

FT-1500A/4 高性能通用微处理器 数据手册

PHYTIUM

2018 年 6 月

版本号 1.7

技术支持邮箱: support@phytium.com.cn

版本历史

版本号	作者	参与者	发布日期	备注
1. 0			2014. 10. 20	
1. 1			2014. 12. 26	
1. 2			2015. 04. 27	
1. 3			2015. 09. 06	更新上电时序
1. 3. 1			2015. 12. 19	
1. 4			2016. 03. 21	增加 GPIO 复用列表和装 焊温度曲线等内容；调整 了文档结构及部分内容
1. 5			2016. 7	
1. 6		应用支持	2018. 2	调整文档结构；增加 FCR 寄存器说明；完善 spi 寄 存器配置寄存器列表；
1. 7		应用支持	2018. 06. 05	完善 spi 寄存器配置寄存 器列表；

目录

目录	3
1 技术指标.....	1
2 功能描述.....	2
3 外围接口电路.....	3
3.1 DDR 接口.....	3
3.2 PCIE 接口.....	3
3.2.1 寄存器说明.....	3
3.2.1.1 基地址.....	3
3.2.1.2 寄存器列表.....	4
3.3 GMAC 接口.....	5
3.4 LPC 接口.....	6
3.4.1 寄存器说明.....	6
3.4.1.1 基地址.....	6
3.4.1.2 寄存器列表.....	6
3.5 SPI 接口.....	7
3.5.1 寄存器说明.....	7
3.5.1.1 基地址.....	7
3.5.1.2 寄存器列表.....	7
3.6 UART 接口.....	9
3.6.1 寄存器说明.....	10
3.6.1.1 基地址.....	10
3.6.1.2 寄存器列表.....	10
3.7 I2C 接口.....	22
3.7.1 寄存器说明.....	22
3.7.1.1 基地址.....	22
3.7.1.2 寄存器列表.....	22
3.8 GPIO 接口.....	34
3.8.1 GPIO 复用说明.....	34
3.8.1.1 GPIO 复用寄存器地址.....	34
3.8.1.2 GPIO 复用寄存器描述.....	34
3.8.2 GPIO 寄存器说明.....	35
3.8.2.1 基地址.....	35
3.8.2.2 寄存器列表.....	36
3.9 上电时序.....	38
4 电气特性.....	40
4.1 极限工作条件.....	40
4.2 典型工作参数.....	40
5 封装数据.....	41
5.1 封装尺寸.....	41
6 装焊温度曲线.....	42
6.1 无铅焊接温度曲线中各温区的作用.....	42

6.2	有铅焊接温度曲线中各温区的作用	43
7	引脚描述	44
7.1	通用 IO 类引脚 (118 PIN)	44
7.2	GMAC 引脚 (28 PIN)	48
7.3	PCIE 引脚 (140 PIN)	49
7.4	DDR3 引脚 (312 PIN)	53
7.5	电源引脚 (552 PIN)	60

PHYTUM

1 技术指标

主要技术指标如下：

- 兼容 ARM V8 指令集
- 集成 4 个 xiaomi 处理器核 (FTC660)
- 核心时钟频率 1.5GHz (标配)
- L1 数据 Cache 32KB
- L2 Cache 2MB
- L3 Cache 8MB
- 峰值性能 24GFlops@1.5GHz
- 最大功耗 15W
- 核电压 0.9V
- 集成 2 个 DDR3 存储控制器
- DDR3 速率可达 1600Mbps，访存带宽可达 25.6GB/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。
- BGA1150 封装，0.914mm 球间距，封装尺寸 37.5mmX37.5mm
- IO 电压 1.8V，包括 UART、I2C、GMAC、GPIOA、GPIOB、GPIOC、GPIOD、LPC
- 支持电源关断、时钟关断和 DVFS
- 支持商业、工业等分级

2 功能描述

表 2-1 功能描述

硬件特性		说明
Core	兼容 ARM V8 指令集的 Xiaomi 内核 (FTC660)	4 个, 标配典型工作频率 1.5GHz。支持电源关断
存储控制器	DDR3 SDRAM 控制器	2 个控制器, 支持带 ECC 的 DDR3 DIMM, 支持 RDIMM、UDIMM、SODIMM, 电压 1.5V
外设	PCIE3.0 RootComplex	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	1 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
	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	0.9W
	存控电源	1.5V
	IO 电源	1.8V

3 外围接口电路

3.1 DDR 接口

DDR 接口支持 X8 和 X4 两种 DDR 颗粒类型，如表 3-1 所示。

表 3-1 DDR 接口支持的内存形式

颗粒芯片位宽	芯片数目
X8	8
X4	16

注：该表芯片数目中，不包含 ECC 内存颗粒

DDR 接口支持 1.5V DDR3 的 RDIMM、UDIMM、SODIMM 等内存条类型。

3.2 PCIE 接口

PCIE 接口支持 PCIE3.0 规范，兼容 2.0、1.0 规范，其特点如下：

- 包含 2 个 x16 rootcomplex 接口，PCIE 0 和 PCIE 1；
- 每个 x16 rootcomplex 接口都可以拆分为两个 x8 接口；
- 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。PCIE0[8:15]、PCIE1[8:15] 不受此约束限制，用户可根据需要进行配置。

3.2.1 寄存器说明

3.2.1.1 基地址

表 3-2 PCIE 基地址

名称	基地址
PCIE	0x2810_0000

3.2.1.2 寄存器列表

表 3-3 寄存器描述

偏移地址	名称	读写类型	说明
0x0E00	REG_PEU_BIF_MODE	RW	拆分模式寄存器
0x0E08	REG_PEU_TX_LANE_FLIP_EN	RW	链路反转设置寄存器

3.2.1.2.1 拆分模式寄存器(0x0E00)

表 3-4 拆分模式寄存器

位	名称	读写方式	初值	说明
[3:2]	PCIE1_bif_mode	RW	0x1	PCIE 1 的分拆模式设置: •00 - 全部关闭 •01 - 一个 16x •10 - 两个 8x
[1:0]	PCIE0_bif_mode	RW	0x1	PCIE 0 的分拆模式设置: •00 - 全部关闭 •01 - 一个 16x •10 - 两个 8x

3.2.1.2.2 链路反转设置寄存器(0x0E08)

表 3-5 链路反转设置寄存器

位	名称	读写方式	初值	说明
[15]	pcie1_c3_tx_lane_flip_en	RW	0x 0	
[14]	pcie1_c2_tx_lane_flip_en	RW	0x 0	
[13]	pcie1_c1_tx_lane_flip_en	RW	0x 0	
[12]	pcie1_c0_tx_lane_flip_en	RW	0x 0	
[11]	pcie1_c3_rx_lane_flip_en	RW	0x 0	
[10]	pcie1_c2_rx_lane_flip_en	RW	0x 0	
[9]	pcie1_c1_rx_lane_flip_en	RW	0x 0	
[8]	pcie1_c0_rx_lane_flip_en	RW	0x 0	
[7]	pcie0_c3_tx_lane_flip_en	RW	0x 0	
[6]	pcie0_c2_tx_lane_flip_en	RW	0x 0	
[5]	pcie0_c1_tx_lane_flip_en	RW	0x 0	
[4]	pcie0_c0_tx_lane_flip_en	RW	0x 0	
[3]	pcie0_c3_rx_lane_flip_en	RW	0x 0	
[2]	pcie0_c2_rx_lane_flip_en	RW	0x 0	
[1]	pcie0_c1_rx_lane_flip_en	RW	0x 0	
[0]	pcie0_c0_rx_lane_flip_en	RW	0x 0	

为 1 表示手动设置 lane 反转

3.3 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 GMII 和 MII 的 PHY 接口。

GMAC 接口特点：

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

芯片包含一路 GMAC 接口，GMAC 接口信号主要分三类：时钟接口、数据接口、管理接口。接口信号列表见 GMAC 引脚。

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 在不使用情况下接地。
- 如果没有使用 GMAC，建议 125MHz 外部时钟输入引脚 GMAC_OSC 接地。
- 时钟需求：GMAC 控制器有 1 个外部时钟输入引脚 GMU_CLK_OSC，频率为 125MHz，需外部提供。
- 外围接口电路与下列引脚连接的信号线，有等长需求，如表 3-5 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 3-5 GMAC

组	I/O PAD
1	G1_CLK_GTX G1_TX0~7 G1_TXEN G1_TXER
2	G1_CLK_RX G1_COL G1_CRS G1_RX0~7 G1_RXDV G1_RXER
3	G1_MDIO G1_MDC

3.4 LPC 接口

芯片集成了一个 LPC 控制器，LPC (Low Pin Count) 是 Intel 公司提出的一种低数量管脚需求的外设接口协议，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。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。因此建议使用四字节访问，详情请参考《LPC 兼容规范》

3.4.1 寄存器说明

3.4.1.1 基地址

表 3-6 LPC 基地址

名称	基地址
LPC	0x27FF_0000

3.4.1.2 寄存器列表

表 3-7 LPC 寄存器说明

偏移地址	名称	属性	默认值	含义
0xFFE8	串行中断配置寄存器[31:0]	RW	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 的寄存器时，请确保 LPC 的外部电路连接正常，否则读写 LPC 寄存器会失败。

3.4.1.2.1 串行中断配置寄存器(0xFFE8)

表 3-8 串行中断配置寄存器

位	名称	读写方式	初值	说明
[31:0]	nu_serirq_config	RW	0x8000_0000	<p>bit31 - 针对读数据每次读 4 bytes 数据使能标志(1'b1, 读 1byte);</p> <p>bit4~3 - 支持的串行中断设备数量(2'b01 代表 32, 否则 16。默认 16);</p> <p>bit2 - 串行中断模式配置默认连续模式(默认为连续模式);</p>

				bit1~0 - 起始周期配置(2'b11, 8clk; 2'b10, 6clk; 否则 4。默认 4 clk);
--	--	--	--	--

3.5 SPI 接口

SPI 接口仅支持连接用于存储 BIOS 用的 FLASH 芯片。

3.5.1 寄存器说明

3.5.1.1 基地址

表 3-9 SPI 基地址

名称	基址
SPI	0x1FFF_FF00

3.5.1.2 寄存器列表

表 3-10 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

3.5.1.2.1 配置寄存器 (0x00)

表 3-11 配置寄存器

位	名称	读写方式	初值	说明
[7:6]	cmd_mode	RW	0x0	设置读写数据空间时所发出的 SPI 命令 •00 - 发 read(03) 和 pp(02); •01 - 发 4read(13) 和 4pp(12);

				<ul style="list-style-type: none"> • 10 - 发 fast_read(0b) 和 pp(02); • 11 - 发 4fast_read(0c) 和 4pp(12);
[5]	addr_mode	RW	0x0	<p>设置发出 read、fast_read 和 pp 命令时的地址模式</p> <ul style="list-style-type: none"> • 0 - 3 字节地址; • 1 - 4 字节地址;
[4]	dummy	RW	0x0	<p>设置发出 fast_read 和 4fast_read 时是否发 dummy 字节</p> <ul style="list-style-type: none"> • 0 - 不发; • 1 - 发;
[3]	wr_mode	RW	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	RW	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; • 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。

3.5.1.2.2 容量寄存器 (0x14)

表 3-12 容量寄存器

位	名称	读写方式	初值	说明
[2:0]	flash_capacity	RW	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

3.5.1.2.3 写缓冲寄存器 (0x18)

表 3-13 写缓冲寄存器

位	名称	读写方式	初值	说明
[0]	flush	RW	0x0	写 1 将产生 flush 操作

3.5.1.2.4 命令端口寄存器 (0x20)

表 3-144 命令端口寄存器

位	名称	读写方式	初值	说明
[17:16]	flash_sel	RW	0x0	选择要操作的 flash

[15:8]	command	RW	0x0	要发出的 SPI 命令
[7]	reserved	RW	0x0	保留
[6]	addr_mode0	RW	0x0	是否带地址 •0 - 不带地址 •1 - 带地址
[5]	addr_mode1	RW	0x0	地址模式 •0 - 3 字节地址 •1 - 4 字节地址
[4]	dummy	RW	0x0	是否发 dummy 字节 •0 - 不发 •1 - 发
[3:0]	rw_num	RW	0x0	读写字节数目, 有效值为 0, 1, 2, 4, 8

3.5.1.2.5 地址端口寄存器 (0x24)

表 3-15 地址端口寄存器

位	名称	读写方式	初值	说明
[31:0]	addr	RW	0x0	地址

3.5.1.2.6 高位寄存器 (0x28)

表 3-165 高位寄存器

位	名称	读写方式	初值	说明
[31:0]	data	RW	0x0	高 4 字节数据

3.5.1.2.7 低位寄存器 (0x2C)

表 3-17 低位寄存器

位	名称	读写方式	初值	说明
[31:0]	data	RW	0x0	低 4 字节数据

3.6 UART 接口

UART (Universal Asynchronous Receiver/Transmitter), 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 线接口 UART0 和 1 个 3 线制接口 UART1, 兼容 16550 标准。

