

Q/LS

Q/LS 0046-2022

龙芯中科技术股份有限公司企业标准

龙芯 CPU 统一系统架构规范

V1.2.1

2022-07-10 发布

2022-07-20 实施

龙芯中科技术股份有限公司 批准

版权声明

本档版权归龙芯中科技术股份有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此档中的任何部分公开、转载或以其他方式散发给第三方。否则，必将追究其法律责任。

免责声明

本档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

龙芯中科技术股份有限公司

Loongson Technology Corporation Limited

地址：北京市海淀区中关村环保科技示范园龙芯产业园 2 号楼

Building No.2, Loongson Industrial Park,

Zhongguancun Environmental Protection Park, Haidian District, Beijing

电话(Tel): 010-62546668

传真(Fax): 010-62600826

文档更新记录		文档名	龙芯 CPU 统一系统架构规范
		版本号	V1.2.1
更新历史			
序号	版本号	更新内容	
1	V1.0	发布文档初始版本 V1.0 版。	
2	V1.1	计划更新《龙芯 3A4000_7A1000 硬件设计规范》、《龙芯 3A4000_7A1000 平台外围功能芯片&模组适配列表》、《龙芯 CPU 内核开发规范》等相关规范，后根据实际情况部分规范未做调整，本版本仅内部发布。	
3	V1.2	<p>1.更新《龙芯 3A4000_7A1000 硬件设计规范》为 V1.1 版</p> <p>2.更新《龙芯 3A4000_7A1000 外围功能芯片&模组支持列表》为 V1.2 版。</p> <p>3.新增《龙芯内置显卡软硬件设计规范》V0.2 版。</p>	
4	V1.2.1	1.更新《龙芯 3A4000_7A1000 通用类板卡硬件设计规范》为 V1.2 版	

一、序言

龙芯 CPU 统一系统架构规范是龙芯产品在通用 PC、服务器领域的开发指导规范，通过规范整机设计、固件开发、内核开发的技术要求，定义了整机、固件、系统的兼容性标准。

二、发布列表

序号	规范名称	版本号
1	龙芯 3A4000_7A1000 通用类板卡硬件设计规范	V1.1
2	龙芯 CPU 开发系统固件与内核接口规范	V2.3
3	AMD780E+710 芯片组固件开发规范	V1.0
4	龙芯 2H 芯片组固件开发规范	V1.0
5	龙芯 7A1000 芯片组固件开发规范	V1.0
6	龙芯 CPU 内核开发规范	V1.0
7	龙芯 7A1000 内置显卡软硬件设计规范	V0.2
8	龙芯 3A4000_7A1000 外围功能芯片&模组支持列表	V1.2
9	龙芯 3A4000_7A1000 通用类板卡硬件设计规范	V1.2

**龙芯 3A4000_7A1000 通用类板
卡**

硬件设计规范

V1.2

文档更新记录	文档名:	龙芯 3A4000_7A1000 通用类板卡硬件设计 规范
	版本号:	V1.2
更新历史		
序号	版本号	更新内容
1	V1.0	<ol style="list-style-type: none"> 1. 初版发布修改 HT 部分控制信号的描述 2. 增加内存 DDR4 的注意事项：不支持组内数据线互换 3. 修改电源部分电压和电流值 4. 增加双路 CPU0 到 7A1000 的 16 位互连拓扑结构模式 5. 增加 PCIE 说明：支持信号反转 6. 修改 STR 电序要求 7. 删除 DDR3 PCB 设计指导 8. 删除外设列表，这部分单独出附录文档 9. 修改一些描述和书写错误 10. 修改处理器 GPIO 复用功能，默认通用系统不支持复用串口功能 11. 添加 PCIE 对 Realtek 网卡支持连接建议 12. 修改 7A1000 桥片 UART0 复用功能，默认通用系统不支持复用 GPIO/I2C 功能 13.修改部分 PCIE、SATA、USB 线长约束规则
2	V1.1	<ol style="list-style-type: none"> 1. 第 2.1.4 节 HT 增加 Hostmode 处理方法、HT 总线不使用时的处理方法 2. 第 2.1.6 节 GPIO 注意事项增加多路互联 GPIO12、GPIO13 拓扑图及等长要求

		<p>3. 第 2.1.8 节 JTAG/EJTAG 增加接线示意图、多路互联拓扑图</p> <p>4. 第 2.2.4 节 DDR3 增加两条注意事项</p> <p>5. 第 3.1 节 DDR4 删除具体线宽、线距描述，增加同通道两 dimm 之间信号线等长要求</p> <p>6. 增加 3.6 节 高速信号 layout 通用设计规则</p> <p>7. 第 2.3.5 节 添加 SCI 备份连接方案</p> <p>8. 增加 2.3.7 节 SE 模块最小设计、2.3.8 节 DDR4 RESET/CKE 设计细节</p> <p>9. RTC 标准电源修改</p> <p>10. 修改公司名称</p> <p>11. 修改 LS3A4000M/LS3A4000-LL 的最大功耗</p> <p>12.增加 LS3A4000I 的最大功耗</p>
3	V1.2	1.更新 2.2.16, CLKSEL8 的推荐设计

目录

概述.....	1
第一章 主板架构.....	2
1.1 单路主板典型应用方案.....	2
1.2 双路主板典型应用方案.....	3
1.3 四路主板典型应用方案.....	4
第二章 原理图设计检查.....	5
2.1 3A4000 checklist.....	5
2.1.1 SPI.....	5
2.1.2 I2C.....	6
2.1.3 UART.....	7
2.1.4 HT.....	8
2.1.5 DDR (DDR4)	25
2.1.6 GPIO.....	37
2.1.7 Strap & Control.....	39
2.1.8 JTAG/EJTAG.....	40
2.1.9 BBG.....	42
2.1.10 POWER.....	43
2.2 7A1000 checklist.....	46
2.2.1 HT.....	46
2.2.2 PCIE.....	51
2.2.3 USB.....	61
2.2.4 DDR3.....	62
2.2.5 SATA.....	65
2.2.6 SPI.....	66
2.2.7 LPC.....	67
2.2.8 I2C.....	68
2.2.9 UART.....	69
2.2.10 GMAC.....	70
2.2.11 PWM.....	73

2.2.12 HDA.....	73
2.2.13 GPIO.....	74
2.2.14 RTC.....	74
2.2.15 DVO.....	75
2.2.16 Strap&Control.....	79
2.2.17 JTAG.....	80
2.2.18 ACPI.....	81
2.2.19 Interrupt.....	84
2.2.20 POWER.....	85
2.3 其他设计注意事项.....	86
2.3.1 时钟.....	86
2.3.2 时序要求.....	89
2.3.3 内置显卡设计.....	89
2.3.4 桥片 SPI 存储 rom.....	89
2.3.5 笔记本、一体机方案注意事项.....	90
2.3.6 接口外设支持列表.....	90
2.3.7 SE 模块最小设计.....	91
2.3.8 DDR4 RESET&CKE 设计细节.....	95
第三章 PCB 设计指导.....	101
3.1 DDR4.....	101
3.2 PCIE.....	104
3.3 USB.....	104
3.4 SATA.....	105
3.5 HT.....	106
3.6 高速信号 layout 通用设计规则.....	107

概述

本文档作为龙芯 3A4000/3B4000_7A1000 平台的单、双、四路的主板硬件设计规范，内含处理器和桥片的硬件 checklist、特殊注意事项以及 layoutguide。设计者请按规范文档认真核对硬件线路和 PCB 设计。

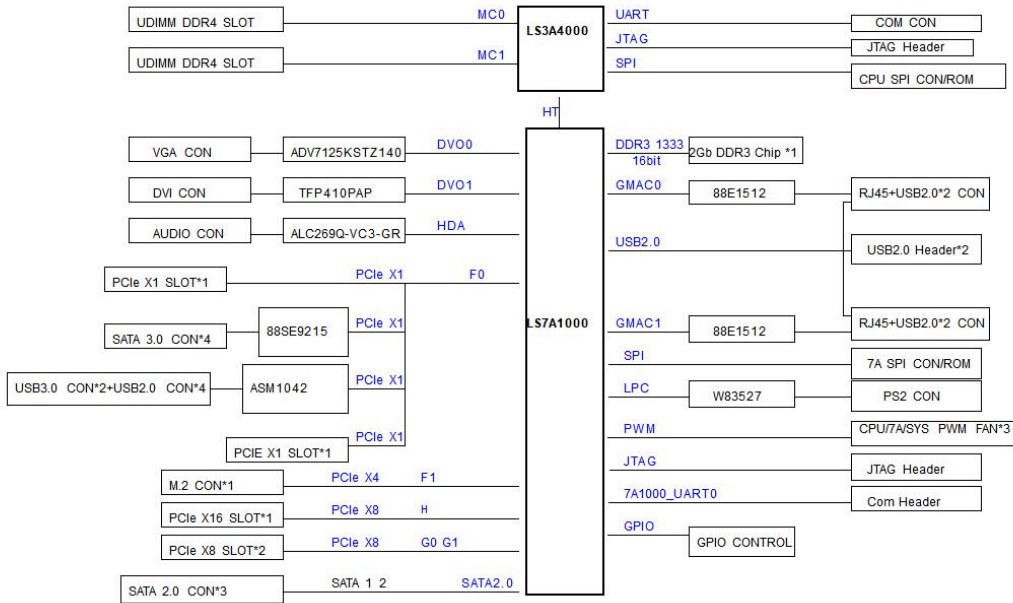
原理图、PCB 设计，推荐使用龙芯官方参考设计的 symbol，如果自己新建 symbol 进行设计，建议多次对照数据手册检查管脚正确性。另：龙芯平台 3A4000 面向单路终端方案，3B4000 面向多路互联服务器方案，功能上两者区别在于 3B4000 可保证 HT0 进行多片互联。本文档内名称上不做区分，统一成 3A4000。

如遇内容错误，欢迎批评指正。

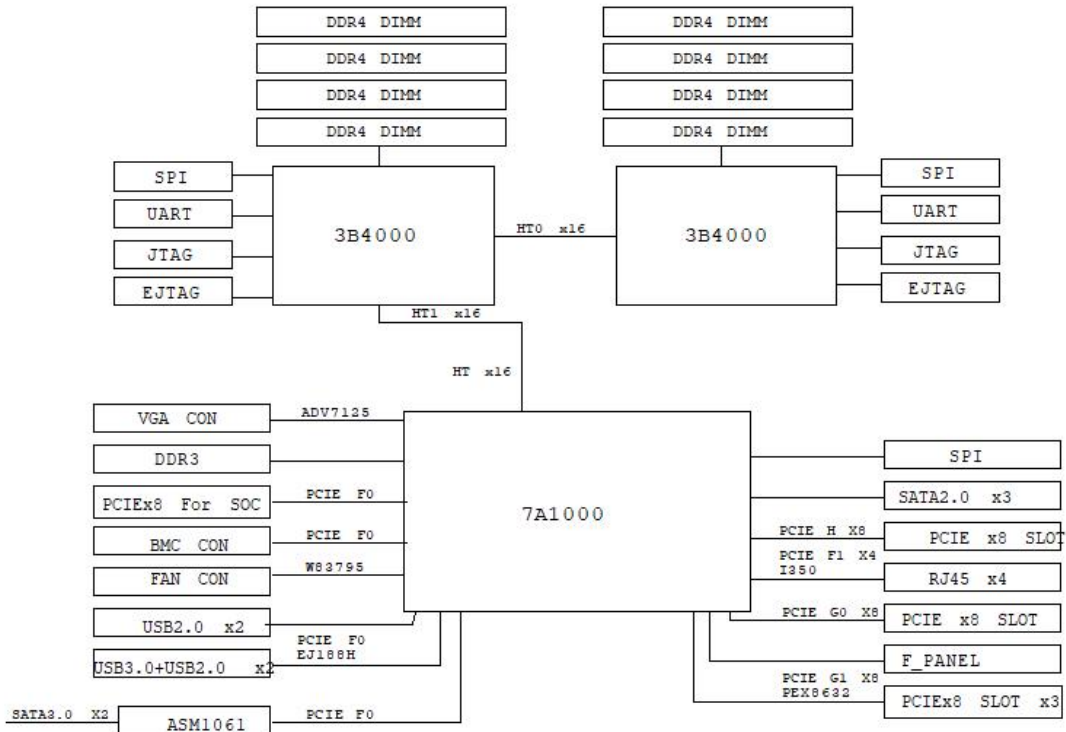
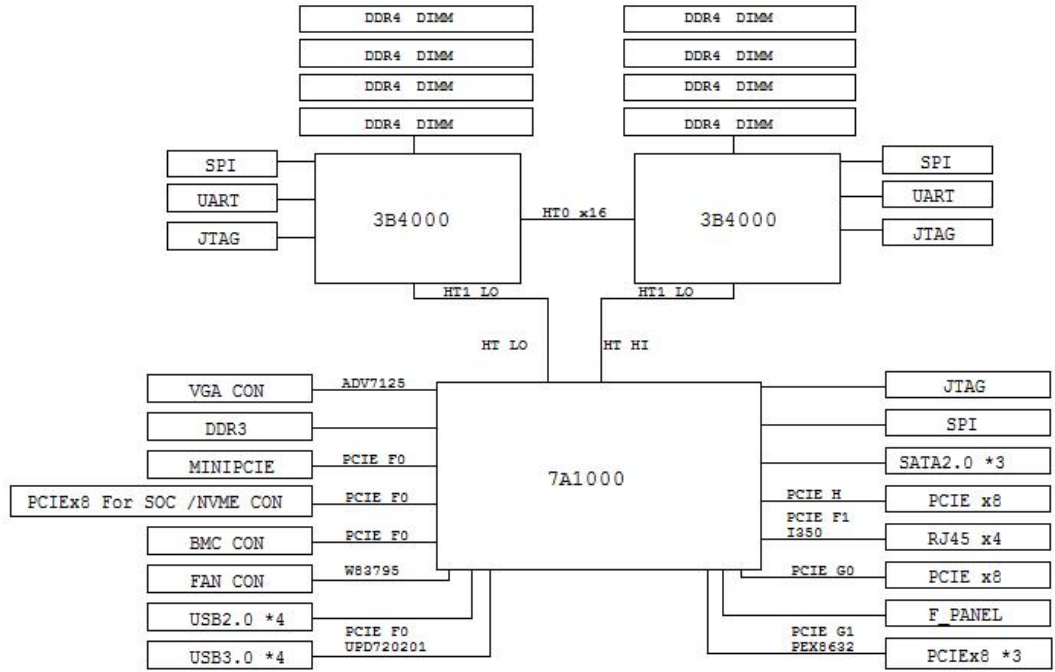
第一章 主板架构

1.1 单路主板典型应用方案

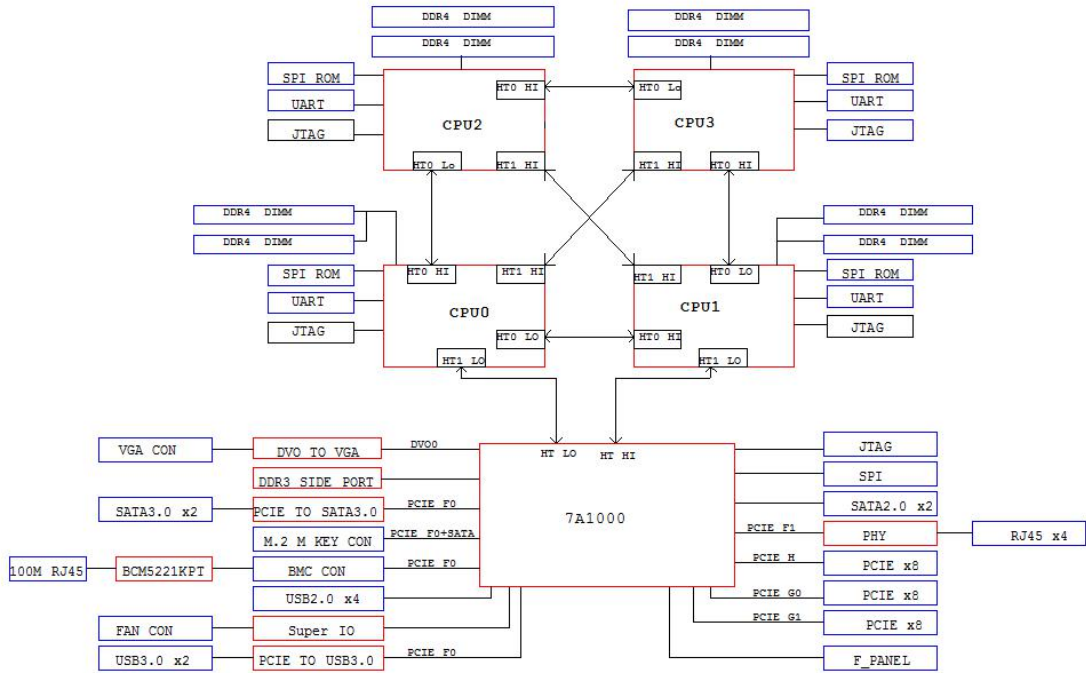
3A4000+7A1000 EVB BOARD SYSTEM ARCHITECTURE



1.2 双路主板典型应用方案



1.3 四路主板典型应用方案



第二章 原理图设计检查

2.1 3A4000 checklist

2.1.1 SPI

Signal Name	Balls	Signal Type	Checked	Recommendations
SPI_SCK	L02	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 时钟输出。 使用时外部需通过 4.7K 电阻上拉到 VDDE_IO。 不用时可悬空
SPI_SDO	L03	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 数据、命令输出。 使用时预留上拉 4.7K 电阻到 VDDE_IO。 不用时可悬空
SPI_SDI	L06	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 数据输入。 使用时预留上拉 4.7K 电阻到 VDDE_IO。 不用时可悬空
SPI_CSN	L01	O	Yes <input type="checkbox"/>	SPI 片选输出，低电平有效。

			No <input type="checkbox"/>	使用时预留上拉 4.7K 电阻到 VDDE_IO。 不使用时可悬空
SPI_HOLDN	L07	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时预留上拉 4.7K 电阻到 VDDE_IO。 不使用时可悬空
SPI_WPN	M06	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时预留上拉 4.7K 电阻到 VDDE_IO。 不使用时可悬空

注：此 SPI 通路必须连接 BIOS ROM,作为 3A4000_7A1000 的启动 ROM,容量建议为 4MB~8MB。

2.1.2 I2C

Signal Name	Balls	Signal Type	Checked	Recommendations
I2C0_SCL	K37	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	固定配置为 master, 用于本地内存槽 SPD 信息读取, 同时用于处理器 VRM 控制。 使用时外部需通过 4.7K 电阻上拉到 VDDE_IO。

I2C0_SDA	L37	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	固定配置为 master, 用于本地内存槽 SPD 信息读取, 同时用于处理器 VRM 控制。 使用时外部需通过 4.7K 电阻上拉到 VDDE_IO。
I2C1_SCL	Y02	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	固定配置为 slave, 可用于 EC、BMC 等外部控制端读取处理器温度等信息。 不用时可悬空
I2C1_SDA	Y01	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	固定配置为 slave, 可用于 EC、BMC 等外部控制端读取处理器温度等信息。 不用时可悬空

2.1.3 UART

Signal Name	Balls	Signal Type	Checked	Recommendations
UART0_TXD	M04	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	串口数据发送信号。 不用时可悬空
UART0_RXD	M05	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	串口数据接收信号。 不用时可悬空

注：3A4000_7A1000 平台调试串口基本固定为 0 号处理器的 UART0。无特殊情况，请不要随意更换。该串口亦可用作系统下普通串口功能。

2.1.4 HT

Signal Name	Balls	Signal Type	Checked	Recommendations
HT0_8x2	J36	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，将 HT0 分成 HT0_Lo 与 HT0_Hi 两个 8 位数据总线使用 为低，将 HT0 作为 16 位总线使用 单路主板可悬空； 双路主板建议外部下拉； 四路以上主板外部上拉到 VDDE_IO；
HT0_Lo_Hostmode	M31	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，将 HT0_Lo 控制器作为主模式，控制复位等信号 为低，将 HT0_Lo 控制器作为从模式，复位等信号仅为输入模式 不用时建议下拉到地。 VDDE_IO 电压域
HT0_Hi_Hostmode	J35	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，将 HT0_Hi 控制器作为主模式，控制复位等信号

				<p>为低,将HT0_Hi 控制器作为从模式, 复位等信号仅为输入模式 不用时建议下拉到地 VDDE_IO 电压域</p>
HT0_Lo_PowerOK	L35	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时为 HT0 总线 PowerOK 信号, 当 HT0_8x2 有效时为 HT0_Lo 总线 PowerOK 信号。 当 HT0_Lo_Hostmode 有效时为双向信号, 作为主设备。 当 HT0_Lo_Hostmode 无效时为输入信号。 不用时建议下拉到地 VDDE_IO 电压域</p>
HT0_Lo_Reset	M30	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时为 HT0 总线 Resetn 信号, 当 HT0_8x2 有效时为 HT0_Lo 总线 Resetn 信号。 当 HT0_Lo_Hostmode 有效时为双向信号, 作为主设备。 当 HT0_Lo_Hostmode 无效时为输入信号。</p>

				不用时处理方式见注 3 VDDE_IO 电压域
HT0_Lo_Ldt_Stopn	L34	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时为 HT0 总线 Ldt_Stopn 信号， 当 HT0_8x2 有效时为 HT0_Lo 总线 Ldt_Stopn 信号。 当 HT0_Lo_Hostmode 有效时为双向信号，作为主设备。 当 HT0_Lo_Hostmode 无效时为输入信号。 不用时必须下拉到地 VDDE_IO 电压域
HT0_Lo_Ldt_reqn	L36	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时为 HT0 总线 Ldt_Reqn 信号， 当 HT0_8x2 有效时为 HT0_Lo 总线 Ldt_Reqn 信号。 不用时建议下拉到地 VDDE_IO 电压域
HT0_Hi_PowerOK	K30	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时该信号无效， 当 HT0_8x2 有效时为 HT0_Hi 总线 PowerOK 信号。 当 HT0_Hi_Hostmode 有效时为

				<p>双向信号，作为主设备。</p> <p>当 HT0_Hi_Hostmode 无效时为输入信号。</p> <p>不用时建议下拉到地</p> <p>VDDE_IO 电压域</p>
HT0_Hi_Reset	J34	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时该信号无效，</p> <p>当 HT0_8x2 有效时为 HT0_Hi 总线 Resetn 信号。</p> <p>当 HT0_Hi_Hostmode 有效时为双向信号，作为主设备。</p> <p>当 HT0_Hi_Hostmode 无效时为输入信号。</p> <p>不用时处理方式见注 3</p> <p>VDDE_IO 电压域</p>
HT0_Hi_Ldt_Stopn	K31	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时该信号无效，</p> <p>当 HT0_8x2 有效时为 HT0_Hi 总线 Ldt_Stopn 信号。</p> <p>当 HT0_Hi_Hostmode 有效时为双向信号，作为主设备。</p> <p>当 HT0_Hi_Hostmode 无效时为输入信号</p> <p>不用时必须下拉到地</p>

HT0_Hi_Ldt_reqn	J37	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时该信号无效， 当 HT0_8x2 有效时为 HT0_Hi 总线 Ldt_Reqn 信号。 不用时建议下拉到地 VDDE_IO 电压域
HT0_Tx_CADp[15:0]	E19 C20 F20 A20 A22 A23 A25 F22 C24 A27 D22 D25 G24 G23 F26 F27	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送数据命令总线， 当 HT0_8x2 有效时， [7:0]位为 HT0_Lo 总线发送数据命令总线， [15:8]位为 HT0_Hi 总线发送数据命令总线。 不用时可悬空
HT0_Tx_CADn[15:0]	D19 B20 G20 A21 B22 B23 A24 G22 B24 B27 D23 E25 F24 H23 G26	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送数据命令总线， 当 HT0_8x2 有效时， [7:0]位为 HT0_Lo 总线发送数据命令总线， [15:8]位为 HT0_Hi 总线发送数据命令总线。

	G27			不用时可悬空
HT0_Tx_CTLp[1:0]	B30 B34	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送控制总线 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线发送控制信号， [1]位为 HT0_Hi 总线发送控制信号。 不用时可悬空
HT0_Tx_CTLn[1:0]	A30 A34	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送控制总线 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线发送控制信号， [1]位为 HT0_Hi 总线发送控制信号。 不用时可悬空
HT0_Tx_CLKp[1:0]	D37 F30	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送时钟总线， 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线发送时钟信

				号。 [1]位为 HT0_Hi 总线发送时钟信号。 不用时可悬空
HT0_Tx_CLKn[1:0]	D36 F30	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线发送时钟总线， 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线发送时钟信号， [1]位为 HT0_Hi 总线发送时钟信号。 不用时可悬空
HT0_Rx_CADp[15:0]	G36 F37 E37 C37 G34 D35 B35 G32 E33 C32 F31 A33 D30 D29 A31 G28	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线接收数据命令总线， 当 HT0_8x2 有效时， [7:0]位为 HT0_Lo 总线接收数据命令总线， [15:8]位为 HT0_Hi 总线接收数据命令总线。 不用时可悬空
HT0_Rx_CADn[15:0]	G37 F36 E36 C36	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线接收数据命令总线，

	F34 D34 A35 F32 D33 B32 E31 A32 D31 E29 B31 F28			<p>当 HT0_8x2 有效时，</p> <p>[7:0]位为 HT0_Lo 总线接收数据命令总线，</p> <p>[15:8]位为 HT0_Hi 总线接收数据命令总线。</p> <p>不用时可悬空</p>
HT0_Rx_CTLp[1:0]	A26 A28	I	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时，该总线为 HT0 总线接收控制总线，</p> <p>当 HT0_8x2 有效时，</p> <p>[0]位为 HT0_Lo 总线接收控制信号，</p> <p>[1]位为 HT0_Hi 总线接收控制信号。</p> <p>不用时可悬空</p>
HT0_Rx_CTLn[1:0]	B26 B28	I	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT0_8x2 无效时，该总线为 HT0 总线接收控制总线，</p> <p>当 HT0_8x2 有效时，</p> <p>[0]位为 HT0_Lo 总线接收控制信号，</p> <p>[1]位为 HT0_Hi 总线接收控制信号。</p> <p>不用时可悬空</p>

HT0_Rx_CLKp[1:0]	D21 D27	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线接收时钟总线， 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线接收时钟信号， [1]位为 HT0_Hi 总线接收时钟信号。 不用时可悬空
HT0_Rx_CLKn[1:0]	E21 D26	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT0_8x2 无效时，该总线为 HT0 总线接收时钟信号， 当 HT0_8x2 有效时， [0]位为 HT0_Lo 总线接收时钟信号， [1]位为 HT0_Hi 总线接收时钟信号。 不用时可悬空
HT0CLK P/N	A28 A29	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	200MHz LVDS 差分输入时钟，可通过 CLKSEL 进行选择是否用此差分时钟作为 HT 部分时钟源。默认可不接，使用 SYSCLK 作为时钟源

Signal	Ball	Sign	Check	Recommendations
--------	------	------	-------	-----------------

Name	Signal	Signal Type	Default	Description
HT1_8x2	H37	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，将 HT1 分成 HT1_Lo 与 HT1_Hi 两个 8 位数据总线使用(多路三角模式互联) 为低，将 HT1 作为 16 位总线使用(单路或双路 16 位模式)
HT1_Lo_Hostmode	H35	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，表示将 HT1_Lo 控制器作为主模式，控制复位等信号 为低，表示将 HT1_Lo 控制器作为从模式，复位等信号仅为输入模式
HT1_Hi_Hostmode	K35	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	为高，表示将 HT1_Hi 控制器作为主模式，控制复位等信号 为低，表示将 HT1_Hi 控制器作为从模式，复位等信号仅为输入模式 不用时建议下拉到地
HT1_Lo_PowerOK	H36	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时为 HT1 总线 PowerOK 信号， 当 HT1_8x2 有效时为 HT1_Lo 总线 PowerOK 信号。 当 HT1_Lo_Hostmode 有效时为双向信号，

				<p>当 HT1_Lo_Hostmode 无效时为输入信号。</p> <p>不用时建议下拉到地</p>
HT1_Lo_Resetn	H32	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时为 HT1 总线 Resetn 信号，</p> <p>当 HT1_8x2 有效时为 HT1_Lo 总线 Resetn 信号。</p> <p>当 HT1_Lo_Hostmode 有效时为双向信号，</p> <p>当 HT1_Lo_Hostmode 无效时为输入信号。</p> <p>不用时处理方式见注 3</p>
HT1_Lo_Ldt_Stopn	H34	I/O	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时为 HT1 总线 Ldt_Stopn 信号，</p> <p>当 HT1_8x2 有效时为 HT1_Lo 总线 Ldt_Stopn 信号。</p> <p>当 HT1_Lo_Hostmode 有效时为双向信号，</p> <p>当 HT1_Lo_Hostmode 无效时为输入信号。</p> <p>不用时必须下拉到地</p>

HT1_Lo_Ldt_reqn	H33	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时为 HT1 总线 Ldt_Reqn 信号， 当 HT1_8x2 有效时为 HT1_Lo 总线 Ldt_Reqn 信号。 不用时建议下拉到地
HT1_Hi_PowerOK	K36	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时该信号无效， 当 HT1_8x2 有效时为 HT1_Hi 总线 PowerOK 信号。 当 HT1_Hi_Hostmode 有效时为双向信号， 当 HT1_Hi_Hostmode 无效时为输入信号。 不用时建议下拉到地
HT1_Hi_Rstn	K32	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时该信号无效， 当 HT1_8x2 有效时为 HT1_Hi 总线 Rstn 信号。 当 HT1_Hi_Hostmode 有效时为双向信号， 当 HT1_Hi_Hostmode 无效时为输入信号。 不用时处理方式见注 3

HT1_Hi_Ldt_Stopn	K34	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>当 HT1_8x2 无效时该信号无效，</p> <p>当 HT1_8x2 有效时为 HT0_Hi 总线 Ldt_Stopn 信号。</p> <p>当 HT1_Hi_Hostmode 有效时为双向信号，</p> <p>当 HT1_Hi_Hostmode 无效时为输入信号</p> <p>不用时必须下拉到地</p>
HT1_Hi_Ldt_reqn	K33	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>当 HT1_8x2 无效时该信号无效，</p> <p>当 HT1_8x2 有效时为 HT1_Hi 总线 Ldt_Reqn 信号。</p> <p>不用时建议下拉到地</p>
HT1_Tx_CADp[15:0]	G02 F01 E01 C01 G04 D03 B03 G06 E05 C06 F07 A05 D08 D09 A07 G10	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送数据命令总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[7:0]位为 HT1_Lo 总线发送数据命令总线，</p> <p>[15:8]位为 HT1_Hi 总线发送数据命令总线。</p> <p>不用时可悬空</p>
HT1_Tx_CADn[15:0]	G01 F02 E02 C02	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送数据命令总线，</p>

	F04 D04 A03 F06 D05 B06 E07 A06 D07 E09 B07 F10			<p>当 HT1_8x2 有效时，</p> <p>[7:0]位为 HT1_Lo 总线发送数据命令总线，</p> <p>[15:8]位为 HT1_Hi 总线发送数据命令总线。</p> <p>不用时可悬空</p>
HT1_Tx_CTLp[1:0]	B04 B08	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送控制总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[0]位为 HT1_Lo 总线发送控制信号，</p> <p>[1]位为 HT1_Hi 总线发送控制信号。</p> <p>不用时可悬空</p>
HT1_Tx_CTLn[1:0]	A04 B08	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送控制总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[0]位为 HT1_Lo 总线发送控制信号，</p> <p>[1]位为 HT1_Hi 总线发送控制信号。</p> <p>不用时可悬空</p>

HT1_Tx_CLKp[1:0]	D01 F08	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送时钟总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[0]位为 HT1_Lo 总线发送时钟信号，</p> <p>[1]位为 HT1_Hi 总线发送时钟信号。</p> <p>不用时可悬空</p>
HT1_Tx_CLKn[1:0]	D02 G08	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线发送时钟总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[0]位为 HT1_Lo 总线发送时钟信号，</p> <p>[1]位为 HT1_Hi 总线发送时钟信号。</p> <p>不用时可悬空</p>
HT1_Rx_CADp[15:0]	B19 C18 F18 A18 A16 A15 A13 F16 C14 A11 D16 D13	1	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT1_8x2 无效时，该总线为 HT1 总线接收数据命令总线，</p> <p>当 HT1_8x2 有效时，</p> <p>[7:0]位为 HT1_Lo 总线接收数据命令总线，</p> <p>[15:8]位为 HT1_Hi 总线接收数据</p>

