_TX15N	AU7	O	
PCIE1_TX15P	AU8	O	

注：

(1)信号类型：I 输入引脚，O 输出引脚，Z 三态或高阻引脚，A 模拟信号脚，GND 地，P 电源引脚，RSV 保留引脚，NC 空引脚；

(2)片内上下拉：Up 片内上拉，Down 片内下拉，空白为既无上拉也无下拉。

8.4 DDR3 引脚 (624 PIN)

表 8-5 DDR3 存储控制器引脚

信号名	引脚编号	信号类型	片内上下拉	功能说明
M0_A00	AY30	O		通道 0 存储器地址, bit 0
M0_A01	AY27	O		通道 0 存储器地址, bit 1
M0_A02	AW27	O		通道 0 存储器地址, bit 2
M0_A03	AR28	O		通道 0 存储器地址, bit 3
M0_A04	AT28	O		通道 0 存储器地址, bit 4
M0_A05	AR27	O		通道 0 存储器地址, bit 5
M0_A06	AT27	O		通道 0 存储器地址, bit 6
M0_A07	AW26	O		通道 0 存储器地址, bit 7
M0_A08	AV26	O		通道 0 存储器地址, bit 8

M0_A09	AR26	O		通道 0 存储器地址, bit 9
M0_A10	AW31	O		通道 0 存储器地址, bit 10
M0_A11	AT26	O		通道 0 存储器地址, bit 11
M0_A12	AV25	O		通道 0 存储器地址, bit 12
M0_A13	AT34	O		通道 0 存储器地址, bit 13
M0_A14	AR25	O		通道 0 存储器地址, bit 14
M0_A15	AW24	O		通道 0 存储器地址, bit 15
M0_ATB0	AT29	I		通道 0 存储器 ATB0/浮空
M0_ATB1	AR29	I		通道 0 存储器 ATB1/浮空
M0_BA0	AT32	O		通道 0 存储器地址, bit 0
M0_BA1	AV31	O		通道 0 存储器地址, bit 1
M0_BA2	AT25	O		通道 0 存储器地址, bit 2
M0_CAL_BLK	AV32	I		通道 0 存储器 CAL_BLK
M0_CAS	AR33	O		通道 0 存储器列地址选通
M0_CKE0	AY24	O		通道 0 存储器时钟使能, cs 0
M0_CKE1	AT24	O		通道 0 存储器时钟使能, cs 1
M0_CKE2	AR24	O		通道 0 存储器时钟使能, cs 2
M0_CKE3	AV23	O		通道 0 存储器时钟使能, cs 3
M0_CLK0N	AR30	O		通道 0 存储器时钟-, cs0
M0_CLK0P	AT30	O		通道 0 存储器时钟+, cs0
M0_CLK1N	AV28	O		通道 0 存储器时钟-, cs1
M0_CLK1P	AW28	O		通道 0 存储器时钟+, cs1
M0_CLK2N	AR31	O		通道 0 存储器时钟-, cs2
M0_CLK2P	AT31	O		通道 0 存储器时钟+, cs2
M0_CLK3N	AV29	O		通道 0 存储器时钟-, cs3
M0_CLK3P	AW29	O		通道 0 存储器时钟+, cs3
M0_CS0	AW32	O		通道 0 存储器片选, cs0
M0_CS1	AR34	O		通道 0 存储器片选, cs1
M0_CS2	AR35	O		通道 0 存储器片选, cs2
M0_CS3	AT35	O		通道 0 存储器片选, cs3
M0_ODT0	AW33	O		通道 0 存储器 ODT, cs0
M0_ODT1	AW34	O		通道 0 存储器 ODT, cs1
M0_ODT2	AY33	O		通道 0 存储器 ODT, cs2
M0_ODT3	AV34	O		通道 0 存储器 ODT, cs3
M0_PAR_ERR	AW25	I		通道 0 存储器的奇偶校验结果
M0_PAR_IN	AW30	O		通道 0 存储器的奇偶校验值

M0_RAS	AR32	O		通道 0 存储器行地址选通
M0_RESET	AW23	O		通道 0 存储器复位
M0_WE	AT33	O		通道 0 存储器写使能
M0_S0_DMN	AV38	I/O		通道 0 存储器数据屏蔽-, slice0
M0_S0_DMP	AW38	I/O		通道 0 存储器数据屏蔽+, slice0
M0_S0_DQ00	AW37	I/O		通道 0 存储器数据, bit0
M0_S0_DQ01	AW41	I/O		通道 0 存储器数据, bit1
M0_S0_DQ02	AW40	I/O		通道 0 存储器数据, bit2
M0_S0_DQ03	AV41	I/O		通道 0 存储器数据, bit3
M0_S0_DQ04	AV40	I/O		通道 0 存储器数据, bit4
M0_S0_DQ05	AV37	I/O		通道 0 存储器数据, bit5
M0_S0_DQ06	AY36	I/O		通道 0 存储器数据, bit6
M0_S0_DQ07	AW36	I/O		通道 0 存储器数据, bit7
M0_S0_DQSN	AY39	I/O		通道 0 存储器数据选通-, slice0
M0_S0_DQSP	AW39	I/O		通道 0 存储器数据选通+, slice0
M0_S1_DMN	AR39	I/O		通道 0 存储器数据屏蔽-, slice1
M0_S1_DMP	AT39	I/O		通道 0 存储器数据屏蔽+, slice1
M0_S1_DQ08	AT37	I/O		通道 0 存储器数据, bit08
M0_S1_DQ09	AR42	I/O		通道 0 存储器数据, bit09
M0_S1_DQ10	AT41	I/O		通道 0 存储器数据, bit10
M0_S1_DQ11	AR41	I/O		通道 0 存储器数据, bit11
M0_S1_DQ12	AR38	I/O		通道 0 存储器数据, bit12
M0_S1_DQ13	AT38	I/O		通道 0 存储器数据, bit13
M0_S1_DQ14	AR37	I/O		通道 0 存储器数据, bit14
M0_S1_DQ15	AT42	I/O		通道 0 存储器数据, bit15
M0_S1_DQSN	AT40	I/O		通道 0 存储器数据选通-, slice1
M0_S1_DQSP	AR40	I/O		通道 0 存储器数据选通+, slice1
M0_S2_DMN	AW18	I/O		通道 0 存储器数据屏蔽-, slice2
M0_S2_DMP	AY18	I/O		通道 0 存储器数据屏蔽+, slice2
M0_S2_DQ16	AW17	I/O		通道 0 存储器数据, bit16
M0_S2_DQ17	AW16	I/O		通道 0 存储器数据, bit17
M0_S2_DQ18	AV16	I/O		通道 0 存储器数据, bit18
M0_S2_DQ19	AV17	I/O		通道 0 存储器数据, bit19
M0_S2_DQ20	AY21	I/O		通道 0 存储器数据, bit20
M0_S2_DQ21	AW21	I/O		通道 0 存储器数据, bit21
M0_S2_DQ22	AW20	I/O		通道 0 存储器数据, bit22

M0_S2_DQ23	AV20	I/O		通道 0 存储器数据, bit23
M0_S2_DQSN	AW19	I/O		通道 0 存储器数据选通-, slice2
M0_S2_DQSP	AV19	I/O		通道 0 存储器数据选通+, slice2
M0_S3_DMN	AV11	I/O		通道 0 存储器数据屏蔽-, slice3
M0_S3_DMP	AW11	I/O		通道 0 存储器数据屏蔽+, slice3
M0_S3_DQ24	AW10	I/O		通道 0 存储器数据, bit24
M0_S3_DQ25	AW14	I/O		通道 0 存储器数据, bit25
M0_S3_DQ26	AW13	I/O		通道 0 存储器数据, bit26
M0_S3_DQ27	AV14	I/O		通道 0 存储器数据, bit27
M0_S3_DQ28	AV13	I/O		通道 0 存储器数据, bit28
M0_S3_DQ29	AV10	I/O		通道 0 存储器数据, bit29
M0_S3_DQ30	AY9	I/O		通道 0 存储器数据, bit30
M0_S3_DQ31	AW9	I/O		通道 0 存储器数据, bit31
M0_S3_DQSN	AY12	I/O		通道 0 存储器数据选通-, slice3
M0_S3_DQSP	AW12	I/O		通道 0 存储器数据选通+, slice3
M0_S4_DMN	AW4	I/O		通道 0 存储器数据屏蔽-, slice4
M0_S4_DMP	AY4	I/O		通道 0 存储器数据屏蔽+, slice4
M0_S4_DQ32	AY3	I/O		通道 0 存储器数据, bit32
M0_S4_DQ33	AW7	I/O		通道 0 存储器数据, bit33
M0_S4_DQ34	AY6	I/O		通道 0 存储器数据, bit34
M0_S4_DQ35	AW6	I/O		通道 0 存储器数据, bit35
M0_S4_DQ36	AY7	I/O		通道 0 存储器数据, bit36
M0_S4_DQ37	AW3	I/O		通道 0 存储器数据, bit37
M0_S4_DQ38	AW2	I/O		通道 0 存储器数据, bit38
M0_S4_DQ39	AV2	I/O		通道 0 存储器数据, bit39
M0_S4_DQSN	AY5	I/O		通道 0 存储器数据选通-, slice4
M0_S4_DQSP	AW5	I/O		通道 0 存储器数据选通+, slice4
M0_S5_DMN	AR19	I/O		通道 0 存储器数据屏蔽-, slice5
M0_S5_DMP	AT19	I/O		通道 0 存储器数据屏蔽+, slice5
M0_S5_DQ40	AT17	I/O		通道 0 存储器数据, bit40
M0_S5_DQ41	AT22	I/O		通道 0 存储器数据, bit41
M0_S5_DQ42	AT21	I/O		通道 0 存储器数据, bit42
M0_S5_DQ43	AR22	I/O		通道 0 存储器数据, bit43
M0_S5_DQ44	AR21	I/O		通道 0 存储器数据, bit44
M0_S5_DQ45	AT18	I/O		通道 0 存储器数据, bit45
M0_S5_DQ46	AR18	I/O		通道 0 存储器数据, bit46