3.6.1 寄存器说明

3.6.1.1 基地址

表 3-18 UART 基地址

名称	基地址
UART0	0x2800_0000
UART1	0x2800_1000

3.6.1.2 寄存器列表

表 3-6 UART 寄存器说明

偏移地址	名称	读写类型	初值	说明
0x00	RBR	R	0x0	接收缓存寄存器 LCR[7] bit = 0
	THR	W	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	R	0x01	中断识别寄存器
	FCR	W	0x0	先进先出控制寄存器
0x0C	LCR	R/W	0x0	行控制寄存器
0x14	LSR	R	0x60	行状态寄存器
0x7C	USR	R	0x6	串口状态寄存器

3.6.1.2.1 RBR(0x00)

表 3-7 RBR

位	名称	读写方式	初值	说明
[31:8]	保留	RO	0x0	保留
[7]	Receive Buffer Register	RO	0x0	UART 模式下串口接收端的数据 buffer。 当 Line Status Register (LCR) 数据就绪位 (DR) 被设置之后该寄存器上的数据才有效。 在 non-FIFO (FIFO_MODE=NONE) 模式或

				<p>者 FIFOs 被禁用（FCR[0]设置为 0）的情况下，RBR 中的数据必须在下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>
--	--	--	--	---

3.6.1.2.2 THR(0x00)

表 3-8 THR

位	名称	读写方式	初值	说明
[31:8]	保留	RW	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）取决于你所配置的 FIFO Depth 的值。当 FIFO 满之后，任何写入都无效。</p>

3.6.1.2.3 DLH(0x04)

表 3-9 DLH

位	名称	读写方式	初值	说明
[31:8]	保留	RW	0x0	保留
[7]	Divisor Latch (High)	RW	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) 的时候这个寄存才能被器访问; 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 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>

3.6.1.2.4 DLL(0x00)

表 3-10 DLL

位	名称	读写方式	初值	说明
[31:8]	保留	RW	0x0	保留
[7]	Divisor Latch (Low)	RW	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) 的时候这个寄存才能被器访问; 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:</p>

				波特率=系统时钟/16*波特率除数。 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
--	--	--	--	---

3.6.1.2.5 IER(0x04)

表 3-11 IER

位	名称	读写方式	初值	说明
[31:8]	保留	RO	0x0	保留
[7]	PTIME	RW	0x0	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。 •0 - 禁用 •1 - 启用
[6:4]	保留	RO	0x0	
[3]	EDSSI	RW	0x0	启用 Modem 状态中断。这位用来启用/禁用 Modem 状态中断的产生。这个中断优先级为 4。 •0 - 禁用 •1 - 启用
[2]	ELSI	RW	0x0	启用 Receiver Line 状态中断。这位用来启用/禁用 Receiver Line 状态中断的产生。这个中断的优先级最高。 •0 - 禁用 •1 - 启用
[1]	ETBEI	R/W	0x0	启用 Transmit Holding Register Empty Interrupt。这位用来启用/禁用 Transmitter Holding Register Empty Interrupt 的产生。这个中断的优先级为 3。 •0 - 禁用 •1 - 启用
[0]	ERBFI	R/W	0x0	启用接收数据可用中断。这位用来启用/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。这个中断的优先级位 2

				•0 - 禁用 •1 - 启用
--	--	--	--	--------------------

3.6.1.2.6 IIR(0x08)

表 3-12 IIR

位	名称	读写方式	初值	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。这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 时这个中断才起作用。

3.6.1.2.7 LCR(0x0C)

表 3-13 LCR

位	名称	读写方式	初值	说明
[31:8]	保留	RO	0x0	保留
[7]	DLAB	RW	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)	RW	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 (保留)	RO	0x0	保留
[4]	EPS	RW	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	RW	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使能或者禁用产生奇偶校验，在接收串行字符时该位被用来使能或者禁用奇偶检查。
[2]	STOP	RW	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，接收端只检测第一个停止位。

				位。 • 0 – 1 stop bit • 1 – 1.5 停止位当 DLS(LCR[1:0]等于 0), 否则 2 停止位 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。
[1:0]	DLS (or CLS, as used in legacy)	RW	0x0	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: • 00 – 5 bits • 01 – 6 bits • 10 – 7 bits • 11 – 8 bits

3.6.1.2.8 FCR (0x08)

位	名称	读写方式	初值	说明
[31:8]	保留	RW	0x0	保留
[7:6]	RCVR Trigger (or RT)	WO	0x0	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: • 00 – 1 character in the FIFO • 01 – FIFO ¼ full • 10 – FIFO ½ full • 11 – FIFO 2 less than full
[5:4]	TX Empty Trigger (or TET)	WO	0x0	TX 空触发, 当 THRE_MODE_USER == 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:

				<ul style="list-style-type: none"> •00 - FIFO empty •01 - 2 characters in the FIFO •10 - FIFO $\frac{1}{4}$ full •11 - FIFO $\frac{1}{2}$ full
[3]	DMA Mode (or DMAM)	WO	0x0	<p>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。</p> <ul style="list-style-type: none"> •0 - 模式 0 •1 - 模式 1
[2]	XMIT FIFO Reset (or XFIFOR)	WO	0x0	<p>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。</p>
[1]	RCVR FIFO Reset (or RFIFOR)	WO	0x0	<p>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。</p>
[0]	FIFO Enable (or FIFOE)	WO	0x0	<p>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。</p>

3.6.1.2.9 LSR(0X14)

表 3-27 LSR

位	名称	读写方式	初值	说明
[31:8]	Reserved	RO	0x0	保留
[7]	RFE	RO	0x0	<p>接收 FIFO 错误位。 仅当 FFIFO_MODE!=NONE 并且 FIFO 使能时有效 (FCR [0]设置为 1)。 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 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。
[6]	TEMT	RO	0x1	发送器空位。 仅当 FIFO_MODE!=NONE 并且 FIFO 使能时有效 (FCR [0]设置为 1)。 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。
[5]	THRE	RO	0x1	发送保持寄存器空标志位。 如果 THRE_MODE_USER == Disabled 或禁用 THRE 模式 (不论 FIFO 的是否正在执行还是与否启用)，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。
[4]	BI	RO	0x0	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，该位置位。 如果红外模式 (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，该位置位。 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 的顶端时，该位被置位。 读 LSR 将清除 BI 位。 在 non-FIFO 模式下，BI 位一直置位，直到 LSD 被读取。
[3]	FE	RO	0x0	<p>帧错误位。</p> <p>这是用来表示在接收器中发生一个帧错误的。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，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。 这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。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。 这是用来指示在 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>

3.6.1.2.10 USR(0x7C)

表 3-28 USR

位	名称	读写方式	初值	说明
[31:5]	保留	RO	0x0	保留
[4]	RFF	RO	0x0	<p>指示接收 FIFO 满。该位只有当 FIFO_STAT== YES 时有效。这是用来指示该接收 FIFO 全满。</p> <ul style="list-style-type: none"> •0 - 接收 FIFO 未满 •1 - 接收 FIFO 满 <p>当 RX FIFO 未满该位被清零。</p>
[3]	RFNE	RO	0x0	<p>指示接收 FIFO 不为空。该位只有当 FIFO_STAT== YES 时有效。这是用来指示接收 FIFO 包含一个或多个条目。</p> <ul style="list-style-type: none"> •0 - 接收 FIFO 为空

				•1 - 接收 FIFO 不为空 当 RX FIFO 为空时该位被清零。
[2]	TFE	RO	0x1	指示发送 FIFO 空。该位只有当 FIFO_STAT== YES 有效。这是用来指示发送 FIFO 完全是空的。 •0 - 发送 FIFO 不为空 •1 - 发送 FIFO 空 当 TX FIFO 不再是空，该位被清零。
[1]	TFNF	RO	0x1	指示发送 FIFO 未满。该位只有当 FIFO_STAT== YES 有效。这是用来指示在未满发送 FIFO。 •0 - 发送 FIFO 满 •1 - 发送 FIFO 未满 当 TX FIFO 满时该位被清零。
[0]	BUSY	RO	0x0	指示 UART 忙。 该位只有当 UART_16550_COMPATIBLE == NO 时有效，指示串口正在进行传输；该位清 0 时表明 DW_apb_uart 空闲或处于 idle 状态。 •0 - DW_apb_uart 空闲或处于 idle 状态。 •1 - DW_apb_uart 忙（主动传输数据）。 该位将在以下任意条件发生时设置为 1（忙）： 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。 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。
--	--	--	--	--

3.7 I2C 接口

芯片包含 2 个 I2C 接口，分别为 I2C 0 和 I2C 1。I2C (Inter-Integrated Circuit) 总线是由 PHILIPS 公司开发的两线式串行总线，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。

3.7.1 寄存器说明

3.7.1.1 基地址

表 3-29 I2C 基地址

名称	基地址
I2C 0	0x2800_2000
I2C 1	0x2800_3000

3.7.1.2 寄存器列表

表 3-30 I2C 寄存器列表

偏移地址	名字	读写类型	初值	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_C MD	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_	R/W	0x1D6	I2C 标准速率时钟计数器，低 32 位

	LCNT				
0x1C	IC_FS_SCL_HCNT	R/W	0x3C	I2C 快速速率时钟计数器, 高 32 位	
0x20	IC_FS_SCL_LCNT	R/W	0x82	I2C 快速速率时钟计数器, 低 32 位	
0x24	IC_HS_SCL_HCNT	R/W	0x6	I2C 高速速率时钟计数器, 高 32 位	
0x28	IC_HS_SCL_LCNT	R/W	0x10	I2C 高速速率时钟计数器, 低 32 位	
0x30	IC_INTR_MASK	R/W	0x8FF	I2C 中断 mask 寄存器	
0x34	IC_RAW_INTR_STAT	R	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	R	0x6	I2C 状态寄存器	

3.7.1.2.1 IC_CON(0x00)

表 3-31 IC_CON(0x00) 寄存器说明

位	名称	读写类型	初值	说明
[15:7]	-	-	-	-
[6]	IC_SLAVE_DISABLE	RW	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。</p> <p>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	RW	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 •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，</p>

				IC_RAW_INTR_STAT 寄存器中的 TX_BART 标志会被置起。
[4]	C_10BITADDR_MASTER or C_10BITADDR_rd_only	RW	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，读写类型为只读状态，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_10BITADDR_SLAVE	RW	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	RW	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_MODE	RW	IC_MASTER_MODE 配置参数	<p>此位是 I2C Master 的使能位。</p> <ul style="list-style-type: none"> •0 - 关闭 master 功能 •1 - 使能 master 功能

3.7.1.2.2 IC_TAR(0x04)

表 3-32 IC_TAR(0x04) 寄存器说明

位	名称	读取类型	初值	说明
[15:13]	-	-	-	-
[12]	IC_10BITADDR_MASTER	RW	IC_10BITADDR_MASTER 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	RW	0x0	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

				•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_START	RW	0x0	如果位 11 (SPECIAL) 为 1，此位设定 DW_apb_i2c 使用广播呼叫地址格式还是 START BYTE 格式。 •0 - 使用广播呼叫地址格式。 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。 •1 - START BYTE 格式
[9:0]	IC_TAR	RW	IC_DEFAULT_TAR_SLAVE_A DDR 配置参数	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。

3.7.1.2.3 IC_SAR(0x08)

表 3-33 IC_SAR(0x08) 寄存器说明

位	名称	读写类型	初值	说明
[15:10]	-	-	-	-
[9:0]	IC_SAR	RW	IC_DEFAULT_SLAVE_ADDR 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。

3.7.1.2.4 IC_DATA_CMD(0x10)

表 3-34 IC_DATA_CMD(0x10) 寄存器说明

位	名称	读取类型	初值	说明
[15:11]	-	-	-	-
[10]	RESTART	WO	-	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。 •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	-	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。

				<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 Slave 发送模式时，CMD=0 表示 IC_DATA_CMD 中的数据将被发送。</p> <p>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。</p>
[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。

3.7.1.2.5 IC_SS_SCL_HCNT(0x14)

表 3-35 IC_SS_SCL_HCNT (0X14) 寄存器

位	名称	读取类型	初值	说明
[15:0]	IC_SS_SC L_HCNT	RW	IC_SS_SCL_H IGH_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 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。</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 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</p>

器只读。

3.7.1.2.6 IC_SS_SCL_LCNT(0x18)

表 3-36 IC_SS_SCL_LCNT(0X18) 寄存器

位	名称	读取类型	初值	说明
[15:0]	IC_SS_SCL_LCNT	RW	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计，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（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 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>

3.7.1.2.7 IC_FS_SCL_HCNT(0x1C)

表 3-37 IC_FS_SCL_HCNT(0X1C) 寄存器

位	名称	读写类型	初值	说明
[15:0]	IC_FS_SCL_HCNT	RW	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，用于明确正确的 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，此寄存器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>

器只读。

3.7.1.2.8 IC_FS_SCL_LCNT(0x20)

表 3-38 IC_FS_SCL_LCNT(0X20) 寄存器

位	名称	读写类型	初值	说明
[15:0]	IC_FS_SCL_LCNT	RW	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，用于明确正确的 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，此寄存器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>