	G14 G15 F12 F11			命令总线。 不用时可悬空
HT1_Rx_CADn[15:0]	A19 B18 G18 A17 B16 B15 A14 G16 B14 B11 D15 E13 F14 H15 G12 G11	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时，该总线为 HT1 总线接收数据命令总线， 当 HT1_8x2 有效时， [7:0]位为 HT1_Lo 总线接收数据命令总线， [15:8]位为 HT1_Hi 总线接收数据命令总线。 不用时可悬空
HT1_Rx_CTLp[1:0]	A12 C10	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时，该总线为 HT1 总线接收控制总线， 当 HT1_8x2 有效时， [0]位为 HT1_Lo 总线接收控制信号， [1]位为 HT1_Hi 总线接收控制信号。 不用时可悬空
HT1_Rx_CTLn[1:0]	B12 B10	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时，该总线为 HT1 总线接收控制总线， 当 HT1_8x2 有效时， [0]位为 HT1_Lo 总线接收控制信

				号， [1]位为 HT1_Hi 总线接收控制信号。 不用时可悬空
HT1_Rx_CLKp[1:0]	D17 D11	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时，该总线为 HT1 总线接收时钟总线， 当 HT1_8x2 有效时， [0]位为 HT1_Lo 总线发送时钟信号， [1]位为 HT1_Hi 总线发送时钟信号。 不用时可悬空
HT1_Rx_CLKn[1:0]	E17 D12	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT1_8x2 无效时，该总线为 HT1 总线接收时钟总线， 当 HT1_8x2 有效时， [0]位为 HT1_Lo 总线接收时钟信号， [1]位为 HT1_Hi 总线接收时钟信号。 不用时可悬空
HT1CLK P/N	A10 A09	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	200MHz LVDS 差分输入时钟，可通过 CLKSEL 进行选择是否用此差

				分时钟作为 HT 部分时钟源。默认可不接，使用 SYSCLK 作为时钟源
--	--	--	--	--------------------------------------

注：

1、 HT0 仅作为处理器间互联使用，不可用于与桥片互联。对于单路和双路方案，HT1 作为与桥片互联使用。对于四路方案，HT1 的高 8 位作为处理器间交叉互联，低 8 位与桥片互联。详细 HT 拓扑结构见第一章主板典型结构。

2、 在多路互联中，相连的两组 HT 必须有一个 HT_HOSTMODE 为高，允许都为高。

3、 HyperTransport 总线不使用时可以悬空。需要注意的是：

- a. 不使用的 HT 通道，LDT_STOP# 必须为低。
- b. 单路和双路设计中，不使用的 HT 通道，RESETn 信号建议下拉；四路设计中，不使用的 HT 通道，RESETn 信号不能下拉。

2.1.5 DDR (DDR4)

Signal Name	Balls	Signal Type	Checked	Recommendations
MC0_DQ [0:63]	AH0 1AH 04AL 02 AL01 AH0 3AH 02AK 01AL	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的数据总线信号

03			
AM0			
1AM			
03AP			
02			
AR0			
1AL0			
4			
AM0			
2AP0			
1AM			
04AP			
04			
AT02			
AU0			
4AR			
05A			
R02			
AP03			
AP05			
AT04			
AP07			
AR0			
7AU			
08A			
R08			
AT05			
AU0			
5AP0			
9			
AT08			
AR3			
0AP3			
0AU			
33A			
R33			
AU3			
0AT3			
0			
AP32			
AT33			
AT34			
AR3			
4AR			

	35A R37 AP33 AU3 4AT3 6 AR3 6AP3 7AN 34AL 36 AL37 AN3 5AP3 6AM 37A M36 AK37 AJ34 AG3 6AG 37 AK35 AK36 AH3 7AH 36			
MC0_CB [0:7]	AP11 AR1 1 AU1 2 AR1 4 AT09 AU0 9 AP12 AT12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的数据总线 ECC 信号 做 PC 设计时不接
MC0_DQ Sp[0:17]	AK02 AN0 4 AU0 3	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的数据选通差分信号 (包括 ECC)，正沿

	AU0 7 AU3 2 AP35 AM3 4 AH3 5 AU1 1 AJ02 AN0 2 AR0 3 AU0 6 AU3 1 AU3 5 AN3 6 AJ36 AU1 0			其中 DQSp[9:17] 复用为 DQM[0:8], 即数据屏蔽信号 DM0-8 (包括 ECC) 其中 DQSp8、DQSp17 在做 PC 设计时不接
MC0_DQ Sn[0:17]	AK03 AN0 3 AT03 AT07 AT32 AP34 AM3 5 AJ35 AT11 AJ01 AN0 1 AR0 4 AT06	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的数据选通差分信号 (包括 ECC), 负沿 其中 DQSn[8:17] PC 设计时不接

	AT31 AT35 AN3 7 AJ37 AT10			
MC0_A[0:13] MC0_A17	AR23 AP20 AR20 AU18 AT18 AR18 AU17 AR17 AT17 AT16 AR24 AU16 AP18 AU27 AP27	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的地址总线信号
MC0_We n/A14	AT25	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的写使能信号，复用地址总线 A14
MC0_CASn/A15	AU26	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的列地址选择信号，复用地址总线 A15
MC0_RASn/A16	AP24	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的行地址选择信号，复用地址总线 A16

MC0_BA [0:1]	AU2 4AT2 4	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的逻辑 Bank 地址 信号
MC0_BG [0:1]	AP17 AT15	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的逻辑 Bank 组地 址信号
MC0_CS n[0:7]	AT26 AT27 AR2 9 AU2 9 AU2 5 AT23 AU2 2 AT29	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的片选信号
MC0_CK E[0:3]	AU1 3 AP15 AT14 AT13	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的时钟使能信号
MC0_OD T[0:3]	AP26 AT28 AR2 7 AP23	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的 ODT 输出信号
MC0_CL KP[0:3]	AU2 0 AU1 9 AU2 1 AR2 1	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的差分时钟输出信 号, 正沿
MC0_CL KN[0:3]	AT20 AT19 AT21 AP21	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的差分时钟输出信 号, 负沿

MC0_Re setn	AT22	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的复位输出信号
MC0_AC TN	AU1 4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的激活命令信号
MC0_AL ERTN	AU1 5	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的 CRC 错或命令奇 偶校验错标志
MC0_CI D	AU2 8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的 CHIP ID bit2 , 3DS 内存使用
MC0_PA R	AU2 3	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的命令与地址的奇偶 校验值
MC0_RE XT	AP14	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 0 的参考输入电阻, 通 过一个 240ohm(1%)的电阻连接 至地。
MC1_DQ [0: 63]	Y32 Y33 AB36 AB37 Y30 Y31 AA3 7 AA3 6 AB31 AB32 AC36 AC35	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的数据总线信号

AB34			
AB33			
AC33			
AC34			
AD3			
6			
AD3			
7			
AF33			
AF30			
AC37			
AD3			
5			
AE37			
AE36			
AF34			
AF35			
AJ32			
AJ33			
AF32			
AF31			
AH3			
0			
AH3			
1			
AN0			
9			
AL09			
AL06			
AM0			
5			
AK09			
AM0			
9			
AN0			
6			
AM0			
6			
AH0			
8			
AJ08			
AG0			
2			
AG0			

	1 AN0 5 AL05 AJ07 AH0 7 AF05 AF06 AE05 AE04 AF07 AF08 AE01 AE02 AD0 3 AE08 AC07 AC06 AE07 AE06 AC05 AC08			
MC1_CB [0:7]	AL33 AL32 AN3 0 AM3 0 AJ31 AJ30 AL30 AK30	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的数据总线 ECC 信号 做 PC 设计时不接
MC1_DQ SP[0:17]	Y35 AB30 AE31 AF36 AM0 8 AJ05 AF04 AD0 2	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的数据选通差分信号 (包括 ECC)，正沿 其中 DQSP[9:17] 复用为 DQM[0:8]，即数据屏蔽信号 DM0-8 (包括 ECC)

	AM3 2 Y37 AC31 AE32 AH3 2 AK08 AH0 5 AF02 AC01 AN3 2			其中 DQSp8、DQSp17 在做 PC 设计 计时不接
MC1_DQ SN[0:17]	Y36 AC32 AE33 AH3 3 AL08 AH0 6 AF01 AC02 AN3 3 Y34 AC30 AE30 AF37 AN0 8 AJ06 AF03 AD0 1 AM3 3	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的数据选通差分信号 (包括 ECC)，负沿 其中 DQSn[8:17] PC 设计时不接
MC1_A[0:13] MC1_A1 7	AK17 AM2 3 AN2 3 AK22	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的地址总线信号

	AN2 4 AM2 4 AL23 AL24 AK23 AM2 6 AM1 5 AN2 6 AL26 AL12 AK12			
MC1_We n/A14	AN1 5	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的写使能信号，复用 地址总线 A14
MC1_CA Sn/A15	AN1 4	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的列地址选择信号， 复用地址总线 A15
MC1_RA Sn/A16	AL15	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的行地址选择信号， 复用地址总线 A16
MC1_BA [0:1]	AL17 AN1 7	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的逻辑 Bank 地址 信号
MC1_BG [1:0]	AL25 AM2 5	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的逻辑 Bank 地址 组信号
MC1_CS n[0;7]	AM1 4 AM1 2 AN1 1 AK11 AK15	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的片选信号

	AM1 8 AK18 AL11			
MC1_CK E[0:3]	AN2 9 AM2 9 AL27 AL29	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的时钟使能信号
MC1_OD T[0:3]	AL14 AM1 1 AK14 AN1 8	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的 ODT 输出信号
MC1_CL Kp[3:0]	AL21 AM2 1 AM2 0 AL20	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的差分时钟输出信号，正沿
MC1_CL Kn[3:0]	AK21 AN2 1 AN2 0 AK20	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的差分时钟输出信号，负沿
MC1_Re setn	AL18	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的复位输出信号
MC1_AC TN	AL28	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的激活命令信号
MC1_AL ERTN	AN2 7	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的 CRC 错或命令奇偶校验错标志
MC1_CI D2	AN1 2	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的 CHIP ID bit2， 3DS 内存使用

MC1_PARR	AM17	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的命令与地址的奇偶校验值
MC1_REXT	AK29	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR 控制器 1 的参考输入电阻，通过一个 240ohm(1%)的电阻连接至地

注：

1、DDR 的复位信号 MC_RESETN 需要经过反向器后连接到 DIMM 条，可以参考评估板使用三极管做反向。

2、每个内存控制器有 8 个 CS 信号 MC_SCSN[7:0]，连线时要求 MC_SCSN[3:0] 连接到第一个 DIMM 条，MC_SCSN[7:4] 连接到第二个 DIMM 条。具体分组要求如下：

DIMM0	MC_SCSN[0:3], ODT0/1, CKE0/1, CLKn/p[0:1]
DIMM1	MC_SCSN[4:7], ODT2/3, CKE2/3, CLKn/p[2:3]

DIMM 条 SPD 地址需固定，MC0 为 0,1；MC1 为 2,3。多路模式下每个处理器对应的 DIMM 条连接自己的 I2C 总线。

3、不支持数据信号组内互换。

4、RESET&CKE 设计细节参考 2.3.8 小节。

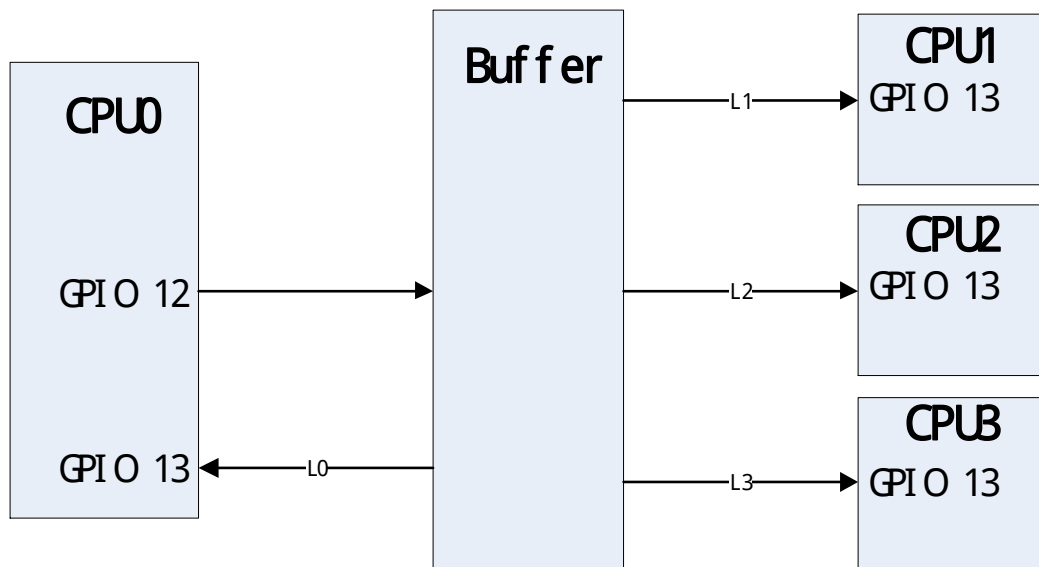
2.1.6 GPIO

Signal Name	Balls	Signal	Checked	Recommendations
-------------	-------	--------	---------	-----------------

		Type		
GPIO[15:0]	T04 T02 T03 T01 U06 U07 U05 U02 U03 U01 V06 V07 V04 V02 V03 V01	IO	Yes <input type="checkbox"/> No <input type="checkbox"/>	默认为高阻态 不用时可悬空 VDDE_IO 电压域

注：

1、两路、四路服务器设计要求：每个 CPU 的 GPIO13 与 CPU0 的 GPIO12 通过 buffer 连在一起，并做等长处理，L0、L1、L2、L3 等长要求 50mils。



2、GPIO0 固定复用为外部中断输入，默认预留跟 7A1000 的 INTO 连接。

3、通用类产品系统默认不支持 GPIO 复用串口功能。如需使用 UART1 功能，需修改内核进行支持。

2.1.7 Strap & Control

Signal Name	Balls	Signal Type	Checked	Recommendations
SYSRESETn	T05	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	处理器复位输入信号，低电平有效。 VDDE_IO 电压域
CHIP_CONFIG[5:0]	N06 N03 N01 N02 P07 P05	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	CHIP_CONFIG[5:0] 推荐配置 6'b000010 详细定义见手册 VDDE_IO 电压域
ICCC_EN	M01	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	多路 cache 一致性使能，多路模式下需上拉。 VDDE_IO 电压域
NODE_ID[2:0]	M02, N07, N04	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	处理器节点指示 000,处理器为 0 号节点 001,处理器为 1 号节点 111 处理器为 7 号节点 VDDE_IO 电压域
NMIIn	M03	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	不可屏蔽中断，上拉处理。 VDDE_IO 电压域
CLKSEL[9:0]	P06 P03 P01	I	Yes <input type="checkbox"/>	CLKSEL[1:0]推荐配置 2'b10。

	P02 R06 R04 R05 R03 R01 R02		No <input type="checkbox"/>	CLKSEL[3:2]推荐配置 2'b10。 CLKSEL[4]根据硬件 SYSCLK 时钟而定, 1-参考时钟采用 25MHz, 0-参考时钟采用 100MHz。 CLKSEL[9:5]推荐配置 5'b01010 更详细定义见手册 VDDE_IO 电压域
SYSCLK	T06	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	处理器时钟输入, 25M/100M 可选。 多路设计时要求同源。(推荐使用 100MHz 时钟)

注：3A4000 处理器的所有 strap 均上拉到 S0 电，主板复位时状态必须稳定。

2.1.8 JTAG/EJTAG

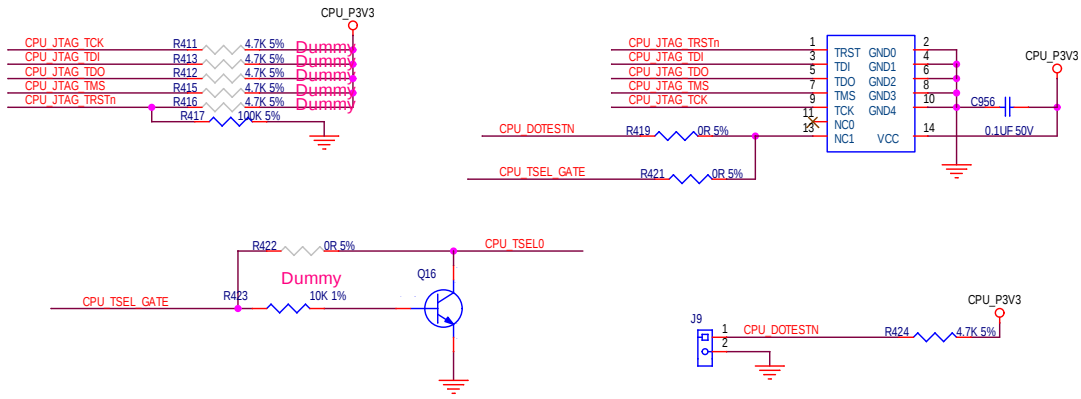
Signal Name	Balls	Signal Type	Checked	Recommendations
TDI	W02	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	不用时可悬空 VDDE_IO 电压域
TDO	W03	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	不用时可悬空 VDDE_IO 电压域
TMS	Y06	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	不用时可悬空 VDDE_IO 电压域

TRST	Y07	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	不用时可悬空 VDDE_IO 电压域
TCK	W01	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	不用时可悬空 VDDE_IO 电压域
TSEL[1:0]	W06, Y03	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 功能选择： 2'b00: GS464V EJTAG 2'b01: JTAG 不用时可悬空 VDDE_IO 电压域
DOTESTn	W05	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	正常模式上拉。 JTAG 模式需要外部拉低。 VDDE_IO 电压域

注：EJTAG 与 JTAG 复用调试接口，EJTAG 用于调试底层应用软件；JTAG 调试接口用于芯片测试，通过 TSEL[1:0]设置输出。如无特殊要求，尽量将调试接口引出。为了兼容龙芯现有仿真器，EJTAG 接口尽量按照参考板设计。可通过软件或者芯片的 FUSE 禁用。

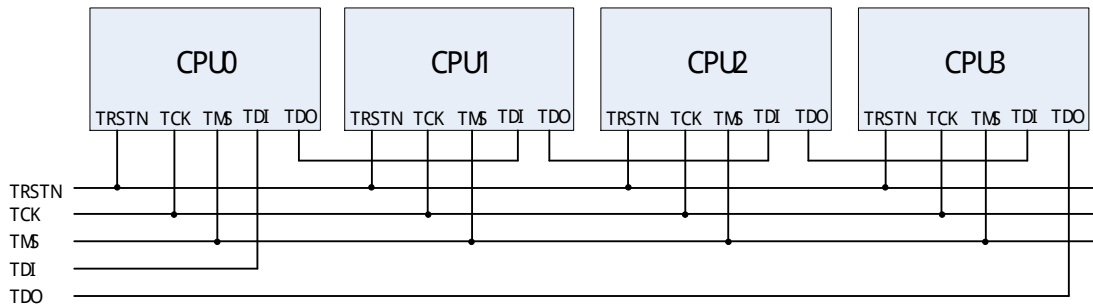
JTAG/EJTAG 接线示意图：

1) 连接示意图



注：CPU_DOTESTN：CPU_PIN.W05，CPU_TSEL：CPU_PIN.W06

2) 多路互联拓扑图



2.1.9 BBG

Signal Name	Balls	Signal Type	Checked	Recommendations
BBG_VD DST[0:3]	V26	O	Yes <input type="checkbox"/>	偏压输出正
	V12 Y26 Y12		No <input type="checkbox"/>	不可悬空
BBG_GN DST[0:3]	V27	O	Yes <input type="checkbox"/>	偏压输出负
	V11 Y27 Y11		No <input type="checkbox"/>	不可悬空
BBG_VD DSR[0:3]	U26 U12 AA2	I	Yes <input type="checkbox"/>	偏压输入正

	6 AA1 2		No <input type="checkbox"/>	不可悬空
BBG_GN DSR[0:3]	U27 U11 AA2 7 AA1 1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	偏压输入负 不可悬空

注：输出正(BBG_VDDST)接输入正(BBG_VDDSR),输出负(BBG_GNDST)接输入负(BBG_GNDSR)为正向偏压；可在 BIOS 下进行设置不偏压或者正偏压。

服务器、台式机等功耗不敏感设计推荐正偏连接方式

输出正(BBG_VDDST)接输入负(BBG_GNDSR),输出负(BBG_GNDST)接输入正(BBG_VDDSR)为反偏。可在 BIOS 下进行设置不偏压或者反偏压。**3a4000M 版 cpu 不支持偏压。**

增加正向偏压有利于提升频率,但是会增大功耗。增加反偏电压有利于降低功耗。

2.1.10 POWER

芯片型号	电源名称	标准电压 (V)	最大功耗 (W)	适用频率
LS3A4000M (笔记本处理器)	VDDN	0.95~1.15V	45	1.5G
LS3A4000-LL (一体机处理器)		1.2~1.25V	45	1.7G
LS3A4000 (台式机处理器)		1.25V	60	1.8G(睿频 2.0G)

LS3B4000 (服务器处理器)		1.25V	70	1.8G
LS3A4000-I (工业级处理器)		1.15V	45	1.5G
		1.0V	45	1.2G

电源名称	标准电压 (V)	实测最大电流 (A)	备注
VDDP	1.3V	2.2	笔记本和一体机
		<10	台式机和服务器
VDDE_IO VDDE_3V3	3.3V	0.5	VDDE_IO 和 VDDE_3V3 支持 3.3V 和 1.8V 供 电模式,且必须使 用相同供电。例 外,此电源域下的 信号电压电平范 围随供电模式而 改变。与外部相连 时需考虑电平转 换,涉及到的信号 组有 GPIO、HT
	1.8V	0.5	

			的控制信号、 I2C、UART、SPI。
MEM_VDDE_0/1	1.5V		DDR3
	1.2V	1	DDR4
MEM_VREF_0/1	0.75V		
	0.6V	0.3	建议使用内部参 考电压
HT_VDDE	1.2V	1	
VDDE_1V8	1.8V	0.5	
PLL_SYS_AVDD	1.15V	0.1	建议使用单独的 LDO 电源供电。
PLL_SYS_DVDD	1.15V		
PLL_DDR_AVDD	1.15V		
PLL_DDR_DVDD	1.15V		
PLL_DDRPHY_VDD	1.15V		
PLL_SE_VDD	1.15V		
PLL_HT0/1_AVDD	1.15V		
PLL_HT0/1_DVDD	1.15V		

注：

- 1、除特殊说明外，所有电源大负载下纹波控制在 $\pm 25\text{mv}$ 以内。
- 2、做 STR 功能时，唤醒时需要 VDDP 早于 MEM_VDDE_0/1 上电，进入 STR 时需要 VDDP 晚于 MEM_VDDE_0/1 掉电。
- 3、VDDN 和 VDDP 设计上建议都支持动态调压。

2.2 7A1000 checklist

2.2.1 HT

Signal Name	Balls	Signal Type	Checked	Recommendations
HT_REXT	H19	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	HT 参考电阻，通过一个 1Kohm(1%)的电阻接地。
HT_8X2	A11	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	不可悬空 为低，HT 作为 16 位总线使用 为高，HT 作为 HT_Hi 和 HT_Lo 两个 8 位总线使用
HT_LO_POWEROK	G14	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时为 HT 总线 PowerOK 信号， 当 HT_8X2 有效时为 HT_Lo 总线 PowerOK 信号
HT_LO_RSTn	G13	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时为 HT 总线 Rstn 信号， 当 HT_8X2 有效时为 HT_Lo 总线 Rstn 信号。
HT_LO_LDT_STOPn	F13	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时为 HT 总线 Ldt_Stopn 信号，

				当 HT_8X2 有效时为 HT_Lo 总线 Ldt_Stopn 信号。
HT_LO_LDT_REQn	A12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时为 HT 总线 Ldt_Reqn 信号， 当 HT_8X2 有效时为 HT_Lo 总线 Ldt_Reqn 信号。
HT_HI_POWEROK	C12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时该信号无效， 当 HT_8X2 有效时为 HT_Hi 总线 PowerOK 信号。
HT_HI_RSTn	E12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时该信号无效， 当 HT_8X2 有效时为 HT_Hi 总线 Resetn 信号。
HT_HI_LDT_STOPn	B12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时该信号无效， 当 HT_8X2 有效时为 HT_Hi 总线 Ldt_Stopn 信号。
HT_HI_LDT_REQn	F12	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时该信号无效， 当 HT_8X2 有效时为 HT_Hi 总线 Ldt_Reqn 信号。
HT_TX_CADP[00~15]	E13 B13 E14 B14 E15 G15 B16 E16	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线发送数据命令总线， 当 HT_8X2 有效时， [7:0]位为 HT_Lo 总线发送数据命

	E17 B18 E18 G17 E19 B20 E20 G19			令总线， [15:8]位为 HT_Hi 总线发送数据命令总线。
HT_TX_C ADN[00 ~15]	D13 A13 D14 A14 D15 G16 A16 D16 D17 A18 D18 G18 D18 A20 D20 G20	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线发送数据命令总线， 当 HT_8X2 有效时， [7:0]位为 HT_Lo 总线发送数据命令总线， [15:8]位为 HT_Hi 总线发送数据命令总线。
HT_TX_C TLP[0~1]	B17 B21	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线发送控制总线， 当 HT_8X2 有效时， [0]位为 HT_Lo 总线发送控制信号， [1]位为 HT_Hi 总线发送控制信号。
HT_TX_C TLN[0~ 1]	A17 A21	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线发送控制总线， 当 HT_8X2 有效时，

				<p>[0]位为 HT_Lo 总线发送控制信号，</p> <p>[1]位为 HT_Hi 总线发送控制信号。</p>
HT_TX_C LKP[0~1]	B15 B19	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT_8X2 无效时，该总线为 HT 总线发送时钟总线，</p> <p>当 HT_8X2 有效时，</p> <p>[0]位为 HT_Lo 总线发送时钟信号，</p> <p>[1]位为 HT_Hi 总线发送时钟信号。</p>
HT_TX_C LKN[0~ 1]	A15 A19	0	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT_8X2 无效时，该总线为 HT 总线发送时钟总线，</p> <p>当 HT_8X2 有效时，</p> <p>[0]位为 HT_Lo 总线发送时钟信号，</p> <p>[1]位为 HT_Hi 总线发送时钟信号</p>
HT_RX_ CADP[0 0~15]	F29 F30 E29 C30 B29 E27 B28 E26 E25 B26	1	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>当 HT_8X2 无效时，该总线为 HT 总线接收数据命令总线，</p> <p>当 HT_8X2 有效时，</p> <p>[7:0]位为 HT_Lo 总线接收数据命令总线，</p>

	E24 B25 G21 E23 B23 E22			[15:8]位为 HT_Hi 总线接收数据命令总线。
HT_RX_CADN[0~15]	F28 E30 D29 B30 A29 D27 A28 D26 D25 A26 D24 A25 G22 D23 A23 D22	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时, 该总线为 HT 总线接收数据命令总线, 当 HT_8X2 有效时, [7:0]位为 HT_Lo 总线接收数据命令总线, [15:8]位为 HT_Hi 总线接收数据命令总线。
HT_RX_CTLP[0~1]	B27 B22	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时, 该总线为 HT 总线接收控制总线, 当 HT_8X2 有效时, [0]位为 HT_Lo 总线接收控制信号, [1]位为 HT_Hi 总线接收控制信号。
HT_RX_CTLN[0~1]	A27 A22	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时, 该总线为 HT 总线接收控制信号, 当 HT_8X2 有效时, [0]位为 HT_Lo 总线接收控制信

				号， [1]位为 HT_Hi 总线接收控制信号。
HT_RX_CLKN[0~1]	D28 A24	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线接收时钟总线， 当 HT_8X2 有效时， [0]位为 HT_Lo 总线接收时钟信号， [1]位为 HT_Hi 总线接收时钟信号。
HT_RX_CLKP[0~1]	E28 B24	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	当 HT_8X2 无效时，该总线为 HT 总线接收时钟总线， 当 HT_8X2 有效时， [0]位为 HT_Lo 总线接收时钟信号， [1]位为 HT_Hi 总线接收时钟信号。

2.2.2 PCIE

Signal Name	Balls	Signal	Checked	Recommendations
-------------	-------	--------	---------	-----------------

		Type		
PCIE_G0_TXP[7:0]	AK14 AK13 AK12 AK11 AK10 AK9 AK8 AF8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	G0 通路 PCIe x8 差分数据输出正。 当 PCIE_G0_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G0_TXN[7:0]	AJ14 AJ13 AJ12 AJ11 AJ10 AJ9 AJ8 AG8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	G0 通路 PCIe x8 差分数据输出负。 当 PCIE_G0_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G0_RXP[7:0]	AF14 AD1 6 AF13 AD1 4 AF11 AF10 AF9 AD1 2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	G0 通路 PCIe x8 差分数据输入正。 当 PCIE_G0_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近设备端摆放 0.1uf 的隔直电容。 不用时可悬空。

PCIE_G0_RXN[7:0]	AG1 4 AD1 5AG 13A D13 AG1 1AG 10A G9 AD1 1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	G0 通路 PCIe8 差分数据输入负。 当 PCIE_G0_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G0_REFRES	AC15	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	外部参考电阻，通过一个 200ohm(1%)的电阻连接至地
PCIE_G0_PRSENTn[1:0]	AC13 AC12	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	控制器使能信号，默认全高 X8 模式时, PCIE_G0_PRSENTn1 必须置高。PCIE_G0_PRSENTn0 作为控制使能信号，当设备接入时置低。 X4 模式时, PCIE_G0_PRSENTn0 作为数据总线的 0~3 使能信号，当有设备接入时置低。 PCIE_G0_PRSENTn1 作为数据总线的 4~7 使能信号，当有设备接入时置低。 不用时建议外部上拉到 3.3VS0。
PCIE_G0_RSTn	AC11	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	G0 通路 PCIe 复位信号，低电平有效。外部预留 4.7K 电阻上拉到 3.3VS0

PCIE_G1_TXP[7:0]	AF1 AJ1 AK2 AK3 AK5 AK4 AK6 AK7	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	G1 通路 PCIe8 差分数据输出正。 当 PCIE_G1_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 电容。 不用时可悬空。
PCIE_G1_TXN[7:0]	AE1 AH1 AJ2 AJ3 AJ5 AJ4 AJ6 AJ7	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	G1 通路 PCIe8 差分数据输出负。 当 PCIE_G1_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G1_RXP[7:0]	AF2 AF3 AE3 AF4 AF5 AF6 AF7 AE7	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	G1 通路 PCIe8 差分数据输入正。 当 PCIE_G1_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近设备端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G1_RXN0	AG2 AG3 AE4 AG4 AG5 AG6 AG7 AE8	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	G1 通路 PCIe8 差分数据输入负。 当 PCIE_G1_PRSENTn1 为低时，此通路拆分为 0~3 和 4~7 两个 X4 通路使用。

				靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_G1_REFRES	AE10	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	外部参考电阻，通过一个 200ohm(1%)的电阻连接至地
PCIE_G1_PRSNTn[1:0]	AE9 AD1 0	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	控制器使能信号，默认全高 X8 模式时, PCIE_G1_PRSNTn1 必须置高。PCIE_G1_PRSNTn0 作为控制使能信号，当设备接入时置低。 X4 模式时, PCIE_G1_PRSNTn0 作为数据总线的 0~3 使能信号，当有设备接入时置低。 PCIE_G1_PRSNTn1 作为数据总线的 4~7 使能信号，当有设备接入时置低。 不用时建议外部上拉到 3.3VS0 处理。
PCIE_G1_RSTn	AD9	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	G1 通路 PCIE 复位信号，低电平有效。外部预留 4.7K 电阻上拉到 3.3VS0。
PCIE_H_TXP[7:0]	AF30 AJ30 AK29 AK28 AK27 AK26	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	H 通路 PCIe8 差分数据输出正。 当 PCIE_H_PRSNTn1 为低时, 此通路拆分为 0~3 和 4~7 两个 X4 通路

	AK25 AK24			使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_H_TXN[7:0]	AE30 AH3 0 AJ29 AJ28 AJ27 AJ26 AJ25 AJ24	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	H 通路 PCIe8 差分数据输出负。 当 PCIE_H_PRSENTn1 为低时,此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_H_RXP[7:0]	AE29 AE27 AF29 AF28 AE25 AF26 AF24 AF25	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	H 通路 PCIe8 差分数据输入正。 当 PCIE_H_PRSENTn1 为低时,此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近设备端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_H_RXN0	AE28 AE26 AG2 9 AG2 8 AE24 AG2 6 AG2 4 AG2 5	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	H 通路 PCIe8 差分数据输入负。 当 PCIE_H_PRSENTn1 为低时,此通路拆分为 0~3 和 4~7 两个 X4 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。