M0_S5_DQ47	AR17	I/O		通道 0 存储器数据, bit47
M0_S5_DQSN	AT20	I/O		通道 0 存储器数据选通-, slice5
M0_S5_DQSP	AR20	I/O		通道 0 存储器数据选通+, slice5
M0_S6_DMN	AR12	I/O		通道 0 存储器数据屏蔽-, slice6
M0_S6_DMP	AT12	I/O		通道 0 存储器数据屏蔽+, slice6
M0_S6_DQ48	AT10	I/O		通道 0 存储器数据, bit48
M0_S6_DQ49	AT15	I/O		通道 0 存储器数据, bit49
M0_S6_DQ50	AT14	I/O		通道 0 存储器数据, bit50
M0_S6_DQ51	AR15	I/O		通道 0 存储器数据, bit51
M0_S6_DQ52	AR14	I/O		通道 0 存储器数据, bit52
M0_S6_DQ53	AT11	I/O		通道 0 存储器数据, bit53
M0_S6_DQ54	AR11	I/O		通道 0 存储器数据, bit54
M0_S6_DQ55	AR10	I/O		通道 0 存储器数据, bit55
M0_S6_DQSN	AT13	I/O		通道 0 存储器数据选通-, slice6
M0_S6_DQSP	AR13	I/O		通道 0 存储器数据选通+, slice6
M0_S7_DMN	AW45	I/O		通道 0 存储器数据屏蔽-, slice7
M0_S7_DMP	AY45	I/O		通道 0 存储器数据屏蔽+, slice7
M0_S7_DQ56	AW44	I/O		通道 0 存储器数据, bit56
M0_S7_DQ57	AW48	I/O		通道 0 存储器数据, bit57
M0_S7_DQ58	AW47	I/O		通道 0 存储器数据, bit58
M0_S7_DQ59	AV47	I/O		通道 0 存储器数据, bit59
M0_S7_DQ60	AV44	I/O		通道 0 存储器数据, bit60
M0_S7_DQ61	AY48	I/O		通道 0 存储器数据, bit61
M0_S7_DQ62	AV43	I/O		通道 0 存储器数据, bit62
M0_S7_DQ63	AW43	I/O		通道 0 存储器数据, bit63
M0_S7_DQSN	AW46	I/O		通道 0 存储器数据选通-, slice7
M0_S7_DQSP	AV46	I/O		通道 0 存储器数据选通+, slice7
M0_S8_DMN	AR46	I/O		通道 0 存储器数据屏蔽-, slice8
M0_S8_DMP	AT46	I/O		通道 0 存储器数据屏蔽+, slice8
M0_S8_DQ64	AT44	I/O		通道 0 存储器数据, bit64
M0_S8_DQ65	AV49	I/O		通道 0 存储器数据, bit65
M0_S8_DQ66	AU49	I/O		通道 0 存储器数据, bit66
M0_S8_DQ67	AU48	I/O		通道 0 存储器数据, bit67
M0_S8_DQ68	AT48	I/O		通道 0 存储器数据, bit68
M0_S8_DQ69	AR45	I/O		通道 0 存储器数据, bit69
M0_S8_DQ70	AT45	I/O		通道 0 存储器数据, bit70

M0_S8_DQ71	AR44	I/O		通道 0 存储器数据, bit71
M0_S8_DQSN	AT47	I/O		通道 0 存储器数据选通-, slice8
M0_S8_DQSP	AR47	I/O		通道 0 存储器数据选通+, slice8
M1_A00	AN29	O		通道 1 存储器地址, bit 0
M1_A01	AK32	O		通道 1 存储器地址, bit 1
M1_A02	AJ32	O		通道 1 存储器地址, bit 2
M1_A03	AM32	O		通道 1 存储器地址, bit 3
M1_A04	AN32	O		通道 1 存储器地址, bit 4
M1_A05	AN33	O		通道 1 存储器地址, bit 5
M1_A06	AM33	O		通道 1 存储器地址, bit 6
M1_A07	AK33	O		通道 1 存储器地址, bit 7
M1_A08	AJ33	O		通道 1 存储器地址, bit 8
M1_A09	AM34	O		通道 1 存储器地址, bit 9
M1_A10	AK28	O		通道 1 存储器地址, bit 10
M1_A11	AN34	O		通道 1 存储器地址, bit 11
M1_A12	AJ34	O		通道 1 存储器地址, bit 12
M1_A13	AM26	O		通道 1 存储器地址, bit 13
M1_A14	AM35	O		通道 1 存储器地址, bit 14
M1_A15	AJ35	O		通道 1 存储器地址, bit 15
M1_ATB0	AK31	I		通道 1 存储器 ATB0
M1_ATB1	AJ31	I		通道 1 存储器 ATB1
M1_BA0	AN28	O		通道 1 存储器地址, bit 0
M1_BA1	AJ28	O		通道 1 存储器地址, bit 1
M1_BA2	AN35	O		通道 1 存储器地址, bit 2
M1_CAL_BLK	AK27	I		通道 1 存储器 CAL_BLK
M1_CAS	AM27	O		通道 1 存储器列地址选通
M1_CKE0	AK35	O		通道 1 存储器时钟使能, cs 0
M1_CKE1	AM36	O		通道 1 存储器时钟使能, cs 1
M1_CKE2	AN36	O		通道 1 存储器时钟使能, cs 2
M1_CKE3	AJ36	O		通道 1 存储器时钟使能, cs 3
M1_CLK0N	AM31	O		通道 1 存储器时钟-, cs0
M1_CLK0P	AN31	O		通道 1 存储器时钟+, cs0
M1_CLK1N	AJ30	O		通道 1 存储器时钟-, cs1
M1_CLK1P	AK30	O		通道 1 存储器时钟+, cs1
M1_CLK2N	AM30	O		通道 1 存储器时钟-, cs2
M1_CLK2P	AN30	O		通道 1 存储器时钟+, cs2

M1_CLK3N	AJ29	O		通道 1 存储器时钟-, cs3
M1_CLK3P	AK29	O		通道 1 存储器时钟+, cs3
M1_CS0	AJ27	O		通道 1 存储器片选, cs0
M1_CS1	AN26	O		通道 1 存储器片选, cs1
M1_CS2	AM25	O		通道 1 存储器片选, cs2
M1_CS3	AN25	O		通道 1 存储器片选, cs3
M1_ODT0	AK26	O		通道 1 存储器 ODT, cs0
M1_ODT1	AJ25	O		通道 1 存储器 ODT, cs1
M1_ODT2	AJ26	O		通道 1 存储器 ODT, cs2
M1_ODT3	AK25	O		通道 1 存储器 ODT, cs3
M1_PAR_ERR	AK34	I		通道 1 存储器的奇偶校验结果
M1_PAR_IN	AM29	O		通道 1 存储器的奇偶校验值
M1_RAS	AM28	O		通道 1 存储器行地址选通
M1_RESET	AK36	O		通道 1 存储器复位
M1_WE	AN27	O		通道 1 存储器写使能
M1_S0_DMN	AJ14	I/O		通道 1 存储器数据屏蔽-, slice0
M1_S0_DMP	AK14	I/O		通道 1 存储器数据屏蔽+, slice0
M1_S0_DQ00	AJ15	I/O		通道 1 存储器数据, bit0
M1_S0_DQ01	AJ16	I/O		通道 1 存储器数据, bit1
M1_S0_DQ02	AJ11	I/O		通道 1 存储器数据, bit2
M1_S0_DQ03	AJ12	I/O		通道 1 存储器数据, bit3
M1_S0_DQ04	AK16	I/O		通道 1 存储器数据, bit4
M1_S0_DQ05	AK15	I/O		通道 1 存储器数据, bit5
M1_S0_DQ06	AK12	I/O		通道 1 存储器数据, bit6
M1_S0_DQ07	AK11	I/O		通道 1 存储器数据, bit7
M1_S0_DQSN	AK13	I/O		通道 1 存储器数据选通-, slice0
M1_S0_DQSP	AJ13	I/O		通道 1 存储器数据选通+, slice0
M1_S1_DMN	AM14	I/O		通道 1 存储器数据屏蔽-, slice1
M1_S1_DMP	AN14	I/O		通道 1 存储器数据屏蔽+, slice1
M1_S1_DQ08	AN15	I/O		通道 1 存储器数据, bit08
M1_S1_DQ09	AM11	I/O		通道 1 存储器数据, bit09
M1_S1_DQ10	AM12	I/O		通道 1 存储器数据, bit10
M1_S1_DQ11	AN12	I/O		通道 1 存储器数据, bit11
M1_S1_DQ12	AM16	I/O		通道 1 存储器数据, bit12
M1_S1_DQ13	AM15	I/O		通道 1 存储器数据, bit13
M1_S1_DQ14	AN11	I/O		通道 1 存储器数据, bit14