3.7.1.2.9 IC_HS_SCL_HCNT(0x24)

表 3-39 IC_HS_SCL_HCNT(0X24) 寄存器

位	名称	读写类型	初值	说明
[15:0]	IC_HS_SCL_HCNT	RW	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，用于明确正确的 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 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</p>

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

3.7.1.2.10 IC_HS_SCL_LCNT(0x28)

表 3-40 IC_HS_SCL_LCNT(0X28) 寄存器

位	名称	读写类型	初值	说明
[15:0]	IC_HS_SCL_LCNT	RW	IC_HS_SCL_LOW_COUNT 配置参数	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。 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。IC_MAX_SPEED_MODE!= high 时, 此寄存器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。 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)。 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。 当 IC_HC_COUNT_VALUES 为 1 时, 该寄存器只读。

3.7.1.2.11 IC_INTR_MASK(0x30)

表 3-41 IC_INTR_MASK(0X30) 寄存器

Bit	Name	Access	Reset value	Description
[15:12]	-	-	-	-
[11]	M_GEN_CALL	RW	0x1	M_GEN_CALL 中断事件标志屏蔽控制。置位时, 如果对应中断事件发生, 不会置位 IC_INTR_STAT 寄存器中对应的中断标志位
[10]	M_START_DET	RW	0x0	
[9]	M_STOP_DET	RW	0x0	
[8]	M_ACTIVITY	RW	0x0	

[7]	M_RX_DONE	RW	0x1
[6]	M_TX_ABRT	RW	0x1
[5]	M_RD_REQ	RW	0x1
[4]	M_TX_EMPTY	RW	0x1
[3]	M_TX_OVER	RW	0x1
[2]	M_RX_FULL	RW	0x1
[1]	M_RX_OVER	RW	0x1
[0]	M_RX_UNDER	RW	0x1

3.7.1.2.12 IC_RAW_INTR_STAT(0x34)

表 3-42 IC_RAW_INTR_STAT(0x34) 寄存器说明

位	名称	读写类型	初值	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	此位标志 I2C 控制器的活动状态。 有 4 种方法可以清楚 ACTIVITY 标志： • 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 一旦被置位则会一致保持置位，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。
[7]	RX_DONE	RO	0x0	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。
[6]	TX_ABRT	RO	0x0	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。 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。
[5]	RD_REQ	RO	0x0	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。
[4]	TX_EMPTY	RO	0x0	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。
[3]	TX_OVER	RO	0x0	在发送过程中，如果发送缓冲区大小达到 IC_RX_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 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，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。

3.7.1.2.13 IC_RX_TL(0x38)

表 3-43 IC_RX_TL(0x38) 寄存器

位	名称	读写类型	初值	说明
[15:8]	-	-	-	-
[7:0]	RX_TL	RW	IC_RX_TL 配	接收缓冲区满中断 (RX_FULL) 触发门限

			置参数	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。
--	--	--	-----	--

3.7.1.2.14 IC_TX_TL(0x3C)

表 3-44 IC_TX_TL (0X3C) 寄存器

位	名称	读写类型	初值	说明
[15:8]	-	-	-	-
[7:0]	TX_TL	RW	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。

3.7.1.2.15 IC_ENABLE(0x6C)

表 3-45 IC_ENABLE (0X6C) 寄存器

位	名称	读写类型	初值	说明
[15:1]	-	-	-	-
[0]	ENABLE	RW	0x0	I2C 控制器使能或关闭控制位。 •0 - 关闭 I2C 控制器功能 •1 - 使能 I2C 控制器功能 以下现象会在 I2C 控制器功能关闭时出现： •TxFIFO 和 RxFIFO 被刷新 •IC_INTR_STAT 寄存器中的状态保持不变。 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。 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 和 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。
--	--	--	--	-----

3.7.1.2.16 IC_STATUS(0x70)

表 3-46 IC_STATUS(0X70)寄存器

位	名称	读写类型	初值	说明
[31:7]	-	-	-	-
[6]	SLV_ACTIVITY	RO	0x0	Slave FSM 活动状态标志。Slave FSM(Slave Finite State Machine) 不在 Idle 状态时被置位 •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。 •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。 •0 - 接收 FIFO 未满 •1 - s 接收 FIFO 全满
[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。 •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。 •0 - 发送 FIFO 不为空 •1 - 发送 FIFO 为空
[1]	TFNF	RO	0x1	发送 FIFO 未满标志。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。 •0 - 发送 FIFO 已满 •1 - 发送 FIFO 未满

[0]	ACTIVITY	RO	0x0	I2C 控制器活动状态标志
-----	----------	----	-----	---------------

3.8 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3-49。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。

3.8.1 GPIO 复用说明

3.8.1.1 GPIO 复用寄存器地址

表 3-47 GPIO 复用寄存器地址

名称	基地址
GPIO 复用控制寄存器	0x2810_0C00

3.8.1.2 GPIO 复用寄存器描述

位	名称	读写类型	初值	说明
[5:0]	GPIO 复用控制位	RW	0x10	每一位控制一组GPIO端口的复用： [0]位控制 cru_sel_uart_func, 1 选择此功能, 0 选择 GPIO 功能。 [1]位控制 cru_sel_lpc_func0, 1 选择此功能, 0 选择 GPIO 功能。 [2]位控制 cru_sel_lpc_func1, 1 选择此功能, 0 选择 GPIO 功能。 [3]位控制 cru_sel_spics, 1 选择此功能, 0 选择 GPIO 功能。 [4]位控制 cru_sel_RST_state, 1 选择此功能, 0 选择 GPIO 功能。 [5]位控制 GMAC 配置信号 cru_gmu_cfg_cttw, 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端口说明表3-49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 3-49 GPIO 复用寄存器说明

控制线	GPIO	复用功能
	cru_sel_xxx=0	cru_sel_xxx=1

CRU_SEL_UART_FUNC	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_FUNC0	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_FUNC1	PortB[7]	LPC.ldrq_n
CRU_SEL_RST_STATE	PortA[0]	clk_ovb_sel[0]
	PortA[1]	clk_ovb_sel[1]
	PortA[2]	clk_ovb_sel[2]
	PortA[3]	clk_ovb_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]
CRU_SEL_SPICS	PortB[0]	spi_csn[2]
CRU_SEL_LINKUP1	PortB[1]	spi_csn[3]
CRU_SEL_LINKUP2	PortB[2]	peu_linkup[1]
CRU_SEL_LINKUP3	PortB[3]	peu_linkup[2]
	PortB[4]	peu_linkup[3]

3.8.2 GPIO 寄存器说明

3.8.2.1 基地址

表 3-50 GPIO 基地址

名称	基地址
GPIO	0x2800_6000

3.8.2.2 寄存器列表

表 3-51 GPIO 内部寄存器描述

偏移地址	名字	读写类型	说明
0x00	GPIOA_DR	R/W	端口 A 数据寄存器 位宽: GPIO_PWIDTH_A 初值: GPIO_SWPORTA_RESET
0x04	GPIOA_DDR	R/W	端口 A 数据方向寄存器 位宽: GPIO_PWIDTH_A 初值: 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 初值: 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 初值: 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 初值: GPIO_SWPORTC_RESET
0x1c	GPIOC_DDR	R/W	端口 C 数据方向寄存器 位宽: GPIO_PWIDTH_C 初值: 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 初值: GPIO_SWPORTD_RESET
0x28	GPIOD_DDR	R/W	端口 D 数据方向寄存器 位宽: GPIO_PWIDTH_D 初值: GPIO_DFLT_DIR_D (for all bits)
0x50	GPIO_EXT_PO RTA	R	端口 A 外部端口寄存器 位宽: GPIO_PWIDTH_A 初值: 0x0
0x54	GPIO_EXT_PO RTB	R	端口 B 外部端口寄存器 位宽: GPIO_PWIDTH_B 初值: 0x0
0x58	GPIO_EXT_PO RTC	R	端口 C 外部端口寄存器 位宽: GPIO_PWIDTH_C 初值: 0x0
0x5c	GPIO_EXT_PO RTD	R	端口 D 外部端口寄存器 位宽: GPIO_PWIDTH_D 初值: 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; GPIO_SWPORTX_RESET=??; GPIO_DFLT_DIR_X=??

其中 X 表示 A/B/C/D。本章内名称以此类推。

3.8.2.2.1 数据寄存器 GPIOX_DR (0x00)

表 3-52 数据寄存器 (GPIOX_DR)

位	名称	读写类型	初值	说明
[31:GPIO_PWIDTH_X]	保留	RW	-	
[GPIO_PWIDTH_X - 1:0]	端口 X 数据寄存器	RW	GPIO_SWPORT_X_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。

3.8.2.2.2 方向寄存器 GPIOX_DDR (0x04)

表 3-53 方向寄存器 (GPIOX_DDR)

位	名称	读写类型	初值	说明
[31:GPIO_PWIDTH_X]	-		-	
[GPIO_PWIDTH_X - 1:0]	端口 X 方向寄存器	RW	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。 •0 - 输入(默认) •1 - 输出

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

表 3-54 外部数据寄存器 (GPIO_EXT_PORTX)

位	名称	读写类型	初值	说明
[31:GPIO_PWIDTH_X]	-	-	-	-
[GPIO_PWIDTH_X - 1:0]	外部端口 X	RO	0x0	当端口 X 设置为输入时，从这个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。

3.9 上电时序

表 3-55 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。

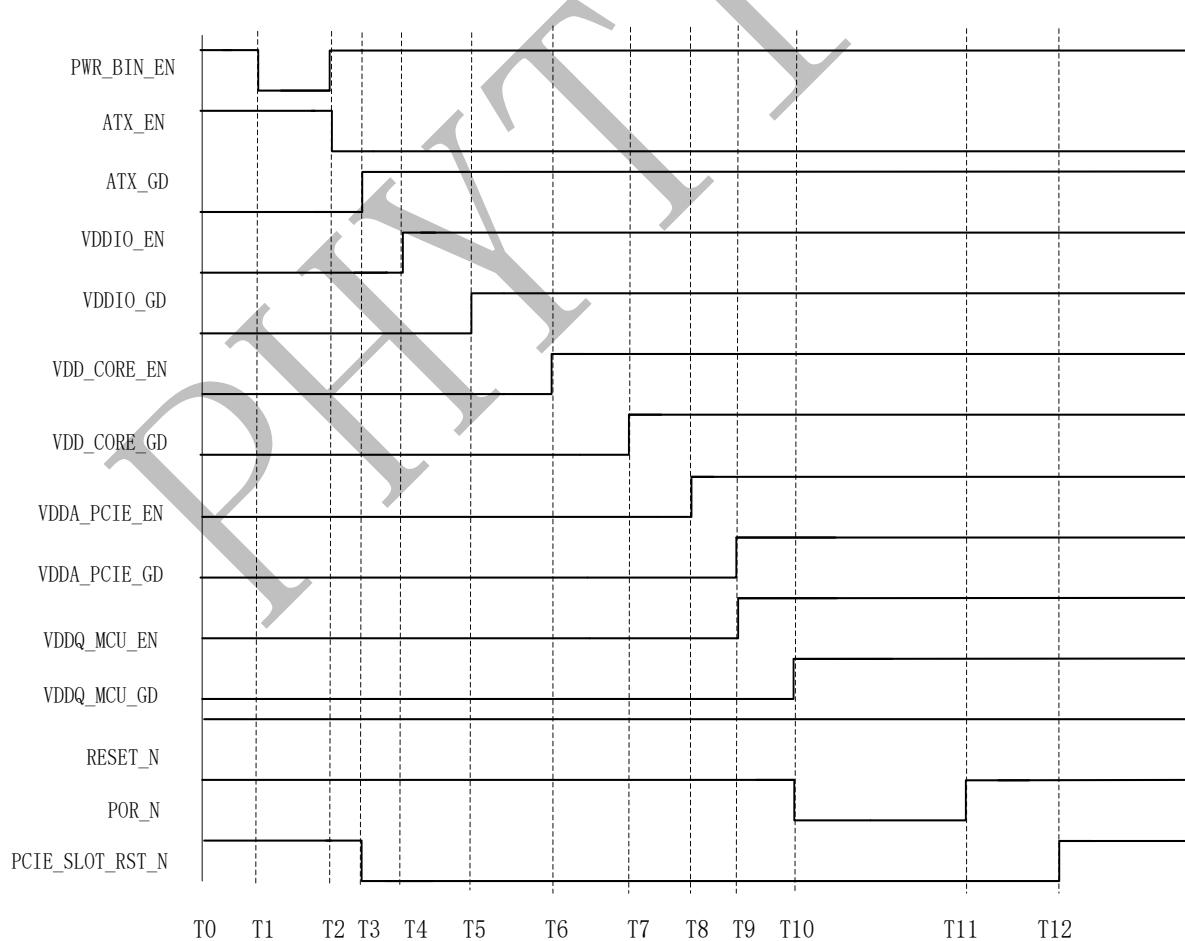


图 3-1 上电复位参考时序图

时序说明：

表 3-56 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

4 电气特性

4.1 极限工作条件

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

4.2 典型工作参数

表 4-1 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 MCUxx			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_PLLx			1.8		V
高电平输入电压	VIH		1.26			V
低电平输入电压	VIL				0.54	V
内核电源电流	IVDD_CORE	1.5GHz, 0.9V		20		A
MCU 电源电流	IVDDQ MCUxx	800MHz, 1.5V		3		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_PCIE	8Gbps, 1.5V		2		A
IO 电源电流	IVDDIO			1		A
锁相环电源电流	IVDDHV_PLLx			0.1		A
高电平输出电流	IOH		4.4	9.6	33	mA
低电平输出电流	IOL		-5.5	-9.1	-25	mA

a) 内核电压 (VDD_090)	$0.9 \times (1 \pm 5\%) V$
b) 0.75V 参考电压 (VDD_075)	$0.75 \times (1 \pm 5\%) V$
c) 0.95V IO 电压 (VDD_095)	$0.95 \times (1 \pm 5\%) V$
d) 1.5V IO 电压 (VDD_15)	$1.5 \times (1 \pm 5\%) V$
e) 1.8V IO 电压 (VDD_18)	$1.8 \times (1 \pm 5\%) V$