PCIE_H_REFRES	AC21	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	外部参考电阻，通过一个 200ohm(1%)的电阻连接至地
PCIE_H_PRSNTn[1:0]	AD2 2AD 23	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	控制器使能信号，默认全高 X8 模式时，PCIE_H_PRSNTn1 必须置高。PCIE_H_PRSNTn0 作为控制使能信号，当设备接入时置低。 X4 模式时，PCIE_H_PRSNTn0 作为数据总线的 0~3 使能信号，当有设备接入时置低。 PCIE_H_PRSNTn1 作为数据总线的 4~7 使能信号，当有设备接入时置低。 不用时建议外部上拉到 3.3VS0 处理。
PCIE_H_RSTN	AE23	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	H 路 PCIE 复位信号，低电平有效。 建议外部预留 4.7K 电阻上拉到 3.3VS0。
PCIE_F0_TXP[3:0]	AK23 AK22 AK21 AK20	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	F0 通路 PCIe4 差分数据输出正。 当 PCIE_F0_PRSNTn[1:3] 其中有一个为低时，此通路拆分为 4 个 X1 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。

				不用时可悬空。
PCIE_F0_TXN[3:0]	AJ23 AJ22 AJ21 AJ20	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	F0 通路 PCIe4 差分数据输出负。 当 PCIE_F0_PRSENTn[1:3] 其中有一个为低时，此通路拆分为 4 个 X1 通路使用。 靠近发送端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_F0_RXP[3:0]	AF22 AF23 AF21 AF20	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	F0 通路 PCIe4 差分数据输入正 当 PCIE_F0_PRSENTn[1:3] 其中有一个为低时，此通路拆分为 4 个 X1 通路使用。 靠近设备端放置 0.1uf 的隔直电容。 不用时可悬空。
PCIE_F0_RXN[3:0]	AG2 2 AG2 3 AG2 1 AG2 0	1	Yes <input type="checkbox"/> No <input type="checkbox"/>	F0 通路 PCIe4 差分数据输入。 当 PCIE_F0_PRSENTn[1:3] 其中有一个为低时，此通路拆分为 4 个 X1 通路使用。 靠近设备端摆放 0.1uf 的隔直电容。 不用时可悬空。
PCIE_F0_REFRES	AC20	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	外部参考电阻，通过一个 200ohm(1%)的电阻连接至地

PCIE_F0_PRSENT_N[1:0]	AG19 AF19	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>控制器使能信号，默认全高</p> <p>X4 模式时，PCIE_F0_PRSENTn[1:3]必须置高。PCIE_F0_PRSENTn0 作为控制使能信号，当设备接入时置低。</p> <p>X1 模式时，PCIE_F0_PRSENTn[0:3] 分别作为数据总线的 0~3 使能信号，当有设备接入时对应的使能信号需要置低。不用时建议外部上拉到 3.3VS0 处理。</p>
PCIE_F0_RSTN	AC20	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>F0 通路 PCIE 复位信号，低电平有效。建议外部预留 4.7K 电阻上拉到 3.3VS0。</p>
PCIE_F1_TXP[3:0]	AK19 AK18 AK16 AK15	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>F1 通路 PCIe4 差分数据输出正。</p> <p>当 PCIE_F1_PRSENTn1 为低时，此通路拆分为 2 个 X1 通路使用。分别为数据组 0、1。</p> <p>靠近发送端摆放 0.1uf 的隔直电容。不用时可悬空。</p>
PCIE_F1_TXN[3:0]	AJ19 AJ18 AJ16 AJ15	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	<p>F1 通路 PCIe4 差分数据输出负。</p> <p>当 PCIE_F1_PRSENTn1 为低时，此</p>

				<p>通路拆分为 2 个 X1 通路使用。分别为数据组 0、1。</p> <p>靠近发送端摆放 0.1uf 的隔直电容。</p> <p>不用时可悬空。</p>
PCIE_F1_RXP[3:0]	AF18 AF17 AF16 AF15	I	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>F1 通路 PCIe4 差分数据输入正。</p> <p>当 PCIE_F1_PRSNTn1 为低时，此通路拆分为 2 个 X1 通路使用。分别为数据组 0、1。</p> <p>靠近设备端摆放 0.1uf 的隔直电容。</p> <p>不用时可悬空。</p>
PCIE_F1_RXN[3:0]	AG18 AG17 AG16 AG15	I	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>F1 通路 PCIe4 差分数据输入负。</p> <p>当 PCIE_F1_PRSNTn1 为低时，此通路拆分为 2 个 X1 通路使用。分别为数据组 0、1。</p> <p>靠近设备端摆放 0.1uf 的隔直电容。</p> <p>不用时可悬空。</p>
PCIE_F1_REFRES	AC16	A	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>外部参考电阻，通过一个 200ohm(1%)的电阻连接至地</p>
PCIE_F1_PRSNTN[1:0]	AE17 AD17	I	<p>Yes <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>控制器使能信号，默认全高</p> <p>X4 模式时，PCIE_F1_PRSNTn1 必须置高。PCIE_F1_PRSNTn0 作为控制使能信号，当设备接入时置低。</p>

				X1 模式时，PCIE_F1_PRSTn[0:1] 分别作为数据总线的 0~1 使能信号，当有设备接入时对应的使能信号需要置低。不用时建议外部上拉到 3.3VS0 处理。
PCIE_F1 RSTN	AD18	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	F1 通路 PCIE 复位信号，低电平有效。建议外部预留 4.7K 电阻上拉到 3.3VS0。

注：

- 1、PCIE 支持信号反转（软件配置，若 X8 通道拆分为两个 X4 使用时，低 4 位不支持 lane reversal，仅高 4 位支持），支持 N/P 互换。
- 2、由于 PCIE 端口内部 buffer 容量差异，如需使用 Realtek 网卡设备，尽量连接到 F0 的 PORT2/3 端口，其他端口不能保证性能最优。

2.2.3 USB

Signal Name	Ball s	Sign al Type	Check ed	Recommendations
USB0TX RTUNE[0:5]	K1 L3 N6 M6 N7	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	参考电阻，通过一个 200ohm(1%) 的电阻连接到地

	M1			
USB0_D P[0:5]	K3 L2 N3 L4 M3 N2	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	USB0 ~ 5 数据正 不用时可悬空
USB0_D M[0:5]	K2 L1 N4 M4 M2 N1	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	USB0 ~ 5 数据负 不用时可悬空
USB0_O C[0:2]	P4 P3 P2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	USB 过流检测，高有效； 与数据信号对应关系为 OC0 对应 USB0 ~ 1 OC1 对应 USB2 ~ 3 OC2 对应 USB4 ~ 5 不使用时下拉到地

2.2.4 DDR3

Signal Name	Balls	Signal Type	Checked	Recommendations
DDR_DQ [15:0]	M27 M28 M29 M30 P28 P27	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 数据信号

	P29 N30 P24 L27 K29 K30 P25 L30 N26 R24			
DDR_DQ SP[1,0]	N28 L28	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 数据选通差分信号, 正沿
DDR_DQ SN[1:0]	N29 L29	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 数据选通差分信号, 负沿
DDR_DQ M[1:0]	N27 N25	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 数据屏蔽信号
DDR_A[15:0]	H29 J27 H27 H28 G29 J26 G30 F27 K25 G26 M25 M24 F26 J29 G25 F25	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 地址总线信号
DDR_BA [2:0]	H30 L25 K24	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 逻辑 bank 地址信号

DDR_WEN	L26	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 写使能信号
DDR_CASN	J30	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 列地址选择信号
DDR_RASN	J28	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 行地址选择信号
DDR_CSNN	K27	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 片选信号
DDR_CKE	H24	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 时钟使能信号
DDR_CKP0	G27	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 差分时钟输出信号，正沿
DDR_CKN0	G28	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 差分时钟输出信号，负沿
DDR_ODT	K28	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM ODT 输出信号
DDR_RESETN	H25	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 SDRAM 复位信号
DDR_REXT	J25	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DDR3 外部参考电阻，通过一个240ohm(1%)的电阻连接至地

注：

- 1、显存颗粒推荐优先选用公版型号。

- 2、DDR3 颗粒如果只使用单颗颗粒，外部可不保留终端电阻。
- 3、如不使用内置显卡 DVO 功能，可不保留 DDR3 颗粒设计，但是 7A1000 桥片内置显卡相关电源设计仍要保留。

2.2.5 SATA

Signal Name	Balls	Signal Type	Checked	Recommendations
SATA_RE FRES	V25	A	Yes <input type="checkbox"/> No <input type="checkbox"/>	外部参考电阻，通过一个 200Ω电阻 接地
SATA_TX P[2:0]	T26 U27 V30	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SATA 差分数据输出信号，正沿 不用时可悬空
SATA0_T XN[2:0]	T27 U26 V29	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SATA 差分数据输出信号，负沿 不用时可悬空
SATA0_R XP[2:0]	R27 R29 U29	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	SATA 差分数据输入信号，正沿 不用时可悬空
SATA0_R XN[2:0]	R26 R30 U30	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	SATA 差分数据输入信号，负沿 不用时可悬空
SATA0_L EDN[2:0]	P30 U24 U28	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SATA 工作状态指示输出信号，低表 示有数据传输 使用时外部需通过 4.7K 电阻上拉到 3.3V。

				不用时可悬空
--	--	--	--	--------

2.2.6 SPI

Signal Name	Balls	Signal Type	Checked	Recommendations
SPI_SCK	W25	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 时钟输出 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
SPI_SDO	Y30	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 数据 命令输出 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
SPI_SDI	W26	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 数据输入 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
SPI_CSN [0:1]	W30 W29	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	SPI 片选输出 使用时外部需通过 4.7K 电阻上拉到 3.3V。

				不用时可悬空
SPI_CSN [2:3]	W28 W27	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	默认通用操作系统用作 DVO 对应的 HP 检测信号，外部上拉，须通过外 部线路默认将该信号拉低。 显示器接入时拉高。 显示器拔出是拉低。

2.2.7 LPC

Signal Name	Balls	Signal Type	Checked	Recommendations
LPC_AD[3:0]	B7 F10 A6 E9	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	LPC 地址/数据输入输出 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
LPC_FRAMEN	D9	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	LPC 帧同步信号，低电平有效 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
LPC_SERIRQ	C7	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	LPC 总线中断输入 使用时外部需通过 4.7K 电阻上拉到

				3.3V。 不用时可悬空
LPC_RES ETN	E10	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	LPC 总线复位输出 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空

注：LPC 设备和控制器要求时钟同源

2.2.8 I2C

Signal Name	Ball s	Sign al Type	Check ed	Recommendations
I2C0_SC L	AC26	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
I2C0_SD A	AD2 6	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
I2C1_SC L	AC24	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空

I2C1_SDA	AC25	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
----------	------	-----	---	-------------------------------------

2.2.9 UART

Signal Name	Balls	Signal Type	Checked	Recommendations
UART_TXD	D11	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 数据发送 不用时可悬空
UART_RXD	B10	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 数据接收 不用时可悬空
UART_RTS	A9	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 请求发送/Uart1 数据发送 不用时可悬空
UART_CTS	B11	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 清除发送/Uart1 数据接收 不用时可悬空
UART_DTR	C11	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 数据终端准备好/Uart2 数据发送 不用时可悬空
UART_DSR	C10	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 数据装置准备好/Uart2 数据接收

				不用时可悬空
UART_RI	E11	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 振铃指示/Uart3 数据发送 不用时可悬空
UART_D CD	A10	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	Uart0 数据载波检查/Uart3 数据接 收 不用时可悬空

注:通用类产品系统默认将 UART0 端口配置串口功能,不可用与 GPIO 或者 I2C 复用使用。如需其他复用功能需修改内核进行支持。

2.2.10 GMAC

Signal Name	Balls	Signal Type	Checked	Recommendations
GMAC0_TXCK	T2	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 发送时钟 信号 不用时可悬空
GMAC0_TCTL	R3	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 发送控制 信号 不用时可悬空
GMAC0_TXD[3:0]	P1 T4 T6 R4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 发送数据 总线

				不用时可悬空
GMAC0_ RXCK	R1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 接收时钟 信号 不用时可悬空
GMAC0_ RCTL	T1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 接收控制 信号 不用时可悬空
GMAC0_ RXD[3:0]	U2 U3 U4 T3	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 RGMII 接收数据 总线 不用时可悬空
GMAC0_ MDCK	R2	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 SMA 接口时钟信 号 不用时可悬空
GMAC0_ MDIO	T7	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 0 的 SMA 接口数据信 号 外部需通过 4.7K 电阻上拉到 3.3V。
GMAC1_ TXCK	V5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 发送时钟 信号 不用时可悬空
GMAC1_ TCTL	U1	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 发送控制 信号

				不用时可悬空
GMAC1_TXD[3:0]	V6 V7 U8 V1	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 发送数据总线 不用时可悬空
GMAC1_RXCK	V3	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 接收时钟信号 不用时可悬空
GMAC1_RCTL	W4	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 接收控制信号 不用时可悬空
GMAC1_RXD[3:0]	W7 W5 V2 W3	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 RGMII 接收数据总线 不用时可悬空
GMAC1_MDCK	U5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 SMA 接口时钟信号 不用时可悬空
GMAC1_MDIO	U6	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	GMAC 控制器 1 的 SMA 接口数据信号 外部需通过 4.7K 电阻上拉到 3.3V。

2.2.11 PWM

Signal Name	Balls	Signal Type	Checked	Recommendations
PWM[3:0]	Y29 Y28 Y27 Y26	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	与 GPIO[7:4]复用 如果用作风扇控制功能，默认 PWM0 为 CPU_FAN PWM1 为 SYS_FAN

2.2.12 HDA

Signal Name	Balls	Signal Type	Checked	Recommendations
HDA_BITCLK	B9	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	HDA_BITCLK 输出、AC97_BITCLK 输入 不用时可悬空
HDA_SDI [0:2]	A7 F11 G12	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	HDA/AC97 数据输入，连接第一个 codec; HDA 数据输入，连接第二个 codec; HDA 数据输入，连接第二个

				codec; 不使用外部需通过 4.7K 电阻下拉至地
HDA_SD O	B8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	数据输出 不用时可悬空
HDA_SY NC	C9	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	同步信号 不用时可悬空
HDA_RE SETN	A8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	复位信号 不用时可悬空

2.2.13 GPIO

Signal Name	Ball s	Sign al Type	Check ed	Recommendations
GPIO0	P6	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	通用输入输出引脚 默认为输出高 不用时可悬空

GPIO 复用功能详见手册

2.2.14 RTC

Signal	Ball	Sign	Check	Recommendations
--------	------	------	-------	-----------------

Name	Ball	Signal Type	Checked	
RTC_XI	AB4	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	RTC 时钟输入引脚 32.768KHz 时钟输入
RTC_XO	AB5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	RTC 时钟输出引脚 32.768KHz 时钟输出
RTC_DO TESTN	AB6	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	默认通过 4.7K 上拉至 RTC_3V
RTC_RS MRSTN	AB3	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	RSM 域复位信号,在 RSM 电源稳定后外部拉起。
RTC_RST N	AB2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	RTC 域寄存器复位信号, 在 RTC 电源稳定后外部拉起。

注：默认通用操作系统使用内部 RTC 功能模块

2.2.15 DVO

Signal Name	Ball	Signal Type	Checked	Recommendations
DVO0_CK P	A4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的时钟输出信号, 正沿(单端全摆幅信号)

				不用时可悬空
DVO0_CK N	B4	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的时钟输出信号, 负沿(单端全摆幅信号), 一般不使用。 不用时可悬空
DVO0_HS YNC	F9	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的水平同步信号 不用时可悬空
DVO0_VS YNC	G10	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的垂直同步信号 不用时可悬空
DVO0_DE	D7	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的数据有效信号 不用时可悬空
DVO0_D[0:23]	E8 C6 B5 F8 D6 C5 A3 B3 C4 D5 E6 F7 G8 B2 A2 C3 D4 B1 C2 C1 D3 D1	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的显示数据信号 不用时可悬空

	D2 E5			
DVO0_SC L	B6	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的 I2C 串行时钟信号 不用时可悬空
DVO0_SD A	A5	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO0 的 I2C 串行数据信号 不用时可悬空
DVO1_CK P	F3	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的时钟输出信号, 正沿(单端 全摆幅信号) 不用时可悬空
DVO1_CK N	F4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的时钟输出信号, 负沿(单端 全摆幅信号), 一般不使用。 不用时可悬空
DVO1_HS YNC	F5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的水平同步信号 不用时可悬空
DVO1_VS YNC	E4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的垂直同步信号 不用时可悬空
DVO1_DE	G6	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的数据有效信号 不用时可悬空
DVO1_D[0:23]	F6 E2 E3 H6 J7 G5 F2 E1	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的显示数据信号 不用时可悬空

	F1 G4 G3 G2 G1 H4 H5 J6 H3 H2 H1 J4 J3 J2 J1 K6			
DVO1_SCL	K5	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的 I2C 串行时钟信号 不用时可悬空
DVO1_SDA	K4	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	DVO1 的 I2C 串行数据信号 不用时可悬空

注：

1. DVO 接口的数据位 DVO_D[23:0]分别对应 RGB888, 即: DVO_D[23:16]对应 RGB 信号的 R[7:0], DVO_D[15:8]对应 RGB 信号的 G[7:0], DVO_D[7:0]对应 RGB 信号的 B[7:0]。

当使用的转换芯片的 RGB 信号宽度小于 8 位时, 必须使用桥片输出信号的高位数据。比如转换芯片的格式为 RGB565, 则需要使用桥片的 DVO_D[23:19]连接 R, 使用 DVO_D[15:10]连接 G, 使用 DVO_D[7:3]连接 B。

2. 板卡必须使用 DVO_SCL/SDA 引脚连接 PHY 转换芯片和显示器接口。

桥片的每个 DVO 接口都自带了一组 I2C 信号, 用于连接 DVO 接口的转换 PHY

芯片和显示器。板卡设计时,必须使用该组 I2C 信号连接 PHY 芯片(如果 PHY 芯片存在 I2C 配置接口的话)和显示器接口,而不使用 PHY 芯片输出的 I2C 引脚。

3.如果板卡需要支持显示器热插拔功能,使用 SPI_CS_n2 和 SPI_CS_n3 (复用为 GPIO 功能)分别作为 DVO0 和 DVO1 通道的显示器热插拔检测引脚使用。

2.2.16 Strap&Control

Signal Name	Balls	Signal Type	Checked	Recommendations
PCIEBRG MODE	AC30	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	默认通过 4.7k 下拉至地
CLKSEL[8:0]	Y25 AA2 9AA 30A A28A B30A B29 AA2 6AA 25AB 28	I/O	Yes <input type="checkbox"/> No <input type="checkbox"/>	CLKSEL[5:0]: 悬空 CLKSEL6: HT PHY 参考时钟选择 0: 使用 200MHz 差分输入时钟(推荐使用) 1: 使用 100MHz 单端系统输入时钟 CLKSEL7: HT 频率配置模式选择, 默认通过 4.7k 下拉至地

				CLKSEL8: HT gen1 模式使能,推荐通过 4.7k 上拉到 3.3V。
--	--	--	--	---

2.2.17 JTAG

Signal Name	Balls	Signal Type	Checked	Recommendations
JTAG_TCK	AC28	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 时钟信号 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
JTAG_TDI	AD30	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 数据输入信号 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
JTAG_TMS	AD28	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 模式 使用时外部需通过 4.7K 电阻上拉到 3.3V。 不用时可悬空
JTAG_TRST	AD27	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 复位信号 内部默认上拉

				推荐板卡设计时外接 4.7k 下拉到地，预留上拉电阻位置。 不用时通过 4.7K 下拉至地
JTAG_TDO	AD29	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	JTAG 数据输出 使用时外部需通过 4.7K 电阻上数据拉到 3.3V。 不用时可悬空

2.2.18 ACPI

Signal Name	Balls	Signal Type	Checked	Recommendations
ACPI_EN	Y1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	正常模式需要上拉到 3.3V _{SB} ，使能 ACPI 信号；该信号为低时，所有 ACPI 信号不起作用（可用于开机自动上电）。
ACPI_SY_SRSTN	AB1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	桥片复位输入，上升沿有效。当桥片接收到该信号后，内部除 ACPI 以外模块寄存器均被复位，并触发后续 PLTRST、PCIE_RST、LPC_RST 和 MEM_RST 等外设复位信号输出。

ACPI_RINGN	AA2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	输入信号，外部振铃唤醒，低电平有效，需与软件配合使用。外部上拉到 3.3VSB。
ACPI_WAKEN	AA1	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	输入信号，外部 PCIE 唤醒信号，低电平有效，需与软件配合使用。外部上拉到 3.3VSB。
ACPI_LID	W2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	输入信号，可用于笔记本屏盖状态监测，软件使能后，该信号跳变会触发系统中断。建议外部上到 3.3VSB。
ACPI_PWRTYPE	Y2	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	输入信号，可用于笔记本方案外部供电类型监测，软件使能后，该信号跳变会触发系统中断。建议外部上到 3.3VSB。
ACPI_BATTLOWN	Y3	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	输入信号，可用于笔记本方案电池低电量保护功能，在开机状态，配合软件使能，该信号可触发系统中断，当关机状态该信号为低时，ACPI_POWERBTN 不能正常唤醒机器。建议外部上到 3.3VSB。
ACPI_S3N	Y7	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	S3 状态输出信号，当系统在 S0 时为高，当在 S5、S4 和 S3 状态下为低。需外部上拉 3.3V SB 电。可用

				作 S0 电源开启控制。在待机 STR 功能中，可作为 3A 端处理器 MEM_VDDE 和内存条 VTT 电源的使能控制。
ACPI_S4N	AA6	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	S4 状态输出信号，当系统在 S0/3 时为高，当在 S5 和 S4 状态下为低。需外部上拉 3.3VSB 电。
ACPI_S5N	AA5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	S5 状态输出信号，当系统在 S0/3/4 时为高，当在 S5 状态下为低。需外部上拉 3.3VSB 电。可用作待机功能时的 3A 端的内存颗粒 MEM_VDDE 和 MEM_VDDP 电源使能。
ACPI_PLTRSTN	W1	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	平台复位输出，可用于复位 IO 外设和 3A 处理器。低电平有效。
ACPI_SLPLANN	AA4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	输出信号，可用于网络电源控制，该信号根据 ACPI_PWRTYPE 的不同，由 ACPI 寄存器中 PMCON_RTC 的 8/7 位决定低功耗时的输出状态。建议外部上到 3.3VSB。
ACPI_PWRBTNN	Y6	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	ACPI 使能时，作为开机唤醒信号，需要在 RSM_RST 拉高稳定后触发。启动信号为大于 20ms 低于 3S 的低

				脉冲(上升沿触发), 桥片接收到该唤醒信号后, 会拉起相应的 ACPI_S3/4/5 信号。4S 以上低脉冲为强制关机信号。该信号在系统启动后可作为中断触发输入功能, 配合软件可实现待机等功能。需上拉 3.3VSB。
ACPI_PWROK	Y4	I	Yes <input type="checkbox"/> No <input type="checkbox"/>	ACPI 使能时, 外部电源正常状态指示输入。当外部所有电源启动正常后, 拉高该信号, 桥片进行内部逻辑的复位; 并可触发后续 PLTRST、PCIE_RST、LPC_RST 和 MEM_RST 等外设复位信号输出。
VSB_GATE	P5	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	S3 功能时, 控制 dual 电转换。此信号使能后, 会早于 ACPI_S3N 被拉低, 按下唤醒键后, 晚于 ACPI_S3N 被拉高。

2.2.19 Interrupt

Signal Name	Balls	Signal	Checked	Recommendations
-------------	-------	--------	---------	-----------------

		Type		
SYS_INT 0	AB25	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	默认预留连接到处理器 GPIO0
SYS_INT 1	AA2 4	0	Yes <input type="checkbox"/> No <input type="checkbox"/>	多路模式下, 默认预留连接到处理器 1 的 GPIO0

2.2.20 POWER

信号名称	标准电源 (V)	实测最大电流 (A)	备注
VDD	1.1V	2.002	
VDD_RSM	1.1V	0.106	
DDR_VDDE	1.5V	0.357	
DDR_VREF	0.75V	< 1mA	
HT_1V2	1.2V	0.498	纹波不能高于 10mV
HT_1V8	1.8V	0.072	
PEST_1V1	1.1V	1.302	
PEST_3V3	3.3V	0.444	
GMAC_VDDE1	3.3V/2.5V	0.023	
USB_A3V3	3.3V	0.174	
IO_3V3	3.3V	0.069	
ACPI_3V3	3.3V	0.045	
RTC_3V	2.5V~2.8V	N/A	
PLL_VDD3	1.25V	< 80mA	建议使用单独的 LDO 电源供电

注:

1. 工作在 3.3V 供电电压下。

2. RTC_3V 在 G3 (物理断电) 状态下的电流小于 5 μ A.
3. PLL_VDD 包括： PLL_HT_VDD 、 PLL_VDDA/VDDD_CORE 、
PLL_VDDA/VDDD_PER、
PLL_VDDA/VDDD_GRAPH。
4. 主板上电时序参见 7A1000 数据手册。

2.3 其他设计注意事项

2.3.1 时钟

3A4000 时钟设计要求

3A4000 有两个时钟输入源：一个为 SYSCLK，输入时钟为 25MHz 或者 100MHz 单端时钟；另外一个为可选的 HT0/1CLKP/N，输入为 200MHz 差分 LVDS 时钟。

SYSCLK 可通过 CLKSEL[4]选择输入源为 25MHz 还是 100MHz (推荐使用 100MHz 时钟)。其中内部处理器核时钟、内存控制器及总线时钟、串口波特率等均为该时钟源产生 (多路方案时，各处理器的 SYSCLK 要求同源)。HT 部分的 PLL 时钟源可通过 CLKSEL[8]进行选择用 SYSCLK 产生还是用外部差分时钟产生。

注意：处理器 SYSCLK 需要添加展频设计，建议展频幅度为正负 0.25%。

7A1000 时钟设计注意事项

信号名称	类型	描述	电源
CLKIN	I	桥片 100 MHz 主参考时钟，除 LPC 以外的所有模块均可由内部锁相环产生。	IO_3V3
TESTCLK	I	测试时钟输入，保留，通过 10Kohm 电阻接地	IO_3V3
LPC_CLKIN	I	LPC 33MHz 参考时钟，不使用 LPC 接口时，可不接	IO_3V3
CLKOUT33M	O	33.3 MHz 单端时钟输出。	IO_3V3
CLKOUT100M	O	100 MHz 单端时钟输出。可用作处理器 SYSCLK。	IO_3V3
CLKOUT25M	O	25 MHz 单端时钟输出，在 S0 状态下输出 25MHz 时钟，其他状态下输出为低电平。可用作处理器 SYSCLK。	ACPI_3V3
CLKOUTFLEX	O	频率可变单端时钟输出，在 S0 状态下输出可变频率时钟，其他状态下输出为低电平。默认为 33.3 MHz。	ACPI_3V3
HTCLKp/n	I	HT 200 MHz 差分参考时钟，	-
PCIE_F0_CLKIN p/n	I	PCIE_F0 100 MHz 差分参考时钟，使用内部时钟时可不接	-

PCIE_F1_CLKIN p/n	I	PCIE_F1 100 MHz 差分参考时钟,使用内部时钟时可不接	-
PCIE_H_CLKIN p/n	I	PCIE_H 100 MHz 差分参考时钟,使用内部时钟时可不接	-
PCIE_G0_CLKI Np/n	I	PCIE_G0 100 MHz 差分参考时钟,使用内部时钟时可不接	-
PCIE_G1_CLKI Np/n	I	PCIE_G1 100 MHz 差分参考时钟,使用内部时钟时可不接	-
SATA0_CLKINp /n	I	SATA0 100 MHz 差分参考时钟, 建议不接	-
SATA1_CLKINp /n	I	SATA1 100 MHz 差分参考时钟, 建议不接	-
SATA2_CLKINp /n	I	SATA2 100 MHz 差分参考时钟, 建议不接	-
USB_XI	I	保留, 通过 10Kohm 电阻接地。	-
USB_CLKIN	I	USB 12 MHz 单端时钟输入	-

注:

1. 不使用的输入参考时钟可以不提供, 但需通过 10Kohm 的电阻接地。
2. 对于 USB 模块的参考时钟 USB_CLKIN, 当不使用 USB 接口唤醒功能时, 可不接, 此时需将引脚通过 10Kohm 的电阻接地; 否则必须提供一个 12MHz 的板上时钟, 注意板上时钟的电压幅值应为 2.5V。
3. 无论 PCIE 控制器使用内部时钟还是外部时钟, PCIE 控制器和外设要求时钟

同源。PCIE 控制器使用外部时钟时，用到的控制器对应时钟输入端口都需要接入。

2.3.2 时序要求

主板复位时序

此部分请详细参照 7A1000 数据手册相关部分。

ACPI 下特殊复位信号设计建议

- 1.目前 7A1000 桥片的 ACPI_SYSRSTN 为上升沿有效，在某些特定应用场景作为系统硬复位不太适合，所以目前推荐使用 ACPI_POWEROK 信号作为系统硬件复位输入源。
- 2.建议用 7A1000 桥片的 PLT_RST 信号通过延迟 200ms 后作为处理器 SYSRESET 使用。
- 3.主板系统复位必须预留外部看门狗方案。

2.3.3 内置显卡设计

参见《龙芯 CPU 统一系统架构规范》中龙芯内置显卡软硬件设计规范相关章节。

2.3.4 桥片 SPI 存储 rom

7A1000 平台主板硬件设计时，7A1000 桥片的 SPI 总线建议连接一个 SPI flash，片选固定为 CS0，读写方式必须完全兼容 SST25VF010，容量不低于

512KB。该 flash 用于存储桥片集成 GMAC 的 MAC 地址、集成 GPU 显示相关参数信息、主板串号等。

2.3.5 笔记本、一体机方案注意事项

平台规范笔记本、一体机方案几个特殊功能引脚定义。

独显方案中 EC 的 SCI 的引脚必须连接到 7A1000 对应管脚，三个背光控制端口由 EC 控制（预留显卡端控制端口做为备份），一体机中 OSD 链接 EC，由 EC 上报系统。SMARTFAN 功能由 EC 控制。

内置显卡方案中 EC 的 SCI 的引脚必须连接到 7A1000 对应管脚，三个背光控制端口由 7A1000 相应管脚控制，一体机中 OSD 链接 EC，由 EC 上报系统。SMARTFAN 功能由 EC 控制。

笔记本、一体机方案中 7A1000 现规定几个特殊管脚定义与之对应：

SCI(System Control Interrupt)	CLKSEL4(GPIO50)默认上拉 备份连接到 7A1000 桥片的 ACPI_LID 管脚
LCD_EN(LCD backlight enable)	CLKSEL0(GPIO46)默认下拉
LCD_VDD_EN(LCD power enable)	CLKSEL1(GPIO47)默认下拉
LCD_PWM(LCD backlight PWM)	PWM3(GPIO07)

注：其中 SCI 中断，目前支持低电平触发。

2.3.6 接口外设支持列表

详见《龙芯 3A4000_7A1000 外围功能芯片&模组支持列表》

2.3.7 SE 模块最小设计

2.3.7.1 概述

本节为龙芯 3A4000 安全处理器 SE 模块硬件最小电路设计参考。

2.3.7.2 硬件参考设计

1) 系统组成

龙芯 3A4000 安全处理器 SE 模块最小接口电路由配置电路、状态指示、UART、QSPI 组成，模块外围电路框图如下图所示。

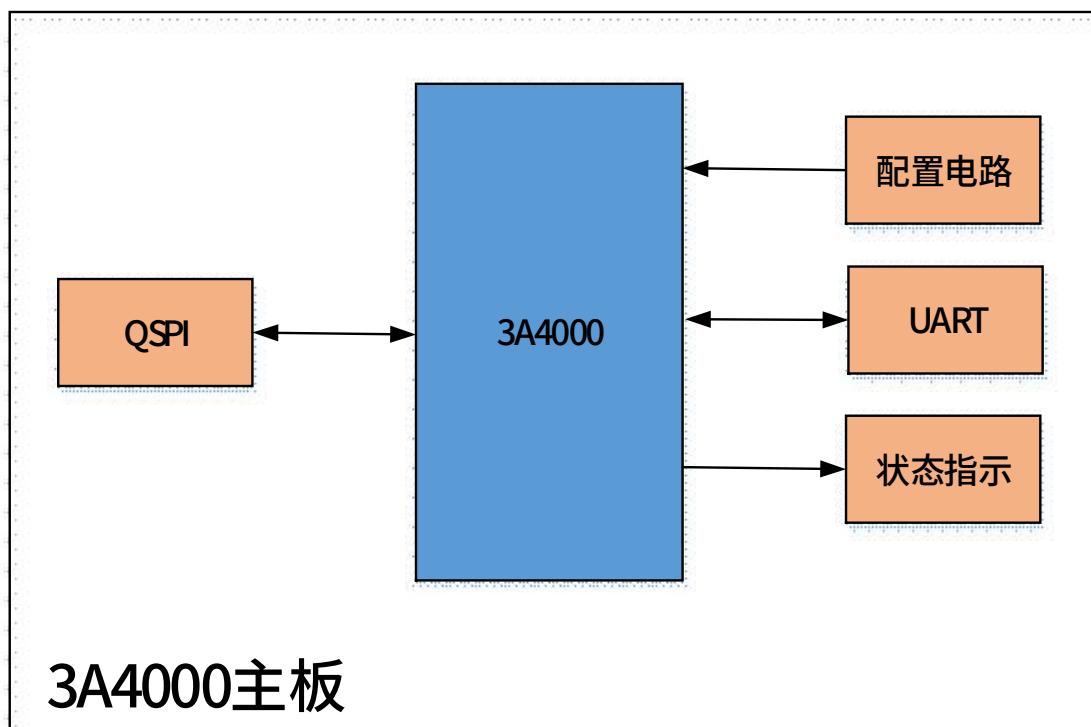


图 2.3.7-1 SE 模块外围电路框图

2) 接口详细参考设计

SE 外围电路参考设计详细描述如下所示。

配置电路：将配置信号管脚（CHIP_CONFIG[0]）接拨码开关，（SE_CLK_SEL）接上拉电阻、（SE_GPIO8、SE_GPIO9）管脚接下拉电阻。参考设计如下图所示。