M1_S1_DQ15	AN16	I/O		通道 1 存储器数据, bit15
M1_S1_DQSN	AN13	I/O		通道 1 存储器数据选通-, slice1
M1_S1_DQSP	AM13	I/O		通道 1 存储器数据选通+, slice1
M1_S2_DMN	AN52	I/O		通道 1 存储器数据屏蔽-, slice2
M1_S2_DMP	AN53	I/O		通道 1 存储器数据屏蔽+, slice2
M1_S2_DQ16	AM53	I/O		通道 1 存储器数据, bit16
M1_S2_DQ17	AL53	I/O		通道 1 存储器数据, bit17
M1_S2_DQ18	AR53	I/O		通道 1 存储器数据, bit18
M1_S2_DQ19	AT53	I/O		通道 1 存储器数据, bit19
M1_S2_DQ20	AT52	I/O		通道 1 存储器数据, bit20
M1_S2_DQ21	AR52	I/O		通道 1 存储器数据, bit21
M1_S2_DQ22	AM52	I/O		通道 1 存储器数据, bit22
M1_S2_DQ23	AL52	I/O		通道 1 存储器数据, bit23
M1_S2_DQSN	AP53	I/O		通道 1 存储器数据选通-, slice2
M1_S2_DQSP	AP52	I/O		通道 1 存储器数据选通+, slice2
M1_S3_DMN	AN48	I/O		通道 1 存储器数据屏蔽-, slice3
M1_S3_DMP	AP48	I/O		通道 1 存储器数据屏蔽+, slice3
M1_S3_DQ24	AP49	I/O		通道 1 存储器数据, bit24
M1_S3_DQ25	AR49	I/O		通道 1 存储器数据, bit25
M1_S3_DQ26	AR50	I/O		通道 1 存储器数据, bit26
M1_S3_DQ27	AP50	I/O		通道 1 存储器数据, bit27
M1_S3_DQ28	AN45	I/O		通道 1 存储器数据, bit28
M1_S3_DQ29	AN46	I/O		通道 1 存储器数据, bit29
M1_S3_DQ30	AM45	I/O		通道 1 存储器数据, bit30
M1_S3_DQ31	AM46	I/O		通道 1 存储器数据, bit31
M1_S3_DQSN	AN47	I/O		通道 1 存储器数据选通-, slice3
M1_S3_DQSP	AM47	I/O		通道 1 存储器数据选通+, slice3
M1_S4_DMN	AK48	I/O		通道 1 存储器数据屏蔽-, slice4
M1_S4_DMP	AL48	I/O		通道 1 存储器数据屏蔽+, slice4
M1_S4_DQ32	AL50	I/O		通道 1 存储器数据, bit32
M1_S4_DQ33	AM49	I/O		通道 1 存储器数据, bit33
M1_S4_DQ34	AM50	I/O		通道 1 存储器数据, bit34
M1_S4_DQ35	AL49	I/O		通道 1 存储器数据, bit35
M1_S4_DQ36	AK46	I/O		通道 1 存储器数据, bit36
M1_S4_DQ37	AK45	I/O		通道 1 存储器数据, bit37
M1_S4_DQ38	AJ46	I/O		通道 1 存储器数据, bit38

M1_S4_DQ39	AJ45	I/O		通道 1 存储器数据, bit39
M1_S4_DQSN	AK47	I/O		通道 1 存储器数据选通-, slice4
M1_S4_DQSP	AJ47	I/O		通道 1 存储器数据选通+, slice4
M1_S5_DMN	AM41	I/O		通道 1 存储器数据屏蔽-, slice5
M1_S5_DMP	AN41	I/O		通道 1 存储器数据屏蔽+, slice5
M1_S5_DQ40	AM42	I/O		通道 1 存储器数据, bit40
M1_S5_DQ41	AN42	I/O		通道 1 存储器数据, bit41
M1_S5_DQ42	AN38	I/O		通道 1 存储器数据, bit42
M1_S5_DQ43	AN39	I/O		通道 1 存储器数据, bit43
M1_S5_DQ44	AM38	I/O		通道 1 存储器数据, bit44
M1_S5_DQ45	AM39	I/O		通道 1 存储器数据, bit45
M1_S5_DQ46	AN43	I/O		通道 1 存储器数据, bit46
M1_S5_DQ47	AM43	I/O		通道 1 存储器数据, bit47
M1_S5_DQSN	AN40	I/O		通道 1 存储器数据选通-, slice5
M1_S5_DQSP	AM40	I/O		通道 1 存储器数据选通+, slice5
M1_S6_DMN	AJ41	I/O		通道 1 存储器数据屏蔽-, slice6
M1_S6_DMP	AK41	I/O		通道 1 存储器数据屏蔽+, slice6
M1_S6_DQ48	AK42	I/O		通道 1 存储器数据, bit48
M1_S6_DQ49	AK43	I/O		通道 1 存储器数据, bit49
M1_S6_DQ50	AJ43	I/O		通道 1 存储器数据, bit50
M1_S6_DQ51	AJ42	I/O		通道 1 存储器数据, bit51
M1_S6_DQ52	AK39	I/O		通道 1 存储器数据, bit52
M1_S6_DQ53	AK38	I/O		通道 1 存储器数据, bit53
M1_S6_DQ54	AJ39	I/O		通道 1 存储器数据, bit54
M1_S6_DQ55	AJ38	I/O		通道 1 存储器数据, bit55
M1_S6_DQSN	AK40	I/O		通道 1 存储器数据选通-, slice6
M1_S6_DQSP	AJ40	I/O		通道 1 存储器数据选通+, slice6
M1_S7_DMN	AJ21	I/O		通道 1 存储器数据屏蔽-, slice7
M1_S7_DMP	AK21	I/O		通道 1 存储器数据屏蔽+, slice7
M1_S7_DQ56	AJ22	I/O		通道 1 存储器数据, bit56
M1_S7_DQ57	AK19	I/O		通道 1 存储器数据, bit57
M1_S7_DQ58	AK22	I/O		通道 1 存储器数据, bit58
M1_S7_DQ59	AK18	I/O		通道 1 存储器数据, bit59
M1_S7_DQ60	AJ18	I/O		通道 1 存储器数据, bit60
M1_S7_DQ61	AJ19	I/O		通道 1 存储器数据, bit61
M1_S7_DQ62	AK23	I/O		通道 1 存储器数据, bit62

M1_S7_DQ63	AJ23	I/O		通道 1 存储器数据, bit63
M1_S7_DQSN	AK20	I/O		通道 1 存储器数据选通-, slice7
M1_S7_DQSP	AJ20	I/O		通道 1 存储器数据选通+, slice7
M1_S8_DMN	AM21	I/O		通道 1 存储器数据屏蔽-, slice8
M1_S8_DMP	AN21	I/O		通道 1 存储器数据屏蔽+, slice8
M1_S8_DQ64	AN22	I/O		通道 1 存储器数据, bit64
M1_S8_DQ65	AN23	I/O		通道 1 存储器数据, bit65
M1_S8_DQ66	AM23	I/O		通道 1 存储器数据, bit66
M1_S8_DQ67	AM22	I/O		通道 1 存储器数据, bit67
M1_S8_DQ68	AN19	I/O		通道 1 存储器数据, bit68
M1_S8_DQ69	AM19	I/O		通道 1 存储器数据, bit69
M1_S8_DQ70	AN18	I/O		通道 1 存储器数据, bit70
M1_S8_DQ71	AM18	I/O		通道 1 存储器数据, bit71
M1_S8_DQSN	AN20	I/O		通道 1 存储器数据选通-, slice8
M1_S8_DQSP	AM20	I/O		通道 1 存储器数据选通+, slice8
M2_A00	B32	O		通道 2 存储器地址, bit 0
M2_A01	B29	O		通道 2 存储器地址, bit 1
M2_A02	C29	O		通道 2 存储器地址, bit 2
M2_A03	F29	O		通道 2 存储器地址, bit 3
M2_A04	E29	O		通道 2 存储器地址, bit 4
M2_A05	F28	O		通道 2 存储器地址, bit 5
M2_A06	E28	O		通道 2 存储器地址, bit 6
M2_A07	A28	O		通道 2 存储器地址, bit 7
M2_A08	B28	O		通道 2 存储器地址, bit 8
M2_A09	F27	O		通道 2 存储器地址, bit 9
M2_A10	B33	O		通道 2 存储器地址, bit 10
M2_A11	E27	O		通道 2 存储器地址, bit 11
M2_A12	C27	O		通道 2 存储器地址, bit 12
M2_A13	E35	O		通道 2 存储器地址, bit 13
M2_A14	F26	O		通道 2 存储器地址, bit 14
M2_A15	C26	O		通道 2 存储器地址, bit 15
M2_ATB0	E30	I		通道 2 存储器 ATB0
M2_ATB1	F30	I		通道 2 存储器 ATB1
M2_BA0	E33	O		通道 2 存储器地址, bit 0
M2_BA1	C33	O		通道 2 存储器地址, bit 1
M2_BA2	E26	O		通道 2 存储器地址, bit 2