5 封装数据

5.1 封装尺寸

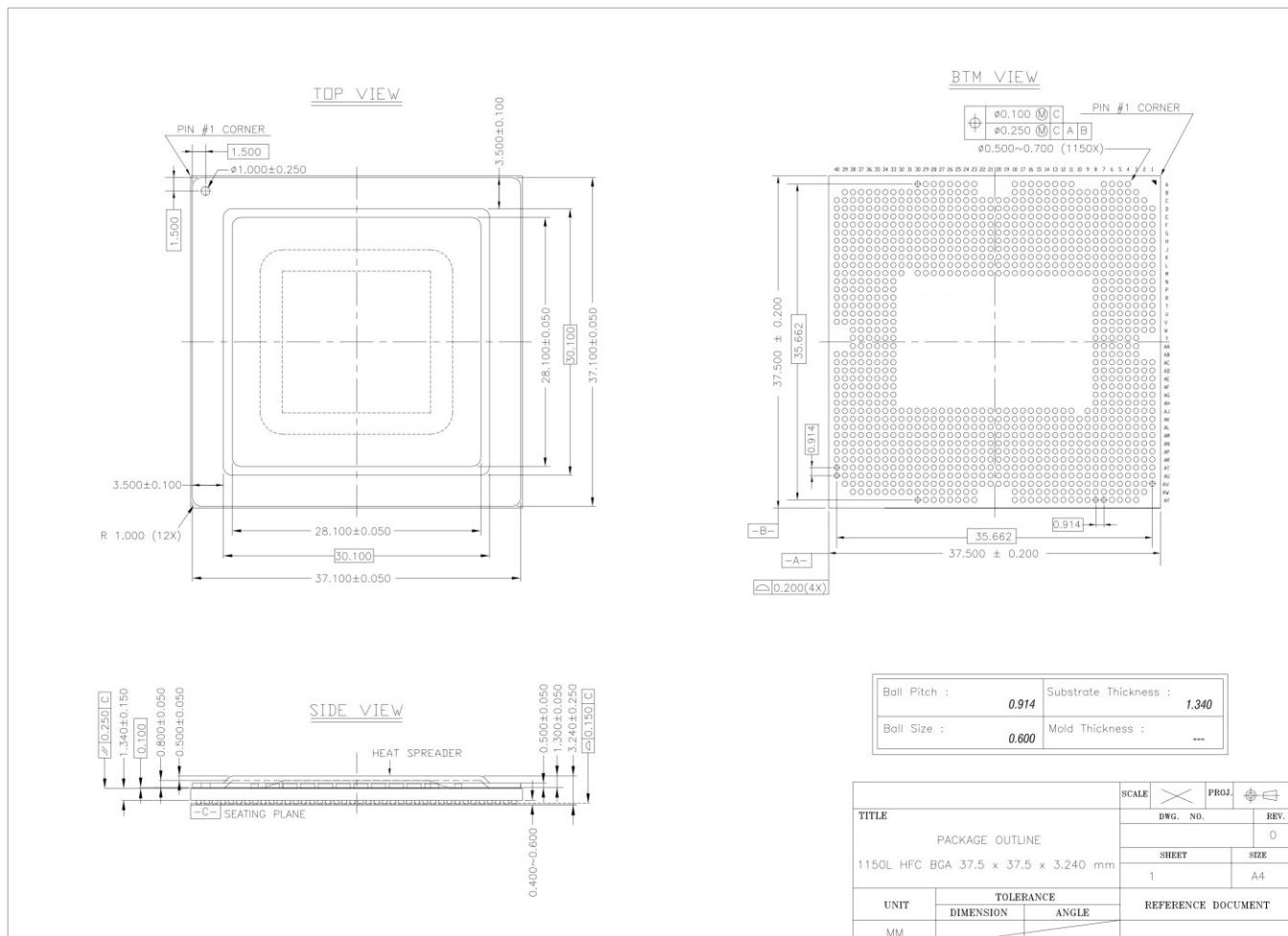


圖 5-1 CPU 封裝尺寸

6 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，焊接时也必须采用无铅焊膏。

6.1 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 6-1 所示的适用于无铅焊料的温度曲线。

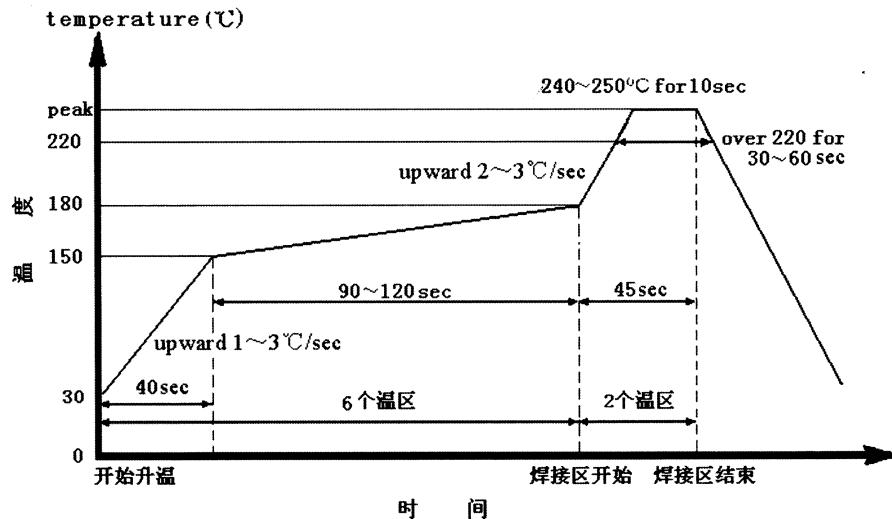


图 6-1 无铅回流焊接曲线

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℃。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℃/Sec 的速率连续上升，最理想为接近 2℃/Sec，温度升得太快，会对元件造成热冲击而导致 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℃ 上升到 180℃，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℃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℃上升到无铅焊料的熔点217℃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℃至峰值温度再回到220℃，升温速率2~3℃/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℃。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，会使升温速率超过2~3℃/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℃/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。

6.2 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。

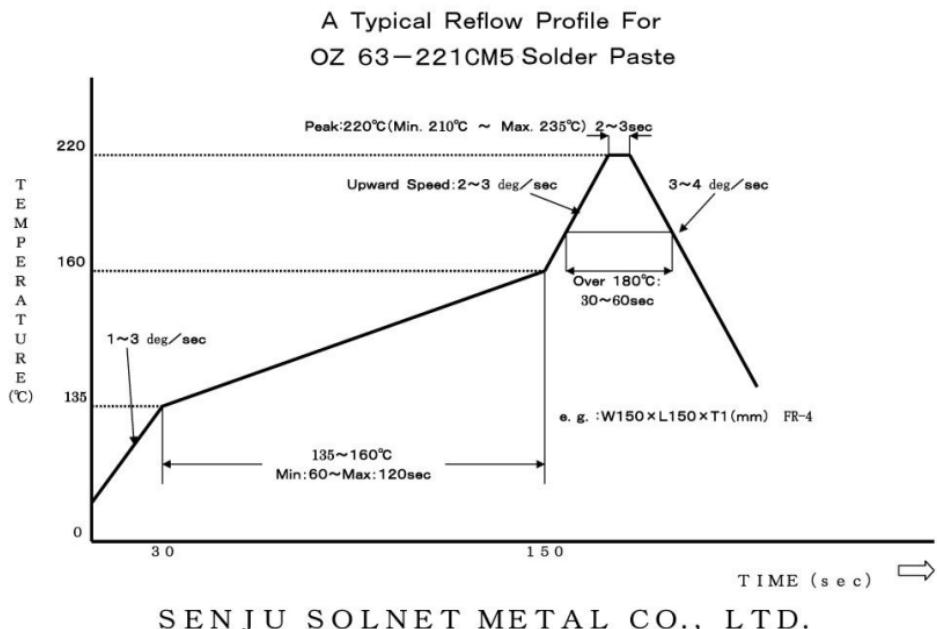


图 6-1 温度曲线

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

7 引脚描述

7.1 通用 IO 类引脚 (118 PIN)

表 7-1 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	AK13	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	AL12	O		1' b1:通知片外 CPLD, CPU 可以关电	
CLK_REF	AN33	I	Down	参考时钟, 默认 50MHz	
RESET_N	AL13	I	Up	热复位信号, 低有效	
POR_N	AK12	I	Up	上电复位信号, 低有效	
CRU_DTI_RST_OK	AJ34	O		复位完成信号	
CRU_I2C_SDA	AP32	B	Up	带外诊断专用 I2C 接口	
CRU_I2C_SCL	AL25	I	Up	带外诊断专用 I2C 接口	
ERROR_INT	AL11	O		错误中断输出, 用于带外诊断	
MCU0_I2C_SCL	AK9			MCU0 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU0_I2C_SDA	AL9			MCU0 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU1_I2C_SCL	AP34			MCU1 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
MCU1_I2C_SDA	AN34			MCU1 的 I2C 接口信号, 用于读取 DIMM 的 SPD	
UART0_SIN	AL7	I	Up	串口 0 数据输入	
UART0_SOUT	AN7	O		串口 0 数据输出	
UART1_SIN	AE7	I	Up	串口 1 数据输入	
UART1_SOUT	AH8	O		串口 1 数据输出	
GPIOCO_UART0_CD	AJ8	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	AP8	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	AH7	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	AF7	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	AL8	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	AF8	B	Up	GPIOC5 输入输出; Ring Indicator Modem 状态 输入信号	
I2C0_SCL	AN16	B	Up	I2C 0 接口的时钟信号	

I2C0_SDA	AP16	B	Up	I2C 0 接口的数据信号	
I2C1_SCL	AN15	B	Up	I2C 1 接口的时钟信号	
I2C1_SDA	AP15	B	Up	I2C 1 接口的数据信号	
SPI_EXT_CSNO	AN14	0		SPI 接口 0 号片选	
SPI_EXT_CSN1	AP14	0		SPI 接口 1 号片选	
SPI_EXT_SCK	AN13	0		SPI 接口时钟信号	
SPI_EXT_SO	AN12	0		SPI 接口数据输出信号	
EXT_SPI_SI	AP13	I	Down	SPI 接口数据输入信号	
GPIOC6_LPC_EXT_IRQ_OU TEN	AG8	B	Up	GPIO C 第 6 位输入输出；	
GPIOC7_LPC_IRQ	AP7	B	Up	GPIO C 第 7 位输入输出；	
GPIODO_LPC_EXT_LFRAME	AP11	B	Up	GPIO D 第 0 位输入输出；	
GPIOD1_EXT_LPC_LRESET	AP12	B	Up	GPIO D 第 1 位输入输出；	
GPIOD2_EXT_LPC_LCK	AN11	B	Up	GPIO D 第 2 位输入输出；	
GPIOD3_LPC_EXT_LAD_OU TEN	AN10	B	Up	GPIO D 第 3 位输入输出；	
GPIOD4_LPC_LAD0	AP10	B	Up	GPIO D 第 4 位输入输出；	
GPIOD5_LPC_LAD1	AN9	B	Up	GPIO D 第 5 位输入输出；	
GPIOD6_LPC_LAD2	AP9	B	Up	GPIO D 第 6 位输入输出；	
GPIOD7_LPC_LAD3	AJ7	B	Up	GPIO D 第 7 位输入输出；	
GPIOB7_EXT_LPC_LDRQ	AM8	B	Up	GPIO B 第 7 位输入输出；	
GPIOAO_CLK_OBV_SEL0	AY17	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	AW18	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	AW29	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	AW30	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	AN25	B	Up	GPIO A 第 4 位输入输出；复位状态机输出信号 0	
GPIOA5_RST_FSM1	AN26	B	Up	GPIO A 第 5 位输入输出；复位状态机输出信号 1	
GPIOA6_RST_FSM2	AN27	B	Up	GPIO A 第 6 位输入输出；复位状态机输出信号 2	
GPIOA7_RST_FSM3	AN28	B	Up	GPIO A 第 7 位输入输出；复位状态机输出信号 3	
GPIOB0_SPI_EXT_CSN2	AW17	B	Up	GPIO B 第 0 位输入输出；SPI 接口 2 号片选	
GPIOB1_SPI_EXT_CSN3	AY18	B	Up	GPIO B 第 1 位输入输出；SPI 接口 3 号片选	
GPIOB2_PEU_LINKUP1	AY29	B	Up	GPIO B 第 2 位输入输出；PCIE linkup 输出信 号	
GPIOB3_PEU_LINKUP2	AY30	B	Up	GPIO B 第 3 位输入输出；PCIE linkup 输出信 号	
GPIOB4_PEU_LINKUP3	AW31	B	Up	GPIO B 第 4 位输入输出；PCIE linkup 输出信 号	