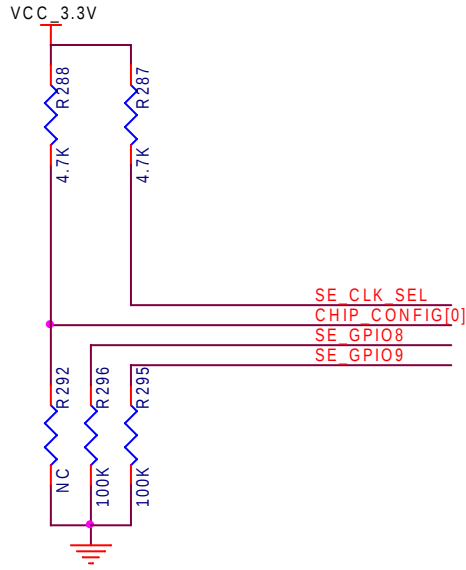


图 2.3.7-2 配置电路参考设计

状态指示：状态指示信号引脚 (SE_GPIO4~7) 外接 LED 灯，SE_GPIO4 接绿色 LED 灯，其他 GPIO 接黄色 LED 灯，信号引脚 (SE_GPIO4~7) 与 LED 灯之间串接阻值 1K 的电阻，状态指示电路图如下图所示。（IO 口驱动能力）

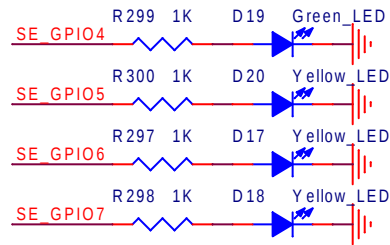


图 2.3.7-3 状态指示参考设计

UART 接口外接 3 芯 2.54mm 单排插针，串口 RX 引脚接 4.7K 上拉电阻，插针引脚定义如下图所示。

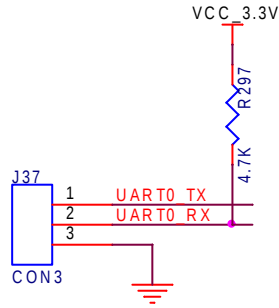


图 2.3.7-4 配置电路参考设计

QSPI 接口外扩兆易创新 SPI FLASH，型号 GD25Q64CSIG，工作电压 3.3V，封装 SOP8，尺寸图参考附录 1，参考电路如下图所示。

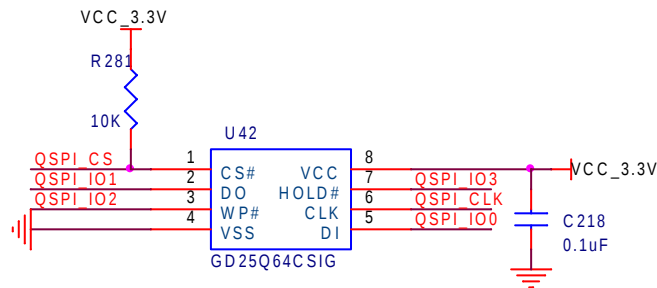


图 2.3.7-5 QSPI 接口参考设计

3) 芯片接口管脚列表

芯片具体管脚设计列表如下表所示。

分类	信号名称	输入输出	管脚	描述	默认处理方式	电压域
Clock	SYS_CLK	I	T06	系统输入时钟，	外接 25MHz 时钟	VDDE_I O
	CLKSEL[4]	I	R06	CPU 参考时钟分频选择： 1：不分频；0：4 分频	0：需外部 1K 下拉	VDDE_I O
Config	SE_CLK_SEL	I	W36	SE 功能时钟选择	1：需外部 4.7K 上拉	VDDE_I O
	CHIP_CONFIG[0]	I	N06	SE Boot 使能： 1：SE 启动；0：龙芯启动	1：使能 SE 可信功能，需外部 4.7K 上拉	VDDE_I O
JTAG	TSEL0	I	W06	JTAG 功能选择：		VDDE_I

				00: GS464V EJTAG 11: SE JTAG	00: 需外部 1K 下拉	O VDDE_I O
	TSEL1	I	Y03			
GPIO	SE_GPIO0	IO	V34	保留		VDDE_I O
	SE_GPIO1	IO	V33	保留		VDDE_I O
	SE_GPIO2	IO	V35	保留		VDDE_I O
	SE_GPIO3	IO	V37	保留		VDDE_I O
	SE_GPIO4	IO	W34	保留		VDDE_I O
	SE_GPIO5	IO	V36	SE 系统状态显示 1	外接指示灯, 高有效	VDDE_I O
	SE_GPIO6	IO	W32	SE 系统状态显示 2	外接指示灯, 高有效	VDDE_I O
	SE_GPIO7	IO	W33	保留		VDDE_I O
	SE_GPIO8	IO	W35	已使用	需外部 1K 下拉	VDDE_I O
	SE_GPIO9	IO	W37	已使用	需外部 1K 下拉	VDDE_I O
UART	SE_UART0_TX	O	M33	SE 串口 0 数据输出	UART0 可引出用于 调试 UART1 保留	VDDE_I O
	SE_UART0_RX	I	M32	SE 串口 0 数据输入		VDDE_I O
	SE_UART1_TX	O	M34	SE 串口 1 数据输出		VDDE_I O
	SE_UART1_RX	I	M36	SE 串口 1 数据输入		VDDE_I O
QSPI	SE_QSPI1_FLASH_CLK	O	P37	SE QSPI 1 时钟信号	外部需连接兆易创新 GD25Q64CSIG 芯片。使用叠封 LS3A4000 芯片时, 此 Flash 芯片不能焊接	VDDE_I O
	SE_QSPI1_FLASH_CS_N	O	R33	SE QSPI 1 片选信号		VDDE_I O
	SE_QSPI1_FLASH_IO0	IO	R34	SE QSPI 1 Flash 数据 0		VDDE_I O
	SE_QSPI1_FLASH_IO1	IO	R36	SE QSPI 1 Flash 数据 1		VDDE_I O
	SE_QSPI1_FLASH_IO2	IO	R37	SE QSPI 1 Flash 数据 2		VDDE_I O
	SE_QSPI1_FLASH_IO3	IO	R35	SE QSPI 1 Flash 数据 3		VDDE_I O

	SE_QSPI2_FLASH_CLK	O	P34	SE QSPI 2 时钟信号	使用可信存储时，外部可选配 SPI FLASH 芯片，容量 128Mb	VDDE_I O
	SE_QSPI2_FLASH_CS_N	O	P33	SE QSPI 2 片选信号		VDDE_I O
	SE_QSPI2_FLASH_IO0	IO	P32	SE QSPI 2 Flash 数据 0		VDDE_I O
	SE_QSPI2_FLASH_IO1	IO	N35	SE QSPI 2 Flash 数据 1		VDDE_I O
	SE_QSPI2_FLASH_IO2	IO	N37	SE QSPI 2 Flash 数据 2		VDDE_I O
	SE_QSPI2_FLASH_IO3	IO	N36	SE QSPI 2 Flash 数据 3		VDDE_I O
SPI	SE_SPI_MOSI	O	M37	SE SPI 总线输出	不用时，无需处理	VDDE_I O
	SE_SPI_MISO	I	N33	SE SPI 总线输入		VDDE_I O
	SE_SPI_CSN	O	M35	SE SPI 片选信号		VDDE_I O
	SE_SPI_CLK	O	N34	SE SPI 总线时钟		VDDE_I O
SCI	SE_SCI_RSTN	O	T36	SE SCI 总线复位信号	不用时，无需处理	VDDE_I O
	SE_SCI_DETECT	I	T32	SE SCI 总线检测信号输入		VDDE_I O
	SE_SCI_DATA	IO	T34	SE SCI 总线数据收发信号		VDDE_I O
	SE_SCI_CLK	O	T33	SE SCI 总线时钟输出		VDDE_I O
I2C	SE_I2C_SCL	O	P36	SE I2C 时钟信号	不用时，无需处理	VDDE_I O
	SE_I2C_SDA	IO	P35	SE I2C 数据		VDDE_I O
RSVD	RSVD1	IO	U34	保留信号	不用时，无需处理	VDDE_I O
	RSVD2	IO	T37			VDDE_I O
	RSVD3	IO	U33			VDDE_I O
	RSVD4	IO	T35			VDDE_I O
	RSVD5	IO	V32			VDDE_I O
	RSVD6	IO	U36			VDDE_I O

	RSVD7	IO	U37			VDDE_I O
	RSVD8	IO	U35			VDDE_I O

2.3.8 DDR4 RESET 和 CKE 设计细节

本节是针对笔记本、台式机等需要支持 S3 睡眠唤醒功能需要特别注意的设计指导。

详细方案

方案 1

采用外部控制 CKE 和 RESET 信号，本方案已验证通过。

具体方案如下：

对于需支持 S3 睡眠唤醒的设计，需增加外部控制电路满足如下图所示时序。

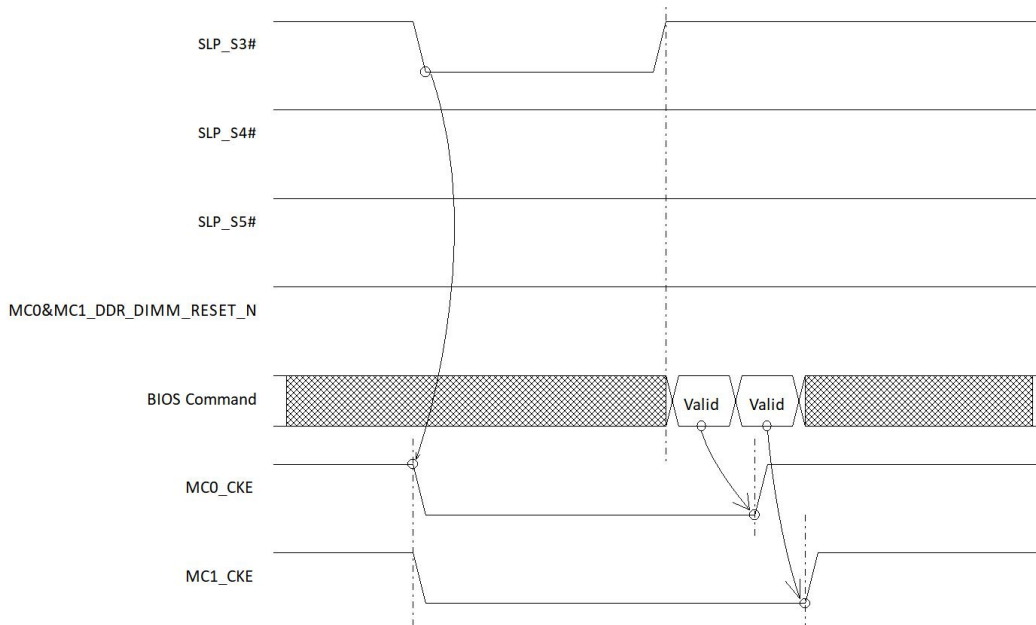


图 2.3.8-1 支持 SLP_S3#睡眠唤醒所需满足的时序图

待机： 外部线路判断进入 S3 时开始控制 MC0 以及 MC1 的 DDR RESETn 和 CKE

信号

唤醒： 图中 BIOS Command 表示 BIOS 会在特定阶段发出命令，外部控制电路根据 BIOS 的命令执行相应操作释放 DDR RESET/CKE 信号的控制。

详细的命令时机和操作见“图 2.3.8-16 本参考设计的时序图”的注释部分。

带 EC 控制的参考设计：

本方案将会引入 EC，对于不需要支持 S3 睡眠唤醒的、或自行设计控制电路并且满足上图所示时序的用户，可不引入本节方案。

对于 MC0：

1) 参考线路图：

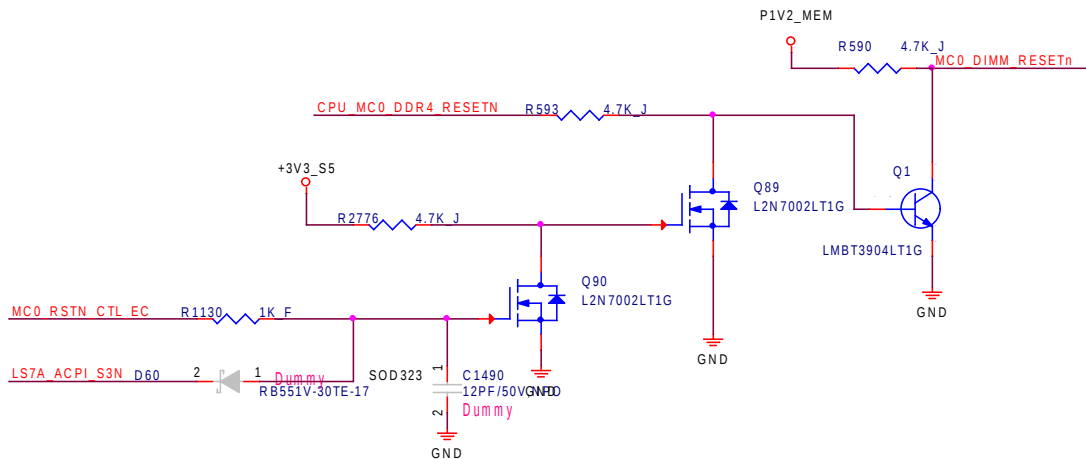


图 2.3.8-2 MC0 RESET 参考线路

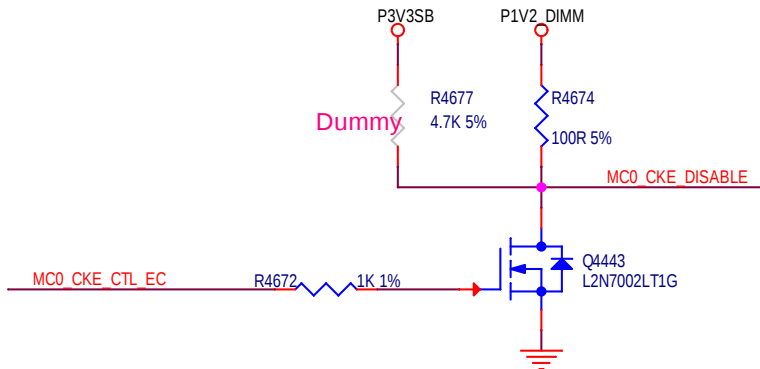


图 2.3.8-3 MC0 CKE 参考线路 1

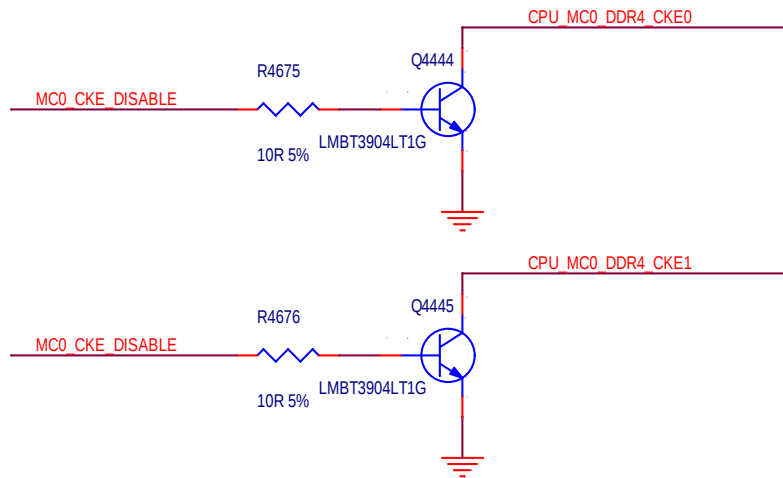


图 2.3.8-4 MC0 CKE 参考线路 2

2) 信号描述:

信号名称	连接位置	输入/输出
CPU_MC0_DDR4_RESETN	CPU MC0 的 PIN.AT22	CPU DDR 控制器 RESET 输出信号
MC0_RSTN_CTL_EC	EC 的 GPIO	EC 输出的 MC0 RESET 控制信号
MC0_CKE_CTL_EC	EC 的 GPIO	EC 输出的 MC0 CKE 控制信号
MC0_DIMM_RESETh	DIMM 的 RESET 信号	DIMM DDR RESET 输入信号
CPU_MC0_DDR4_CKE0/CKE1	CPU MC0 的 CKE0/1 和 DIMM 的 CKE0/1 信号	CPU DDR 控制器 CKE 输出信号, DIMM DDR CKE 输入信号

对于 MC1:

1) 参考线路图:

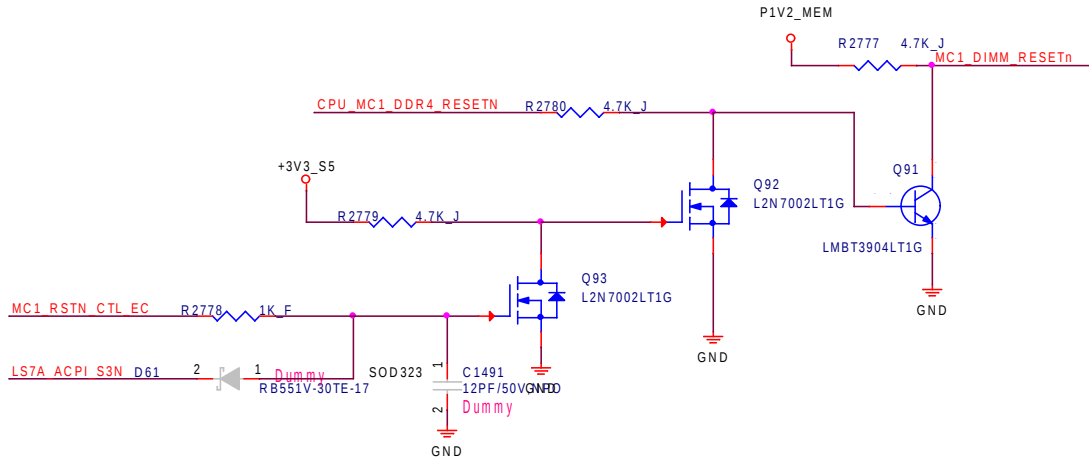


图 2.3.8-5 MC1 RESET 参考线路

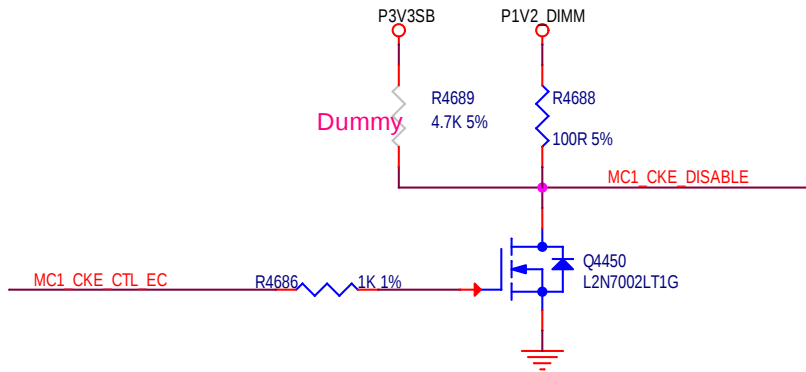


图 2.3.8-6 MC1 CKE 参考线路 1

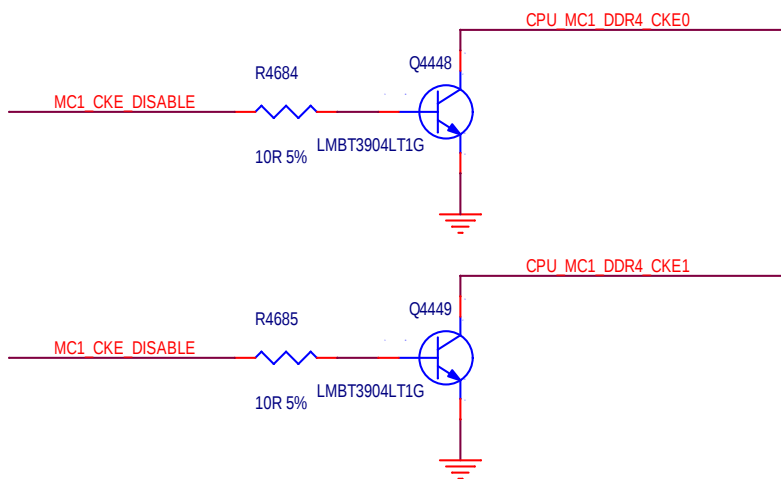


图 2.3.8-7 MC1 CKE 参考线路 2

2) 信号描述:

信号名称	连接位置	输入/输出
CPU_MC1_DDR4_RESETN	CPU MC1 的 PIN.AT22	CPU DDR 控制器 RESET 输出信号
MC1_RSTN_CTL_EC	EC 的 GPIO	EC 输出的 MC0 RESET 控制信号
MC1_CKE_CTL_EC	EC 的 GPIO	EC 输出的 MC1 CKE 控制信号
MC1_DIMM_RESETN	DIMM 的 RESET 信号	DIMM DDR RESET 输入信号
CPU_MC1_DDR4_CKE0/CKE1	CPU MC1 的 CKE0/1 和 DIMM 的 CKE0/1 信号	CPU DDR 控制器 CKE 输出信号, DIMM DDR CKE 输入信号

EC 需根据 SLP_S3#/SLP_S4#/SLP_S5# 状态以及 BIOS Command 的命令, 对 MC0_CKE_CTL_EC / MC1_CKE_CTL_EC / MC0_RSTN_CTL_EN / MC1_RSTN_CTL_EN 进行操作, 操作时序满足下图。

图中 BIOS Command 是在执行唤醒操作后, 由 BIOS 在特定阶段发出命令, 在 EC 获取到命令后对 MC0_CKE_CTL_EC / MC1_CKE_CTL_EC / MC0_RSTN_CTL_EN / MC1_RSTN_CTL_EN 执行相应的控制或释放操作。

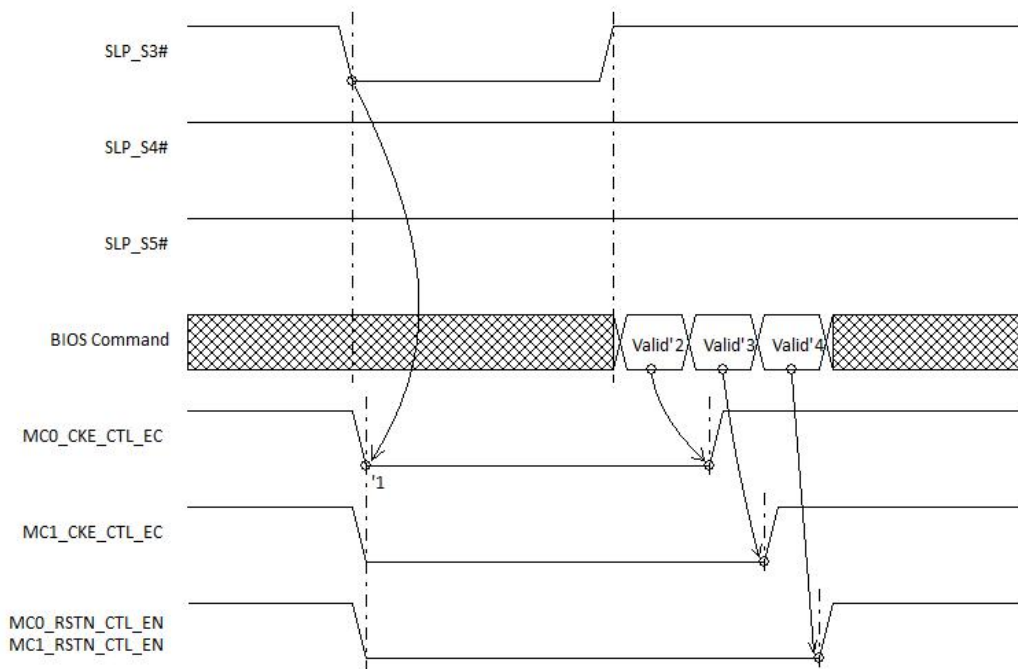


图 2.3.8-8 本参考设计的时序图

- 注：'1 EC 判别进入 S3 状态时开始控制两个通道的 DDR RESETn 和 CKE 信号。
'2 MC0 内存初始化开始前释放 EC 对 MC0 的 CKE 控制。
'3 MC1 内存初始化开始前释放 EC 对 MC1 的 CKE 控制。
'4 两个内存通道初始化都完成后释放 EC 对两个内存通道的 RESETn 控制。

2.3.8.2.2 方案 2

处理器中内存模块的电源在 S3 下保持有电,本方案可不需要用外部 EC 等控制 RESET 和 CKE 信号,但存在对处理器的其他 VDDP 电源引脚漏电的风险,待验证。

具体方案如下:

1)

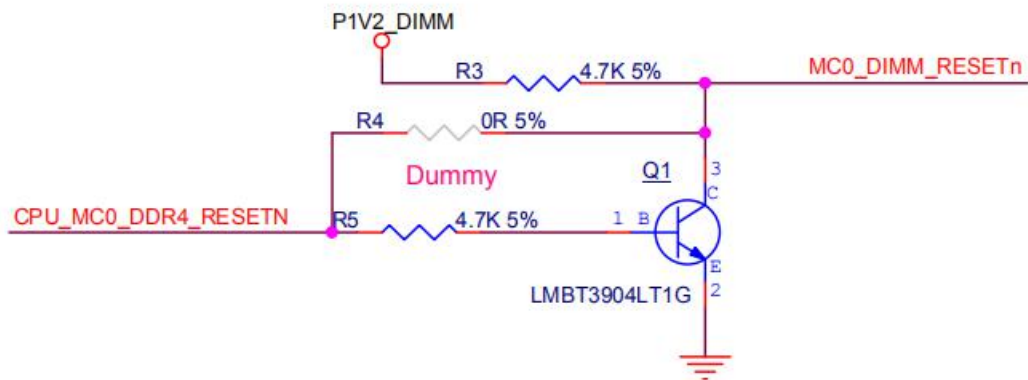
AB17	VDDMC
AB21	VDDMC
AC16	VDDMC
AC17	VDDMC
AC21	VDDMC
AC22	VDDMC
AD11	VDDMPHY
AD12	VDDMPHY
AD15	VDDMPHY
AD18	VDDMPHY
AD20	VDDMPHY
AD23	VDDMPHY
AD26	VDDMPHY

AD27	VDDMPHY
AE13	VDDMPHY
AE14	VDDMPHY
AE16	VDDMPHY
AE17	VDDMPHY
AE21	VDDMPHY
AE22	VDDMPHY
AE24	VDDMPHY
AE25	VDDMPHY

处理器的以上 pin 脚使用 S3 域电源供电。

2) VDDE_DDR 与 DIMM VDD 电源合并

3) RESET 信号接法如下：



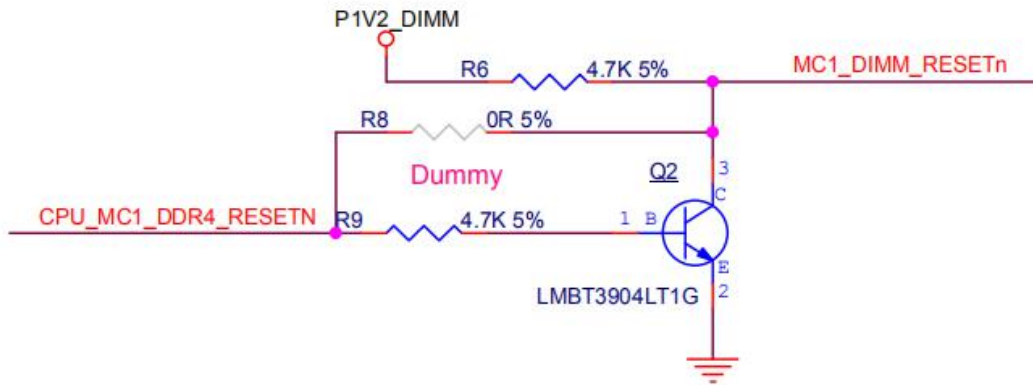


图 2.3.8-9 本方案设计 RESET 连接示意图

4) CKE 信号直连即可。

第三章 PCB 设计指导

3.1 DDR4

信号名称	走线要求
CLOCK	阻抗：70Ω +- 15% 同一个 DIMM 条上时钟之间等长误差 5mils, 差分对内等长误差 2mils
A[17:0], BA[1:0], RAS_N, CAS_N, WE_N, BG[1:0], PAR, ACT, C2	阻抗：40Ω +- 10% 组内等长误差 25mils PC 要求：走线长度 < 3500mils
CS[7:0], ODT[3:0], CKE[3:0]	阻抗：40Ω +- 10% 与时钟线等长，等长误差 25mils

DQ	<p>阻抗：50Ω +- 10%</p> <p>同一 Byte 内等长误差 10mils, 相邻两个 Byte 间走线长度相差 400mils 以内</p> <p>对于 PC 主板要求：DQ 走线总长度小于 5000mils</p> <p>对于服务器主板要求：DQ 走线总长度小于 6000mils</p>
DQS	<p>阻抗：83Ω +- 15%</p> <p>与同组 DQ 等长误差 10mils, 差分对内等长误差 5mils</p> <p>DQS-CLK < 1500mils</p>

注：

1. PC 主板建议单通道单 DIMM 设计。
2. 走线要求严格的同组同层，DQ、DQS 参考平面为 GND；时钟、地址、命令及控制线尽量参考 VDDE_DDR。

对于 PC

(1) 如果处理器和 DIMM 是两个电源域（即两个电源芯片供电），推荐参考 GND；

(2) 如果处理器和 DIMM 是同一电源域，推荐参考 VDDE_DDR(处理器端 DDR IO 电)，切换 MOS 放置在处理器和 DIMM 中间。

3. 差分信号跨层时，差分信号对设置一个 GND 平面穿引孔，在距离信号过孔 150mil 范围内设置 GND 孔。

4. 走线满足 3W 原则。

5. 参考层禁止跨分割，相邻非参考层要求满足 3H 原则。

6. 推荐处理器端放置去耦电容组合：

22uF 0603 10 个

1uF 0402 8 个

0.1uF 0402 20 个

放置在处理器电源 pin 附近

7. 推荐每个 DIMM 端去耦电容组合：

560uF 1 个

22uF 5 个

0.47uF 5 个

0.1uF 5 个

放在每个 dimm 电源 pin 附近

8.对于每通道双 dimm 的设计，要求同通道的两个 dimm 之间所有信号线等长控制在 5mils 以内。

3.2 PCIE

信号名称	描述
PCIE_G0_TXp[7:0] PCIE_G0_TXn[7:0]	差分阻抗 100Ω±15%
PCIE_G1_TXp[7:0] PCIE_G1_TXn[7:0]	每对差分等长 < 5mil
PCIE_H_TXp[7:0] PCIE_H_TXn[7:0]	信号走线总长度小于 10”
PCIE_F0_TXp[3:0] PCIE_F0_TXn[3:0]	ASIC 引线 (0.5” 以内) =1:1 在时钟信号对两侧，到所有其他信号的间距至少为 4:1

<p>PCIE_F1_TXp[3:0] PCIE_F1_TXn[3:0]</p>	<p>差分信号要参考 GND，且不能跨分割，避免跨层， 当走线参考地边沿时，保持线与参考 GND 最边沿的 距离>20mil</p>
<p>PCIE_G0_RXp[7:0] PCIE_G0_RXn[7:0]</p> <p>PCIE_G1_RXp[7:0] PCIE_G1_RXn[7:0]</p> <p>PCIE_H_RXp[7:0] PCIE_H_RXn[7:0]</p> <p>PCIE_F0_RXp[3:0] PCIE_F0_RXn[3:0]</p> <p>PCIE_F1_RXp[3:0] PCIE_F1_RXn[3:0]</p>	<p>若必须跨层，每两个 PCIE 差分信号对设置一个 GND 平面穿引孔，在距离信号过孔 150mil 范围内 设置 GND 孔</p>