M2_CAL_BLK	B34	I		通道 2 存储器 CAL_BLK
M2_CAS	F34	O		通道 2 存储器列地址选通
M2_CKE0	B26	O		通道 2 存储器时钟使能, cs 0
M2_CKE1	E25	O		通道 2 存储器时钟使能, cs 1
M2_CKE2	F25	O		通道 2 存储器时钟使能, cs 2
M2_CKE3	B25	O		通道 2 存储器时钟使能, cs 3
M2_CLK0N	F31	O		通道 2 存储器时钟-, cs0
M2_CLK0P	E31	O		通道 2 存储器时钟+, cs0
M2_CLK1N	C30	O		通道 2 存储器时钟-, cs1
M2_CLK1P	B30	O		通道 2 存储器时钟+, cs1
M2_CLK2N	F32	O		通道 2 存储器时钟-, cs2
M2_CLK2P	E32	O		通道 2 存储器时钟+, cs2
M2_CLK3N	B31	O		通道 2 存储器时钟-, cs3
M2_CLK3P	A31	O		通道 2 存储器时钟+, cs3
M2_CS0	A34	O		通道 2 存储器片选, cs0
M2_CS1	F35	O		通道 2 存储器片选, cs1
M2_CS2	F36	O		通道 2 存储器片选, cs2
M2_CS3	E36	O		通道 2 存储器片选, cs3
M2_ODT0	C35	O		通道 2 存储器 ODT, cs0
M2_ODT1	B36	O		通道 2 存储器 ODT, cs1
M2_ODT2	B35	O		通道 2 存储器 ODT, cs2
M2_ODT3	C36	O		通道 2 存储器 ODT, cs3
M2_PAR_ERR	B27	I		通道 2 存储器的奇偶校验结果
M2_PAR_IN	C32	O		通道 2 存储器的奇偶校验值
M2_RAS	F33	O		通道 2 存储器行地址选通
M2_RESET	A25	O		通道 2 存储器复位
M2_WE	E34	O		通道 2 存储器写使能
M2_S0_DMN	B40	I/O		通道 2 存储器数据屏蔽-, slice0
M2_S0_DMP	A40	I/O		通道 2 存储器数据屏蔽+, slice0
M2_S0_DQ00	B39	I/O		通道 2 存储器数据, bit0
M2_S0_DQ01	B43	I/O		通道 2 存储器数据, bit1
M2_S0_DQ02	B42	I/O		通道 2 存储器数据, bit2
M2_S0_DQ03	C42	I/O		通道 2 存储器数据, bit3
M2_S0_DQ04	A43	I/O		通道 2 存储器数据, bit4
M2_S0_DQ05	C39	I/O		通道 2 存储器数据, bit5
M2_S0_DQ06	B38	I/O		通道 2 存储器数据, bit6

M2_S0_DQ07	C38	I/O		通道 2 存储器数据, bit7
M2_S0_DQSN	B41	I/O		通道 2 存储器数据选通-, slice0
M2_S0_DQSP	C41	I/O		通道 2 存储器数据选通+, slice0
M2_S1_DMN	F40	I/O		通道 2 存储器数据屏蔽-, slice1
M2_S1_DMP	E40	I/O		通道 2 存储器数据屏蔽+, slice1
M2_S1_DQ08	F39	I/O		通道 2 存储器数据, bit08
M2_S1_DQ09	E43	I/O		通道 2 存储器数据, bit09
M2_S1_DQ10	F42	I/O		通道 2 存储器数据, bit10
M2_S1_DQ11	E42	I/O		通道 2 存储器数据, bit11
M2_S1_DQ12	F43	I/O		通道 2 存储器数据, bit12
M2_S1_DQ13	E39	I/O		通道 2 存储器数据, bit13
M2_S1_DQ14	F38	I/O		通道 2 存储器数据, bit14
M2_S1_DQ15	E38	I/O		通道 2 存储器数据, bit15
M2_S1_DQSN	E41	I/O		通道 2 存储器数据选通-, slice1
M2_S1_DQSP	F41	I/O		通道 2 存储器数据选通+, slice1
M2_S2_DMN	C20	I/O		通道 2 存储器数据屏蔽-, slice2
M2_S2_DMP	B20	I/O		通道 2 存储器数据屏蔽+, slice2
M2_S2_DQ16	A19	I/O		通道 2 存储器数据, bit16
M2_S2_DQ17	B18	I/O		通道 2 存储器数据, bit17
M2_S2_DQ18	B19	I/O		通道 2 存储器数据, bit18
M2_S2_DQ19	C18	I/O		通道 2 存储器数据, bit19
M2_S2_DQ20	A22	I/O		通道 2 存储器数据, bit20
M2_S2_DQ21	B22	I/O		通道 2 存储器数据, bit21
M2_S2_DQ22	B23	I/O		通道 2 存储器数据, bit22
M2_S2_DQ23	C23	I/O		通道 2 存储器数据, bit23
M2_S2_DQSN	B21	I/O		通道 2 存储器数据选通-, slice2
M2_S2_DQSP	C21	I/O		通道 2 存储器数据选通+, slice2
M2_S3_DMN	B13	I/O		通道 2 存储器数据屏蔽-, slice3
M2_S3_DMP	A13	I/O		通道 2 存储器数据屏蔽+, slice3
M2_S3_DQ24	B12	I/O		通道 2 存储器数据, bit24
M2_S3_DQ25	B16	I/O		通道 2 存储器数据, bit25
M2_S3_DQ26	B15	I/O		通道 2 存储器数据, bit26
M2_S3_DQ27	C15	I/O		通道 2 存储器数据, bit27
M2_S3_DQ28	A16	I/O		通道 2 存储器数据, bit28
M2_S3_DQ29	C12	I/O		通道 2 存储器数据, bit29
M2_S3_DQ30	B11	I/O		通道 2 存储器数据, bit30

M2_S3_DQ31	C11	I/O		通道 2 存储器数据, bit31
M2_S3_DQSN	B14	I/O		通道 2 存储器数据选通-, slice3
M2_S3_DQSP	C14	I/O		通道 2 存储器数据选通+, slice3
M2_S4_DMN	B6	I/O		通道 2 存储器数据屏蔽-, slice4
M2_S4_DMP	A6	I/O		通道 2 存储器数据屏蔽+, slice4
M2_S4_DQ32	A5	I/O		通道 2 存储器数据, bit32
M2_S4_DQ33	B9	I/O		通道 2 存储器数据, bit33
M2_S4_DQ34	A8	I/O		通道 2 存储器数据, bit34
M2_S4_DQ35	B8	I/O		通道 2 存储器数据, bit35
M2_S4_DQ36	A9	I/O		通道 2 存储器数据, bit36
M2_S4_DQ37	B5	I/O		通道 2 存储器数据, bit37
M2_S4_DQ38	A4	I/O		通道 2 存储器数据, bit38
M2_S4_DQ39	B4	I/O		通道 2 存储器数据, bit39
M2_S4_DQSN	A7	I/O		通道 2 存储器数据选通-, slice4
M2_S4_DQSP	B7	I/O		通道 2 存储器数据选通+, slice4
M2_S5_DMN	F20	I/O		通道 2 存储器数据屏蔽-, slice5
M2_S5_DMP	E20	I/O		通道 2 存储器数据屏蔽+, slice5
M2_S5_DQ40	F19	I/O		通道 2 存储器数据, bit40
M2_S5_DQ41	E23	I/O		通道 2 存储器数据, bit41
M2_S5_DQ42	E22	I/O		通道 2 存储器数据, bit42
M2_S5_DQ43	F22	I/O		通道 2 存储器数据, bit43
M2_S5_DQ44	F23	I/O		通道 2 存储器数据, bit44
M2_S5_DQ45	E19	I/O		通道 2 存储器数据, bit45
M2_S5_DQ46	E18	I/O		通道 2 存储器数据, bit46
M2_S5_DQ47	F18	I/O		通道 2 存储器数据, bit47
M2_S5_DQSN	E21	I/O		通道 2 存储器数据选通-, slice5
M2_S5_DQSP	F21	I/O		通道 2 存储器数据选通+, slice5
M2_S6_DMN	F13	I/O		通道 2 存储器数据屏蔽-, slice6
M2_S6_DMP	E13	I/O		通道 2 存储器数据屏蔽+, slice6
M2_S6_DQ48	F12	I/O		通道 2 存储器数据, bit48
M2_S6_DQ49	E16	I/O		通道 2 存储器数据, bit49
M2_S6_DQ50	E15	I/O		通道 2 存储器数据, bit50
M2_S6_DQ51	F16	I/O		通道 2 存储器数据, bit51
M2_S6_DQ52	F15	I/O		通道 2 存储器数据, bit52
M2_S6_DQ53	E12	I/O		通道 2 存储器数据, bit53
M2_S6_DQ54	E11	I/O		通道 2 存储器数据, bit54