				号	
GPIOB5	AP25	B	Up	GPIO B 第 5 位输入输出;	
GPIOB6	AP26	B	Up	GPIO B 第 6 位输入输出;	
GPIOA_DDR_RST_FSM4	AP28	0		GPIO A 数据方向; 复位状态机输出信号 4	
GPIOB_DDR_CRU_LK_OBV	AP27	0		GPIO B 数据方向; CRU 观测时钟输出信号	
PCIE_LINKUP	AK11			PCIE 控制器 0 的 Linkup 信号	
RSV1	AL16			保留引脚	浮空
RSV2	AL15			保留引脚	浮空
RSV3	AL14			保留引脚	浮空
RSV4	AK15			保留引脚	浮空
RSV5	AK16			保留引脚	浮空
RSV6	AK27	I	Down	保留引脚	接地
RSV7	AL27	I	Down	保留引脚	接 VDD_IO
RSV8	AK28	I	Down	保留引脚	接地
RSV9	AL28	I	Down	保留引脚	接地
RSV10	AF34	I	Down	保留引脚	接地
RSV11	AL34	I	Down	保留引脚	接地
RSV12	AP33	I	Down	保留引脚	接地
RSV13	AN32	I	Down	保留引脚	接地
RSV14	AK30	I	Down	保留引脚	接地
RSV15	AK31	I	Down	保留引脚	接地
RSV16	AL31	I	Down	保留引脚	接地
RSV17	AK14	I	Down	保留引脚	接地
RSV18	AN8	I	Down	保留引脚	接地
RSV19	AE8	0		保留引脚	浮空
RSV20	AM7	0		保留引脚	浮空
RSV21	AK7	0		保留引脚	浮空
RSV22	AG7	0		保留引脚	浮空
RSV23	AK8	0		保留引脚	浮空
RSV24	AD34	I	Down	保留引脚	接地
RSV25	AH34	I	Down	保留引脚	接地
RSV26	AM34	I	Down	保留引脚	接地
RSV27	AM33	I	Down	保留引脚	接地
RSV28	AJ33	I	Down	保留引脚	接地
RSV29	AG34	0		保留引脚	浮空
RSV30	AL32	I	Down	保留引脚	接地
RSV31	AK10	I	Down	保留引脚	接地
RSV32	AL10	I	Down	保留引脚	接地
RSV33	AF33	I	Down	保留引脚	接地
RSV34	AP30	I	Up	保留引脚	接地
RSV35	AN30	I	Up	保留引脚	接地
RSV36	AP31	I	Up	保留引脚	接地
RSV37	AN31	0		保留引脚	浮空
RSV38	AK32	I	Down	保留引脚	接地

RSV39	AK34	I	Down	保留引脚	接地
RSV40	AK33	I	Down	保留引脚	接地
RSV41	AH33	I	Down	保留引脚	接地
RSV42	AL26	I	Down	保留引脚	接地
RSV43	AE33	I	Down	保留引脚	接地
RSV44	AK26	I	Down	保留引脚	接地
RSV45	AE34	I	Down	保留引脚	接地
RSV46	AL33	I	Up	保留引脚	接地
RSV47	AK25	I	Down	保留引脚	接地
RSV48	AG33	I	Down	保留引脚	接地
RSV49	AK29	I	Up	保留引脚	接地
RSV50	AL30	I	Up	保留引脚	接地
RSV51	AK17	I	Down	保留引脚	接地
RSV52	AL17	I	Down	保留引脚	接地
RSV53	AN17	I	Up	保留引脚	接地
RSV54	AP17	O		保留引脚	浮空
RSV55	AL29	O		保留引脚	浮空
RSV56	AN29	O		保留引脚	浮空
RSV57	AP29	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 7-2。

表 7-2 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
	GPIOD1	LPC.lreset_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_ovb_sel[0], 选择被观察时钟的选择信号
	GPIOA1	clk_ovb_sel[1], 选择被观察时钟的选择信号
	GPIOA2	clk_ovb_sel[2], 选择被观察时钟的选择信号
	GPIOA3	clk_ovb_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_ovb, 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

注意：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。

7.2 GMAC 引脚 (28 PIN)

表 7-3 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	AK18	I		输入 osc 时钟, 125MHz
G1_CLK_GTX	AP18	O		GMACO 向 PHY 输出 TX 时钟
G1_CLK_RX	AL18	I		GMACO 输入 RX 时钟
G1_CLK_TX	AN18	I		GMACO 输入 TX 时钟
G1_COL	AL23	I		GMACO PHY 冲突检测信号
G1_CRS	AK23	I		GMACO PHY 发送接收非 IDLE 状态
G1_MDC	AN23	O		GMACO 管理接口时钟
G1_MDIO	AP23	B	Down	GMACO 管理接口输入输出数据
G1_RX0	AK19	I		GMACO 数据输入 bit0

G1_RX1	AL19	I		GMAC0 数据输入 bit1
G1_RX2	AK20	I		GMAC0 数据输入 bit2
G1_RX3	AL20	I		GMAC0 数据输入 bit3
G1_RX4	AK21	I		GMAC0 数据输入 bit4
G1_RX5	AL21	I		GMAC0 数据输入 bit5
G1_RX6	AL22	I		GMAC0 数据输入 bit6
G1_RX7	AK22	I		GMAC0 数据输入 bit7
G1_RXDV	AL24	I		GMAC0 数据输入有效
G1_RXER	AK24	I		GMAC0 接收 Error
G1_TX0	AN19	0		GMAC0 数据输出 bit
G1_TX1	AP19	0		GMAC0 数据输出 bit
G1_TX2	AN20	0		GMAC0 数据输出 bit
G1_TX3	AP20	0		GMAC0 数据输出 bit
G1_TX4	AN21	0		GMAC0 数据输出 bit
G1_TX5	AP21	0		GMAC0 数据输出 bit
G1_TX6	AP22	0		GMAC0 数据输出 bit
G1_TX7	AN22	0		GMAC0 数据输出 bit
G1_TXEN	AP24	0		GMAC0 数据输出有效
G1_TXER	AN24	0		GMAC0 发送 Error

注:

- (1) 信号类型: I 输入引脚, 0 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。

7.3 PCIE 引脚 (140 PIN)

表 7-4 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	J33	I		100MHz 差分时钟输入
PCIE0_CLKP	K33	I		
PCIE0_RBIAST0	J23	A		参考电阻偏置
PCIE0_RBIAST1	J25	A		
PCIE0_RBIAST2	J27	A		
PCIE0_RBIAST3	J29	A		
PCIE0_RX00N	C22	I		输入链路差分信号
PCIE0_RX00P	D22	I		
PCIE0_RX01N	A23	I		
PCIE0_RX01P	B23	I		
PCIE0_RX02N	C24	I		
PCIE0_RX02P	D24	I		
PCIE0_RX03N	A25	I		

PCIE0_RX03P	B25	I	
PCIE0_RX04N	C26	I	
PCIE0_RX04P	D26	I	
PCIE0_RX05N	A27	I	
PCIE0_RX05P	B27	I	
PCIE0_RX06N	C28	I	
PCIE0_RX06P	D28	I	
PCIE0_RX07N	A29	I	
PCIE0_RX07P	B29	I	
PCIE0_RX08N	C30	I	
PCIE0_RX08P	D30	I	
PCIE0_RX09N	E31	I	
PCIE0_RX09P	F31	I	
PCIE0_RX10N	B32	I	
PCIE0_RX10P	C32	I	
PCIE0_RX11N	D33	I	
PCIE0_RX11P	E33	I	
PCIE0_RX12N	B34	I	
PCIE0_RX12P	C34	I	
PCIE0_RX13N	D35	I	
PCIE0_RX13P	E35	I	
PCIE0_RX14N	B36	I	
PCIE0_RX14P	C36	I	
PCIE0_RX15N	D37	I	
PCIE0_RX15P	E37	I	
PCIE0_TX00N	G20	0	
PCIE0_TX00P	H20	0	
PCIE0_TX01N	E21	0	
PCIE0_TX01P	F21	0	
PCIE0_TX02N	G22	0	
PCIE0_TX02P	H22	0	
PCIE0_TX03N	E23	0	
PCIE0_TX03P	F23	0	
PCIE0_TX04N	G24	0	
PCIE0_TX04P	H24	0	
PCIE0_TX05N	E25	0	
PCIE0_TX05P	F25	0	
PCIE0_TX06N	G26	0	
PCIE0_TX06P	H26	0	
PCIE0_TX07N	E27	0	
PCIE0_TX07P	F27	0	
PCIE0_TX08N	G28	0	
PCIE0_TX08P	H28	0	
PCIE0_TX09N	E29	0	

输出链路差分信号

PCIE0_TX09P	F29	0	
PCIE0_TX10N	G30	0	
PCIE0_TX10P	H30	0	
PCIE0_TX11N	G32	0	
PCIE0_TX11P	H32	0	
PCIE0_TX12N	F34	0	
PCIE0_TX12P	G34	0	
PCIE0_TX13N	H35	0	
PCIE0_TX13P	J35	0	
PCIE0_TX14N	F36	0	
PCIE0_TX14P	G36	0	
PCIE0_TX15N	H37	0	
PCIE0_TX15P	J37	0	
PCIE1_CLKN	J31	I	100MHz 差分时钟输入
PCIE1_CLKP	K31	I	
PCIE1_RBIAST0	J21	A	参考电阻偏置
PCIE1_RBIAST1	J19	A	
PCIE1_RBIAST2	J17	A	
PCIE1_RBIAST3	J15	A	
PCIE1_RX00N	C20	I	输入链路差分信号
PCIE1_RX00P	D20	I	
PCIE1_RX01N	C18	I	
PCIE1_RX01P	D18	I	
PCIE1_RX02N	A17	I	
PCIE1_RX02P	B17	I	
PCIE1_RX03N	C16	I	
PCIE1_RX03P	D16	I	
PCIE1_RX04N	A15	I	
PCIE1_RX04P	B15	I	
PCIE1_RX05N	C14	I	
PCIE1_RX05P	D14	I	
PCIE1_RX06N	A13	I	
PCIE1_RX06P	B13	I	
PCIE1_RX07N	C12	I	
PCIE1_RX07P	D12	I	
PCIE1_RX08N	A11	I	
PCIE1_RX08P	B11	I	
PCIE1_RX09N	D10	I	
PCIE1_RX09P	E10	I	
PCIE1_RX10N	B9	I	
PCIE1_RX10P	C9	I	
PCIE1_RX11N	D8	I	
PCIE1_RX11P	E8	I	
PCIE1_RX12N	A7	I	

PCIE1_RX12P	B7	I	
PCIE1_RX13N	C6	I	
PCIE1_RX13P	D6	I	
PCIE1_RX14N	A5	I	
PCIE1_RX14P	B5	I	
PCIE1_RX15N	C4	I	
PCIE1_RX15P	D4	I	
PCIE1_TX00N	E19	0	
PCIE1_TX00P	F19	0	
PCIE1_TX01N	G18	0	
PCIE1_TX01P	H18	0	
PCIE1_TX02N	E17	0	
PCIE1_TX02P	F17	0	
PCIE1_TX03N	G16	0	
PCIE1_TX03P	H16	0	
PCIE1_TX04N	E15	0	
PCIE1_TX04P	F15	0	
PCIE1_TX05N	G14	0	
PCIE1_TX05P	H14	0	
PCIE1_TX06N	E13	0	
PCIE1_TX06P	F13	0	
PCIE1_TX07N	H12	0	
PCIE1_TX07P	J12	0	
PCIE1_TX08N	F11	0	
PCIE1_TX08P	G11	0	
PCIE1_TX09N	H10	0	
PCIE1_TX09P	J10	0	
PCIE1_TX10N	F9	0	
PCIE1_TX10P	G9	0	
PCIE1_TX11N	H8	0	
PCIE1_TX11P	J8	0	
PCIE1_TX12N	F7	0	
PCIE1_TX12P	G7	0	
PCIE1_TX13N	H6	0	
PCIE1_TX13P	J6	0	
PCIE1_TX14N	E5	0	
PCIE1_TX14P	F5	0	
PCIE1_TX15N	G4	0	
PCIE1_TX15P	H4	0	

输出链路差分信号

注:

(1) 信号类型: I 输入引脚, 0 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。

7.4 DDR3 引脚 (312 PIN)

表 7-5 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
MO_A00	AP1	0		通道0存储器地址, bit 0
MO_A01	AL1	0		通道0存储器地址, bit 1
MO_A02	AL2	0		通道0存储器地址, bit 2
MO_A03	AK5	0		通道0存储器地址, bit 3
MO_A04	AK4	0		通道0存储器地址, bit 4
MO_A05	AJ5	0		通道0存储器地址, bit 5
MO_A06	AJ4	0		通道0存储器地址, bit 6
MO_A07	AK1	0		通道0存储器地址, bit 7
MO_A08	AK2	0		通道0存储器地址, bit 8
MO_A09	AH5	0		通道0存储器地址, bit 9
MO_A10	AR1	0		通道0存储器地址, bit 10
MO_A11	AH4	0		通道0存储器地址, bit 11
MO_A12	AJ2	0		通道0存储器地址, bit 12
MO_A13	AT4	0		通道0存储器地址, bit 13
MO_A14	AG5	0		通道0存储器地址, bit 14
MO_A15	AH2	0		通道0存储器地址, bit 15
MO_ATB0	AL4	0		通道0存储器ATB0, 模拟测试总线信号, 留待IP公司观察使用。PCB设计需保证该信号可观察。
MO_ATB1	AL5	0		通道0存储器ATB1, 模拟测试总线信号, 留待IP公司观察使用。PCB设计需保证该信号可观察。
MO_BA0	AP4	0		通道0存储器地址, bit 0
MO_BA1	AR2	0		通道0存储器地址, bit 1
MO_BA2	AG4	0		通道0存储器地址, bit 2
MO_CAL_BLK	AT2	I		通道0存储器CAL_BLK
MO_CAS	AR5	0		通道0存储器列地址选通
MO_CKE0	AH1	0		通道0存储器时钟使能, cs 0
MO_CKE1	AF4	0		通道0存储器时钟使能, cs 1
MO_CKE2	AF5	0		通道0存储器时钟使能, cs 2
MO_CKE3	AG2	0		通道0存储器时钟使能, cs 3
MO_CLKON	AM5	0		通道0存储器时钟-, cs0
MO_CLKOP	AM4	0		通道0存储器时钟+, cs0
MO_CLK1N	AM2	0		通道0存储器时钟-, cs1
MO_CLK1P	AM1	0		通道0存储器时钟+, cs1
MO_CLK2N	AN5	0		通道0存储器时钟-, cs2
MO_CLK2P	AN4	0		通道0存储器时钟+, cs2
MO_CLK3N	AN2	0		通道0存储器时钟-, cs3

MO_CLK3P	AN1	0		通道 0 存储器时钟+, cs3
MO_CS0	AT1	0		通道 0 存储器片选, cs0
MO_CS1	AT5	0		通道 0 存储器片选, cs1
MO_CS2	AU5	0		通道 0 存储器片选, cs2
MO_CS3	AU4	0		通道 0 存储器片选, cs3
MO_ODT0	AU2	0		通道 0 存储器 ODT, cs0
MO_ODT1	AV1	0		通道 0 存储器 ODT, cs1
MO_ODT2	AU1	0		通道 0 存储器 ODT, cs2
MO_ODT3	AV2	0		通道 0 存储器 ODT, cs3
MO_PAR_ERR	AJ1	I		通道 0 存储器的奇偶校验结果
MO_PAR_IN	AP2	0		通道 0 存储器的奇偶校验值
MO_RAS	AP5	0		通道 0 存储器行地址选通
MO_RESET	AG1	0		通道 0 存储器复位
MO_WE	AR4	0		通道 0 存储器写使能
MO_SO_DMN	AT9	I/O		通道 0 存储器数据屏蔽-, slice0
MO_SO_DMP	AU9	I/O		通道 0 存储器数据屏蔽+, slice0
MO_SO_DQ00	AU7	I/O		通道 0 存储器数据, bit0
MO_SO_DQ01	AU12	I/O		通道 0 存储器数据, bit1
MO_SO_DQ02	AU11	I/O		通道 0 存储器数据, bit2
MO_SO_DQ03	AT12	I/O		通道 0 存储器数据, bit3
MO_SO_DQ04	AT11	I/O		通道 0 存储器数据, bit4
MO_SO_DQ05	AT8	I/O		通道 0 存储器数据, bit5
MO_SO_DQ06	AU8	I/O		通道 0 存储器数据, bit6
MO_SO_DQ07	AT7	I/O		通道 0 存储器数据, bit7
MO_SO_DQSN	AU10	I/O		通道 0 存储器数据选通-, slice0
MO_SO_DQSP	AT10	I/O		通道 0 存储器数据选通+, slice0
MO_S1_DMN	AW5	I/O		通道 0 存储器数据屏蔽-, slice1
MO_S1_DMP	AY5	I/O		通道 0 存储器数据屏蔽+, slice1
MO_S1_DQ08	AY4	I/O		通道 0 存储器数据, bit08
MO_S1_DQ09	AW8	I/O		通道 0 存储器数据, bit09
MO_S1_DQ10	AY7	I/O		通道 0 存储器数据, bit10
MO_S1_DQ11	AW7	I/O		通道 0 存储器数据, bit11
MO_S1_DQ12	AW4	I/O		通道 0 存储器数据, bit12
MO_S1_DQ13	AY3	I/O		通道 0 存储器数据, bit13
MO_S1_DQ14	AW3	I/O		通道 0 存储器数据, bit14
MO_S1_DQ15	AY8	I/O		通道 0 存储器数据, bit15
MO_S1_DQSN	AY6	I/O		通道 0 存储器数据选通-, slice1
MO_S1_DQSP	AW6	I/O		通道 0 存储器数据选通+, slice1
MO_S2_DMN	W2	I/O		通道 0 存储器数据屏蔽-, slice2
MO_S2_DMP	W1	I/O		通道 0 存储器数据屏蔽+, slice2
MO_S2_DQ16	V1	I/O		通道 0 存储器数据, bit16
MO_S2_DQ17	AE2	I/O		通道 0 存储器数据, bit17
MO_S2_DQ18	AD1	I/O		通道 0 存储器数据, bit18
MO_S2_DQ19	AD2	I/O		通道 0 存储器数据, bit19

MO_S2_DQ20	AE1	I/O		通道 0 存储器数据, bit20
MO_S2_DQ21	V2	I/O		通道 0 存储器数据, bit21
MO_S2_DQ22	U2	I/O		通道 0 存储器数据, bit22
MO_S2_DQ23	U1	I/O		通道 0 存储器数据, bit23
MO_S2_DQSN	AC1	I/O		通道 0 存储器数据选通-, slice2
MO_S2_DQSP	AC2	I/O		通道 0 存储器数据选通+, slice2
MO_S3_DMN	AA5	I/O		通道 0 存储器数据屏蔽-, slice3
MO_S3_DMP	AA4	I/O		通道 0 存储器数据屏蔽+, slice3
MO_S3_DQ24	W4	I/O		通道 0 存储器数据, bit24
MO_S3_DQ25	AD4	I/O		通道 0 存储器数据, bit25
MO_S3_DQ26	AC4	I/O		通道 0 存储器数据, bit26
MO_S3_DQ27	AD5	I/O		通道 0 存储器数据, bit27
MO_S3_DQ28	AC5	I/O		通道 0 存储器数据, bit28
MO_S3_DQ29	Y5	I/O		通道 0 存储器数据, bit29
MO_S3_DQ30	Y4	I/O		通道 0 存储器数据, bit30
MO_S3_DQ31	W5	I/O		通道 0 存储器数据, bit31
MO_S3_DQSN	AB4	I/O		通道 0 存储器数据选通-, slice3
MO_S3_DQSP	AB5	I/O		通道 0 存储器数据选通+, slice3
MO_S4_DMN	P5	I/O		通道 0 存储器数据屏蔽-, slice4
MO_S4_DMP	P4	I/O		通道 0 存储器数据屏蔽+, slice4
MO_S4_DQ32	M4	I/O		通道 0 存储器数据, bit32
MO_S4_DQ33	U5	I/O		通道 0 存储器数据, bit33
MO_S4_DQ34	T4	I/O		通道 0 存储器数据, bit34
MO_S4_DQ35	T5	I/O		通道 0 存储器数据, bit35
MO_S4_DQ36	U4	I/O		通道 0 存储器数据, bit36
MO_S4_DQ37	N5	I/O		通道 0 存储器数据, bit37
MO_S4_DQ38	N4	I/O		通道 0 存储器数据, bit38
MO_S4_DQ39	M5	I/O		通道 0 存储器数据, bit39
MO_S4_DQSN	R4	I/O		通道 0 存储器数据选通-, slice4
MO_S4_DQSP	R5	I/O		通道 0 存储器数据选通+, slice4
MO_S5_DMN	M2	I/O		通道 0 存储器数据屏蔽-, slice5
MO_S5_DMP	M1	I/O		通道 0 存储器数据屏蔽+, slice5
MO_S5_DQ40	L1	I/O		通道 0 存储器数据, bit40
MO_S5_DQ41	R1	I/O		通道 0 存储器数据, bit41
MO_S5_DQ42	P1	I/O		通道 0 存储器数据, bit42
MO_S5_DQ43	R2	I/O		通道 0 存储器数据, bit43
MO_S5_DQ44	P2	I/O		通道 0 存储器数据, bit44
MO_S5_DQ45	K1	I/O		通道 0 存储器数据, bit45
MO_S5_DQ46	L2	I/O		通道 0 存储器数据, bit46
MO_S5_DQ47	K2	I/O		通道 0 存储器数据, bit47
MO_S5_DQSN	N1	I/O		通道 0 存储器数据选通-, slice5
MO_S5_DQSP	N2	I/O		通道 0 存储器数据选通+, slice5
MO_S6_DMN	E2	I/O		通道 0 存储器数据屏蔽-, slice6
MO_S6_DMP	E1	I/O		通道 0 存储器数据屏蔽+, slice6

MO_S6_DQ48	D1	I/O		通道 0 存储器数据, bit48
MO_S6_DQ49	H1	I/O		通道 0 存储器数据, bit49
MO_S6_DQ50	G1	I/O		通道 0 存储器数据, bit50
MO_S6_DQ51	H2	I/O		通道 0 存储器数据, bit51
MO_S6_DQ52	G2	I/O		通道 0 存储器数据, bit52
MO_S6_DQ53	C2	I/O		通道 0 存储器数据, bit53
MO_S6_DQ54	D2	I/O		通道 0 存储器数据, bit54
MO_S6_DQ55	B3	I/O		通道 0 存储器数据, bit55
MO_S6_DQSN	F1	I/O		通道 0 存储器数据选通-, slice6
MO_S6_DQSP	F2	I/O		通道 0 存储器数据选通+, slice6
MO_S7_DMN	AT16	I/O		通道 0 存储器数据屏蔽-, slice7
MO_S7_DMP	AU16	I/O		通道 0 存储器数据屏蔽+, slice7
MO_S7_DQ56	AU14	I/O		通道 0 存储器数据, bit56
MO_S7_DQ57	AT19	I/O		通道 0 存储器数据, bit57
MO_S7_DQ58	AU18	I/O		通道 0 存储器数据, bit58
MO_S7_DQ59	AT18	I/O		通道 0 存储器数据, bit59
MO_S7_DQ60	AT15	I/O		通道 0 存储器数据, bit60
MO_S7_DQ61	AU19	I/O		通道 0 存储器数据, bit61
MO_S7_DQ62	AT14	I/O		通道 0 存储器数据, bit62
MO_S7_DQ63	AU15	I/O		通道 0 存储器数据, bit63
MO_S7_DQSN	AU17	I/O		通道 0 存储器数据选通-, slice7
MO_S7_DQSP	AT17	I/O		通道 0 存储器数据选通+, slice7
MO_S8_DMN	AW12	I/O		通道 0 存储器数据屏蔽-, slice8
MO_S8_DMP	AY12	I/O		通道 0 存储器数据屏蔽+, slice8
MO_S8_DQ64	AY11	I/O		通道 0 存储器数据, bit64
MO_S8_DQ65	AY15	I/O		通道 0 存储器数据, bit65
MO_S8_DQ66	AW15	I/O		通道 0 存储器数据, bit66
MO_S8_DQ67	AY14	I/O		通道 0 存储器数据, bit67
MO_S8_DQ68	AW14	I/O		通道 0 存储器数据, bit68
MO_S8_DQ69	AW11	I/O		通道 0 存储器数据, bit69
MO_S8_DQ70	AY10	I/O		通道 0 存储器数据, bit70
MO_S8_DQ71	AW10	I/O		通道 0 存储器数据, bit71
MO_S8_DQSN	AY13	I/O		通道 0 存储器数据选通-, slice8
MO_S8_DQSP	AW13	I/O		通道 0 存储器数据选通+, slice8
M2_A00	AN40	0		通道 2 存储器地址, bit 0
M2_A01	AK40	0		通道 2 存储器地址, bit 1
M2_A02	AK39	0		通道 2 存储器地址, bit 2
M2_A03	AJ36	0		通道 2 存储器地址, bit 3
M2_A04	AJ37	0		通道 2 存储器地址, bit 4
M2_A05	AH36	0		通道 2 存储器地址, bit 5
M2_A06	AH37	0		通道 2 存储器地址, bit 6
M2_A07	AJ40	0		通道 2 存储器地址, bit 7
M2_A08	AJ39	0		通道 2 存储器地址, bit 8
M2_A09	AG36	0		通道 2 存储器地址, bit 9