3.3 USB

信号名称	描述
USB_DM[5:0]	差分阻抗 90Ω±15%
USB_DP[5:0]	每对差分等长 < 10mil

	信号总长度小于 12.5"
	ASIC 引线 (0.5"以内) =1:1, 在信号对两侧, 到所有其他信号的间距至少为 4:1; 蛇形信号间距至少=4:1
	ESD 器件离 connector 不超过 500mil
	共模扼流圈离 connector 不超过 1000mil
	信号线与 clock、power 等信号距离>500mil
	信号线尽量少打 via, 不能有直角, 不能偶 stub
	信号线不能走在晶振、clock synthesizers 和使用 clk 的 IC 下面。
	跨层时, 每个信号对设置一个 GND。GND via 距离 signal via 在 300mil 内。

Signals	W/S1	Trace Length	Zo(single)	Reference
USB[5:0]_TXR TUNE	TBD/BD	尽量短	50 Ohm	GND or PWR

Signals	W/S1	Breakout	Zo(single)	Reference
USB[2:0]_OC	TBD/BD	BD/TBD<0.5"	50 Ohm	GND or PWR

3.4 SATA

Signals	Trace Length	Breakout	Zo(diff)	Diff. match	Reference
SATA[2:0]_T Xp/n SATA[2:0]_R Xp/n	≤8"	TBD/TBD/TBD <0.5"	100 Ohm	<5mils	GND

SATA 信号不允许跨切割，不可有分支，trace 远离晶振、clock 信号，总长度≤ 8”。

Sata differential pair 上的耦合电容需放到 sata 连接器端，对称放置。

SATA_REFRES 走线宽度不小于 12mils。

3.5 HT

信号名称	描述
HT_RX_CADn[15:0]/CA Dp[15:0] HT_TX_CADn[15:0]/CA Dp[15:0] HT_RX/TX_CLKn/p[1:0] HT_RX/TX_CTLn/p[1:0]	若用作 16 位 HT 总线，每个通道（HT0 或 HT1）的这些信号分为两个 Group（TX、RX），若用作 8 位 HT 总线，信号分为四个 group（TX0\TX1、RX0\RX1）
	每对差分线等长误差<5mil（尽量不要绕线），同一个 Group 内非成对差分线等长误差<50mil
	在 breakout 区 Space : Height ≥ 1:1
	在 breakout 区外：若走线长度小于 5”，Space : Height ≥ 3:1 若走线长度小于 8”，Space : Height ≥ 4:1 若走线长度大于 8”，Space : Height ≥ 5:1
	差分阻抗 100 Ω ± 10%
	Group 间长度差 <1500mils

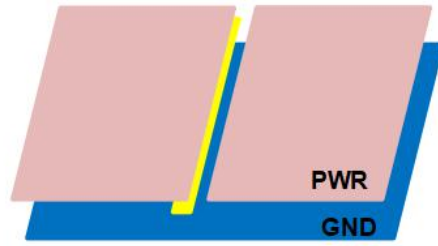
	同一个 Group 的信号必须走在同一 layer；作为 16 位使用时，如不能满足同组同层走线，要保证每 8bit（即 bit0-7 和 bit8-15）同层走线
	所有信号都参考 GND，且不能跨分割
	差分信号最多只能有两个过孔
	信号走线总长度大于 1”，小于 12.5”
LDT_STOPn LDT_PG (Hi/Lo_POWEROK) LDT_RSTn (Hi/Lo_RSTn)	在 breakout 区外，Space:Height \geq 3:1
	走线长度小于 8”
	走线不允许跨电源层分割

3.6 高速信号 layout 通用设计规则

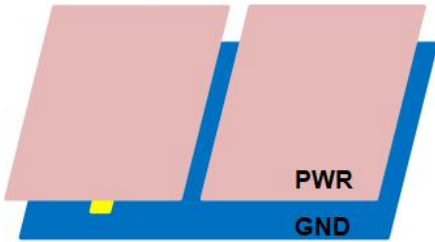
- 1) 避免走线参考平面跨分割



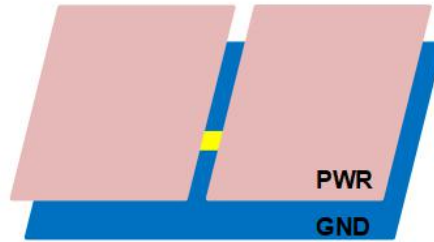
避免此种走线方式



避免此种走线方式



推荐此种走线方式



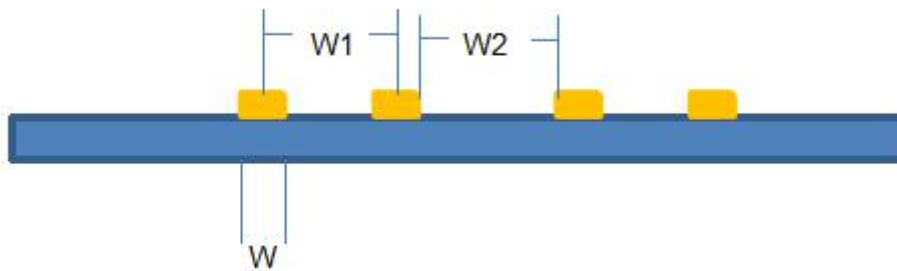
可以接受，但要保证信号线到
PWR 平面的距离是信号线到
GND 平面距离的 3 倍以上，
即满足 3H 原则

2) 高速差分信号线换层时打回流 GND 孔

在高速差分信号布线换层时，需要在换层过孔附近打回流 GND 孔。推荐几
何对称、规则的打孔方式。



3) 差分对线宽线距规则



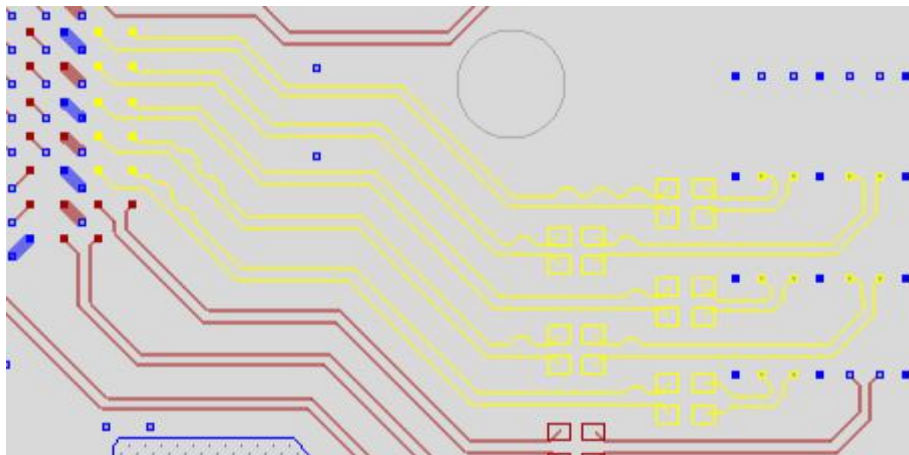
A) 差分对之间布线间距推荐满足 $3W$ 原则：相邻两条信号线的边沿到边沿的距离大于等于线宽的 3 倍，即 $W2 \geq 3W$.

B) 差分对内布线间距推荐满足 $3W$ 原则：差分对信号线中心距中心的距离大于等于线宽的 3 倍，即 $W1 \geq 3W$.

4) 差分对绕线规则:



A) 差分对内等长绕线补偿方式: 对于差分对内的等长, 如果产生不等长现象, 建议在产生不等长变化处或靠近产生不等长变化处绕线补偿回来。



B) 分段等长: 对于发生类似于换层、跨耦合电容的分段式走线方式, 建议分段做等长约束, 然后再整体做等长约束。

C) 差分对组内等长: 建议同组同层走线; 对于涉及表、底层走线换层的情况, 建议在表、底层做分段等长, (对于表、底层不具备绕线空间的情况, 可以将表、底层走线长度差换算为内层走线长度, 在内层补偿回来)。

5) 避免从电感、晶振、开关电源 mos、phase 平面下走线。

6) 远离存在大电压、大电流变化的 via 孔。

龙芯 3A4000_7A1000 外围功能 芯片&模组支持列表

V1.2

文档更新记录	文档名:	龙芯 3A4000_7A1000 外围功能芯片&模组 支持列表
	版本号:	V1.2
更新历史		
序号	版本号	更新内容
1	V1.0	初版发行
2	V1.1	删除内存模组列表信息
3	V1.2	增加多个 DVO 转接片 增加 codec 型号

《龙芯 3A4000_7A1000 外围功能芯片&模组支持列表》

类型	厂商	型号	备注
千兆网卡	Intel	82574	
		I210/I211	
		I350	
		82580	
	Realtek	RTL8111F/G/H	
	网迅	WX1860AL-W	
万兆网卡	网迅	WX1820AL	
千兆 PHY	Mavell	88E1510	
	Mavell	88E1111-BAB1I000	
	普维特电子	RPC101	无法关闭 EEE 模式
	Marvell	88E1512	
	Realtek	RTL8211E	
	Microchip	KSZ9031	
	裕太车通	YT8511	
USB3.0	瑞萨	UDP720201	四口
	钰创	EJ188H	四口
	祥硕	ASM1042	双口
USB3.1	祥硕	ASM3142	双口
SATA3.0	Mavell	9215	
	祥硕	ASM1061	
	祥硕	ASM1064	
PCIE SWITCH	Broadcom	PEX8632	
	祥硕	ASM8024	
	Broadcom	PEX8619	

PCI SWITCH	Broadcom	PEX8112	PCIE 转 PCI
SIO	Winbond	W83527	LPC
		W83795	I2C 接口
	Fintek	F81866AD	LPC
	Nuvoton	NCT6106D	LPC
DVO 编码器	Chrontel	CH7055A	DVO 转 VGA
	Anolog device	ADV7125KSTZ140	DVO 转 VGA
	振芯科技	GM7123	DVO 转 VGA
	深圳晶格微电子	SDA7123	DVO 转 VGA
	Anolog device	ADV7513	DVO 转 HDMI
	ITE	IT66121	DVO 转 HDMI
	Silicon image	Sii9022ACNU	DVO 转 HDMI
	龙迅	LT8618SXB	DVO 转 HDMI
	TI	TFP410PAP	DVO 转 DVI
	振芯科技	GM7510	DVO 转 DVI
	NewCoSemi	ncs8805	DVO 转 EDP
	振芯科技	GM8285C	DVO 转 LVDS
GPU	景嘉微	JM7201	PCIE x8
Codec	Realtek	ALC662	EOL
		ALC269	
		ALC897	
Rpid io	IDT	TSI721	PCIE 转 Rpid io
CLock gen	IDT	6P41505	
	Aura	Au5329	

	微禽半导体	MSC1421	
	IDT	9FG108EFILFT	
CLock buff	Aura	Au5411	
UART	江苏沁恒	CH384L	PCIE 转 UART
7A1000-DD R3 颗粒	三星	K4B2G1646F-BYM A	
		K4B2G1646Q-BCK 0	
		K4B1G1646G-BCK 0	
		K4B1G1646I-BCN B	
	ESMT	M15T1G1664A	
	紫光国芯	HXI15H4G160AF- 13K	EOL
3A4000-DD R4 颗粒	紫光国芯	SCB12Q4G160AF- 07Q	
	合肥长鑫	CXDQ3BFAM-CG	
板载 SSD	安信物联	AXH-US064MTW	
	三顿电子	SCUD128GMTWT	
	鸿秦科技	HTUSMU064G-W M	64GB
测温芯片	申矽凌	CT75MR	
	圣邦微	SGM452	

龙芯 CPU 开发系统固件与内核接口规范

V2.3

文档更新记录	文档名	龙芯 CPU 开发系统固件与内核接口规范
	版本号	V2.3
更新历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。
2	V1.1	修订版本 V1.1 1、重新修订了地址规范约束； 2、重新修订了 SMBIOS 规范中 CPU 型号的约定；
3	V1.2	修订版本 V1.2 1、添加传参新成员 cpuname 及其格式规范； 2、添加新传参功能：表示桥片个数、DMA cache/uncache； 3、添加新的结构体描述（efi_reset_system_t）； 4、添加成员 DoSuspend；
4	V2.0	修订版本 V2.0 1、重新修订附录 B bootparam.h 文件； 2、修正附录 A 中的几处错误；
5	V2.1	修订版本 V2.1

		<ol style="list-style-type: none">1、第 4 节修改，地址规范相关修改；2、增加附录 E，添加 3A/3B+7A 描述 (boardinfo、地址、中断) ，增加 7A GPU 使用地址段约束；3、完善 4.2 节地址规范约束；4、添加 4.3 节 DMA 规范描述以及 C、D、E 附录中 DMA 规范约束表；5、修改一些编写错误；
6	V2.2	<p>修订版本 V2.2</p> <ol style="list-style-type: none">1、第 4.2 节，低端内存的高 16M 地址空间相对 UEFI 做了约束；2、第 7 节，对 SMBIOS 产品信息 (Type2) 第六字段增加了命名约束，附录 A.6 board_devices.name 受同样约束；3、附录 A.2，删除现在接口中已经废弃的 screen_info 和 sys_desc_table；4、附录 A.3，删除现在接口中已经废弃的 systab 和 UEFI runtime 服务相关的接口；5、附录 A.4，规范了使用龙芯内置显卡情况 vbios 的处理；6、附录 A.8，固件传参接口的内存映射表中对应物理地址空间范围增加了对应的 DMA 地址空间范围；7、附录 A.9，固件传参接口添加 of_dtb_addr 成员；8、附录 A.12，对固件传参接口 cpuname 的使用进行补充完善；9、修改了一些语法错误和排版问题。

7	V2.3	<p>修订版本 V2.3</p> <ol style="list-style-type: none">1、 4.2 节，低端内存搞 16M 地址空间 UEFI 参考分布更新；2、 4.3 节，DMA 地址规范删除，参考不同平台的固件设计规范；3、 5.1.2 节，修改中断处理方式，参考各平台固件开发规范；4、 5.2 节，约束 LPC 接口使用方式；5、 第 7 章，修改 SMBIOS 约定必须传递的信息以及实现参考；6、 附录 A.4，更名 <code>smbios_tables</code> 为 <code>sysinfo_tables</code>；7、 附录 A.7，修改 <code>vers</code> 成员的定义、根据实际使用情况修改其他成员定义；8、 附录 B，更新 <code>bootparam.h</code> 文件；9、 删除附录 C、D、E 各平台中断及地址空间约定，请分别参考对应平台的固件开发规范； <p>修改了一些语法错误和排版问题。</p>
---	------	---

目 录

1 范围.....	3
2 术语与定义.....	3
3 架构关系.....	4
4 地址空间规范.....	4
4.1 芯片地址空间表.....	4
4.2 地址空间表.....	4
4.3 DMA 地址映射规范.....	7
5 中断配置规范.....	7
5.1 固件与内核关于中断的划分界面.....	7
5.1.1 固件与内核关于中断的分工.....	7
5.1.2 中断的处理方式.....	7
5.2 与板级硬件设计有关的约定.....	8
6 固件与内核接口传参规范.....	8
6.1 传参数据结构的约定.....	8
6.2 传参的实现.....	8
6.2.1 概述.....	8
6.2.2 固件中实现.....	8
6.2.3 参数的传递.....	8
6.2.4 内核中实现.....	9
7 SMBIOS 规范的实现约定.....	9
8 总结.....	12

附录 A 龙芯 CPU 开发系统传参数据结构.....	13
A.1 固件与内核接口定义.....	13
A.2 boot_params.....	13
A.3 efi_loongson.....	14
A.4 sysinfo_tables.....	15
A.5 loongson_params.....	16
A.6 board_devices.....	16

A.7 interface_info.....	17
A.8 efi_memory_map_loongson.....	18
A.9 system_loongson.....	19
A.10 uart_device.....	20
A.11 sensor_device.....	21
A.12 efi_cpuinfo_loongson.....	22
A.13 irq_source_routing_table.....	24
A.14 loongson_special_attribute.....	25
A.15 小结.....	25
附录 B 传参数据结构头文件 bootparam.h.....	27
B.1 说明.....	27
B.2 bootparam.h.....	27

前 言

本规范是龙芯中科技术股份有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范涉及到固件与内核之间的传参、中断分配及地址空间划分等方面，主要介绍龙芯板卡固件与内核之间的接口定义，地址空间分配，中断在内核和固件之间的划分、中断号的分配以及 SMBIOS 中需要实现的类型约定。接口定义涉及到固件与内核间信息传递的数据结构定义，固件使用何种方式传递这些数据结构及内核如何解析等方面。

1 范围

本规范规定了龙芯 CPU 开发系统的地址空间、中断配置、固件与内核接口传参实现及 SMBIOS 实现约定的要求。本规范适用于 3 号系列 CPU 开发系统。建议其它系统厂商遵循此规范开发相关产品。

本规范正文及附录 A、B 为通用规范，通用规范描述一般性的约定。详细规范为具体平台的固件开发规范：《AMD780E_710 芯片组固件开发规范》、《龙芯 2H 芯片组固件开发规范》、《龙芯 7A1000 芯片组固件开发规范》，针对不同开发系统进行具体约定。当通用规范与详细规范不一致时，以详细规范为准。

2 术语与定义

本规范所用术语定义如下：

- a) 固件：Firmware，写入 ROM、EPROM 等非易失存储器中的程序，负责控制和协调集成电路。
- b) BIOS：基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。
- c) UEFI：统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。
- d) PMON：MIPS 架构机器上使用的一种兼有 BIOS 和 boot loader 部分功能的开放源码软件。
- e) SMBIOS(System Management BIOS)：是主板或系统制造者以标准格式显示产品管理信息所需遵循的统一规范。DMI (Desktop Management Interface)是帮助收集电脑系统信息的管理系统，DMI 信息的收集必须在严格遵照 SMBIOS 规范的前提下进行。SMBIOS 和 DMI 是由行业指导机构 Desktop Management Task Force (DMTF)起草的开放性的技术标准。
- f) HT (HyperTransport)：是一种为主板上的集成电路互连而设计的端到端总线技术，目的是加快芯片间的数据传输速度。HT 通常指 CPU 到主板芯片（或北桥）之间的连接总线，即 HT 总线。类似于 Intel 平台中的前端总线（FSB），HT 按技术规格分有 HT1.0、HT2.0、HT3.0、HT3.1。
- g) PCI (Peripheral Component Interconnect)：是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。

3 架构关系

龙芯 PC 产品的固件与内核接口在系统各软件之间所处的层次关系如图 1 所示：

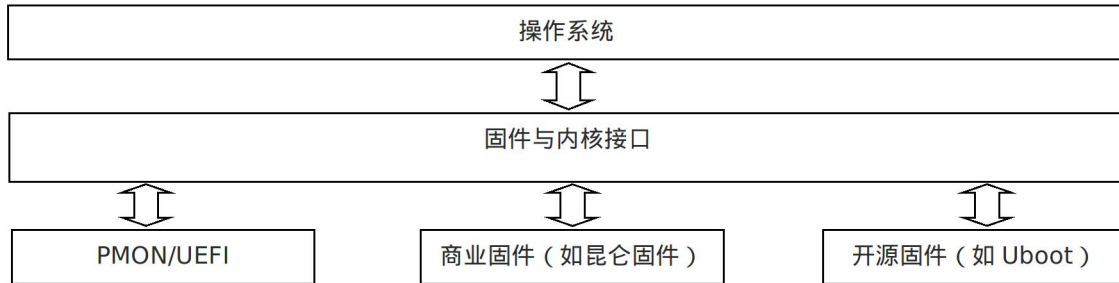


图 1 接口与内核和固件之间的关系

4 地址空间规范

4.1 芯片地址空间表

龙芯处理器可以通过芯片内的 PCI 接口连接外部桥片，也可以使用 HT 端口连接桥片。由于 HT 端口速度更快、位长更宽，现在多使用 HT1 端口连接桥片的模式。不同芯片的地址空间配置约定如各自的附录所示。

4.2 地址空间表

地址空间设计的规则约定如下：

- a) 0x0000_0000 ~ 0x0FFF_FFFF 的低 256MB 空间为低端内存空间。
其中 0x000 0000 ~ 0x001F FFFF 为兼容老版本固件保留的 2M 地址空间；0x0F00_0000 ~ 0x0FFF_FFFF 是为固件保留的 16M 地址空间，用于与内核的信息交互，如固件传参、关机和重启功能、Smbios、Vbios 等。当使用龙芯 UEFI 作为固件时，该 16M 地址空间要做特殊规划，防止上述数据分配到 16M 以下空间。参考实现如图 2。

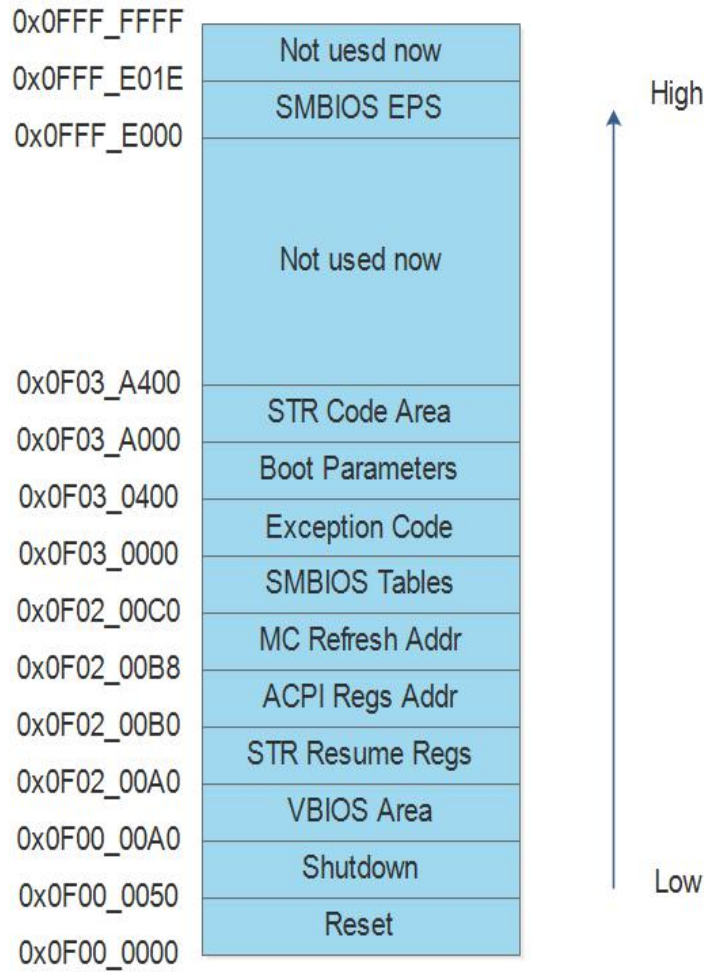


图 2 16MB 空间参考分布

- b) 0x1000_0000 ~ 0x1FFF_FFFF 为 PCI 等 IO 设备空间及部分芯片配置寄存器空间；
- c) 0x3000_0000 ~ 0x3FFF_FFFF 为窗口配置寄存器的空间范围；
- d) 0x4000_0000 ~ 0x7FFF_FFFF 为 PCI 设备 memory 空间范围；
- e) 0x2000_0000 ~ 0x2FFF_FFFF 和 0x8000_0000 ~ 高端内存基址-0x0000_0001 为保留空洞；
- f) 高端内存基址缺省值为 0x9000_0000，各系统如有特殊需要可从 emap 结构传递修改的值；
- g) 高端内存基址约束如下

对齐要求： $2^n + 0x10000000$ ， $31 \leq n < 40$ ；

高端内存基址参考示例：

- 0x9000_0000
- 0x11000_0000
- 0x41000_0000
- 0x81000_0000

下文，高端内存基址以 0x9000_0000 为例（本规范中内存地址相关示例均适用）进行说明。

- h) 内存地址信息必须由固件赋值后传参至内核，内核不可擅自修改。内存地址信息传参结构体描述详见附录 A.8 节。

地址空间划分如图 3 所示：

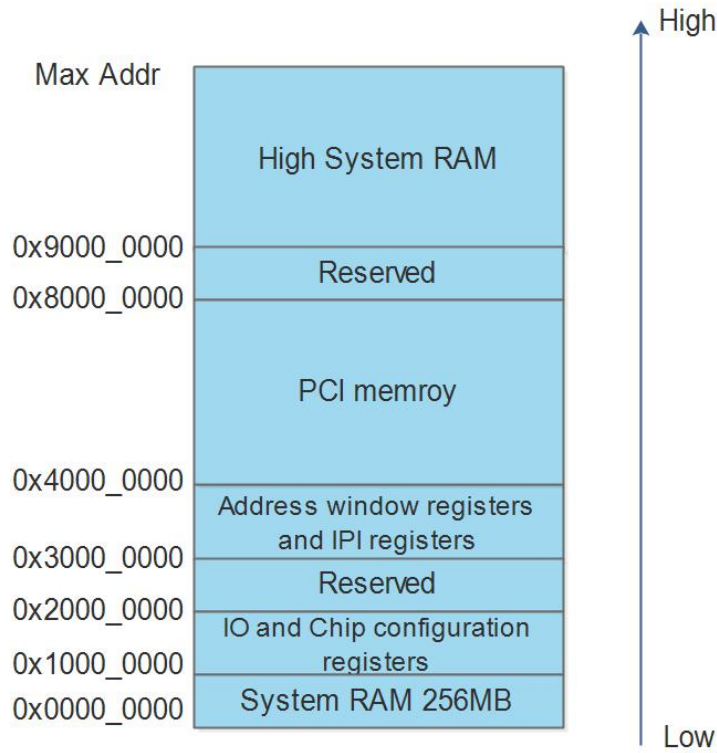


图 3 地址空间分布图

系统内存低 256MB 内，0x0F00_0000-0x0FFF_FFFF 为固件保留使用，0x0000_0000-0x001F_FFFF 为兼容老版本固件保留使用。

下面举例来说明高端内存的使用情况，

a) 如果内存大小为 1GB，则内存在系统中的地址空间表 1：

表 1 1GB 内存地址分布表

	起始地址	结束地址	物理内存
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	0 - 256MB
地址 1	0x0000_0000_9000_0000	0x0000_0000_BFFF_FFFF	256MB - 1GB

b) 如果内存大小为 2GB，则内存在系统中的地址空间如表 2：

表 2 2GB 内存地址分布表

	起始地址	结束地址	物理内存
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	0 - 256MB
地址 1	0x0000_0000_9000_0000	0x0000_0000_FFFF_FFFF	256MB - 2GB

c) 如果内存 4GB 以上，则内存在系统中的地址空间如表 3：

表 3 4GB 内存地址分布表

	起始地址	结束地址	物理内存
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	0 - 256MB

地址 1	0x0000_0000_9000_0000	0x0000_0001_7FFF_FFFF	256MB - 4GB
------	-----------------------	-----------------------	-------------

4.3 DMA 地址映射规范

参考具体平台的固件开发规范。

5 中断配置规范

5.1 固件与内核关于中断的划分界面

5.1.1 固件与内核关于中断的分工

龙芯芯片内部集成了中断控制器用于处理自带串口、PCI 总线等中断，又可以通过 PCI、HT 接口所连接桥片上的中断控制器来处理外接设备中断。因此，龙芯芯片系统的中断控制较为复杂。中断配置规范的目的是划清固件和操作系统关于中断的配置范围，统一龙芯芯片操作系统的中断分配。为此，进行如下约定：

- 桥片的中断路由由固件来做，其他的中断分配和路由由内核来做；
- 固件负责完成桥片上的 PCI/PCIE 设备和插槽的中断号申请和路由，操作系统内核完成中断号的分配和芯片内部的中断路由，好处是固件只和具体的板卡相关，内核负责 CPU 相关的初始化（包括 CPU 核间中断），这样一个内核可以适配不同板卡的固件；
- 固件完成 PCI 设备扫描和配置后，根据桥片手册、设备类型及硬件原理图，向每一个 PCI 设备或 PCI/PCIE 插槽上可能的设备的中断线寄存器写入申请的中断线号，同时需要对桥片上的中断控制器进行配置，保证它能路由到 HT 的 0-16 位中断向量的相应位置；
- 操作系统内核在分配中断号时，根据中断线寄存器的值进行分配。此外，操作系统内核还负责对芯片内部的中断控制器进行设置；
- 内核不再负责 CPU、板卡相关中断的具体配置，所有信息均由固件传递过来，并通过 `cpu_type` 的类型选择不同的平台支持，这样可以做到一个内核二进制在不同平台和 CPU 上都适用；
- 如果使用 MSI/MSI-X，由内核根据桥片的特性来分配中断，固件不再负责。

5.1.2 中断的处理方式

如果有 8259，其中断处理方式约定如下：

- 每次从 HT 中断向量寄存器中读取一个中断位，然后进行中断处理，并且清除 HT 相应的中断向量位，这种方式使用了 8259 默认的中断优先级；
- 固件需要将当前的设备中断号传递给内核，内核首先获知有哪些外设中断，并且给相应位使能。其他芯片的中断参考具体平台的固件开发规范。

5.2 与板级硬件设计有关的约定

为了保证兼容性，不建议使用处理器上的 LPC 接口，建议使用桥片上的 LPC 接口。

6 固件与内核接口传参规范

6.1 传参数据结构的约定

为便于同一个内核适配不同板卡、处理器，增强内核兼容性，龙芯平台规定由固件向内核提供 CPU 类型、中断号、总线信息、内存大小、pci-mem 地址、video_bios 地址、busclock 等信息。提供方式为：传递参数。该信息为内核必须获取信息，供内核解析结构体信息并启动运行。其中，同板卡不同芯片之间的区别可以通过 cpu_type、processor_id、PIC_type、ht_enable 以及 board_devices 等结构联合描述清楚。

龙芯 3 号开发系统采用附录 A 中的数据结构进行传参,其对应的头文件参见附录 B 所示,以后的扩展需要兼容此结构。

6.2 传参的实现

6.2.1 概述

本节的描述以 PMON 和 Linux 内核的实现为例来说明,但传参初始结构体指针 bp 规定放在 a2 寄存器中。

6.2.2 固件中实现

在固件中,每一个结构体针对一类型的功能或服务,具体定义在 boot_param.c 中,如 efi_memory_map 是针对系统的 memory 信息,efi_cpuinfo_loongson 处理 CPU 的相关信息等,为了给 UEFI 的模块开发预留空间,对每一类的服务或功能提供一个初始化函数,如:

```
int init_boot_param(struct boot_params *bp)
{
    init_efi(&(bp->efi));
    init_reset_system(&(bp->reset_system));

    return bp;
}
```

struct efi_memory_map_loongson *init_memory_map(){}中给每一项具体赋值。

6.2.3 参数的传递

在固件中初始化好所有的结构体或服务,最终封装到 boot_params 的结构体中,将该结构体指针 bp

赋值给 a2 寄存器。在 main.c 中调用 boot_params 的指针，并将指针赋值给 a2 寄存器。

6.2.4 内核中实现

在内核中，内核的实现主要是对该结构的解析工作，目前是在 env.c 中直接解析。首先定义和固件完全一样的结构体，在 head.S 中把 a2 寄存器读出，赋值给 fw_arg2，依次读出固件传过的信息，并对相应的结构体变量赋值即可；如：

```
struct boot_params *bp;  
bp = (struct boot_params)fw_arg2;  
emap = bp + bp->memory_offset;
```

等等逐次赋值解析，则相应的信息均可被内核使用。

注意：内核与固件的结构体必须完全一致，任何类型改动，都会导致内核无法启动。

7 SMBIOS 规范的实现约定

SMBIOS 是主板或系统制造者以标准格式显示产品管理信息所需遵循的统一规范。DMI (Desktop Management Interface, DMI)就是帮助收集电脑系统信息的管理系统，DMI 信息的收集必须在严格遵照 SMBIOS 规范的前提下进行。SMBIOS 和 DMI 是由行业指导机构 Desktop Management Task Force (DMTF) 起草的开放性的技术标准，其中，DMI 设计适用于任何的平台和操作系统。DMI 充当了管理工具和系统层之间接口的角色。它建立了标准的可管理系统更加方便了电脑厂商和用户对系统的了解。DMI 的主要组成部分是 Management Information Format (MIF)数据库。这个数据库包括了所有有关电脑系统和配件的信息。通过 DMI，用户可以获取序列号、电脑厂商、串口信息以及其它系统配件信息。

龙芯固件平台必须实现的 SMBIOS 类别如下：兼容 SMBIOS 2.3 规范版本，必须实现包含以下 11 个数据表结构：

- 1) BIOS 信息(Type 0)
- 2) 系统信息 (Type 1)
- 3) 产品信息 (Type 2)
- 4) 系统外围或底架 (Type3)
- 5) 处理器信息 (Type 4)
- 6) 高速缓存信息 (Type 7)
- 7) 系统插槽 (Type 9)
- 8) 物理存储阵列 (Type 16)
- 9) 存储设备(Type 17)
- 10) 存储阵列映射信息 (Type 19)
- 11) 表格结束指示 (Type 127)