M2_S6_DQ55	F11	I/O		通道 2 存储器数据, bit55
M2_S6_DQSN	E14	I/O		通道 2 存储器数据选通-, slice6
M2_S6_DQSP	F14	I/O		通道 2 存储器数据选通+, slice6
M2_S7_DMN	C47	I/O		通道 2 存储器数据屏蔽-, slice7
M2_S7_DMP	B47	I/O		通道 2 存储器数据屏蔽+, slice7
M2_S7_DQ56	A46	I/O		通道 2 存储器数据, bit56
M2_S7_DQ57	B49	I/O		通道 2 存储器数据, bit57
M2_S7_DQ58	A50	I/O		通道 2 存储器数据, bit58
M2_S7_DQ59	B50	I/O		通道 2 存储器数据, bit59
M2_S7_DQ60	B46	I/O		通道 2 存储器数据, bit60
M2_S7_DQ61	A49	I/O		通道 2 存储器数据, bit61
M2_S7_DQ62	B45	I/O		通道 2 存储器数据, bit62
M2_S7_DQ63	C45	I/O		通道 2 存储器数据, bit63
M2_S7_DQSN	B48	I/O		通道 2 存储器数据选通-, slice7
M2_S7_DQSP	C48	I/O		通道 2 存储器数据选通+, slice7
M2_S8_DMN	F47	I/O		通道 2 存储器数据屏蔽-, slice8
M2_S8_DMP	E47	I/O		通道 2 存储器数据屏蔽+, slice8
M2_S8_DQ64	F46	I/O		通道 2 存储器数据, bit64
M2_S8_DQ65	E50	I/O		通道 2 存储器数据, bit65
M2_S8_DQ66	D50	I/O		通道 2 存储器数据, bit66
M2_S8_DQ67	E49	I/O		通道 2 存储器数据, bit67
M2_S8_DQ68	D49	I/O		通道 2 存储器数据, bit68
M2_S8_DQ69	E46	I/O		通道 2 存储器数据, bit69
M2_S8_DQ70	E45	I/O		通道 2 存储器数据, bit70
M2_S8_DQ71	F45	I/O		通道 2 存储器数据, bit71
M2_S8_DQSN	E48	I/O		通道 2 存储器数据选通-, slice8
M2_S8_DQSP	F48	I/O		通道 2 存储器数据选通+, slice8
M3_A00	H29	O		通道 3 存储器地址, bit 0
M3_A01	L32	O		通道 3 存储器地址, bit 1
M3_A02	M32	O		通道 3 存储器地址, bit 2
M3_A03	J32	O		通道 3 存储器地址, bit 3
M3_A04	H32	O		通道 3 存储器地址, bit 4
M3_A05	H33	O		通道 3 存储器地址, bit 5
M3_A06	J33	O		通道 3 存储器地址, bit 6
M3_A07	L33	O		通道 3 存储器地址, bit 7
M3_A08	M33	O		通道 3 存储器地址, bit 8

M3_A09	J34	O		通道 3 存储器地址, bit 9
M3_A10	L28	O		通道 3 存储器地址, bit 10
M3_A11	H34	O		通道 3 存储器地址, bit 11
M3_A12	M34	O		通道 3 存储器地址, bit 12
M3_A13	J26	O		通道 3 存储器地址, bit 13
M3_A14	J35	O		通道 3 存储器地址, bit 14
M3_A15	M35	O		通道 3 存储器地址, bit 15
M3_ATB0	L31	I		通道 3 存储器 ATB0
M3_ATB1	M31	I		通道 3 存储器 ATB1
M3_BA0	H28	O		通道 3 存储器地址, bit 0
M3_BA1	M28	O		通道 3 存储器地址, bit 1
M3_BA2	H35	O		通道 3 存储器地址, bit 2
M3_CAL_BLK	L27	I		通道 3 存储器 CAL_BLK
M3_CAS	J27	O		通道 3 存储器列地址选通
M3_CKE0	L35	O		通道 3 存储器时钟使能, cs 0
M3_CKE1	J36	O		通道 3 存储器时钟使能, cs 1
M3_CKE2	H36	O		通道 3 存储器时钟使能, cs 2
M3_CKE3	M36	O		通道 3 存储器时钟使能, cs 3
M3_CLK0N	J31	O		通道 3 存储器时钟-, cs0
M3_CLK0P	H31	O		通道 3 存储器时钟+, cs0
M3_CLK1N	M30	O		通道 3 存储器时钟-, cs1
M3_CLK1P	L30	O		通道 3 存储器时钟+, cs1
M3_CLK2N	J30	O		通道 3 存储器时钟-, cs2
M3_CLK2P	H30	O		通道 3 存储器时钟+, cs2
M3_CLK3N	M29	O		通道 3 存储器时钟-, cs3
M3_CLK3P	L29	O		通道 3 存储器时钟+, cs3
M3_CS0	M27	O		通道 3 存储器片选, cs0
M3_CS1	H26	O		通道 3 存储器片选, cs1
M3_CS2	J25	O		通道 3 存储器片选, cs2
M3_CS3	H25	O		通道 3 存储器片选, cs3
M3_ODT0	L26	O		通道 3 存储器 ODT, cs0
M3_ODT1	M25	O		通道 3 存储器 ODT, cs1
M3_ODT2	M26	O		通道 3 存储器 ODT, cs2
M3_ODT3	L25	O		通道 3 存储器 ODT, cs3
M3_PAR_ERR	L34	I		通道 3 存储器的奇偶校验结果
M3_PAR_IN	J29	O		通道 3 存储器的奇偶校验值

M3_RAS	J28	O		通道 3 存储器行地址选通
M3_RESET	L36	O		通道 3 存储器复位
M3_WE	H27	O		通道 3 存储器写使能
M3_S0_DMN	M14	I/O		通道 3 存储器数据屏蔽-, slice0
M3_S0_DMP	L14	I/O		通道 3 存储器数据屏蔽+, slice0
M3_S0_DQ00	L16	I/O		通道 3 存储器数据, bit0
M3_S0_DQ01	M16	I/O		通道 3 存储器数据, bit1
M3_S0_DQ02	M15	I/O		通道 3 存储器数据, bit2
M3_S0_DQ03	L15	I/O		通道 3 存储器数据, bit3
M3_S0_DQ04	L12	I/O		通道 3 存储器数据, bit4
M3_S0_DQ05	M12	I/O		通道 3 存储器数据, bit5
M3_S0_DQ06	M11	I/O		通道 3 存储器数据, bit6
M3_S0_DQ07	L11	I/O		通道 3 存储器数据, bit7
M3_S0_DQSN	L13	I/O		通道 3 存储器数据选通-, slice0
M3_S0_DQSP	M13	I/O		通道 3 存储器数据选通+, slice0
M3_S1_DMN	J14	I/O		通道 3 存储器数据屏蔽-, slice1
M3_S1_DMP	H14	I/O		通道 3 存储器数据屏蔽+, slice1
M3_S1_DQ08	H15	I/O		通道 3 存储器数据, bit08
M3_S1_DQ09	J11	I/O		通道 3 存储器数据, bit09
M3_S1_DQ10	H12	I/O		通道 3 存储器数据, bit10
M3_S1_DQ11	J12	I/O		通道 3 存储器数据, bit11
M3_S1_DQ12	J16	I/O		通道 3 存储器数据, bit12
M3_S1_DQ13	J15	I/O		通道 3 存储器数据, bit13
M3_S1_DQ14	H11	I/O		通道 3 存储器数据, bit14
M3_S1_DQ15	H16	I/O		通道 3 存储器数据, bit15
M3_S1_DQSN	H13	I/O		通道 3 存储器数据选通-, slice1
M3_S1_DQSP	J13	I/O		通道 3 存储器数据选通+, slice1
M3_S2_DMN	H52	I/O		通道 3 存储器数据屏蔽-, slice2
M3_S2_DMP	H53	I/O		通道 3 存储器数据屏蔽+, slice2
M3_S2_DQ16	J53	I/O		通道 3 存储器数据, bit16
M3_S2_DQ17	K53	I/O		通道 3 存储器数据, bit17
M3_S2_DQ18	E53	I/O		通道 3 存储器数据, bit18
M3_S2_DQ19	F53	I/O		通道 3 存储器数据, bit19
M3_S2_DQ20	E52	I/O		通道 3 存储器数据, bit20
M3_S2_DQ21	F52	I/O		通道 3 存储器数据, bit21
M3_S2_DQ22	J52	I/O		通道 3 存储器数据, bit22

M3_S2_DQ23	K52	I/O		通道 3 存储器数据, bit23
M3_S2_DQSN	G53	I/O		通道 3 存储器数据选通-, slice2
M3_S2_DQSP	G52	I/O		通道 3 存储器数据选通+, slice2
M3_S3_DMN	J48	I/O		通道 3 存储器数据屏蔽-, slice3
M3_S3_DMP	H48	I/O		通道 3 存储器数据屏蔽+, slice3
M3_S3_DQ24	G50	I/O		通道 3 存储器数据, bit24
M3_S3_DQ25	G49	I/O		通道 3 存储器数据, bit25
M3_S3_DQ26	H50	I/O		通道 3 存储器数据, bit26
M3_S3_DQ27	H49	I/O		通道 3 存储器数据, bit27
M3_S3_DQ28	H46	I/O		通道 3 存储器数据, bit28
M3_S3_DQ29	J46	I/O		通道 3 存储器数据, bit29
M3_S3_DQ30	J45	I/O		通道 3 存储器数据, bit30
M3_S3_DQ31	H45	I/O		通道 3 存储器数据, bit31
M3_S3_DQSN	H47	I/O		通道 3 存储器数据选通-, slice3
M3_S3_DQSP	J47	I/O		通道 3 存储器数据选通+, slice3
M3_S4_DMN	M48	I/O		通道 3 存储器数据屏蔽-, slice4
M3_S4_DMP	L48	I/O		通道 3 存储器数据屏蔽+, slice4
M3_S4_DQ32	K49	I/O		通道 3 存储器数据, bit32
M3_S4_DQ33	K50	I/O		通道 3 存储器数据, bit33
M3_S4_DQ34	L50	I/O		通道 3 存储器数据, bit34
M3_S4_DQ35	L49	I/O		通道 3 存储器数据, bit35
M3_S4_DQ36	L46	I/O		通道 3 存储器数据, bit36
M3_S4_DQ37	L45	I/O		通道 3 存储器数据, bit37
M3_S4_DQ38	M46	I/O		通道 3 存储器数据, bit38
M3_S4_DQ39	M45	I/O		通道 3 存储器数据, bit39
M3_S4_DQSN	L47	I/O		通道 3 存储器数据选通-, slice4
M3_S4_DQSP	M47	I/O		通道 3 存储器数据选通+, slice4
M3_S5_DMN	J41	I/O		通道 3 存储器数据屏蔽-, slice5
M3_S5_DMP	H41	I/O		通道 3 存储器数据屏蔽+, slice5
M3_S5_DQ40	H42	I/O		通道 3 存储器数据, bit40
M3_S5_DQ41	H43	I/O		通道 3 存储器数据, bit41
M3_S5_DQ42	H38	I/O		通道 3 存储器数据, bit42
M3_S5_DQ43	H39	I/O		通道 3 存储器数据, bit43
M3_S5_DQ44	J38	I/O		通道 3 存储器数据, bit44
M3_S5_DQ45	J39	I/O		通道 3 存储器数据, bit45
M3_S5_DQ46	J42	I/O		通道 3 存储器数据, bit46