M2_A10	AP40	0		通道 2 存储器地址, bit 10
M2_A11	AG37	0		通道 2 存储器地址, bit 11
M2_A12	AH39	0		通道 2 存储器地址, bit 12
M2_A13	AR37	0		通道 2 存储器地址, bit 13
M2_A14	AF36	0		通道 2 存储器地址, bit 14
M2_A15	AG39	0		通道 2 存储器地址, bit 15
M2_ATB0	AK37	0		通道 2 存储器 ATB0, 模拟测试总线信号, 留待 IP 公司观察使用。PCB 设计需保证该信号可观察。
M2_ATB1	AK36	0		通道 2 存储器 ATB1, 模拟测试总线信号, 留待 IP 公司观察使用。PCB 设计需保证该信号可观察。
M2_BA0	AN37	0		通道 2 存储器地址, bit 0
M2_BA1	AP39	0		通道 2 存储器地址, bit 1
M2_BA2	AF37	0		通道 2 存储器地址, bit 2
M2_CAL_BLK	AR39	I		通道 2 存储器 CAL_BLK
M2_CAS	AP36	0		通道 2 存储器列地址选通
M2_CKE0	AG40	0		通道 2 存储器时钟使能, cs 0
M2_CKE1	AE37	0		通道 2 存储器时钟使能, cs 1
M2_CKE2	AE36	0		通道 2 存储器时钟使能, cs 2
M2_CKE3	AF39	0		通道 2 存储器时钟使能, cs 3
M2_CLKON	AL36	0		通道 2 存储器时钟-, cs0
M2_CLKOP	AL37	0		通道 2 存储器时钟+, cs0
M2_CLK1N	AL39	0		通道 2 存储器时钟-, cs1
M2_CLK1P	AL40	0		通道 2 存储器时钟+, cs1
M2_CLK2N	AM36	0		通道 2 存储器时钟-, cs2
M2_CLK2P	AM37	0		通道 2 存储器时钟+, cs2
M2_CLK3N	AM39	0		通道 2 存储器时钟-, cs3
M2_CLK3P	AM40	0		通道 2 存储器时钟+, cs3
M2_CS0	AR40	0		通道 2 存储器片, cs0
M2_CS1	AR36	0		通道 2 存储器片选, cs1
M2_CS2	AT36	0		通道 2 存储器片选, cs2
M2_CS3	AT37	0		通道 2 存储器片选, cs3
M2_ODT0	AT39	0		通道 2 存储器 ODT, cs0
M2_ODT1	AU40	0		通道 2 存储器 ODT, cs1
M2_ODT2	AT40	0		通道 2 存储器 ODT, cs2
M2_ODT3	AU39	0		通道 2 存储器 ODT, cs3
M2_PAR_ERR	AH40	I		通道 2 存储器的奇偶校验结果
M2_PAR_IN	AN39	0		通道 2 存储器的奇偶校验值
M2_RAS	AN36	0		通道 2 存储器行地址选通
M2_RESET	AF40	0		通道 2 存储器复位
M2_WE	AP37	0		通道 2 存储器写使能
M2_SO_DMN	AT31	I/O		通道 2 存储器数据屏蔽-, slice0
M2_SO_DMP	AU31	I/O		通道 2 存储器数据屏蔽+, slice0
M2_SO_DQ00	AT32	I/O		通道 2 存储器数据, bit0
M2_SO_DQ01	AT28	I/O		通道 2 存储器数据, bit1

M2_SO_DQ02	AU29	I/O		通道 2 存储器数据, bit2
M2_SO_DQ03	AT29	I/O		通道 2 存储器数据, bit3
M2_SO_DQ04	AU28	I/O		通道 2 存储器数据, bit4
M2_SO_DQ05	AU32	I/O		通道 2 存储器数据, bit5
M2_SO_DQ06	AT33	I/O		通道 2 存储器数据, bit6
M2_SO_DQ07	AT34	I/O		通道 2 存储器数据, bit7
M2_SO_DQSN	AU30	I/O		通道 2 存储器数据选通-, slice0
M2_SO_DQSP	AT30	I/O		通道 2 存储器数据选通+, slice0
M2_S1_DMN	AW36	I/O		通道 2 存储器数据屏蔽-, slice1
M2_S1_DMP	AV36	I/O		通道 2 存储器数据屏蔽+, slice1
M2_S1_DQ08	AV37	I/O		通道 2 存储器数据, bit08
M2_S1_DQ09	AV33	I/O		通道 2 存储器数据, bit09
M2_S1_DQ10	AV34	I/O		通道 2 存储器数据, bit10
M2_S1_DQ11	AW33	I/O		通道 2 存储器数据, bit11
M2_S1_DQ12	AW34	I/O		通道 2 存储器数据, bit12
M2_S1_DQ13	AW37	I/O		通道 2 存储器数据, bit13
M2_S1_DQ14	AV38	I/O		通道 2 存储器数据, bit14
M2_S1_DQ15	AW38	I/O		通道 2 存储器数据, bit15
M2_S1_DQSN	AV35	I/O		通道 2 存储器数据选通-, slice1
M2_S1_DQSP	AW35	I/O		通道 2 存储器数据选通+, slice1
M2_S2_DMN	V39	I/O		通道 2 存储器数据屏蔽-, slice2
M2_S2_DMP	V40	I/O		通道 2 存储器数据屏蔽+, slice2
M2_S2_DQ16	U40	I/O		通道 2 存储器数据, bit16
M2_S2_DQ17	AD40	I/O		通道 2 存储器数据, bit17
M2_S2_DQ18	AD39	I/O		通道 2 存储器数据, bit18
M2_S2_DQ19	AC39	I/O		通道 2 存储器数据, bit19
M2_S2_DQ20	AC40	I/O		通道 2 存储器数据, bit20
M2_S2_DQ21	U39	I/O		通道 2 存储器数据, bit21
M2_S2_DQ22	T40	I/O		通道 2 存储器数据, bit22
M2_S2_DQ23	T39	I/O		通道 2 存储器数据, bit23
M2_S2_DQSN	AB40	I/O		通道 2 存储器数据选通-, slice2
M2_S2_DQSP	AB39	I/O		通道 2 存储器数据选通+, slice2
M2_S3_DMN	Y36	I/O		通道 2 存储器数据屏蔽-, slice3
M2_S3_DMP	Y37	I/O		通道 2 存储器数据屏蔽+, slice3
M2_S3_DQ24	W36	I/O		通道 2 存储器数据, bit24
M2_S3_DQ25	AC36	I/O		通道 2 存储器数据, bit25
M2_S3_DQ26	AB37	I/O		通道 2 存储器数据, bit26
M2_S3_DQ27	AB36	I/O		通道 2 存储器数据, bit27
M2_S3_DQ28	AC37	I/O		通道 2 存储器数据, bit28
M2_S3_DQ29	W37	I/O		通道 2 存储器数据, bit29
M2_S3_DQ30	V37	I/O		通道 2 存储器数据, bit30
M2_S3_DQ31	V36	I/O		通道 2 存储器数据, bit31
M2_S3_DQSN	AA37	I/O		通道 2 存储器数据选通-, slice3
M2_S3_DQSP	AA36	I/O		通道 2 存储器数据选通+, slice3

M2_S4_DMN	N36	I/O		通道 2 存储器数据屏蔽-, slice4
M2_S4_DMP	N37	I/O		通道 2 存储器数据屏蔽+, slice4
M2_S4_DQ32	M36	I/O		通道 2 存储器数据, bit32
M2_S4_DQ33	T36	I/O		通道 2 存储器数据, bit33
M2_S4_DQ34	R37	I/O		通道 2 存储器数据, bit34
M2_S4_DQ35	R36	I/O		通道 2 存储器数据, bit35
M2_S4_DQ36	T37	I/O		通道 2 存储器数据, bit36
M2_S4_DQ37	M37	I/O		通道 2 存储器数据, bit37
M2_S4_DQ38	L37	I/O		通道 2 存储器数据, bit38
M2_S4_DQ39	L36	I/O		通道 2 存储器数据, bit39
M2_S4_DQSN	P37	I/O		通道 2 存储器数据选通-, slice4
M2_S4_DQSP	P36	I/O		通道 2 存储器数据选通+, slice4
M2_S5_DMN	L39	I/O		通道 2 存储器数据屏蔽-, slice5
M2_S5_DMP	L40	I/O		通道 2 存储器数据屏蔽+, slice5
M2_S5_DQ40	K40	I/O		通道 2 存储器数据, bit40
M2_S5_DQ41	P40	I/O		通道 2 存储器数据, bit41
M2_S5_DQ42	N40	I/O		通道 2 存储器数据, bit42
M2_S5_DQ43	N39	I/O		通道 2 存储器数据, bit43
M2_S5_DQ44	P39	I/O		通道 2 存储器数据, bit44
M2_S5_DQ45	K39	I/O		通道 2 存储器数据, bit45
M2_S5_DQ46	J40	I/O		通道 2 存储器数据, bit46
M2_S5_DQ47	J39	I/O		通道 2 存储器数据, bit47
M2_S5_DQSN	M40	I/O		通道 2 存储器数据选通-, slice5
M2_S5_DQSP	M39	I/O		通道 2 存储器数据选通+, slice5
M2_S6_DMN	D39	I/O		通道 2 存储器数据屏蔽-, slice6
M2_S6_DMP	D40	I/O		通道 2 存储器数据屏蔽+, slice6
M2_S6_DQ48	C40	I/O		通道 2 存储器数据, bit48
M2_S6_DQ49	G40	I/O		通道 2 存储器数据, bit49
M2_S6_DQ50	F40	I/O		通道 2 存储器数据, bit50
M2_S6_DQ51	G39	I/O		通道 2 存储器数据, bit51
M2_S6_DQ52	F39	I/O		通道 2 存储器数据, bit52
M2_S6_DQ53	C39	I/O		通道 2 存储器数据, bit53
M2_S6_DQ54	B39	I/O		通道 2 存储器数据, bit54
M2_S6_DQ55	B38	I/O		通道 2 存储器数据, bit55
M2_S6_DQSN	E40	I/O		通道 2 存储器数据选通-, slice6
M2_S6_DQSP	E39	I/O		通道 2 存储器数据选通+, slice6
M2_S7_DMN	AT24	I/O		通道 2 存储器数据屏蔽-, slice7
M2_S7_DMP	AU24	I/O		通道 2 存储器数据屏蔽+, slice7
M2_S7_DQ56	AT25	I/O		通道 2 存储器数据, bit56
M2_S7_DQ57	AT22	I/O		通道 2 存储器数据, bit57
M2_S7_DQ58	AU21	I/O		通道 2 存储器数据, bit58
M2_S7_DQ59	AT21	I/O		通道 2 存储器数据, bit59
M2_S7_DQ60	AU25	I/O		通道 2 存储器数据, bit60
M2_S7_DQ61	AU22	I/O		通道 2 存储器数据, bit61

M2_S7_DQ62	AU26	I/O		通道 2 存储器数据, bit62
M2_S7_DQ63	AT26	I/O		通道 2 存储器数据, bit63
M2_S7_DQSN	AU23	I/O		通道 2 存储器数据选通-, slice7
M2_S7_DQSP	AT23	I/O		通道 2 存储器数据选通+, slice7
M2_S8_DMN	AW26	I/O		通道 2 存储器数据屏蔽-, slice8
M2_S8_DMP	AY26	I/O		通道 2 存储器数据屏蔽+, slice8
M2_S8_DQ64	AY27	I/O		通道 2 存储器数据, bit64
M2_S8_DQ65	AW24	I/O		通道 2 存储器数据, bit65
M2_S8_DQ66	AY23	I/O		通道 2 存储器数据, bit66
M2_S8_DQ67	AW27	I/O		通道 2 存储器数据, bit67
M2_S8_DQ68	AY24	I/O		通道 2 存储器数据, bit68
M2_S8_DQ69	AW28	I/O		通道 2 存储器数据, bit69
M2_S8_DQ70	AY28	I/O		通道 2 存储器数据, bit70
M2_S8_DQ71	AW23	I/O		通道 2 存储器数据, bit71
M2_S8_DQSN	AY25	I/O		通道 2 存储器数据选通-, slice8
M2_S8_DQSP	AW25	I/O		通道 2 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。

7.5 电源引脚 (552 PIN)

表 7-6 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	AA7			0.9V 内核电源 (104 PIN)
VDD_CORE	AA8			
VDD_CORE	AA33			
VDD_CORE	AA34			
VDD_CORE	AB6			
VDD_CORE	AB8			
VDD_CORE	AB33			
VDD_CORE	AJ12			
VDD_CORE	AJ14			
VDD_CORE	AJ15			
VDD_CORE	AJ16			
VDD_CORE	AJ17			
VDD_CORE	AJ18			
VDD_CORE	AJ19			
VDD_CORE	AJ20			
VDD_CORE	AJ21			