其中：

Type0, BIOS 信息；此表的信息**必须实现，需要固件厂家或者主板厂家自行填写**，其中的固件版本(BIOS Version)约定格式见表 4。

表 4 BIOS 信息约定格式说明及举例

举例说明	厂商代号	连字符	固件名称	连字符	版本	连字符	发布日期
PMON3	Loongson	-	PMON	-	V3.0.1	-	20120808
UEFI	Loongson	-	UDK2018	-	V0.3.0		

Type2, 产品(Product)信息; 格式约定见表 5。

表 5 产品信息约定格式说明及举例

举例说明	厂商代号	连字符	CPU 型号	连字符	桥片	连字符	CPU 路数	连字符	板卡版本号	连字符	板卡标识
3A20002H 单路开发板	Loongson	-	LS3A2000	-	2H	-	1w	-	V1.01	-	ATX_EVB
3A780E 双路开发板	Loongson	-	LS3A	-	RS780E	-	2w	-	V1.03	-	ATX_EVB
龙梦 3A780E 双路带 BMC 控制服务器板	Lemote	-	LS3A	-	RS780E	-	2wBM C	-	V1.00	-	EATX_MB
3A30007A100 0 单路开发板	Loongson	-	LS3A3000	-	7A1000	-	1w	-	V1.00	-	ATX_EVB

表 5 注:

a) CPU 路数的后缀” w” 是小写以便和后面跟的诸如 BMC(Baseboard Management Controller,远程管理控制器)或 TCM(*Trust Cryptography Module*, 可信模块)区分。

b) CPU 型号约定如下:

3A 系列: LS3A1000、LS3A2000、LS3A3000、LS3A4000

3B 系列: LS3B1000、LS3B1500、LS3B3000、LS3B4000

c)板卡标识字段, 其命名规则约定如下:

产品类型/结构类型_主板属性/编码

产品类型 / 结构类型: 产品类型如笔记本电脑使用 laptop; 主板形态使用 PCI/VPX/ATX/MATX/ITX/等, 异形主板使用 YX。

主板属性/编码: 主板属性使用 MB/IO/CRB/EVB/POWER/等; 定制需要可使用编码

举例: laptop_a1001, PCI_1000, MATX_A1601, ATX_EVB 等等。

Type3, 系统外围或底架, **必须实现**。其中第六个字节的信息, 里面包含了这个主机的类型。见表 6。

表 6 SMBIOS Type3 第六字节信息含义

值	意义
01h	Other
02h	Unknown
03h	Desktop
04h	Low Profile Desktop

05h	Pizza Box
06h	Mini Tower
07h	Tower
08h	Portable
09h	Laptop
0Ah	Notebook
0Bh	Hand Held
0Ch	Docking Station
0Dh	All in One
0Eh	Sub Notebook
0Fh	Space-saving
10h	Lunch Box
11h	Main Server Chassis
12h	Expansion Chassis
13h	SubChassis
14h	Bus Expansion Chassis
15h	Peripheral Chassis
16h	RAID Chassis
17h	Rack Mount Chassis
18h	Sealed-case PC
19h	Multi-system chassis
1Ah	Compact PCI
1Bh	Advanced TCA
1Ch	Blade
1Dh	Blade Enclosure
1Eh	Tablet
1Fh	Convertible
20h	Detachable
21h	IoT Gateway
22h	Embedded PC
23h	Mini PC
24h	Stick PC

龙芯建议不同厂家的 platform driver 使用 DMI 的方式解析 SMBIOS 来获取当前主机信息，必须填写，否则可能相应驱动无法加载执行。

Type4, CPU 信息；描述 CPU 表项信息，由龙芯实现，无需固件和主板厂商填写，龙芯固件针对不同节点数量实现相应的参考代码。

Type7, Cache 信息; 描述 Cache 组织结构信息。由龙芯实现, 无需固件和主板厂商填写。

Type9, 系统插槽; 描述主板的 PCI、PCIE 插槽的信息, 该表项需要主板厂商根据各自设计自行完成; 可参考龙芯固件代码实现。

Type16, 物理存储阵列; 描述内存的信息, 如大小、DIMM 槽数量、错误信息 Handle 等。该表项需要主板厂商根据各自设计自行完成, 可参考龙芯固件代码实现。

Type17, 存储设备; 描述每个内存槽的信息, 比如类型、大小、是否 ECC 等。该表项需要主板厂商根据各自设计自行完成, 可参考龙芯固件代码实现。

Type19, 存储阵列映射信息; 描述内存映射到物理地址的范围。需要根据二级交叉开关的映射关系以及主存实际大小填写相应的地址范围; 该表项需要固件和主板厂商根据自己 BIOS 地址映射关系进行填写, 可参考龙芯固件代码实现。

Type127, 表格结束标识; 标识 SMBIOS 表的结尾, 无需固件和主板厂商填写, 龙芯代码已经实现。

8 总结

本传参规范的提出和制定旨在规范龙芯的固件和内核接口, 重点规定了地址空间、中断分配、传参数据结构、SMBIOS 实现约定等方面, 使得内核具有更广泛的适应性及兼容性, 能有效解决内核对具体板卡设备的依赖性, 有利于龙芯系列产品基础软件的规范和统一。

附录 A 龙芯 CPU 开发系统传参数据结构

A.1 固件与内核接口定义

本节规定了固件在加载内核阶段的接口定义，定义了接口信息结构体、预留了 UEFI 可扩展信息。信息结构体定义如下：

```
#define u64 phys_t
#define u64 unsigned long long
#define u32 unsigned int
#define u16 unsigned short
#define u8 unsigned char
```

注：为避免固件 32 位，内核 64 位的问题，在代码实现时，建议把指针替换成 u64，并在使用处注明 u64 代表什么结构体或定义，在本规范中不给出直接替换，否则会混淆结构含义，且对 UEFI/内核或 PMON64/内核 64 不具备兼容性。

传参主要结构体，如表 A.1 所示：

表 A.1 传参结构一览表

结构体	描述
struct boot_params	启动参数结构体，必选
struct efi	Efi 入口结构体，必选
struct sysinfo_tables	sysinfo 相关信息描述，必选
struct loongson_params	其他结构体相对的偏移地址
struct interface_info	规范本身的信息，描述规范和 BIOS 相关信息，必选
struct efi_loongson_memory_map	龙芯内存地址空间信息，必选
struct system_loongson	龙芯 system 的信息，如单路还是双路，是否用 numa 等
struct efi_cpuinfo_loongson	与龙芯 CPU 相关的信息，如频率，型号等，必选
struct irq_source_routing_table	中断信息，包含中断控制器，中断号等等
struct board_devices	板载设备的描述结构
struct uart_device	串口相关的设备结构
struct sensor_device	温度、风扇传感器相关的设备结构
struct loongson_special_attribute	龙芯特殊应用的预留接口

注：各板卡可根据需要选择必须实现的结构体，但如果选用，需按照本章约定的结构体进行传参，不能自行重新设计结构体，为保证兼容性，如果需要扩充本规范没有的结构体，可加在结构体 boot_params 的最后添加，并在各自的附录中给以说明。

A.2 boot_params

```
struct boot_params {
    struct efi_loongson efi; /*efi struct */
    struct efi_reset_system_t reset_system; /*information of reset*/
};
```

固件最终是将 boot_params 的结构体指针 bp 传递给内核，这样有一个统一的接口结构体，screen_info,sys_desc_table, efi_info 等可以暂时保留。

必须定义：strcut efi_loongson efi。

表 A.2.1 boot_params 结构体描述

成员	意义
efi	efi_loongson 结构
reset_system	系统重启、关机信息

其中 efi_reset_system_t 结构体：

```
struct efi_reset_system_t {
    u64 ResetCold;
    u64 ResetWarm;
    u64 ResetType;
    u64 Shutdown;
    u64 DoSuspend; /* NULL if not support */
};
```

表 A.2.2 efi_reset_system_t 结构体描述

成员	意义
ResetCold	冷启动
ResetWarm	热启动
ResetType	重启类型（保留）
Shutdown	关机
DoSuspend	挂起/休眠

注：各成员的定义据具体情况而定，例如笔记本电脑结构中的 Shutdown、ResetWarm 以及 DoSuspend 必须定义，ResetCold 、ResetType 可保留。

A.3 efi_loongson

```
struct efi_loongson {
```

```

u64 mps;                /* MPS table */
u64 acpi;                /* ACPI table (IA64 ext 0.71) */
u64 acpi20;             /* ACPI table (ACPI 2.0) */
struct sysinfo_tables sysinfo; /* sysinfo table */
u64 sal_systab;         /* SAL system table */
u64 boot_info;          /* boot info table */
};

```

该结构体中包含了与 efi 服务相关的很多信息，如 system_table, acpi,smbios,time 服务等，这些信息在当前 PMON 的基础上无法实现，是作为 UEFI 的传参结构体预留扩展预留的，目前主要使用的信息定义在 sysinfo 中。

必须定义：strcut sysinfo_tables sysinfo。

表 A.3 efi 结构体描述

成员	意义
mps	电源管理相关信息；
acpi	Acpi 的接口；
acpi20	Acpi 补充信息；
sysinfo	描述与厂商相关的信息，如 CPU,memory,irq 等；
sal_systable	SAL_Table 的信息；
boot_info	与启动相关的参数信息；

A.4 sysinfo_tables

```

struct sysinfo_tables {
    u16 vers;            /* version of sysinfo */
    u64 vga_bios;       /*vga_bios address */
    struct loongson_params lp;
};

```

sysinfo_tables 中的 loongson_params 成员部分内容可遵循 SMBIOS 规范填写，SMBIOS 是主板或系统制造者以标准格式显示产品管理信息所需遵循的统一规范，可以给出序列号、电脑厂商信息、固件、操作系统、主板、内存、扩展槽及扩展接口的详细规范。

sysinfo 给出版本信息、vga_bios 基地址以及龙芯平台相关的结构体信息。其中 vers 是标识本结构体的版本号，vga_bios 保存 vga_bios 的基地址，loongson_params 则是与龙芯体系相关的结构体；vga_bios 告诉内核 vga_bios 的基地址，在龙芯上使用 x86 模拟的显卡（如 RS780E 内置显卡），或者使用龙芯桥片内置显卡（如 7A1000 内置显卡），该地址必须也是龙芯板卡的特性之一,如果当前使用的是自带 vbios 的独立显卡，vga_bios 的值默认设定为 0，而不能是个未赋值的随机数。

必须定义：原则上，这结构体的每一项都必须定义，任何一个如果不定义或者定义出错，内核就无法启动。如 vga_bios，如果不定义此项图形界面无法正常启动。

注意：使用龙芯桥片(如 7A1000)内置显卡的情况下，固件必须为内置显卡准备 vbios 并存储在内存特

定区域中，把 vbios 内存地址通过 vga_bios 传递给内核。

A.5 loongson_params

```
struct loongson_params{
    u64 memory_offset;      /*efi_memory_map_loongson struct offset*/
    u64 cpu_offset;        /*efi_cpuinfo_loongson struct offset*/
    u64 system_offset;     /*system_loongson struct offset*/
    u64 irq_offset;        /*irq_source_routing_table struct offset*/
    u64 interface_offset;  /*interface_info struct offset*/
    u64 special_offset;    /*loongson_special_attribute struct offset*/
    u64 boarddev_table_offset; /*board_devices offset*/
};
```

该结构保存所有结构体相对 loongson_params 结构体地址的偏移，memory_offset 是内存结构体的偏移，cpu_offset 是 CPU 结构体的偏移，system_offset 是系统结构体的偏移，irq_offset 是龙芯中断控制器相关的偏移，special_offset 是龙芯特殊特性结构体的偏移，boarddev_table_offset 是龙芯板载设备结构体偏移。该结构体中只存放各个结构体的偏移，因此不能在填充其他变量。表 A.4 为和龙芯传参相关信息一览表。

表 A.4 loongson_params 成员相关结构体描述

成员	意义
efi_memory_map_loongson	龙芯内存地址空间信息；
efi_cpuinfo_loongson	与龙芯 CPU 相关的信息，如频率，型号等；
system_loongson	龙芯 system 的信息，如单路还是双路，是否用 numa 等；
irq_source_routing_table	中断信息，包含中断控制器，中断号等等；
interface_info	规范本身的信息，描述规范和 BIOS 相关信息信息；
loongson_special_attribute	龙芯的其他特殊属性；
board_devices	板载设备的描述结构；

A.6 board_devices

```
#define MAX_RESOUOCR_NUMBER 128
struct resource_loongson {
    u64 start; /*resource start address*/
    u64 end; /*resource end address*/
    u8 name[64];
    u32 flags;
};
```

```
struct archdev_data {};
struct board_devices {
    u8 name[64];           /*hold the device name*/
    u32 num_resources;    /*number of device_resource*/
    struct resource_loongson resource[MAX_RESOUOCR_NUMBER]; /*for each device`s
resource*/
    struct archdev_data archdata; /*arch specific additions*/
};
```

表 A.5 board_devices 结构体描述

成员	意义
name[64]	设备名称;
num_resources	设备资源个数;
resource[MAX_RESOUOCR_NUMBER]	设备资源信息;
archdata	设备相关的数据信息;

注：name 成员是用于固件传递主板设备名称给内核使用，必须实现；具体实现和约定参考本规范第 7 节表 6。

A.7 interface_info

```
struct interface_info {
    union {
        u16 vers; /*version of the specificition*/
        struct {
            u8 minor;
            u8 major;
        } version;
    };
    u16 size;      /*size of BIOS ROM*/
    u8 flag;      /*use or unuse*/
    u8 description[64]; /*BIOS description*/
}__attribute__((packed));
```

本结构提供 interface 版本信息，vers 表示本规范的版本信息，可以用来区别不同版本下不同的功能，重大改动版本使用 major 表示。小改动版本使用 minor 表示；flag 标识当前内核或固件有没有使用本规范。这个作为接口升级和变更标记，同时区别可以规范前后固件和内核的情况，使得内核的处理和使用有一定的兼容性。size 表示 BIOS 的 ROM 大小，单位是 KB。description[64]用于描述 BIOS 版本。

必须定义：vers、flag。

表 A.6 interface_info 结构体描述

成员	意义
----	----

vers	interface_info 的版本信息，标识规范的版本更迭；
size	BIOS ROM 大小，单位是 KB；
flag	标识当前有没有使用规范；
description[64]	BIOS 版本描述。

注：description 成员用于传递 BIOS 版本信息供内核使用，必须实现，具体实现和约定见本规范第 7 节表 4。

A.8 efi_memory_map_loongson

针对龙芯平台，定义了如下内存类型：

```
#define SYSTEM_RAM_LOW 1
#define SYSTEM_RAM_HIGH 2
#define MEM_RESERVED 3
#define PCI_IO 4
#define PCI_MEM 5
#define LOONGSON_CFG_REG 6
#define VIDEO_ROM 7
#define ADAPTER_ROM 8
#define ACPI_TABLE 9
#define SMBIOS_TABLE 10
#define UMA_VIDEO_RAM 11
#define VUMA_VIDEO_RAM 12
#define SYSTEM_RAM_DMA_LOW 13
#define SYSTEM_RAM_DMA_HIGH 14
#define MAX_MEMORY_TYPE 15
struct efi_memory_map_loongson {
    u16 vers; /*version of efi_memory_map*/
    u32 nr_map; /*number of memory_maps*/
    u32 mem_freq; /*memory frequency*/
    struct mem_map{
        u32 node_id; /*recorde the node_id*/
        u32 mem_type; /*recorde the memory type*/
        u64 mem_start; /*memory map start address*/
        u32 mem_size; /*for each memory_map size,not the total size*/
    }map[LOONGSON3_BOOT_MEM_MAP_MAX];
}__attribute__((packed));
```

龙芯内存是统一编址的，整个内存是一张表，在本结构中，vers 表示当前 memory_map 的版本号，nr_map 是传递过来的内存表的个数，内核扫描传过来的全部的内存表，然后根据下面结构体中 node_id,mem_type 等两项判断传递过来的内存表是什么表，挂在哪个节点上。从而得到每个内存表的起始

地址，大小等等信息，sysinfo_tables 使用此结构体向内存发布。

当 vers>=2 时，DMA 的地址空间范围允许由固件传递给内核，通过传递 DMA 地址范围来表达 DMA 地址和物理地址的对应关系。这种情况需要 map 数组顺序满足如下约束：

1. 物理地址先于 DMA 地址
2. 物理地址顺序与 DMA 地址顺序一一对应

举例：

```
[0] {Low Physical memory region},
[1] {High Physical memory region},
[2] {SMBIOS_TABLE memory region},
[3] {other memory region},
[4] {Low DMA memory region},
[5] {High DMA memory region}}
```

数组中，[0][1]作为物理地址段优先于[4][5]DMA 地址段；物理地址与 DMA 地址对应关系[0]-[4] [1]-[5]。

注：当传递的 DMA 地址被检测无效时，内核会使用缺省的 DMA 地址和物理内存地址转换关系。

表 A.7 efi_memory_map_loongson 结构体描述

成员	意义
vers	Efi_memory_map_loongson 的版本信息；
nr_map	内存表的个数，记录从固件传递过来的所有表的个数；
mem_freq	内存的频率， ddr 频率；
struct mem_map	表示每张表的信息的结构体；
	node_id :当前内存表挂在那个节点上； mem_type:当前内存表的类型； mem_start:当前表的起始地址； mem_size : 当前表的大小；

注：固件中此结构体必须定义和赋值，内存地址信息必须通过此结构体传递给内核。

A.9 system_loongson

```
struct system_loongson {
    u16 vers;                //version of system_loongson
    u32 ccnuma_smp ;        //0:smp;1:numa;
    u32 sing_double_channel; //1:single; 2:double
    u32 nr_uarts;           //number of uarts
    struct uart_device uarts[MAX_UARTS];
    u32 nr_sensors;        //number of sensors
};
```

```

struct sensor_device sensors[MAX_SENSORS];
u8 has_ec;
u8 ec_name[64];
u64 ec_base_addr;
u8 has_tcm;
u8 tcm_name[64];
u64 tcm_base_addr;
u64 workarounds;
u64 of_dtb_addr
}_attribute__((packed));

```

这个结构体主要包含龙芯特性，如：系统是否用 ccnuma，当前是单路还是双路等等，也可以添加其他的特性。

ccnuma 用来判断，启动 numa-os 还是 smp；single_double 则选择单路还是双路；

在多核芯片中必须定义：ccnuma_smp，不定义影响性能；

uarts 用来传递主板的串口配置，包括串口地址和时钟频率，要求至少传递一个 CPU0 的 UART0 配置；

当 vers>=2 时，操作系统依赖固件传递的 FDT 信息，固件必须传递此信息。

表 A.8 system_loongson 结构体描述

成员	意义
vers	龙芯系统版本信息；
ccnuma_smp	0: 代表 smp, 1: 代表 numa;
single_double_channel	1: 代表单路, 2: 代表双路;
nr_uarts	串口数量
uarts[MAX_UARTS]	串口设备数组
nr_sensors	传感器数量
sensors[MAX_SENSORS]	传感器设备数组
has_ec	是否存在 EC
ec_name[64]	用于创建 platform_device 的 EC 名字
ec_base_addr	EC 寄存器基地址
has_tcm	是否存在 TCM
tcm_name[64]	用于创建 platform_device 的 TCM 名字
tcm_base_addr	TCM 寄存器基地址
workarounds	根据有缺陷的机型做 workarounds, 有厂家自定义, 一般在 workarounds.h 里定义
of_dtb_addr	DTB 地址, 0 代表固件无 DTB 信息

A.10 uart_device


```
#define MAX_UARTS 64
struct uart_device {
    u32 iotype; /* see include/linux/serial_core.h */
    u32 uartclk;
    u32 int_offset;
    u64 uart_base;
}__attribute__((packed));
```

该结构体为描述串口相关信息 MAX_UARTS 为最大串口数量，本结构体是可选结构体，该结构体的访问通过 system_loongson 结构体寻址得到，而不是通过 smbios 结构体直接访问。

表 A.9 uart_device 结构体描述

成员	意义
iotype	标示 IO 类型是 MMIO 还是 PortIO;参照内核下的 Iotype;
uartclk	串口的时钟频率;
int_offset	串口 IRQ 号的偏移量, MMIO 相对于 56, PortIO 是相对于 0;
uart_base	串口寄存器的基地址。

A.11 sensor_device

```
#define MAX_SENSORS 64
#define SENSOR_TEMPER 0x00000001
#define SENSOR_VOLTAGE 0x00000002
#define SENSOR_FAN 0x00000004
struct sensor_device {
    u8 name[32]; /* a formal name */
    u8 label[64]; /* a flexible description */
    u32 type; /* SENSOR_* */
    u32 id; /* instance id of a sensor-class */
    u32 fan_policy; /* see arch/mips/include/asm/mach-loongson/loongson_hwmon.h */
    u32 fan_percent; /* only for constant speed policy */
    u64 base_addr; /* base address of device registers */
}__attribute__((packed));
```

该结构体包括温度、电压传感器和风扇的描述，MAX_SENSORS 是最大传感器数量，本结构体为可选结构体。该结构体的访问通过 system_loongson 结构体寻址得到，而不是通过 smbios 结构体直接访问。

表 A.10 sensor_device 结构体描述

成员	意义
name[32]	用于创建 platform_device 的名字;

label[64]	自定义的一个描述名;
type	传感器类型;
id	区分同等设备的不同实例;
fan_policy	三种策略，定速 CONSTANT_SPEED_POLICY、变速 STEP_SPEED_POLICY 和内核辅助 KERNEL_HELPER (EC 控制);
fan_percent	定速策略专用，用于风扇转速的百分比;
base_addr	传感器寄存地址的基地址。

A.12 efi_cpuinfo_loongson

```
typedef enum loongson_cpu_type {
    Legacy_2F = 0x1,
    Legacy_3A = 0x2,
    Legacy_3B = 0x3,
    Legacy_1A = 0x4,
    Legacy_1B = 0x5,
    Legacy_2H = 0x7,
    Loongson_1A = 0x100,
    Loongson_1B = 0x101,
    Loongson_2F = 0x201,
    Loongson_2H = 0x203,
    Loongson_3A = 0x300,
    Loongson_3B = 0x301
};

/*
 * Capability and feature descriptor structure for MIPS CPU
 */
struct efi_cpuinfo_loongson {
    u16  vers;                /*version of efi_cpinfo_loongson*/
    u32  processor_id;       /* PRID, e.g. 6305, 6306 */
    enum loongson_cpu_type  cputype; /*3a-3b-2f-2e-1a-1b*/
    u32  total_node;        /* physical core number */
    u16  cpu_startup_core_id; /* Core id*/
    u16  reserved_cores_mask; /*Reserved Core mask*/
    u32  cpu_clock_freq;    /*cpu_clock */
    u32  nr_cpus;          /*number of cpus*/
};
```

```
u8  cpuname[64];          /*cpu name*/
}_attribute__((packed));
```

注： loongson_cpu_type 中的 Legacy_3B 和 Loongson_3B 特指处理器型号为 3B1500。

Processor_id 是从 PRID 寄存器里读取的值,以 16 进制表示, cputype 传递是龙芯哪个型号的, total_node 保存当前 CPU 中包含几个节点, cpu_startup_core_id 保存启动核的 id, cpu_clock_freq 是当前 CPU 的频率, 与 CPU 相关的信息在传递过程中要非常小心, 如果弄错一点, 内核绝对无法启动。

必须定义： loongson_cpu_type、cpu_clock_freq、nr_cpus、total_node、cpuname 等。

表 A.11 efi_cpuinfo_loongson 结构体描述

成员	意义
vers	CPU 结构的版本号;
processor_id	标识处理器的型号, 具体的批次, 如 3a3,3b2 等;
loongson_cpu_type	CPU 的类型, 如 1A,1B,2E,2F,3A,3B 等;
total_node	总结点数;
cpu_startup_core_id	CPU 启动核的 id 号,用其 10 进制值表示;
reserved_cores_mask	可工作的处理器核掩码, 核序和字节的位序相同, 即 0~15 位分别对应 0~15 号核的状态; 0 表该核可用, 1 表不可用;
cpu_clock_freq	CPU 频率;
nr_cpus	CPU 物理核数;
cpuname[64]	CPU 名称

vers 决定内核中 cpuname 的来源。当 vers>=2 时, cpuname 由固件生成, 内核直接接收使用; vers<2 时, cpuname 由内核自己判断并填充。

cpuname 约束如下:

- 1) 结构约束: cpuname 分为两部分, 括号外为核心代号, 括号内为商品名称;
- 2) 内容约束:

核心代号和商品名称的起始字符串必须为 Loongson;

商品名称格式为 Loongson-版本号(小版本号)-(工业等级)。(注: 芯片等级说明: I, 专业工业级, 芯片有唯一编号; i, 普通工业级。)

举例:

```
3A1000: "Loongson-3A R1 (Loongson-3A1000)";
3A2000: "Loongson-3A R2 (Loongson-3A2000)";
3A3000: "Loongson-3A R3 (Loongson-3A3000)";
3A3000F: "Loongson-3A R3 (Loongson-3A3000F)";
```

```
3B1500: "Loongson-3B R2 (Loongson-3B1500)";
3B2000: "Loongson-3A R2 (Loongson-3B2000)";
3B3000: "Loongson-3A R3 (Loongson-3B3000)";
3B4000: "Loongson-3B R4 (Loongson-3B4000)";
```

A.13 irq_source_routing_table

该结构体描述龙芯上中断控制的信息。

```
struct irq_source_routing_table {
    u16 vers ; /*version of irq_source_routing_table*/
    u16 size ; /*bridge number*/
    u16 rtr_bus; /*bus number*/
    u16 rtr_devfn; /* Where the interrupt router lies */
    u32 vendor;
    u32 device;
    u32 PIC_type; /* Vendor and device ID of interrupt router PIC */
    u64 ht_int_bit; //3a: 1<<24; 3b:1<<16
    u64 ht_enable;//all irq's used in HT PIC which from 8259 or other PICS
    u32 node_id; //the PIC interrupter attach to the cpu_node
    u64 pci_mem_start_addr;
    u64 pci_mem_end_addr;
    u64 pci_io_start_addr;
    u64 pci_io_end_addr;
    u64 pci_config_addr;
    u16 dma_mask_bits;/*dma 的位数*/
    u16 dma_noncoherent; /* DMA cache or uncach*/
}__attribute__((packed));
```

vers 是当前结构体版本号，rtr_bus 等表示中断控制器连接在哪个总线上，vendor,device 表示当前中断控制器的设备信息,PIC_type 告诉内核当前用的是什么类型的中断控制器，ht_int_bit 告诉内核当前的中断控制器是哪一位置 1 便使能 HT 中断控制器。ht_enable 表示板上所有设备的中断号，node_id 表示中断控制器接在哪个 CPU 节点上。pci_mem 相关信息传递当前 pci 的 mem,io 的起始，结束信息等，与 efi_memory_map 一起，就把规范了内存地址使用信息传递给内核，以后内核不再再次扫描分配，而直接使用规定好的内存段即可。Irq_info 传递了当前板卡上 PCI、PCIE、min-PCI 槽的中断，接的 bus 号，设备名称等，目前该 slots 暂时不用，目前的处理是把槽号当做 PCI 设备，其中断号等直接写入 PCI 设备的寄存器，内核扫描 PCI 设备即可获得。内核接收到该信息后，只处理固件传递过来的中断信息及设备中断，而不再做判断和扫描。

必须定义：pci_mem_start_addr、pci_mem_end_addr、pci_io_start_addr、pci_io_end_addr。

采用 HT 总线的必须定义:ht_int_bit、ht_enable。

表 A.12 irq_source_routing_table 结构体描述

成员	意义
vers	本结构体版本号；
size	表示桥片个数
ht_int_bit	中断控制器哪一位使能；
rtr_bus,rtr_devfn	中断路由所在的 bus 号；

vendor, device, PIC_type	中断控制器的厂商号, 设备号, 及其本身的类型;
ht_enable	Ht 中断控制器使能的中断位, i8259 也可用其标识;
node_id	中断控制器接到哪一个 CPU 核上;
pci_mem_start_addr	Pci_mem 空间的起始地址;
pci_mem_end_addr	Pci_mem 空间的结束地址;
pci_io_start_addr	Pci_io 空间的起始地址;
pci_io_end_addr	Pci_io 空间的结束地址;
pci_config_start_addr	Pci_config 空间的起始地址;
dma_mask_bits	DMA 的位数: 32 位或 64 位
dma_noncoherent	表示 DMA cache 或 uncache 0: 表示 cache DMA 1: 表示 Ucache DMA

A.14 loongson_special_attribute

```
struct loongson_special_attribute{
    u16 vers; /*version of this special*/
    u8 special_name[64]; /*special_attribute_name*/
    u32 loongson_special_type; /*type of special device*/
    struct resource_loongson resource[MAX_RESOURCE_NUMBER]; /*special_vlaue
resource*/
}__attribute__((packed));
```

该结构体为龙芯的特殊应用留下接口, 定义该特殊应用的版本, 名称, 类型, 地址及 data 的其他属性。本结构暂不实现。

表 A.13 loongson_special_attribute 结构体描述

成员	意义
vers	本结构版本号;
special_name[64]	特殊属性的名称;
loongson_special_type	特殊属性的类型;
resource[MAX_RESOURCE_NUMBER]	特殊属性是否又含有自身的资源、属性之类;

本结构作为扩展可存放在龙芯平台上的一些特殊的属性, 如之前 vga_bios 的基地址等。special_name 成员作为 BIOS 的发布时间使用, 参考龙芯固件实现。

A.15 小结

固件的接口结构体, 能满足当前 PMON 基础上固件和内核的交互应用, 预留的 UEFI 扩展, 涵盖 UEFI 基本服务信息, 详细、精确的扩展需要在使用 UEFI 的时候具体调试, 定义。

固件接口中各模块的具体包含关系如下:

```
Struct boot_params{->struct efi{->struct smbios{->  
|->struct efi_memory_map;  
|->struct efi_cpuinfo_loongson;  
|->struct irq_source_routing_table;  
|->struct system_loongson;  
|->struct uart_device;  
|->struct sensor_device;  
|->struct interface_info;  
|->struct board_device;
```

所有的结构体均定义在 bootparam.h 中，所需要保证的是固件和内核中的 bootparam.h 完全一致，否则可能导致内核解析不正确，其中 uart_device 和 sensor_device 的访问是通过结构体 system_loongson，而不是通过结构 smbios 的偏移访问。