M3_S5_DQ47	J43	I/O		通道 3 存储器数据, bit47
M3_S5_DQSN	H40	I/O		通道 3 存储器数据选通-, slice5
M3_S5_DQSP	J40	I/O		通道 3 存储器数据选通+, slice5
M3_S6_DMN	M41	I/O		通道 3 存储器数据屏蔽-, slice6
M3_S6_DMP	L41	I/O		通道 3 存储器数据屏蔽+, slice6
M3_S6_DQ48	L42	I/O		通道 3 存储器数据, bit48
M3_S6_DQ49	L43	I/O		通道 3 存储器数据, bit49
M3_S6_DQ50	M43	I/O		通道 3 存储器数据, bit50
M3_S6_DQ51	M42	I/O		通道 3 存储器数据, bit51
M3_S6_DQ52	L39	I/O		通道 3 存储器数据, bit52
M3_S6_DQ53	M39	I/O		通道 3 存储器数据, bit53
M3_S6_DQ54	L38	I/O		通道 3 存储器数据, bit54
M3_S6_DQ55	M38	I/O		通道 3 存储器数据, bit55
M3_S6_DQSN	L40	I/O		通道 3 存储器数据选通-, slice6
M3_S6_DQSP	M40	I/O		通道 3 存储器数据选通+, slice6
M3_S7_DMN	M21	I/O		通道 3 存储器数据屏蔽-, slice7
M3_S7_DMP	L21	I/O		通道 3 存储器数据屏蔽+, slice7
M3_S7_DQ56	L23	I/O		通道 3 存储器数据, bit56
M3_S7_DQ57	L19	I/O		通道 3 存储器数据, bit57
M3_S7_DQ58	L22	I/O		通道 3 存储器数据, bit58
M3_S7_DQ59	L18	I/O		通道 3 存储器数据, bit59
M3_S7_DQ60	M18	I/O		通道 3 存储器数据, bit60
M3_S7_DQ61	M19	I/O		通道 3 存储器数据, bit61
M3_S7_DQ62	M22	I/O		通道 3 存储器数据, bit62
M3_S7_DQ63	M23	I/O		通道 3 存储器数据, bit63
M3_S7_DQSN	L20	I/O		通道 3 存储器数据选通-, slice7
M3_S7_DQSP	M20	I/O		通道 3 存储器数据选通+, slice7
M3_S8_DMN	J21	I/O		通道 3 存储器数据屏蔽-, slice8
M3_S8_DMP	H21	I/O		通道 3 存储器数据屏蔽+, slice8
M3_S8_DQ64	H22	I/O		通道 3 存储器数据, bit64
M3_S8_DQ65	H23	I/O		通道 3 存储器数据, bit65
M3_S8_DQ66	J23	I/O		通道 3 存储器数据, bit66
M3_S8_DQ67	J22	I/O		通道 3 存储器数据, bit67
M3_S8_DQ68	H19	I/O		通道 3 存储器数据, bit68
M3_S8_DQ69	J19	I/O		通道 3 存储器数据, bit69
M3_S8_DQ70	H18	I/O		通道 3 存储器数据, bit70

M3_S8_DQ71	J18	I/O		通道 3 存储器数据, bit71
M3_S8_DQSN	H20	I/O		通道 3 存储器数据选通-, slice8
M3_S8_DQSP	J20	I/O		通道 3 存储器数据选通+, slice8

注:

(1)信号类型: I 输入引脚, O 输出引脚, I/O 双向引脚, Z 三态或高阻引脚, A 模拟信号脚, GND 地, P 电源引脚, RSV 保留引脚, NC 空引脚;

(2)片内上下拉: IPU 片内上拉, IPD 片内下拉。

8.5 电源引脚 (983 PIN)

表 8-6 电源引脚

信号名	引脚编号	信号类型	片内上下拉	功能说明
VDD_CORE	AA43			0.9V 内核电源
VDD_CORE	AA45			
VDD_CORE	AA47			
VDD_CORE	AA49			
VDD_CORE	AA51			
VDD_CORE	AA53			
VDD_CORE	AA55			
VDD_CORE	AA57			
VDD_CORE	AB44			
VDD_CORE	AB46			
VDD_CORE	AC43			
VDD_CORE	AC45			
VDD_CORE	AD44			
VDD_CORE	AE43			
VDD_CORE	AE45			
VDD_CORE	AF12			
VDD_CORE	AF14			
VDD_CORE	AF16			
VDD_CORE	AF18			
VDD_CORE	AF20			
VDD_CORE	AF22			
VDD_CORE	AF24			
VDD_CORE	AF26			
VDD_CORE	AF28			

VDD_CORE	AF30		
VDD_CORE	AF34		
VDD_CORE	AF36		
VDD_CORE	AF42		
VDD_CORE	AF44		
VDD_CORE	AG11		
VDD_CORE	AG13		
VDD_CORE	AG15		
VDD_CORE	AG17		
VDD_CORE	AG19		
VDD_CORE	AG21		
VDD_CORE	AG23		
VDD_CORE	AG25		
VDD_CORE	AG27		
VDD_CORE	AG29		
VDD_CORE	AG31		
VDD_CORE	AG33		
VDD_CORE	AG35		
VDD_CORE	AG37		
VDD_CORE	AG43		
VDD_CORE	AG45		
VDD_CORE	AH12		
VDD_CORE	AH14		
VDD_CORE	AH16		
VDD_CORE	AH18		
VDD_CORE	AH20		
VDD_CORE	AH22		
VDD_CORE	AH24		
VDD_CORE	AH42		
VDD_CORE	AH44		
VDD_CORE	AJ17		
VDD_CORE	L51		
VDD_CORE	M44		
VDD_CORE	M50		
VDD_CORE	N11		
VDD_CORE	N13		

VDD_CORE	N15		
VDD_CORE	N17		
VDD_CORE	N19		
VDD_CORE	N21		
VDD_CORE	N23		
VDD_CORE	N37		
VDD_CORE	N39		
VDD_CORE	N41		
VDD_CORE	N43		
VDD_CORE	N45		
VDD_CORE	N47		
VDD_CORE	N49		
VDD_CORE	N51		
VDD_CORE	P12		
VDD_CORE	P14		
VDD_CORE	P16		
VDD_CORE	P18		
VDD_CORE	P20		
VDD_CORE	P22		
VDD_CORE	P24		
VDD_CORE	P26		
VDD_CORE	P28		
VDD_CORE	P30		
VDD_CORE	P32		
VDD_CORE	P34		
VDD_CORE	P36		
VDD_CORE	P38		
VDD_CORE	P40		
VDD_CORE	P42		
VDD_CORE	P44		
VDD_CORE	P46		
VDD_CORE	P48		
VDD_CORE	P50		
VDD_CORE	R11		
VDD_CORE	R13		
VDD_CORE	R15		

VDD_CORE	R17		
VDD_CORE	R19		
VDD_CORE	R21		
VDD_CORE	R23		
VDD_CORE	R25		
VDD_CORE	R27		
VDD_CORE	R29		
VDD_CORE	R33		
VDD_CORE	R35		
VDD_CORE	R37		
VDD_CORE	R39		
VDD_CORE	R41		
VDD_CORE	R43		
VDD_CORE	R45		
VDD_CORE	R47		
VDD_CORE	R49		
VDD_CORE	R51		
VDD_CORE	R53		
VDD_CORE	R55		
VDD_CORE	R57		
VDD_CORE	T43		
VDD_CORE	T44		
VDD_CORE	T45		
VDD_CORE	T46		
VDD_CORE	T47		
VDD_CORE	T48		
VDD_CORE	T49		
VDD_CORE	T50		
VDD_CORE	T51		
VDD_CORE	T52		
VDD_CORE	T53		
VDD_CORE	T54		
VDD_CORE	T55		
VDD_CORE	T56		
VDD_CORE	T57		
VDD_CORE	U43		

VDD_CORE	U44		
VDD_CORE	U45		
VDD_CORE	U46		
VDD_CORE	U47		
VDD_CORE	U48		
VDD_CORE	U49		
VDD_CORE	U50		
VDD_CORE	U51		
VDD_CORE	U52		
VDD_CORE	U53		
VDD_CORE	U54		
VDD_CORE	U55		
VDD_CORE	U56		
VDD_CORE	U57		
VDD_CORE	V44		
VDD_CORE	V46		
VDD_CORE	V48		
VDD_CORE	V50		
VDD_CORE	V52		
VDD_CORE	V54		
VDD_CORE	V56		
VDD_CORE	W43		
VDD_CORE	W44		
VDD_CORE	W45		
VDD_CORE	W46		
VDD_CORE	W47		
VDD_CORE	W48		
VDD_CORE	W49		
VDD_CORE	W50		
VDD_CORE	W51		
VDD_CORE	W52		
VDD_CORE	W53		
VDD_CORE	W54		
VDD_CORE	W55		
VDD_CORE	W56		
VDD_CORE	W57		