VDD_CORE	AJ22		
VDD_CORE	AJ23		
VDD_CORE	AJ27		
VDD_CORE	AJ28		
VDD_CORE	AJ29		
VDD_CORE	AJ30		
VDD_CORE	AM10		
VDD_CORE	AM12		
VDD_CORE	AM14		
VDD_CORE	AM16		
VDD_CORE	AM18		
VDD_CORE	AM20		
VDD_CORE	AM26		
VDD_CORE	AM28		
VDD_CORE	AM30		
VDD_CORE	AR10		
VDD_CORE	AR12		
VDD_CORE	AR14		
VDD_CORE	AR16		
VDD_CORE	AR26		
VDD_CORE	AR28		
VDD_CORE	AR30		
VDD_CORE	L19		
VDD_CORE	L21		
VDD_CORE	L23		
VDD_CORE	L25		
VDD_CORE	L27		
VDD_CORE	L31		
VDD_CORE	L33		
VDD_CORE	M6		
VDD_CORE	M8		
VDD_CORE	M9		
VDD_CORE	M10		
VDD_CORE	M11		
VDD_CORE	M12		
VDD_CORE	M13		
VDD_CORE	M14		
VDD_CORE	M15		
VDD_CORE	M16		
VDD_CORE	M17		
VDD_CORE	M18		
VDD_CORE	M19		
VDD_CORE	M20		
VDD_CORE	M21		

VDD_CORE	M22		
VDD_CORE	M23		
VDD_CORE	M24		
VDD_CORE	M25		
VDD_CORE	M26		
VDD_CORE	M27		
VDD_CORE	M28		
VDD_CORE	M29		
VDD_CORE	M30		
VDD_CORE	M32		
VDD_CORE	M33		
VDD_CORE	M35		
VDD_CORE	N7		
VDD_CORE	N8		
VDD_CORE	N33		
VDD_CORE	N34		
VDD_CORE	P6		
VDD_CORE	P8		
VDD_CORE	P33		
VDD_CORE	P35		
VDD_CORE	R7		
VDD_CORE	R8		
VDD_CORE	R33		
VDD_CORE	R34		
VDD_CORE	T6		
VDD_CORE	T8		
VDD_CORE	T33		
VDD_CORE	T35		
VDD_CORE	U7		
VDD_CORE	U8		
VDD_CORE	U33		
VDD_CORE	U34		
VDD_CORE	V6		
VDD_CORE	V8		
VDD_CORE	V33		
VDD_CORE	V35		
VDD_CORE	W7		
VDD_CORE	W8		
VDD_CORE	W33		
VDD_CORE	W34		
VDD_CORE	Y6		
VDD_CORE	Y8		
VDD_CORE	Y33		
VDD_CORE	Y35		

VDD_IO	AJ24			1.8V 通用 IO 类管脚供电电源 (4 PIN)
VDD_IO	AJ25			
VDD_IO	AM22			
VDD_IO	AM24			
VDDA_PCIE	K6			0.95V PCIE 电源 (16 PIN)
VDDA_PCIE	K8			
VDDA_PCIE	K10			
VDDA_PCIE	K12			
VDDA_PCIE	K14			
VDDA_PCIE	K16			
VDDA_PCIE	K18			
VDDA_PCIE	K20			
VDDA_PCIE	K22			
VDDA_PCIE	K24			
VDDA_PCIE	L7			
VDDA_PCIE	L9			
VDDA_PCIE	L11			
VDDA_PCIE	L13			
VDDA_PCIE	L15			
VDDA_PCIE	L17			
VDDHV_PCIE	K26			1.5V 的 PCIE 电源 (6 PIN)
VDDHV_PCIE	K28			
VDDHV_PCIE	K30			
VDDHV_PCIE	K32			
VDDHV_PCIE	K34			
VDDHV_PCIE	L29			
VDDHV_PLL0	AJ13			1.8V 的模拟电源 (2 PIN)
VDDHV_PLL1	AJ26			
VDDQ MCU01	AC7			1.5V DDR3 DIMM 接口电源 (20 PIN)
VDDQ MCU01	AC8			
VDDQ MCU01	AD6			
VDDQ MCU01	AD8			
VDDQ MCU01	AE3			
VDDQ MCU01	AF6			
VDDQ MCU01	AG3			
VDDQ MCU01	AH6			
VDDQ MCU01	AJ3			
VDDQ MCU01	AJ9			
VDDQ MCU01	AJ11			
VDDQ MCU01	AK6			
VDDQ MCU01	AL3			
VDDQ MCU01	AM6			
VDDQ MCU01	AN3			
VDDQ MCU01	AP6			

VDDQ_MCU01	AR3		
VDDQ_MCU01	AT6		
VDDQ_MCU01	AU3		
VDDQ_MCU01	AV6		
VDDQ_MCU23	AB35		
VDDQ_MCU23	AC33		
VDDQ_MCU23	AC34		
VDDQ_MCU23	AD33		
VDDQ_MCU23	AD35		
VDDQ_MCU23	AE38		
VDDQ_MCU23	AF35		
VDDQ_MCU23	AG38		
VDDQ_MCU23	AH35		
VDDQ_MCU23	AJ31		
VDDQ_MCU23	AJ32		
VDDQ_MCU23	AJ38		
VDDQ_MCU23	AK35		
VDDQ_MCU23	AL38		
VDDQ_MCU23	AM35		
VDDQ_MCU23	AN38		
VDDQ_MCU23	AP35		
VDDQ_MCU23	AR38		
VDDQ_MCU23	AT35		
VDDQ_MCU23	AU38		
VSS	A4		
VSS	A6		
VSS	A12		
VSS	A14		
VSS	A16		
VSS	A18		
VSS	A24		
VSS	A26		
VSS	A28		
VSS	A30		
VSS	AA3		
VSS	AA6		
VSS	AA35		
VSS	AA38		
VSS	AB3		
VSS	AB7		
VSS	AB34		
VSS	AB38		
VSS	AC3		
VSS	AC6		

1.5V DDR3 DIMM 接口电源 (20 PIN)

数字地 (380 PIN)

VSS	AC35		
VSS	AC38		
VSS	AD3		
VSS	AD7		
VSS	AD36		
VSS	AD37		
VSS	AD38		
VSS	AE4		
VSS	AE5		
VSS	AE6		
VSS	AE35		
VSS	AE39		
VSS	AE40		
VSS	AF1		
VSS	AF2		
VSS	AF3		
VSS	AF38		
VSS	AG6		
VSS	AG35		
VSS	AH3		
VSS	AH38		
VSS	AJ6		
VSS	AJ35		
VSS	AK3		
VSS	AK38		
VSS	AL6		
VSS	AL35		
VSS	AM3		
VSS	AM9		
VSS	AM11		
VSS	AM13		
VSS	AM15		
VSS	AM17		
VSS	AM19		
VSS	AM21		
VSS	AM23		
VSS	AM25		
VSS	AM27		
VSS	AM29		
VSS	AM31		
VSS	AM32		
VSS	AM38		
VSS	AN6		
VSS	AN35		

VSS	AP3		
VSS	AP38		
VSS	AR6		
VSS	AR7		
VSS	AR8		
VSS	AR9		
VSS	AR11		
VSS	AR13		
VSS	AR15		
VSS	AR17		
VSS	AR18		
VSS	AR19		
VSS	AR20		
VSS	AR21		
VSS	AR22		
VSS	AR23		
VSS	AR24		
VSS	AR25		
VSS	AR27		
VSS	AR29		
VSS	AR31		
VSS	AR32		
VSS	AR33		
VSS	AR34		
VSS	AR35		
VSS	AT3		
VSS	AT13		
VSS	AT20		
VSS	AT27		
VSS	AT38		
VSS	AU6		
VSS	AU13		
VSS	AU20		
VSS	AU27		
VSS	AU33		
VSS	AU34		
VSS	AU35		
VSS	AU36		
VSS	AU37		
VSS	AV3		
VSS	AV4		
VSS	AV5		
VSS	AV7		
VSS	AV8		

VSS	AV9		
VSS	AV10		
VSS	AV11		
VSS	AV12		
VSS	AV13		
VSS	AV14		
VSS	AV15		
VSS	AV16		
VSS	AV17		
VSS	AV18		
VSS	AV19		
VSS	AV20		
VSS	AV21		
VSS	AV22		
VSS	AV23		
VSS	AV24		
VSS	AV25		
VSS	AV26		
VSS	AV27		
VSS	AV28		
VSS	AV29		
VSS	AV30		
VSS	AV31		
VSS	AV32		
VSS	AV39		
VSS	AW2		
VSS	AW9		
VSS	AW16		
VSS	AW32		
VSS	AY9		
VSS	AY16		
VSS	B4		
VSS	B6		
VSS	B8		
VSS	B10		
VSS	B12		
VSS	B14		
VSS	B16		
VSS	B18		
VSS	B24		
VSS	B26		
VSS	B28		
VSS	B30		
VSS	B31		

VSS	B33		
VSS	B35		
VSS	B37		
VSS	C3		
VSS	C5		
VSS	C7		
VSS	C8		
VSS	C10		
VSS	C11		
VSS	C13		
VSS	C15		
VSS	C17		
VSS	C19		
VSS	C21		
VSS	C23		
VSS	C25		
VSS	C27		
VSS	C29		
VSS	C31		
VSS	C33		
VSS	C35		
VSS	C37		
VSS	C38		
VSS	D3		
VSS	D5		
VSS	D7		
VSS	D9		
VSS	D11		
VSS	D13		
VSS	D15		
VSS	D17		
VSS	D19		
VSS	D21		
VSS	D23		
VSS	D25		
VSS	D27		
VSS	D29		
VSS	D31		
VSS	D32		
VSS	D34		
VSS	D36		
VSS	D38		
VSS	E3		
VSS	E4		

VSS	E6		
VSS	E7		
VSS	E9		
VSS	E11		
VSS	E12		
VSS	E14		
VSS	E16		
VSS	E18		
VSS	E20		
VSS	E22		
VSS	E24		
VSS	E26		
VSS	E28		
VSS	E30		
VSS	E32		
VSS	E34		
VSS	E36		
VSS	E38		
VSS	F3		
VSS	F4		
VSS	F6		
VSS	F8		
VSS	F10		
VSS	F12		
VSS	F14		
VSS	F16		
VSS	F18		
VSS	F20		
VSS	F22		
VSS	F24		
VSS	F26		
VSS	F28		
VSS	F30		
VSS	F32		
VSS	F33		
VSS	F35		
VSS	F37		
VSS	F38		
VSS	G3		
VSS	G5		
VSS	G6		
VSS	G8		
VSS	G10		
VSS	G12		

VSS	G13		
VSS	G15		
VSS	G17		
VSS	G19		
VSS	G21		
VSS	G23		
VSS	G25		
VSS	G27		
VSS	G29		
VSS	G31		
VSS	G33		
VSS	G35		
VSS	G37		
VSS	G38		
VSS	H3		
VSS	H5		
VSS	H7		
VSS	H9		
VSS	H11		
VSS	H13		
VSS	H15		
VSS	H17		
VSS	H19		
VSS	H21		
VSS	H23		
VSS	H25		
VSS	H27		
VSS	H29		
VSS	H31		
VSS	H33		
VSS	H34		
VSS	H36		
VSS	H38		
VSS	H39		
VSS	H40		
VSS	J1		
VSS	J2		
VSS	J3		
VSS	J4		
VSS	J5		
VSS	J7		
VSS	J9		
VSS	J11		
VSS	J13		

VSS	J14		
VSS	J16		
VSS	J18		
VSS	J20		
VSS	J22		
VSS	J24		
VSS	J26		
VSS	J28		
VSS	J30		
VSS	J32		
VSS	J34		
VSS	J36		
VSS	J38		
VSS	K3		
VSS	K4		
VSS	K5		
VSS	K7		
VSS	K9		
VSS	K11		
VSS	K13		
VSS	K15		
VSS	K17		
VSS	K19		
VSS	K21		
VSS	K23		
VSS	K25		
VSS	K27		
VSS	K29		
VSS	K35		
VSS	K36		
VSS	K37		
VSS	K38		
VSS	L3		
VSS	L4		
VSS	L5		
VSS	L6		
VSS	L8		
VSS	L10		
VSS	L12		
VSS	L14		
VSS	L16		
VSS	L18		
VSS	L20		
VSS	L22		

VSS	L24		
VSS	L26		
VSS	L28		
VSS	L30		
VSS	L32		
VSS	L34		
VSS	L35		
VSS	L38		
VSS	M3		
VSS	M7		
VSS	M34		
VSS	M38		
VSS	N3		
VSS	N6		
VSS	N35		
VSS	N38		
VSS	P3		
VSS	P7		
VSS	P34		
VSS	P38		
VSS	R3		
VSS	R6		
VSS	R35		
VSS	R38		
VSS	R39		
VSS	R40		
VSS	T1		
VSS	T2		
VSS	T3		
VSS	T7		
VSS	T34		
VSS	T38		
VSS	U3		
VSS	U6		
VSS	U35		
VSS	U36		
VSS	U37		
VSS	U38		
VSS	V3		
VSS	V4		
VSS	V5		
VSS	V7		
VSS	V34		
VSS	V38		

VSS	W3		
VSS	W6		
VSS	W35		
VSS	W38		
VSS	Y3		
VSS	Y7		
VSS	Y34		
VSS	Y38		

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。