附录 B 传参数据结构头文件 bootparam.h

B.1 说明

此头文件在内核和固件中须保持一致，其结构体的描述参见附录 A。

B.2 bootparam.h

```
#ifndef __ASM_MACH_LOONGSON_BOOT_PARAM_H_
#define __ASM_MACH_LOONGSON_BOOT_PARAM_H_

#define SYSTEM_RAM_LOW          1
#define SYSTEM_RAM_HIGH        2
#define MEM_RESERVED            3
#define PCI_IO                  4
#define PCI_MEM                 5
#define LOONGSON_CFG_REG        6
#define VIDEO_ROM               7
#define ADAPTER_ROM             8
#define ACPI_TABLE              9
#define SMBIOS_TABLE            10
#define UMA_VIDEO_RAM           11
#define VUMA_VIDEO_RAM          12
#define SYSTEM_RAM_LOW_DMA      13
#define SYSTEM_RAM_HIGH_DMA     14
#define MAX_MEMORY_TYPE         15

#define LOONGSON3_BOOT_MEM_MAP_MAX 128
struct efi_memory_map_loongson{
    u16 vers;          /* version of efi_memory_map */
    u32 nr_map;        /* number of memory_maps */
    u32 mem_freq;     /* memory frequency */
    struct mem_map{
        u32 node_id;   /* node_id which memory attached to */
        u32 mem_type;  /* system memory, pci memory, pci io, etc. */
        u64 mem_start; /* memory map start address */
        u32 mem_size;  /* each memory_map size, not the total size */
    }map[LOONGSON3_BOOT_MEM_MAP_MAX];
};
```

```
}__attribute__((packed));

enum loongson_cpu_type
{
    Legacy_2F = 0x1,
    Legacy_3A = 0x2,
    Legacy_3B = 0x3,
    Legacy_1A = 0x4,
    Legacy_1B = 0x5,
    Legacy_2H = 0x7,
    Loongson_1A = 0x100,
    Loongson_1B = 0x101,
    Loongson_2F = 0x201,
    Loongson_2H = 0x203,
    Loongson_3A = 0x300,
    Loongson_3B = 0x301
};

/*
 * Capability and feature descriptor structure for MIPS CPU
 */
struct efi_cpuinfo_loongson {
    u16 vers;      /* version of efi_cpuinfo_loongson */
    u32 processor_id; /* PRID, e.g. 6305, 6306 */
    enum loongson_cpu_type cputype; /* 3A, 3B, etc. */
    u32 total_node; /* num of total numa nodes */
    u16 cpu_startup_core_id; /* Core id */
    u16 reserved_cores_mask;
    u32 cpu_clock_freq; /* cpu_clock */
    u32 nr_cpus;
    u8 cpuname[64]; /*cpu name*/
}__attribute__((packed));

#define MAX_UARTS 64
struct uart_device {
    u32 iotype; /* see include/linux/serial_core.h */
    u32 uartclk;
    u32 int_offset;
    u64 uart_base;
```



```
}_attribute__((packed));

#define MAX_SENSORS 64
#define SENSOR_TEMPER 0x00000001
#define SENSOR_VOLTAGE 0x00000002
#define SENSOR_FAN 0x00000004
struct sensor_device {
    char name[32]; /* a formal name */
    char label[64]; /* a flexible description */
    u32 type; /* SENSOR_* */
    u32 id; /* instance id of a sensor-class */
    u32 fan_policy; /* see arch/mips/include/asm/mach-loongson/loongson_hwmon.h */
    u32 fan_percent; /* only for constant speed policy */
    u64 base_addr; /* base address of device registers */
}_attribute__((packed));

struct system_loongson{
    u16 vers; /* version of system_loongson */
    u32 ccnuma_smp; /* 0: no numa; 1: has numa */
    u32 sing_double_channel; /* 1:single; 2:double */
    u32 nr_uarts;
    struct uart_device uarts[MAX_UARTS];
    u32 nr_sensors;
    struct sensor_device sensors[MAX_SENSORS];
    char has_ec;
    char ec_name[32];
    u64 ec_base_addr;
    char has_tcm;
    char tcm_name[32];
    u64 tcm_base_addr;
    u64 workarounds; /* see workarounds.h */
    u64 of_dtb_addr; /* NULL if not support */
}_attribute__((packed));

struct irq_source_routing_table {
    u16 vers;
    u16 size;
    u16 rtr_bus;
    u16 rtr_devfn;
```

```
u32 vendor;
u32 device;
u32 PIC_type; /* conform use HT or PCI to route to CPU-PIC */
u64 ht_int_bit; /* 3A: 1<<24; 3B: 1<<16 */
u64 ht_enable; /* irqs used in this PIC */
u32 node_id; /* node id: 0x0-0; 0x1-1; 0x10-2; 0x11-3 */
u64 pci_mem_start_addr;
u64 pci_mem_end_addr;
u64 pci_io_start_addr;
u64 pci_io_end_addr;
u64 pci_config_addr;
u16 dma_mask_bits;
u16 dma_noncoherent;
}_attribute__((packed));

struct interface_info{
    union {
        u16 vers; /*version of the specification*/
        struct {
            u8 minor;
            u8 major;
        } version;
    };

    u16 size;
    u8 flag;
    char description[64];
}_attribute__((packed));

#define MAX_RESOURCE_NUMBER 128
struct resource_loongson {
    u64 start; /* resource start address */
    u64 end; /* resource end address */
    char name[64];
    u32 flags;
};

struct archdev_data {}; /* arch specific additions */
```

```
struct board_devices{
    char name[64];    /* hold the device name */
    u32 num_resources; /* number of device_resource */
    struct resource_loongson resource[MAX_RESOURCE_NUMBER]; /* for each device's
resource */
    /* arch specific additions */
    struct archdev_data archdata;
};

struct loongson_special_attribute{
    u16 vers;    /* version of this special */
    char special_name[64]; /* special_attribute_name */
    u32 loongson_special_type; /* type of special device */
    struct resource_loongson resource[MAX_RESOURCE_NUMBER]; /* for each device's
resource */
};

struct loongson_params{
    u64 memory_offset;    /* efi_memory_map_loongson struct offset */
    u64 cpu_offset;    /* efi_cpuinfo_loongson struct offset */
    u64 system_offset;    /* system_loongson struct offset */
    u64 irq_offset;    /* irq_source_routing_table struct offset */
    u64 interface_offset; /* interface_info struct offset */
    u64 special_offset;    /* loongson_special_attribute struct offset */
    u64 boarddev_table_offset; /* board_devices offset */
};

struct smbios_tables {
    u16 vers;    /* version of smbios */
    u64 vga_bios; /* vga_bios address */
    struct loongson_params lp;
};

struct efi_reset_system_t{
    u64 ResetCold;
    u64 ResetWarm;
    u64 ResetType;
    u64 Shutdown;
    u64 DoSuspend; /* NULL if not support */
};
```

```
};

struct efi_loongson {
    u64 mps;          /* MPS table */
    u64 acpi;        /* ACPI table (IA64 ext 0.71) */
    u64 acpi20;     /* ACPI table (ACPI 2.0) */
    struct sysinfo_tables sysinfo; /* sysinfo table */
    u64 sal_systab; /* SAL system table */
    u64 boot_info; /* boot info table */
};

struct boot_params{
    struct efi_loongson efi;
    struct efi_reset_system_t reset_system;
};

#endif
```

AMD780E+710 芯片组
固件开发规范
V1.0

目 录

1 DMA 地址映射.....	1
2 中断.....	1
2.1 中断系统描述.....	1
2.2 中断号分配约定.....	1
3 地址空间分布表.....	2

前 言

本规范是龙芯中科技术股份有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范面向固件软件开发人员，在《龙芯 CPU 开发系统固件与内核接口规范》基础上针对 AMD780+710 芯片组的固件软件开发要点进行补充说明，正文提到的 3A/3B 包括 3A/3B4000 处理器及该型号之前的 3 号处理器。

1 DMA 地址映射

龙芯平台搭配 AMD780E 芯片组 DMA 地址与内存物理地址对应关系分为两种情况：

第一种，内存地址在低 256MB 映射关系为： $\text{dmaaddr}=\text{phyaddr} | 0\text{x}8000_0000$;

第二种，内存地址大于 256MB 的部分，映射关系为 1:1 映射。

2 中断

2.1 中断系统描述

3A/B+780E 方案，所有的外设中断都是路由到 0 号核（即 CPU 0），其中片上的 UART 控制器的中断直接进入 Int Controller 的 UART/LPC 脚，最后路由到 CPU 0 的 IP3，而片外的 780E 桥片上的中断则是通过 HT1 控制器进入到 Int Controller 的 HT1-0 脚，最后路由到 CPU 0 的 IP2，其示意图如下。

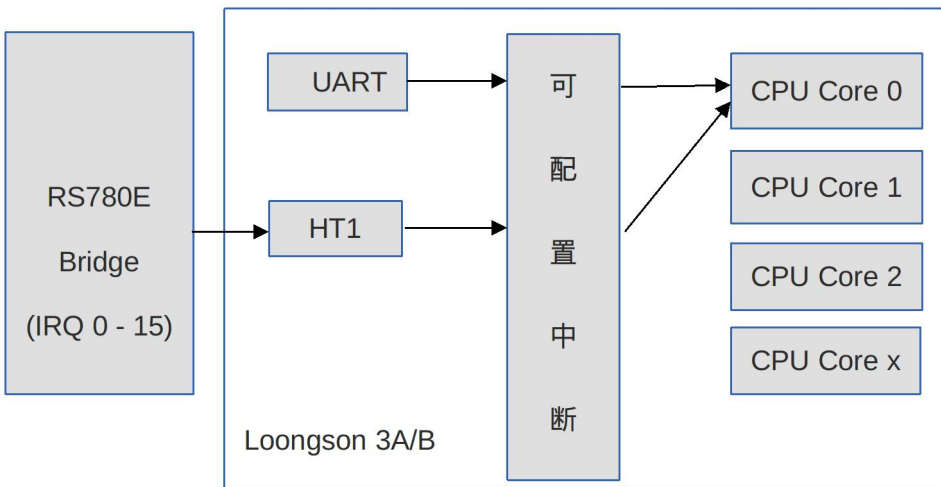


图 1 3A/B+780E 中断路由示意图

2.2 中断号分配约定

0-15 号保留分配给连接诸如 8259 的传统设备的中断。原则上，龙芯芯片 HT 总线接口和芯片 PCI 总线接口不会同时使用，因此，8259 中断控制寄存器只可能唯一出现在 HT 或者 PCI 总线上。由于历史原因，

8259 中断控制寄存器也不是所有的中断号都可以使用，如表下所示的中断号就被保留。此外，剩下的中断号有些会被芯片组保留，具体的需要查询芯片组的中断控制器相关说明。

中断号	中断源	说明
0	HPET	HPET 高精度定时器
1	I8042	XT-PIC 键盘
2	级联	XT-PIC
3		
4		
5		
6		
7	SCI	系统控制中断（用于笔记本电脑 Fn 功能键）
8	RTC	XT-PIC 实时时钟
9		
10		
11		
12	I8042	XT-PIC 鼠标
13		
14	ide0	XT-PIC 硬盘
15	Ide1	XT-PIC 硬盘

16-47 号对应分配给芯片内部的中断控制器，其中级联部分不直接连接设备，不需要进行中断号分配。

64-319 号：使用 780E 芯片组的约定如下，64-319 号分配给芯片内部的 HT 中断控制器。但由于 HT 套片上的 8259 控制器通过 HT 的 0-15 位中断向量连接 CPU，64-79 号中断实际作为对应 8259 的 16 个中断源，不对应中断设备。

3 地址空间分布表

使用这种模式，为了兼容之前的 32 位的固件代码，需要配置一级交叉开关，把 44 位的访问 HT 空间的地址转换成 32 位地址，芯片的地址空间如下表所示：

	起始地址	结束地址	说明
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	内存控制器 0
地址 1	0x0000_0000_1000_0000	0x0000_0000_17FF_FFFF	保留
地址 2	0x0000_0000_1800_0000	0x0000_0000_19FF_FFFF	HT1 IO 空间

地址 3	0x0000_0000_1A00_0000	0x0000_0000_1BFF_FFFF	HT1 配置空间
地址 4	0x0000_0000_1C00_0000	0x0000_0000_1DFF_FFFF	LPC Memory
地址 5	0x0000_0000_1FC0_0000	0x0000_0000_1FCF_FFFF	LPC Boot
地址 6	0x0000_0000_1FD0_0000	0x0000_0000_1FDF_FFFF	PCI IO 空间
地址 7	0x0000_0000_1FE0_0000	0x0000_0000_1FE0_00FF	PCI 控制器配置空间
地址 8	0x0000_0000_1FE0_0100	0x0000_0000_1FE0_01DF	IO 寄存器空间
地址 9	0x0000_0000_1FE0_01E0	0x0000_0000_1FE0_01E7	UART 0
地址 10	0x0000_0000_1FE0_01E8	0x0000_0000_1FE0_01EF	UART 1
地址 11	0x0000_0000_1FE0_01F0	0x0000_0000_1FE0_01FF	SPI
地址 12	0x0000_0000_1FE0_0200	0x0000_0000_1FE0_02FF	LPC Register
地址 13	0x0000_0000_1FE8_0000	0x0000_0000_1FE8_FFFF	PCI 配置空间
地址 14	0x0000_0000_1FF0_0000	0x0000_0000_1FF0_FFFF	LPC I/O
地址 15	0x0000_0000_4000_0000	0x0000_0000_7FFF_FFFF	HT1 MEM 空间
地址 16	0x0000_0000_8000_0000	0x0000_0000_8FFF_FFFF	保留
地址 17	0x0000_0000_9000_0000	0x0000_0001_7FFF_FFFF	以内存大小 4G 为例,其中 0x0000_0000_FE00_0000 - 0x0000_0000_FFFF_FFFF 为 780E 桥片用做 MSI 中断等地址空间, 必须保留
地址 18	0x0000_0C00_0000_0000	0x0000_0FFF_FFFF_FFFF	HT1 控制器空间
地址 19	0x0000_1000_0000_0000	0x0000_3FFF_FFFF_FFFF	保留
地址 20	其它地址		系统配置空间

龙芯 3A/B+780E 方案 DMA 地址空间关系, 以内存大小 4GB 为例, 如下表:

	内存地址	DMA 地址	关系	物理内存
地址 0	0x0000_0000_0000_0000 -0x0000_0000_0FFF_FFFF	0x0000_0000_8000_000 -0x0000_0000_8FFF_FFFF	内存地址 = DMA 地址 0x80000000	0-256MB
地址 1	0x0000_0000_9000_0000 -0x0000_0001_7FFF_FFFF	0x0000_0000_9000_000 -0x0000_0001_7FFF_FFFF	内存地址 = DMA 地址	256MB-4 GB

龙芯 2H 芯片组 固件开发规范

V1.0

文档更新记录	文档名	龙芯 2H 芯片组固件开发规范
	版本号	V1.0
更新历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。

目 录

1.DMA 地址映射.....	1
2. 中断.....	1
2.1 中断系统描述.....	1
2.2 中断号分配约定.....	1
2.3 中断路由约定.....	2
3. 芯片地址空间分布表.....	4

前 言

本规范是龙芯中科技术股份有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范面向固件软件开发人员，在《龙芯 CPU 开发系统固件与内核接口规范》基础上针对龙芯 2H 芯片组的固件软件开发要点进行补充说明，正文提到的 3A/3B 包括 3A/3B4000 处理器及该型号之前的 3 号处理器。。

1. DMA 地址映射

龙芯 2H 芯片组平台 DMA 地址与内存物理地址对应关系分为两种情况：

第一种，内存地址在低 256MB 时，映射关系为 1:1 映射。

第二种，内存地址大于 256MB 时，映射关系为 $damaddr = phyaddr - 0x8000_0000$ 。

2. 中断

2.1 中断系统描述

2H 芯片组有 5 组中断控制寄存器，每组中断控制寄存器控制 32 个中断源，总共可以控制 160 个中断源。3A/B+2H 使用 3A/B 的 INTn0 中断。3A/B 芯片的 INTN0 管脚与 2H 芯片的 SYS_INTN 管脚相连，将 2H 上的 160 个中断源连接到 3A/B 的 INTn0 中断。因此通过配置 3A/B 的中断路由寄存器可以把 INTn0 中断路由到 Core0 的 IP3。当 2H 的中断触发时，3A/B Core0 的 Cause 寄存器 IP3 位就会被置 1，这时查询 2H 的中断状态寄存器就可以判断出中断来源，其示意图如下。

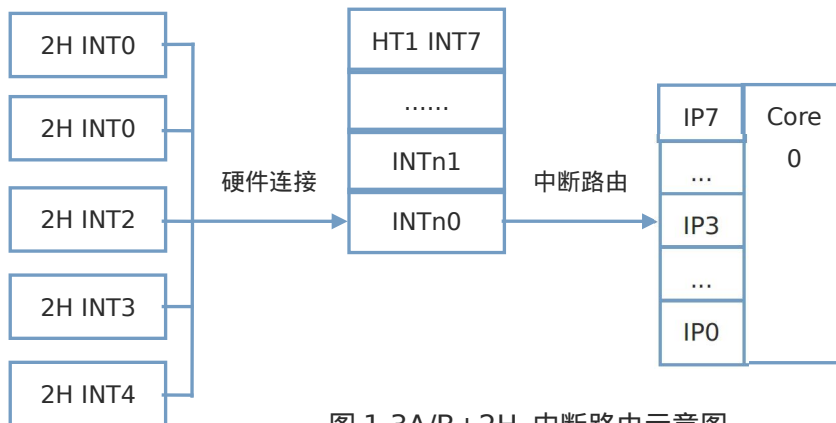


图 1 3A/B+2H 中断路由示意图

2.2 中断号分配约定

0-15 号保留下表所示的中断。剩下的中断号有些会被芯片组保留，具体的需要查询芯片组的中断控制器相关说明。

中断号	中断源	说明
0	HPET	HPET 高精度定时器
1	I8042	XT-PIC 键盘
2	级联	XT-PIC
3		
4		
5		
6		
7		
8	RTC	XT-PIC 实时时钟
9		
10		
11		
12	I8042	XT-PIC 鼠标
13		
14	ide0	XT-PIC 硬盘
15	ide1	XT-PIC 硬盘

16-47 号对应分配给芯片内部的中断控制器，其中级联部分不直接连接设备，不需要进行中断号分配。

64-224 号： 分配给 2H 芯片组。

2.3 中断路由约定

IP0-IP7 用于标识发生了哪些中断。IP0 和 IP1 是软件中断位，不提供对外的中断引脚。IP2-IP5 随着硬件输入引脚上的信号而变化，标识哪些设备发生了中断。IP6 用于处理器核间中断，负责多核处理器的通信。

IP7 一方面用于 MIPS 内部的定时器中断，另一方面用于性能计数器中断。具体描述如下：

- IP0：软件中断（Linux 内核暂未使用）
- IP1：软件中断（Linux 内核暂未使用）
- IP2：CPU 内部 LPC 总线和 UART 设备
- IP3：桥片中断
- IP4：保留（以后扩展）
- IP5：保留（以后扩展）

- IP6: 多核处理器核间中断
- IP7: 定时器和性能计数器中断

下表是根据上述约定给出的处理器中断路由配置寄存器的配置表，在此不考虑中断负载均衡，芯片内部所有中断源都路由到第一个处理器核上。

中断源	中断号	路由至	路由 entry 值	说明
Sys_int0	16/级联	0 号核 IP3	0x21	直接连接中断设备时进行分配。若连接 PCI 南桥则仅用于级联，不分配中断设备
Sys_int1	17/级联	0 号核 IP3	0x21	
Sys_int2	18/级联	0 号核 IP3	0x21	
Sys_int3	19/级联	0 号核 IP3	0x21	
Pci_int0	20	0 号核 IP5	0x81	中断号与具体槽位相关,中断号对应 INTA 的号码
Pci_int1	21	0 号核 IP5	0x81	
Pci_int2	22	0 号核 IP5	0x81	
Pci_int3	23	0 号核 IP5	0x81	
MT0	24	0 号核 IP5	0x81	
MT1	25	0 号核 IP5	0x81	
LPC/UART	26	0 号核 IP2	0x11	
DRR INT0	27	0 号核 IP5	0x81	
DRR INT1	28	0 号核 IP5	0x81	
Barrier	29	0 号核 IP5	0x81	
保留	-	0 号核 IP5	0x81	
PCI-perr& serr	31	0 号核 IP5	0x81	
HT0 INT0	级联	0 号核 IP3	0x21	0 号 HT 对应的中断位。其中连接 HT 南桥时，HT0-INT0 与 HT0-INT1 实际作为级联通过，不直接对应中断设备。
HT0 INT1	级联	0 号核 IP3	0x21	
HT0 INT2	级联	0 号核 IP3	0x21	
HT0 INT3	级联	0 号核 IP3	0x21	
HT0 INT4	级联	0 号核 IP3	0x21	
HT0 INT5	级联	0 号核 IP3	0x21	
HT0 INT6	级联	0 号核 IP3	0x21	
HT0 INT7	级联	0 号核 IP3	0x21	
HT1 INT0	级联	0 号核 IP3	0x2F	1 号 HT 对应的中断位
HT1 INT1	级联	0 号核 IP3	0x2F	
HT1 INT2	级联	0 号核 IP3	0x21	
HT1 INT3	级联	0 号核 IP3	0x21	
HT1 INT4	级联	0 号核 IP3	0x21	
HT1 INT5	级联	0 号核 IP3	0x21	

HT1 INT6	级联	0号核 IP3	0x21	
HT1 INT7	级联	0号核 IP3	0x21	

3. 芯片地址空间分布表

龙芯 3A/B+2H 方案地址空间分布如下表：

	起始地址	结束地址	说明
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	内存 0 ~ 256M
地址 1	0x0000_0000_1000_0000	0x0000_0000_17FF_FFFF	HT1 MEM, LS2H PCIE MEM
地址 2	0x0000_0000_1800_0000	0x0000_0000_18FF_FFFF	HT1 MEM, LS2H PCIE IO
地址 3	0x0000_0000_1900_0000	0x0000_0000_1AFF_FFFF	保留
地址 4	0x0000_0000_1B00_0000	0x0000_0000_1BFF_FFFF	HT1 MEM, LS2H REGISTER
地址 5	0x0000_0000_1C00_0000	0x0000_0000_1DFF_FFFF	LPC MEM
地址 6	0x0000_0000_1FC0_0000	0x0000_0000_1FCF_FFFF	LPC Boot
地址 7	0x0000_0000_1FD0_0000	0x0000_0000_1FDF_FFFF	PCI IO 空间
地址 8	0x0000_0000_1FE0_0000	0x0000_0000_1FE0_00FF	PCI 控制器配置空间
地址 9	0x0000_0000_1FE0_0100	0x0000_0000_1FE0_01DF	IO 寄存器空间
地址 10	0x0000_0000_1FE0_01E0	0x0000_0000_1FE0_01E7	UART 0
地址 11	0x0000_0000_1FE0_01E8	0x0000_0000_1FE0_01EF	UART 1
地址 12	0x0000_0000_1FE0_01F0	0x0000_0000_1FE0_01FF	SPI
地址 13	0x0000_0000_1FE0_0200	0x0000_0000_1FE0_02FF	LPC REGISTER
地址 14	0x0000_0000_1FE8_0000	0x0000_0000_1FE8_FFFF	PCI 配置空间
地址 15	0x0000_0000_1FF0_0000	0x0000_0000_1FF0_FFFF	LPC I/O
地址 16	0x0000_0000_4000_0000	0x0000_0000_7FFF_FFFF	HT1 MEM 空间
地址 17	0x0000_0C00_0000_0000	0x0000_0FFF_FFFF_FFFF	HT1 控制器空间
地址 18	0x0000_1000_0000_0000	0x0000_3FFF_FFFF_FFFF	保留
地址 19	0x0000_0000_8000_0000	0x0000_0000_FFFF_FFFF	保留
地址 20	0x0000_0000_9000_0000	0x0000_0001_7FFF_FFFF	内存 256M ~ 4G

龙芯 3A/B+2H 方案 DMA 地址空间关系，以内存大小 4GB 为例，如下表：

	内存地址	DMA 地址	关系	物理内存
地址 0	0x0000_0000_0000_0000 -0x0000_0000_0FFF_FFFF	0x0000_0000_0000_0000 -0x0000_0000_0FFF_FFFF	内存地址 = DMA 地址	0-256MB
地址 1	0x0000_0000_9000_0000 -0x0000_0001_7FFF_FFFF	0x0000_0000_1000_0000 -0x0000_0000_FFFF_FFFF	内存地址 = DMA 地址 - 0x80000000	256MB-4GB

龙芯 7A1000 芯片组
固件开发规范
V1.0

文档更新记录	文档名	龙芯 7A1000 芯片组固件开发规范
	版本号	V1.0
更新历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。

目 录

1 Audio Codec.....	1
2 LPC 控制器.....	1
3 VBIOS.....	2
4 GMAC.....	2
5 DMA 地址映射.....	3
6 FDT.....	3
7 中断.....	3
7.1 中断系统描述.....	3
7.2 中断号分配约定.....	4
8 芯片地址空间分布表.....	6

前 言

本规范是龙芯中科技术股份有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范面向固件软件开发人员，在《龙芯 CPU 开发系统固件与内核接口规范》基础上针对 7A1000 芯片组的固件软件开发要点进行补充说明，正文提到的 3A/3B 包括 3A/3B4000 处理器及该型号之前的 3 号处理器。

1 Audio Codec

固件通过 Verb Table 机制实现不同板卡中对声卡的差异化设计。固件开发人员需要根据具体主板的声卡设计正确填写 Verb Table。具体参考龙芯代码中关于 Realtek 的 Verb Table 配置实现。

注意：须确认芯片组 HDA 引脚工作在 HDA 模式（配置寄存器偏移 0x0440 寄存器的 12:11 为 x1b。芯片组配置寄存器基址需要固件下进行配置，参考代码配置寄存器基址为：0x90000e0010010000）。

2 LPC 控制器

LPC 控制器一共支持 17 个 SIRQ 中断，当支持 FDT 时，中断极性以及中断使能需要通过 FDT 传递，交由内核进行配置；当不支持 FDT 时，其中断触发极性以及中断使能需要在固件内配置。配置极性的寄存器位于 LPC 控制器的控制寄存器偏移 0x10 处。

具体寄存器定义：

位域	名称	访问	描述
16:0	SIRQ_INT_POLARITY	RW	LPC SIRQ 中断极性寄存器，每个比特位对应一个中断源，对于每个中断源， 0：低电平触发； 1：高电平触发。

SIRQ 中断使能控制寄存器位于偏移 0x0 的 31 位。

具体定义：

位域	名称	访问	描述
31	SIRQ_EN	RW	SIRQ 中断使能控制

LPC 中断使能寄存器位于偏移 0x4 的[17:0]位。

具体定义：

位域	名称	访问	描述
17:0	LPC_INT_EN	RW	LPC 中断使能,每个比特位对应一个中断源。对于每个中断源， 0:关闭中断；

			1:使能中断。
--	--	--	---------

LPC 中断清除寄存器位于偏移 0xC。

具体定义：

位域	名称	访问	描述
17:0	LPC_TIMEOUT_INT_CLEAR	WO	LPC 访问超时中断清除(写 1 清除)。比特 17 对应 LPC 访问超时中断,写 1 清除,写 0 无效。

LPC 控制器控制寄存器基地址定义位于 7A1000 用户手册第二十四章第二节,物理地址为 0x1200,0000。

使用 64bit 访问地址为 0x90000e0012000000。

3 VBIOS

7A1000 内部包含图形处理器 (以下简称 GPU), 位于 D6:F0; 内核下 GPU 需要 Video BIOS (以下简称 VBIOS) 的支撑才可以工作正常。

固件对 VBIOS 处理逻辑如下：

1. 固件从 7A1000 SPI flash 偏移 0x1000 处读取 VBIOS, 并判断是否合法;
2. 如果合法, 将 VBIOS 全部读取到内存保留区域, 并将首地址通过固件传参接口传递给内核;
3. 如果非法, 将固件内默认 VBIOS 传递给内核。

4 GMAC

7A1000 包含两个 GMAC 控制器, 固件分别从 7A1000 SPI flash 偏移 0x0 和 0x10 处读取 MAC0 和 MAC1 地址信息, 并写入相应 GMAC 的 MAC 地址寄存器中。

具体寄存器定义：

GMAC0/GMAC1 Address High Register(Offset 0x40):

位域	名称	默认值	描述
----	----	-----	----

31	MO: Always 1	0x0	保留
30:1 6	Reserved	0x0	保留
15:0	MAC Address0[47:32] MAC 地址高 16 位	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

GMAC0/GMAC1 Address Low Register(Offset 0x44):

位域	名称	默认值	描述
31:0	MAC Address0[31:0] MAC 地址低 32 位	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

5 DMA 地址映射

对于不支持超过 44 位 DMA 地址的设备，固件需要调整 7A1000 的 DMA 路由配置以支持此类设备。参考 7A1000 用户手册第四章第一节，DMA 路由配置寄存器（0x041C）的[12:8]位，具体实现参考龙芯代码。

寄存器定义：

位域	名称	访问	描述
12:8	dma_node_id_offset	RW	DMA 访问中节点号所在的地址偏移（相对于 bit36）

6 FDT

FDT 表用作描述设备差异，选填；如果 system_loongson 结构内 vers 成员值 ≥ 2 时，说明固件需要传递 FDT 信息给内核，这时候需要填充好 of_dtb_addr，否则内核使用缺省 FDT。参考《龙芯 CPU 开发系统固件与内核接口详细规范》附录 A.9。

7 中断

7.1 中断系统描述

7A1000 芯片组有 1 组中断控制寄存器，控制 64 个中断源。3A/B+7A1000 方案，芯片组的中断则是通过 HT1 控制器进入到 Int Controller 的 HT1-2 和 HT1-3 脚，所有经过 7A1000 中断控制器的外设中断都是

路由到 0 号核（即 CPU0）的 IP3，而 7A1000 芯片组上外接的 PCIe 设备默认采用了 MSI 中断，可以绕过 7A1000 的中断控制器直接发送到 HT1 的中断向量 HT1-0 和 HT1-1 上，并最终通过轮转机制发送到 CPU0-CPU3 上。当 7A1000 芯片组的中断触发时，3A/B Core0 的 Cause 寄存器的 IP3 被置位，这时查询 HT 的中断状态寄存器就可以判断中断来源。见图 1 所示。

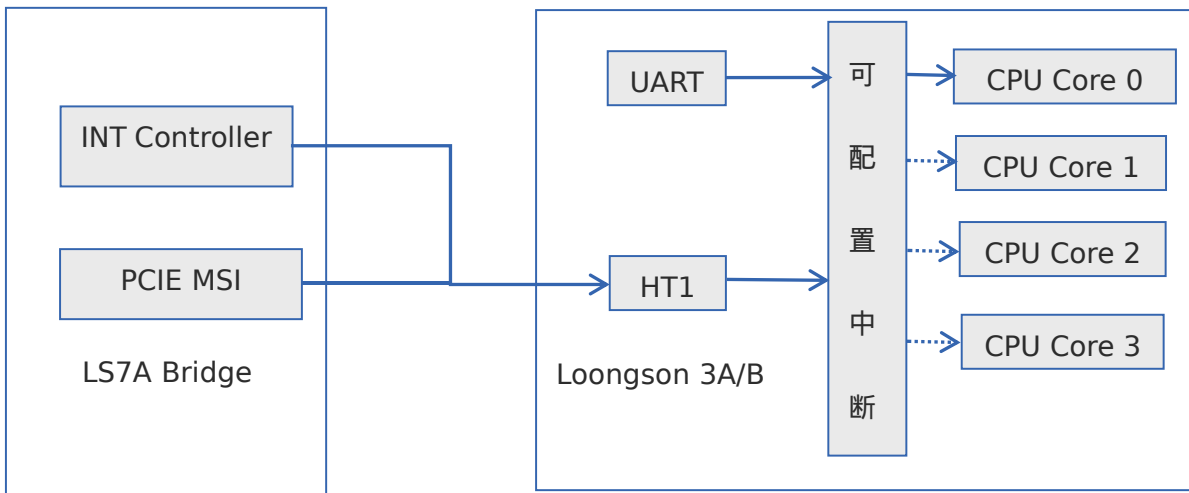


图 1 3A/B+7A1000 中断路由示意图

7.2 中断号分配约定

0-15 号保留下表所示的中断。剩下的中断号有些会被芯片组保留，具体的需要查询芯片组的中断控制器相关说明。

中断号	中断源	说明
0	HPET	HPET 高精度定时器
1	I8042	XT-PIC 键盘
2	级联	XT-PIC
3		
4		
5		
6		
7		
8	RTC	XT-PIC 实时时钟
9		
10		
11		

12	I8042	XT-PIC 鼠标
13		
14	ide0	XT-PIC 硬盘
15	Ide1	XT-PIC 硬盘

56-63 号中断保留给 CPU 内部中断源，如下表：

中断号	中断源	说明
56		
57		
58	uart	cpu 中的 uart 中断
59		
60		
61		
62		
63	timer	timer 中断

64-127 号中断，分配给芯片组。7A1000 集成或外接的 PCI 兼容设备分配的中断号如下：

中断号	中断源	中断号	中断源
64		96	pcie_f0_0
65		97	pcie_f0_1
66		98	pcie_f0_2
67		99	pcie_f0_3
68		100	pcie_f1_0
69		101	pcie_f1_1
70		102	pcie_h_lo
71		103	pcie_h_hi
72		104	pcie_g0_ho
73		105	pcie_g0_li
74		106	pcie_g1_lo
75		107	pcie_g1_hi
76	gmac0_sbd	108	
77	gmac0_pmt	109	
78	gmac1_sbd	110	
79	gmac1_pmt	111	
80	sata[0]	112	usb_0_ehci
81	sata[1]	113	usb_0_ohci
82	sata[2]	114	usb_1_ehci
83		115	usb_1_ohci
84		116	

85		117	
86		118	
87		119	
88		120	
89		121	
90		122	ac97/hda
91		123	SCI
92	dc	124	
93	gpu	125	
94		126	
95		127	

128-319 号中断保留给芯片组上的 PCIe 设备发送 MSI 中断。

注意：根据 7A1000 硬件规范，若使用 EC 时需要使用 SCI 中断，此中断需要硬件连接到 GPIO50，对应中断源为 59 号。遵循《统一系统架构规范》中断号规则，7A1000 中断源从 64 开始编码，SCI 中断编号为 123 (59+64)。