VDD_CORE	Y43				
VDD_CORE	Y44				
VDD_CORE	Y45				
VDD_CORE	Y46				
VDD_CORE	Y47				
VDD_CORE	Y48				
VDD_CORE	Y49				
VDD_CORE	Y50				
VDD_CORE	Y51				
VDD_CORE	Y52				
VDD_CORE	Y53				
VDD_CORE	Y54				
VDD_CORE	Y55				
VDD_CORE	Y56				
VDD_CORE	Y57				
VDD_IO	AF38				1.8V 通用 IO 类管脚供电电源
VDD_IO	AF40				
VDD_IO	AG39				
VDD_IO	AG41				
VDD_IO	AH38				
VDD_IO	AH40				
VDDA_PCIE	AA12			0.95V PCIE 电源	
VDDA_PCIE	AA14				
VDDA_PCIE	AB11				
VDDA_PCIE	AB13				
VDDA_PCIE	AB15				
VDDA_PCIE	AC12				
VDDA_PCIE	AC14				
VDDA_PCIE	AD11				
VDDA_PCIE	AD13				
VDDA_PCIE	AD15				
VDDA_PCIE	AE12				
VDDA_PCIE	AE14				
VDDA_PCIE	W12				
VDDA_PCIE	W14				
VDDA_PCIE	Y11				

VDDA_PCIE	Y13			1.5V 的 PCIE 电源
VDDA_PCIE	Y15			
VDDHV_PCI E	T11			
VDDHV_PCI E	T13			
VDDHV_PCI E	T15			
VDDHV_PCI E	U12			
VDDHV_PCI E	U14			
VDDHV_PCI E	V11			
VDDHV_PCI E	V13			
VDDHV_PCI E	V15			
VDDHV_PL L0	AF32			1.8V 的模拟电源
VDDHV_PL L1	R31			
VDDQ_MCU 01	AH26			1.5V DDR3 DIMM 接口电源
VDDQ_MCU 01	AH28			
VDDQ_MCU 01	AH30			
VDDQ_MCU 01	AH32			
VDDQ_MCU 01	AH34			
VDDQ_MCU 01	AH36			
VDDQ_MCU 01	AL25			

VDDQ_MCU 01	AL27			
VDDQ_MCU 01	AL29			
VDDQ_MCU 01	AL31			
VDDQ_MCU 01	AL33			
VDDQ_MCU 01	AL35			
VDDQ_MCU 01	AP25			
VDDQ_MCU 01	AP27			
VDDQ_MCU 01	AP29			
VDDQ_MCU 01	AP31			
VDDQ_MCU 01	AP33			
VDDQ_MCU 01	AP35			
VDDQ_MCU 01	AU24			
VDDQ_MCU 01	AU27			
VDDQ_MCU 01	AU30			
VDDQ_MCU 01	AU33			
VDDQ_MCU 23	D25			1.5V DDR3 DIMM 接口电源
VDDQ_MCU 23	D28			
VDDQ_MCU 23	D31			

VDDQ_MCU 23	D34		
VDDQ_MCU 23	G26		
VDDQ_MCU 23	G28		
VDDQ_MCU 23	G30		
VDDQ_MCU 23	G32		
VDDQ_MCU 23	G34		
VDDQ_MCU 23	G36		
VDDQ_MCU 23	K26		
VDDQ_MCU 23	K28		
VDDQ_MCU 23	K30		
VDDQ_MCU 23	K32		
VDDQ_MCU 23	K34		
VDDQ_MCU 23	K36		
VDDQ_MCU 23	N25		
VDDQ_MCU 23	N27		
VDDQ_MCU 23	N29		
VDDQ_MCU 23	N31		
VDDQ_MCU 23	N33		

VDDQ_MCU 23	N35			
VSS	A10			数字地
VSS	A37			
VSS	A51			
VSS	AA1			
VSS	AA2			
VSS	AA5			
VSS	AA6			
VSS	AA9			
VSS	AA10			
VSS	AA11			
VSS	AA13			
VSS	AA15			
VSS	AA44			
VSS	AA46			
VSS	AA48			
VSS	AA50			
VSS	AA52			
VSS	AA54			
VSS	AA56			
VSS	AB3			
VSS	AB4			
VSS	AB7			
VSS	AB8			
VSS	AB12			
VSS	AB14			
VSS	AB43			
VSS	AB45			
VSS	AB54			
VSS	AC1			
VSS	AC2			
VSS	AC5			
VSS	AC6			
VSS	AC9			
VSS	AC10			

VSS	AC11		
VSS	AC13		
VSS	AC15		
VSS	AC44		
VSS	AC48		
VSS	AC51		
VSS	AC54		
VSS	AD3		
VSS	AD4		
VSS	AD7		
VSS	AD8		
VSS	AD12		
VSS	AD14		
VSS	AD43		
VSS	AD45		
VSS	AD48		
VSS	AD51		
VSS	AD54		
VSS	AE1		
VSS	AE2		
VSS	AE5		
VSS	AE6		
VSS	AE9		
VSS	AE10		
VSS	AE11		
VSS	AE13		
VSS	AE15		
VSS	AE44		
VSS	AE48		
VSS	AE51		
VSS	AE54		
VSS	AF3		
VSS	AF4		
VSS	AF7		
VSS	AF8		
VSS	AF11		

VSS	AF13		
VSS	AF15		
VSS	AF17		
VSS	AF19		
VSS	AF21		
VSS	AF23		
VSS	AF25		
VSS	AF27		
VSS	AF29		
VSS	AF31		
VSS	AF33		
VSS	AF35		
VSS	AF37		
VSS	AF39		
VSS	AF41		
VSS	AF43		
VSS	AF45		
VSS	AF48		
VSS	AF51		
VSS	AF54		
VSS	AG1		
VSS	AG2		
VSS	AG5		
VSS	AG6		
VSS	AG9		
VSS	AG10		
VSS	AG12		
VSS	AG14		
VSS	AG16		
VSS	AG18		
VSS	AG20		
VSS	AG22		
VSS	AG24		
VSS	AG26		
VSS	AG28		
VSS	AG30		

VSS	AG32		
VSS	AG34		
VSS	AG36		
VSS	AG38		
VSS	AG40		
VSS	AG42		
VSS	AG44		
VSS	AG48		
VSS	AG51		
VSS	AG54		
VSS	AH3		
VSS	AH4		
VSS	AH7		
VSS	AH8		
VSS	AH11		
VSS	AH13		
VSS	AH15		
VSS	AH17		
VSS	AH19		
VSS	AH21		
VSS	AH23		
VSS	AH25		
VSS	AH27		
VSS	AH29		
VSS	AH31		
VSS	AH33		
VSS	AH35		
VSS	AH37		
VSS	AH39		
VSS	AH41		
VSS	AH43		
VSS	AH45		
VSS	AH46		
VSS	AH47		
VSS	AH51		
VSS	AH54		

VSS	AJ1		
VSS	AJ2		
VSS	AJ5		
VSS	AJ6		
VSS	AJ9		
VSS	AJ10		
VSS	AJ24		
VSS	AJ37		
VSS	AJ44		
VSS	AJ48		
VSS	AJ51		
VSS	AJ54		
VSS	AK3		
VSS	AK4		
VSS	AK7		
VSS	AK8		
VSS	AK10		
VSS	AK17		
VSS	AK24		
VSS	AK37		
VSS	AK44		
VSS	AK49		
VSS	AK50		
VSS	AK52		
VSS	AK53		
VSS	AK54		
VSS	AL1		
VSS	AL2		
VSS	AL5		
VSS	AL6		
VSS	AL9		
VSS	AL11		
VSS	AL12		
VSS	AL13		
VSS	AL14		
VSS	AL15		

VSS	AL16		
VSS	AL17		
VSS	AL18		
VSS	AL19		
VSS	AL20		
VSS	AL21		
VSS	AL22		
VSS	AL23		
VSS	AL24		
VSS	AL26		
VSS	AL28		
VSS	AL30		
VSS	AL32		
VSS	AL34		
VSS	AL36		
VSS	AL37		
VSS	AL38		
VSS	AL39		
VSS	AL40		
VSS	AL41		
VSS	AL42		
VSS	AL43		
VSS	AL44		
VSS	AL45		
VSS	AL46		
VSS	AL47		
VSS	AL51		
VSS	AL54		
VSS	AM3		
VSS	AM4		
VSS	AM7		
VSS	AM8		
VSS	AM10		
VSS	AM17		
VSS	AM24		
VSS	AM37		

VSS	AM44		
VSS	AM48		
VSS	AM51		
VSS	AM54		
VSS	AN1		
VSS	AN2		
VSS	AN5		
VSS	AN6		
VSS	AN9		
VSS	AN10		
VSS	AN17		
VSS	AN24		
VSS	AN37		
VSS	AN44		
VSS	AN49		
VSS	AN50		
VSS	AN51		
VSS	AN54		
VSS	AP3		
VSS	AP4		
VSS	AP7		
VSS	AP8		
VSS	AP10		
VSS	AP11		
VSS	AP12		
VSS	AP13		
VSS	AP14		
VSS	AP15		
VSS	AP16		
VSS	AP17		
VSS	AP18		
VSS	AP19		
VSS	AP20		
VSS	AP21		
VSS	AP22		
VSS	AP23		

VSS	AP24		
VSS	AP26		
VSS	AP28		
VSS	AP30		
VSS	AP32		
VSS	AP34		
VSS	AP36		
VSS	AP37		
VSS	AP38		
VSS	AP39		
VSS	AP40		
VSS	AP41		
VSS	AP42		
VSS	AP43		
VSS	AP44		
VSS	AP45		
VSS	AP46		
VSS	AP47		
VSS	AP51		
VSS	AP54		
VSS	AR1		
VSS	AR2		
VSS	AR5		
VSS	AR6		
VSS	AR9		
VSS	AR16		
VSS	AR23		
VSS	AR36		
VSS	AR43		
VSS	AR48		
VSS	AR51		
VSS	AR54		
VSS	AT3		
VSS	AT4		
VSS	AT7		
VSS	AT8		

VSS	AT9		
VSS	AT16		
VSS	AT23		
VSS	AT36		
VSS	AT43		
VSS	AT49		
VSS	AT50		
VSS	AT51		
VSS	AT54		
VSS	AU1		
VSS	AU2		
VSS	AU5		
VSS	AU6		
VSS	AU9		
VSS	AU10		
VSS	AU11		
VSS	AU12		
VSS	AU13		
VSS	AU14		
VSS	AU15		
VSS	AU16		
VSS	AU17		
VSS	AU18		
VSS	AU19		
VSS	AU20		
VSS	AU21		
VSS	AU22		
VSS	AU23		
VSS	AU25		
VSS	AU26		
VSS	AU28		
VSS	AU29		
VSS	AU31		
VSS	AU32		
VSS	AU34		
VSS	AU35		

VSS	AU36		
VSS	AU37		
VSS	AU38		
VSS	AU39		
VSS	AU40		
VSS	AU41		
VSS	AU42		
VSS	AU43		
VSS	AU44		
VSS	AU45		
VSS	AU46		
VSS	AU47		
VSS	AU50		
VSS	AU52		
VSS	AU53		
VSS	AV1		
VSS	AV3		
VSS	AV4		
VSS	AV5		
VSS	AV6		
VSS	AV7		
VSS	AV8		
VSS	AV9		
VSS	AV12		
VSS	AV15		
VSS	AV18		
VSS	AV21		
VSS	AV22		
VSS	AV24		
VSS	AV27		
VSS	AV30		
VSS	AV33		
VSS	AV35		
VSS	AV36		
VSS	AV39		
VSS	AV42		

VSS	AV45		
VSS	AV48		
VSS	AV50		
VSS	AW8		
VSS	AW15		
VSS	AW22		
VSS	AW35		
VSS	AW42		
VSS	AW49		
VSS	AY8		
VSS	AY15		
VSS	AY42		
VSS	AY49		
VSS	B2		
VSS	B3		
VSS	B10		
VSS	B17		
VSS	B24		
VSS	B37		
VSS	B44		
VSS	B51		
VSS	C1		
VSS	C2		
VSS	C3		
VSS	C4		
VSS	C5		
VSS	C6		
VSS	C7		
VSS	C8		
VSS	C9		
VSS	C10		
VSS	C13		
VSS	C16		
VSS	C17		
VSS	C19		
VSS	C22		

VSS	C24		
VSS	C25		
VSS	C28		
VSS	C31		
VSS	C34		
VSS	C37		
VSS	C40		
VSS	C43		
VSS	C44		
VSS	C46		
VSS	C49		
VSS	C50		
VSS	C51		
VSS	D3		
VSS	D4		
VSS	D7		
VSS	D8		
VSS	D11		
VSS	D12		
VSS	D13		
VSS	D14		
VSS	D15		
VSS	D16		
VSS	D17		
VSS	D18		
VSS	D19		
VSS	D20		
VSS	D21		
VSS	D22		
VSS	D23		
VSS	D24		
VSS	D26		
VSS	D27		
VSS	D29		
VSS	D30		
VSS	D32		

VSS	D33		
VSS	D35		
VSS	D36		
VSS	D37		
VSS	D38		
VSS	D39		
VSS	D40		
VSS	D41		
VSS	D42		
VSS	D43		
VSS	D44		
VSS	D45		
VSS	D46		
VSS	D47		
VSS	D48		
VSS	D51		
VSS	D52		
VSS	D53		
VSS	D54		
VSS	E1		
VSS	E2		
VSS	E5		
VSS	E6		
VSS	E9		
VSS	E10		
VSS	E17		
VSS	E24		
VSS	E37		
VSS	E44		
VSS	E51		
VSS	E54		
VSS	F3		
VSS	F4		
VSS	F7		
VSS	F8		
VSS	F10		

VSS	F17		
VSS	F24		
VSS	F37		
VSS	F44		
VSS	F49		
VSS	F50		
VSS	F51		
VSS	F54		
VSS	G1		
VSS	G2		
VSS	G5		
VSS	G6		
VSS	G9		
VSS	G11		
VSS	G12		
VSS	G13		
VSS	G14		
VSS	G15		
VSS	G16		
VSS	G17		
VSS	G18		
VSS	G19		
VSS	G20		
VSS	G21		
VSS	G22		
VSS	G23		
VSS	G24		
VSS	G25		
VSS	G27		
VSS	G29		
VSS	G31		
VSS	G33		
VSS	G35		
VSS	G37		
VSS	G38		
VSS	G39		

VSS	G40		
VSS	G41		
VSS	G42		
VSS	G43		
VSS	G44		
VSS	G45		
VSS	G46		
VSS	G47		
VSS	G48		
VSS	G51		
VSS	G54		
VSS	H3		
VSS	H4		
VSS	H7		
VSS	H8		
VSS	H10		
VSS	H17		
VSS	H24		
VSS	H37		
VSS	H44		
VSS	H51		
VSS	H54		
VSS	J1		
VSS	J2		
VSS	J5		
VSS	J6		
VSS	J9		
VSS	J10		
VSS	J17		
VSS	J24		
VSS	J37		
VSS	J44		
VSS	J49		
VSS	J50		
VSS	J51		
VSS	J54		

VSS	K3		
VSS	K4		
VSS	K7		
VSS	K8		
VSS	K11		
VSS	K12		
VSS	K13		
VSS	K14		
VSS	K15		
VSS	K16		
VSS	K17		
VSS	K18		
VSS	K19		
VSS	K20		
VSS	K21		
VSS	K22		
VSS	K23		
VSS	K24		
VSS	K25		
VSS	K27		
VSS	K29		
VSS	K31		
VSS	K33		
VSS	K35		
VSS	K37		
VSS	K38		
VSS	K39		
VSS	K40		
VSS	K41		
VSS	K42		
VSS	K43		
VSS	K44		
VSS	K45		
VSS	K46		
VSS	K47		
VSS	K48		

VSS	K51		
VSS	K54		
VSS	L1		
VSS	L2		
VSS	L5		
VSS	L6		
VSS	L9		
VSS	L10		
VSS	L17		
VSS	L24		
VSS	L37		
VSS	L44		
VSS	L52		
VSS	L53		
VSS	L54		
VSS	M3		
VSS	M4		
VSS	M7		
VSS	M8		
VSS	M10		
VSS	M17		
VSS	M24		
VSS	M37		
VSS	M49		
VSS	M51		
VSS	N1		
VSS	N2		
VSS	N5		
VSS	N6		
VSS	N9		
VSS	N10		
VSS	N12		
VSS	N14		
VSS	N16		
VSS	N18		
VSS	N20		

VSS	N22		
VSS	N24		
VSS	N26		
VSS	N28		
VSS	N30		
VSS	N32		
VSS	N34		
VSS	N36		
VSS	N38		
VSS	N40		
VSS	N42		
VSS	N44		
VSS	N46		
VSS	N48		
VSS	N50		
VSS	P3		
VSS	P4		
VSS	P7		
VSS	P8		
VSS	P11		
VSS	P13		
VSS	P15		
VSS	P17		
VSS	P19		
VSS	P21		
VSS	P23		
VSS	P25		
VSS	P27		
VSS	P29		
VSS	P31		
VSS	P33		
VSS	P35		
VSS	P37		
VSS	P39		
VSS	P41		
VSS	P43		

VSS	P45		
VSS	P47		
VSS	P49		
VSS	P51		
VSS	R1		
VSS	R2		
VSS	R5		
VSS	R6		
VSS	R9		
VSS	R10		
VSS	R12		
VSS	R14		
VSS	R16		
VSS	R18		
VSS	R20		
VSS	R22		
VSS	R24		
VSS	R26		
VSS	R28		
VSS	R30		
VSS	R32		
VSS	R34		
VSS	R36		
VSS	R38		
VSS	R40		
VSS	R42		
VSS	R44		
VSS	R46		
VSS	R48		
VSS	R50		
VSS	R52		
VSS	R54		
VSS	R56		
VSS	T3		
VSS	T4		
VSS	T7		

VSS	T8		
VSS	T12		
VSS	T14		
VSS	U1		
VSS	U2		
VSS	U5		
VSS	U6		
VSS	U9		
VSS	U10		
VSS	U11		
VSS	U13		
VSS	U15		
VSS	V3		
VSS	V4		
VSS	V7		
VSS	V8		
VSS	V12		
VSS	V14		
VSS	V43		
VSS	V45		
VSS	V47		
VSS	V49		
VSS	V51		
VSS	V53		
VSS	V55		
VSS	V57		
VSS	W1		
VSS	W2		
VSS	W5		
VSS	W6		
VSS	W9		
VSS	W10		
VSS	W11		
VSS	W13		
VSS	W15		
VSS	Y3		

VSS	Y4			
VSS	Y7			
VSS	Y8			
VSS	Y12			
VSS	Y14			

注:

(1)信号类型: I 输入引脚, O 输出引脚, Z 三态或高阻引脚, A 模拟信号脚, GND 地, P 电源引脚, RSV 保留引脚, NC 空引脚;

(2)片内上下拉: IPU 片内上拉, IPD 片内下拉。