8 芯片地址空间分布表

龙芯 3A/B+7A1000 方案地址空间分布如下表：

	起始地址	结束地址	说明
地址 0	0x0000_0000_0000_0000	0x0000_0000_0FFF_FFFF	内存控制器 0
地址 1	0x0000_0000_1000_0000	0x0000_0000_17FF_FFFF	保留
地址 2	0x0000_0000_1800_0000	0x0000_0000_1801_FFFF	32 位模式下 7A LPC 的 IO 空间
地址 3	0x0000_0000_1802_0000	0x0000_0000_19FF_FFFF	32 位模式下 7A PCI 的 IO 空间
地址 4	0x0000_0000_1A00_0000	0x0000_0000_1BFF_FFFF	32 位模式下 7A PCI 的配置空间
地址 5	0x0000_0000_1C00_0000	0x0000_0000_1DFF_FFFF	LPC Memory
地址 6	0x0000_0000_1FC0_0000	0x0000_0000_1FCF_FFFF	LPC Boot
地址 7	0x0000_0000_1FD0_0000	0x0000_0000_1FDF_FFFF	PCI IO 空间
地址 8	0x0000_0000_1FE0_0000	0x0000_0000_1FE0_00FF	PCI 控制器配置空间
地址 9	0x0000_0000_1FE0_0100	0x0000_0000_1FE0_01DF	IO 寄存器空间
地址 10	0x0000_0000_1FE0_01E0	0x0000_0000_1FE0_01E7	UART 0
地址 11	0x0000_0000_1FE0_01E8	0x0000_0000_1FE0_01EF	UART 1
地址 12	0x0000_0000_1FE0_01F0	0x0000_0000_1FE0_01FF	SPI
地址 13	0x0000_0000_1FE0_0200	0x0000_0000_1FE0_02FF	LPC Register
地址 14	0x0000_0000_1FE8_0000	0x0000_0000_1FE8_FFFF	PCI 配置空间
地址 15	0x0000_0000_1FF0_0000	0x0000_0000_1FF0_FFFF	LPC I/O
地址 16	0x0000_0000_4000_0000	0x0000_0000_7FFF_FFFF	HT1 MEM 空间

地址 17	0x0000_0000_8000_0000	0x0000_0000_8FFF_FFFF	保留
地址 18	0x0000_0000_9000_0000	0x0000_0001_5FFF_FFFF	以内存大小 4G 为例
地址 19	0x0000_0001_6000_0000	0x0000_0001_7FFF_FFFF	7A GPU 访问空间 (以内存大小 4G 为例) 7A GPU 只支持低 2G 内存直接访问, 但此区间内存 (0 ~ 0x10000000) 无法满足应用需求, 需单独开辟一段地址供其使用; 现约定 0x20000000 ~ 0x40000000-1 地址固定为 7A GPU 发出地址, 通过 HT 转换映射内存的高 512M
地址 20	0x0000_0E00_1000_0000	0x0000_0E00_1000_0FFF	7A 中断控制器空间
地址 21	0x0000_0E00_1000_1000	0x0000_0E00_1000_1FFF	7A 中 HPET 寄存器空间
地址 22	0x0000_0E00_1000_2000	0x0000_0E00_1000_2FFF	7A 中 LPC 控制寄存器空间
地址 23	0x0000_0E00_1001_0000	0x0000_0E00_1001_FFFF	7A 中 confbus 空间
地址 24	0x0000_0E00_1008_0000	0x0000_0E00_100F_FFFF	7A 中 misc 设备寄存器空间
地址 25	0x0000_0E00_1200_0000	0x0000_0E00_13FF_FFFF	7A 中 LPC MEM 空间
地址 26	0x0000_0E00_4000_0000	0x0000_0E00_7FFF_FFFF	7A 中 PCI MEM 空间
地址 27	0x0000_0EFD_FC00_0000	0x0000_0EFD_FC01_FFFF	64 位模式下 7A LPC 的 IO 空间
地址 28	0x0000_0EFD_FC02_0000	0x0000_0EFD_FDFF_FFFF	64 位模式下 7A PCI 的 IO 空间
地址 29	0x0000_0EFD_FE00_0000	0x0000_0EFD_FF00_0000	64 位模式下 7A PCI 的配置访问空间
地址 30	0x0000_0C00_0000_0000	0x0000_0FFF_FFFF_FFFF	HT1 控制器空间
地址 31	0x0000_1000_0000_0000	0x0000_3FFF_FFFF_FFFF	保留
地址 32	其它地址		系统配置空间

龙芯 3A/B+7A1000 方案 DMA 地址空间关系, 以内存大小 4GB 为例, 如下表:

	内存地址	DMA 地址	关系	物理内存
地址 0	0x0000_0000_0000_0000 -0x0000_0000_0FFF_FFFF	0x0000_0000_0000_0000 -0x0000_0000_0FFF_FFFF	内存地址=DMA 地址	0-256MB
地址 1	0x0000_0000_9000_0000 -0x0000_0001_7FFF_FFFF	0x0000_0000_9000_0000 -0x0000_0001_7FFF_FFFF	内存地址=DMA 地址	256MB-4GB

龙芯 CPU 内核开发规范

V1.0

文档更新记录	文档名	龙芯 CPU 内核开发规范
	版本号	V1.0
更新历史		
序号	版本号	更新内容
1	V1.0	发布文档初始版本 V1.0 版。

目 录

一、适用范围.....	1
二、中断系统设计规范.....	1
1 中断路由.....	1
1.1 3A-7A1000 开发系统中断路由.....	1
1.2 3A-RS780/5690 开发系统中断路由.....	2
2 中断号分配.....	4
2.1 3A-7A1000 开发系统中断号分配.....	4
2.2 3A-RS780/5690 开发系统中断号分配.....	5
3 LPC 中断.....	6
3.1 3A-7A1000 开发系统 LPC 中断.....	6
3.2 3A-RS780/5690 开发系统 LPC 中断.....	7
3.3 休眠唤醒处理.....	7
三、DMA 配置规范.....	7
1 3A-7A1000 开发系统 DMA 配置.....	7
2 3A-RS780/5690 开发系统 DMA 配置.....	8
四、外设设计规范.....	9
1 串口设计规范.....	9
1.1 3A-7A1000 开发系统串口.....	9
1.2 3A-RS780/5690 开发系统串口.....	9
2 集成网卡设计规范.....	9
2.1 3A-7A1000 开发系统集成网卡 MAC 地址获取.....	9

3 I2C 设计规范.....	9
3.1 3A7A1000 开发系统 I2C 设计规范.....	9
3.2 3A-RS780/5690 开发系统 I2C 设计规范.....	10
4 RTC 设计规范.....	10
4.1 3A7A1000 开发系统 RTC 设计规范.....	10
4.2 3A-RS780/5690 开发系统 RTC 设计规范.....	10

一、适用范围

本文档是对龙芯平台 Linux 内核的固件接口、中断系统配置、DMA 管理、外设支持等的设计规范，适用于龙芯 3 号系列 CPU (3A/3B2000、3A/3B3000、3A/3B4000) ，配套桥片包括 7A (7A1000) 、RS780/5690。

二、中断系统设计规范

1 中断路由

1.1 3A-7A1000 开发系统中断路由

CPU 核 Cause 寄存器的 IP0-IP7 用于标识发生了哪些中断。IP0 和 IP1 是软件中断位，不提供对外的中断引脚。IP2-IP5 随着硬件输入引脚上的信号而变化，标识哪些设备发生了中断。IP6 用于处理器核间中断，负责多核处理器的通信。IP7 一方面用于 MIPS 内部的定时器中断，另一方面用于性能计数器中断。具体描述如下：

- IP0：软件中断（Linux 内核暂未使用）
- IP1：软件中断（Linux 内核暂未使用）
- IP2：CPU 内部 LPC 总线和 UART 设备
- IP3：桥片设备中断
- IP4：保留（以后扩展）
- IP5：保留（以后扩展）
- IP6：多核处理器核间中断
- IP7：定时器和性能计数器中断

下表是根据上述约定给出的处理器中断路由配置寄存器的配置表，在此不考虑中断负载均衡，芯片内部所有中断源都路由到第一个处理器核上。

中断源	中断号	路由至	路由 entry 值	说明
Sys_int0	16/级联	0 号核 IP3	0x21	
Sys_int1	17/级联	0 号核 IP3	0x21	
Sys_int2	18/级联	0 号核 IP3	0x21	
Sys_int3	19/级联	0 号核 IP3	0x21	
Pci_int0	20	0 号核 IP5	0x81	中断号与具体槽位相关，中断号对应 INTA 的号码
Pci_int1	21	0 号核 IP5	0x81	
Pci_int2	22	0 号核 IP5	0x81	
Pci_int3	23	0 号核 IP5	0x81	
MT0	24	0 号核 IP5	0x81	

MT1	25	0号核 IP5	0x81	
LPC/UART	26	0号核 IP2	0x11	
DRR INTO	27	0号核 IP5	0x81	
DRR INT1	28	0号核 IP5	0x81	
Barrier	29	0号核 IP5	0x81	
保留	-	0号核 IP5	0x81	
PCI-perr&serr	31	0号核 IP5	0x81	
HT0 INTO	级联	0号核 IP3	0x21	0号 HT 对应的中断位。其中连接 HT 南桥时，HT0-INT0 与 HT0-INT1 实际作为级联通过，不直接对应中断设备。
HT0 INT1	级联	0号核 IP3	0x21	
HT0 INT2	级联	0号核 IP3	0x21	
HT0 INT3	级联	0号核 IP3	0x21	
HT0 INT4	级联	0号核 IP3	0x21	
HT0 INT5	级联	0号核 IP3	0x21	
HT0 INT6	级联	0号核 IP3	0x21	
HT0 INT7	级联	0号核 IP3	0x21	
HT1 INTO	级联	0号核 IP3	0x2F	1号 HT 对应的中断位
HT1 INT1	级联	0号核 IP3	0x2F	
HT1 INT2	级联	0号核 IP3	0x21	
HT1 INT3	级联	0号核 IP3	0x21	
HT1 INT4	级联	0号核 IP3	0x21	
HT1 INT5	级联	0号核 IP3	0x21	
HT1 INT6	级联	0号核 IP3	0x21	
HT1 INT7	级联	0号核 IP3	0x21	

1.2 3A-RS780/5690 开发系统中断路由

CPU 核 Cause 寄存器的 IP0-IP7 用于标识发生了哪些中断。IP0 和 IP1 是软件中断位，不提供对外的中断引脚。IP2-IP5 随着硬件输入引脚上的信号而变化，标识哪些设备发生了中断。IP6 用于处理器核间中断，负责多核处理器的通信。IP7 一方面用于 MIPS 内部的定时器中断，另一方面用于性能计数器中断。具体描述如下：

- IP0：软件中断（Linux 内核暂未使用）
- IP1：软件中断（Linux 内核暂未使用）
- IP2：CPU 内部 LPC 总线和 UART 设备
- IP3：南桥设备中断
- IP4：传统 Bonito 南桥中断

- IP5: 保留 (以后扩展)
- IP6: 多核处理器核间中断
- IP7: 定时器和性能计数器中断

下表是根据上述约定给出的处理器中断路由配置寄存器的配置表, 在此不考虑中断负载均衡, 芯片内部所有中断源都路由到第一个处理器核上。

中断源	中断号	路由至	路由 entry 值	说明
Sys_int0	16/级联	0号核 IP3	0x21	直接连接中断设备时进行分配。若连接 PCI 南桥则仅用于级联, 不分配中断设备
Sys_int1	17/级联	0号核 IP3	0x21	
Sys_int2	18/级联	0号核 IP3	0x21	
Sys_int3	19/级联	0号核 IP3	0x21	
Pci_int0	20	0号核 IP5	0x81	中断号与具体槽位相关, 中断号对应 INTA 的号码
Pci_int1	21	0号核 IP5	0x81	
Pci_int2	22	0号核 IP5	0x81	
Pci_int3	23	0号核 IP5	0x81	
MT0	24	0号核 IP5	0x81	
MT1	25	0号核 IP5	0x81	
LPC/UART	26	0号核 IP2	0x11	
DRR INT0	27	0号核 IP5	0x81	
DRR INT1	28	0号核 IP5	0x81	
Barrier	29	0号核 IP5	0x81	
保留	-	0号核 IP5	0x81	
PCI-perr&serr	31	0号核 IP5	0x81	
HT0 INT0	级联	0号核 IP3	0x21	0号 HT 对应的中断位。其中连接 HT 南桥时, HT0-INT0 与 HT0-INT1 实际作为级联通过, 不直接对应中断设备。
HT0 INT1	级联	0号核 IP3	0x21	
HT0 INT2	级联	0号核 IP3	0x21	
HT0 INT3	级联	0号核 IP3	0x21	
HT0 INT4	级联	0号核 IP3	0x21	
HT0 INT5	级联	0号核 IP3	0x21	
HT0 INT6	级联	0号核 IP3	0x21	
HT0 INT7	级联	0号核 IP3	0x21	
HT1 INT0	级联	0号核 IP3	0x2F	1号 HT 对应的中断位
HT1 INT1	级联	0号核 IP3	0x2F	

HT1 INT2	级联	0号核 IP3	0x21	
HT1 INT3	级联	0号核 IP3	0x21	
HT1 INT4	级联	0号核 IP3	0x21	
HT1 INT5	级联	0号核 IP3	0x21	
HT1 INT6	级联	0号核 IP3	0x21	
HT1 INT7	级联	0号核 IP3	0x21	

2 中断号分配

内核根据平台的不同分配的中断号也有所不同，应遵循如下规则：

2.1 3A-7A1000 开发系统中断号分配

3A-7A1000 平台中断号分配规则如下：

- (1) 0~15 号分配给 LPC 设备中断；
- (2) 16~63 号分配给 CPU 内部中断控制器；
- (3) 64 ~ 127 号分配给桥片上的设备中断；
- (4) 128~319 号分配给 PCIE-MSI/MSI-X 中断；

分配情况如下表所示：

中断号	中断源	备注
0	HPET	HPET 高精度定时器
1	I8042	XT-PIC 键盘
2	级联	XT-PIC
3		
4		
5		
6		
7		
8	RTC	XT-PIC 实时时钟
9		
10		
11		
12	I8042	XT-PIC 鼠标
13		

14	Ide0	XT-PIC 硬盘
15	Ide1	XT-PIC 硬盘
16 ~ 47		cpu 中断控制器
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58	uart	cpu 中的 uart 中断
59		
60		
61		
62		
63	timer	timer 中断
64 ~ 127	7A1000 桥片	7A1000 中断控制器
128~191	PCIE-MSI	3A3000/3A4000 PCIE-MSI/MSI-X 中断
192~319	PCIE-MSI	3A4000 PCIE-MSI/MSI-X 中断

注意：

(1) 桥片上的 PCI 设备中断号由固件分配，写入 PCI 配置空间的 0x3C 寄存器。内核读取此寄存器值作为对应设备的中断号。

(2) 64 ~ 127 号中断应依据 7A1000 手册中断控制器章节的规定进行配置。

2.2 3A-RS780/5690 开发系统中断号分配

3A-780/5690 平台中断号分配规则如下：

(1) 0~15 号保留分配给连接诸如 8259 的传统设备的中断或者 CPU 外接 LPC 设备中断；

(2) 16~63 号分配给 CPU 内部中断控制器；

(3) 64 ~ 319 号中断保留；

分配情况如下表所示：

中断号	中断源	备注
0	HPET	HPET 高精度定时器

1	I8042	XT-PIC 键盘
2	级联	XT-PIC
3		
4		
5		
6		
7	SCI	系统控制中断（用于笔记本电脑 Fn 功能键）
8	RTC	XT-PIC 实时时钟
9		
10		
11		
12	I8042	XT-PIC 鼠标
13		
14	Ide0	XT-PIC 硬盘
15	Ide1	XT-PIC 硬盘
16 ~ 47		cpu 中断控制器
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58	uart	cpu 中的 uart 中断
59		
60		
61		
62		
63	timer	timer 中断
64 ~ 319		保留

注意：

(1) 桥片上的 PCI 设备中断号由固件分配，写入 PCI 配置空间的 0x3C 寄存器。内核读取此寄存器值作为对应设备的中断号。

(2) 3A-780/5690 系统不支持 MSI/MSI-X 中断。

3 LPC 中断

内核使用 0 ~ 15 作为 LPC 设备中断。

3.1 3A-7A1000 开发系统 LPC 中断

当适配桥片为 7A1000 时，LPC 设备必须连接到 7A1000 的 LPC 控制器上。LPC 中断通过 7A1000 中断控制器路由到 CPU 上。

3.1.1 通过 FDT 配置 LPC 中断电平极性

内核支持通过 FDT 配置 LPC 的中断电平极性：支持高电平和低电平触发。内核解析 DTB 传递的 LPC 中断电平极性并进行相应的配置。如果 DTB 中支持 LPC 设备，但没有配置 LPC 的中断电平极性，那么内核默认配置成高电平触发。

3.1.2 内核默认配置方式

如果固件不支持 FDT，内核不对中断触发电平（高/低电平）进行配置，交由固件根据外接不同的 LPC 设备配置成高电平或低电平有效并设置 LPC 控制器中断使能。

3.2 3A-RS780/5690 开发系统 LPC 中断

内核支持从桥片或者 CPU 的 LPC 控制器连接 LPC 设备。由于桥片和 CPU 的 LPC 设备共享 0~15 号中断，为了避免冲突，内核只支持从桥片和 CPU 的 LPC 控制器选择一个连接设备，推荐连接到桥片的 LPC 控制器上。

3.3 休眠唤醒处理

系统执行休眠唤醒操作时，LPC 控制器的状态保存和恢复需要在内核中执行。

三、DMA 配置规范

1 3A-7A1000 开发系统 DMA 配置

内核中 DMA 地址和物理地址一般情况下采用 1:1 的转换关系，考虑到有些设备不支持超过 44 位的 DMA 地址，固件需要调整 7A1000 的 DMA 路由配置以适应此类设备。内核需要读取“HT 时钟使能与 DMA 路由配置”寄存器，获取 dma_node_id_offse(bit12~bit8)位的值。根据此值计算节点号存放的位置。示例代码如下：

```
static dma_addr_t loongson_ls7a_phys_to_dma(struct device *dev, phys_addr_t paddr)
{
```

```
#ifdef CONFIG_PHYS48_TO_HT40
    long nid;
    /* We extract 2bit node id (bit 44~47, only bit 44~45 used now) from
     * Loongson3's 48bit address space and embed it into 40bit */
    nid = (paddr >> 44) & 0x3;
    paddr = ((nid << 44) ^ paddr) | (nid << (36 + node_id_offset));
#endif

return paddr;
}

static phys_addr_t loongson_ls7a_dma_to_phys(struct device *dev, dma_addr_t daddr)
{
#ifdef CONFIG_PHYS48_TO_HT40
    long nid;
    nid = (daddr >> (36 + node_id_offset)) & 0x3;
    daddr = ((nid << (36 + node_id_offset)) ^ daddr) | (nid << 44);
#endif

return daddr;
}

node_id_offset = (*(volatile u32 *)NODE_ID_OFFSET_ADDR >> 8) & 0x1F;
NODE_ID_OFFSET_ADDR 为“HT 时钟使能与 DMA 路由配置”寄存器地址。
```

2 3A-RS780/5690 开发系统 DMA 配置

因为 HT 总线最高支持 40 位地址，所以对桥片上的设备进行 DMA 相关操作时需要将节点号保存到 37 位 (phy_to_dma) 或者从 37 位恢复到 44 位(dma_to_phys)。示例代码如下：

```
static dma_addr_t loongson_rs780_phys_to_dma(struct device *dev, phys_addr_t paddr)
{
    long nid;
    dma_addr_t daddr;

    daddr = (paddr < 0x10000000) ?
            (paddr | 0x0000000080000000) : paddr;
#ifdef CONFIG_PHYS48_TO_HT40
    /* We extract 2bit node id (bit 44~47, only bit 44~45 used now) from
     * Loongson3's 48bit address space and embed it into 40bit */
    nid = (paddr >> 44) & 0x3;
    daddr = ((nid << 44) ^ daddr) | (nid << 37);
#endif

return daddr;
}
```

```
}  
  
static phys_addr_t loongson_rs780_dma_to_phys(struct device *dev, dma_addr_t daddr)  
{  
    long nid;  
  
    daddr = (daddr < 0x90000000 && daddr >= 0x80000000) ?  
            (daddr & 0x0fffffff) : daddr;  
#ifdef CONFIG_PHYS48_TO_HT40  
    nid = (daddr >> 37) & 0x3;  
    daddr = ((nid << 37) ^ daddr) | (nid << 44);  
#endif  
    return daddr;  
}
```

四、外设设计规范

1 串口设计规范

调试串口使用 CPU 集成的 UART0，默认时钟为 33MHz。如果外接其他频率时钟，当固件支持的《龙芯 CPU 开发系统固件与内核接口详细规范》版本小于 3.0 时，内核解析固件传递的“uart_device”参数并根据“uartclk”成员的值配置串口时钟频率。

1.1 3A-7A1000 开发系统串口

当适配桥片为 7A1000 时，除了 CPU 的 UART0 作为调试串口外，内核默认支持 7A1000 桥片上集成的 UART0 作为全功能串口。

1.2 3A-RS780/5690 开发系统串口

只支持 CPU 的 UART0 作为调试串口，不支持桥片上的串口。

2 集成网卡设计规范

2.1 3A-7A1000 开发系统集成网卡 MAC 地址获取

内核运行前，固件将 MAC 地址存储在“MAC Address0 High Register”和“MAC Address0 Low

Register” 寄存器中。内核驱动从这两个寄存器中的值作为 MAC 地址，如果值非法，则从 7A1000 外接的 SPIFLASH 读取。如果 SPIFLASH 中存储 MAC 地址也为非法值，则使用软件生成的随机值作为 MAC 地址。

3 I2C 设计规范

3.1 3A7A1000 开发系统 I2C 设计规范

内核提供对 7A1000 集成的 I2C0 和 I2C1 控制器的支持。

7A1000 显示控制器集成了 DVO0/1_SDA 和 DVO0/1_SCL 两组专用 GPIO，内核使用这两组 GPIO 模拟 I2C，用于 7A1000 内置显卡驱动获取显示器” EDID” 信息。

3.2 3A-RS780/5690 开发系统 I2C 设计规范

内核不对桥片上的 I2C 控制器提供支持。

4 RTC 设计规范

4.1 3A7A1000 开发系统 RTC 设计规范

当桥片为 7A1000 时，内核只支持 7A1000 集成 RTC 控制器，不支持外接 RTC 控制器。

4.2 3A-RS780/5690 开发系统 RTC 设计规范

3A-RS780/5690 支持外接 MC146818 兼容的 RTC 控制器，内核默认使用外部 RTC。

龙芯 7A1000 内置显卡软硬件设计规范

V0.2

文档更新记录	文档名	龙芯 7A1000 内置显卡软硬件接口规范
	版本号	V0.2
更新历史		
序号	版本号	更新内容
1	V0.1	发布文档初始版本 V0.1 版。
2	V0.2	<ol style="list-style-type: none">1. 调整编码器链接方案2. 增加关于热插拔方案的说明

目录

1. 范围.....	1
2. 术语与定义.....	1
3. 架构关系.....	2
4. 显示方案.....	2
4.1 显示模式.....	2
4.2 DDC 通道硬件连接方案.....	3
4.3 EDID 获取方案.....	4
4.4 热插拔探测方案.....	4
4.2.1 轮询.....	5
4.2.2 中断.....	5
4.5 背光控制.....	6
5. VBIOS 存储约定.....	6

1. 范围

本规范规定龙芯内置显卡的硬件设计要求，VBIOS 固件的存放位置等。本规范适用于龙芯 7A1000 系列桥片主板。建议其它系统厂商遵循此规范开发相关产品。

2. 术语与定义

本规范所用术语定义如下：

- a. Firmware：固件，写入 ROM、EEPROM 等非易失存储器中的程序，负责控制和协调集成电路。
- b. BIOS：基本输入输出系统，Basic Input Output System，一组固化到主板上一个 ROM 芯片上的程序，它保存着计算机基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。BIOS 与硬件系统集成在一起，也被称为固件，本规范中固件和 BIOS 不做区分。
- c. UEFI：统一的可扩展固定接口，Unified Extensible Firmware Interface，是 Intel 为全新类型的 PC 固件的体系结构、接口和服务提出的建议标准。主要目的是提供在 OS 加载之前在所有平台上一致、正确指定的启动服务，被看做是有近 20 多年历史的 PC BIOS 的继任者。
- d. PMON：MIPS 架构机器上使用的一种兼有 BIOS 和 boot loader 部分功能的开放源码软件。
- e. VBIOS(Video BIOS)：VBIOS 是显卡的 BIOS，VBIOS 提供一些和显示相关的功能，并存放显示芯片与驱动程序之间的控制程序，另外还存放有显示卡型号、规格、生产厂家、出厂时间等信息。
- f. PCI (Peripheral Component Interconnect)：是连接电子计算机主板和外部设备的总线标准，用于定义局部总线的标准。此标准允许在计算机内安装多达 10 个遵从 PCI 标准的扩展卡。
- g. Encoder: 型号编码器，用于将 DVO 型号转换为其他的显示信号，如模拟信号、LVTM 或 TMDS 等显示信号。
- h. Crtc: 显示控制器，显示控制器把数据从显存中将要显示数据取出来,然后对这些数据做一定的处理送到显示屏。
- i. Connector: 显示接口连接器，指显示接口硬件，例如 VGA 连接器，HDMI 连接器等。
- j. GPU：图形处理器。

3. 架构关系

龙芯 7A1000 桥片中集成显示控制器(DC)和图形处理器 (GPU)，龙芯 7A1000 显示驱动包括内核驱动和 VBIOS 固件，由内核驱动和 VBIOS 固件配合控制龙芯显示控制器和图形处理器 (GPU)，实现显示，渲染，背光调节等功能。VBIOS 固件需要使用 VBIOS 生成工具生成，烧写到固定的位置(详见第 5 章)，VBIOS 生成工具的使用详见《龙芯 VBIOS 工具用户手册》，龙芯 7A1000 内置显卡内核驱动初始化设备时，从 VBIOS 固件中解析硬件信息、配置、代码等内容。龙芯 7A1000 内置显卡内核驱动与 VBIOS 固件层次关系如图 1 所示：

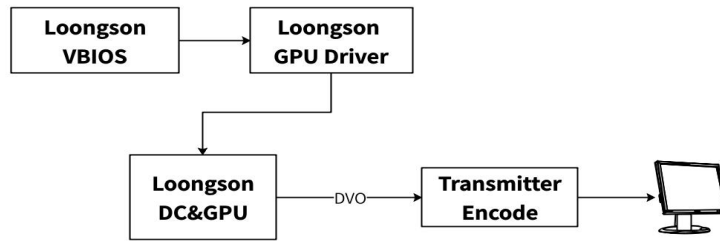


图 1 显示驱动，固件与显示硬件的关系

4. 显示方案

龙芯 7A1000 桥片中集成的显示控制器包含两路 DVO 端口，支持 DVO 信号输出，支持模拟信号或数字信号输出时需要连接额外的编码芯片来实现。龙芯 7A1000 显示控制器支持多种分辨率，每路显示最大支持 1920x1080@60Hz，支持 RGB565, RGB888 两种色深，龙芯 7A1000 桥片内置显卡支持的显示模式，如表 1 所示。

显示类型	描述
CRT	通过 DVO 端口连接额外的转换芯片支持模拟信号显示器(VGA)。
显示面板	通过 DVO 端口连接额外的转换芯片支持 TMDS(DVI,HDMI)和 LVDS 兼容显示面板。

表 1 显示信号的支持类型

4.1 显示模式

龙芯 7A1000 显示驱动支持单路显示或者两路同时显示，其中两路显示时支持几种不同的模式，支持的情况如表 2 所示。

显示配置模式	描述	支持情况
Single	支持单独一个显示器	支持
Twin	支持两个显示器，以相同分辨率和时序，显示相同画面	支持
Clone	支持两个显示器，以不同分辨率和时序，显示相同的画面	不支持

Extended	支持两个显示器，以左右或上下扩展的形式显示完整画面	支持
----------	---------------------------	----

表 2 多屏支持情况

龙芯 7A1000 显示驱动支持多种硬件链接方案，增加主板设计时的灵活性。硬件差异记录在 VBIOS 中，由内核态驱动解析。

方案 A：单路显示，选择两路显示控制器中的一路连接视频编解码转换芯片，如图 2。

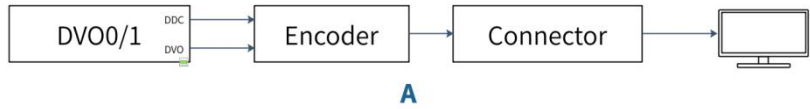


图 2 单屏链接方案图

方案 B：两路独立的双屏显示，两路显示控制器分别连接视频编解码转换芯片。如图 3，实现多屏扩展(Extended)和多屏镜像(Twin)功能。支持单屏-多屏切换，多屏-单屏切换和多屏不同模式间的切换等功能，支持显示控制器和视频编解码转换芯片之间交叉连接。注意 DDC 通道和 DVO 通道需要同时交叉。通过 VBIOS 生成工具生成符合硬件连线的配置。

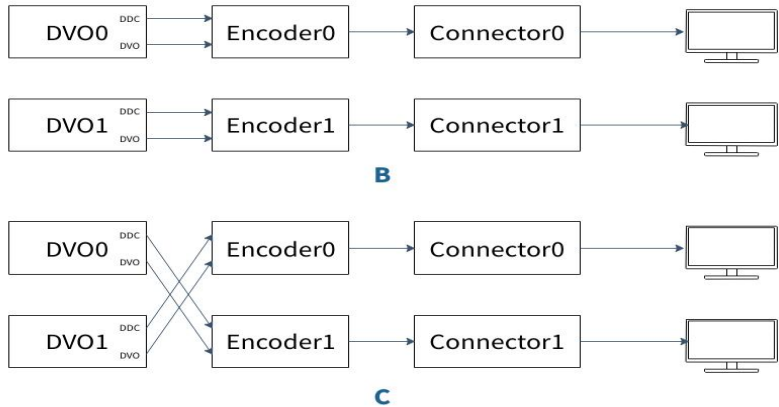


图 3 双屏显示硬件连接方案图

4.2 DDC 通道硬件连接方案

龙芯 7A1000 桥片对每个显示通路的 DDC 连接方案做了具体规定。龙芯 7A1000 内置显卡推荐使用二种 DDC 通道连接方案。

方案 A：主板设计时采用不需要软件配置的编码芯片，例如 VGA 编码芯片，将 7A1000 桥片上 DDC 通道的 DVO_SCL 和 DVO_SDA 经过电平转换后与连接器直连，硬件连接方案如图 4 方案 A 所示，该方案的热插拔探测只能采用轮询的方式。

方案 B：主板设计时采用需要软件配置的编码芯片，且编码芯片支持 EDID 的读取功能和热插拔探测中断，则需要将 DDC 通道的 DVO_SCL 和 DVO_SDA 作为编码器芯片的配置通道，通过向编码器芯片发送命令来读取 EDID 信息，DDC 通道只与编码器芯片连接，如图 4 方案 B 所示，该方案的热插拔探测只能采用中断的方式。如果采用需要软件配置的编码器芯片，我们优先推荐方案 B。方案 B 的连接方式更加合理，但是需要为编码器开发驱动，有额外的工作量。

方案 C：主板设计时采用需要软件配置的编码芯片，将 DDC 通道的 DVO_SCL 和

DVO_SDA 同时连接到编码器芯片和连接器，如图 4 方案 C 所示，该方案的热插拔探测采用轮询方式，如果新设计的主板，我们推荐使用方案 B。

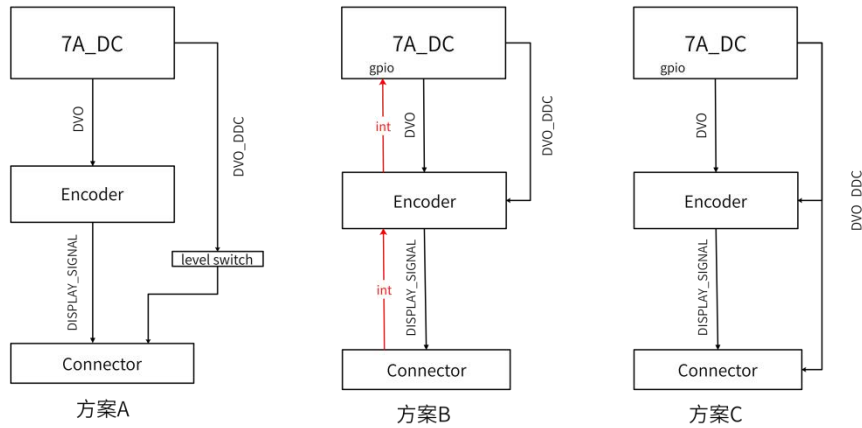


图 4 DDC 通道连接图

4.3 EDID 获取方案

龙芯 7A1000 内置显卡提供三种 EDID 获取方案，可以根据具实际的需求任意选择。

方案 A：通过 DDC 通道读取外部 EDID 信息。

方案 B：通过通道向编码器芯片发送命令读取 EDID 信息。

方案 C：在 VBIOS 中预存由显示屏厂商提供的 EDID 信息，无需外部设备获取，该方案适用于以下两种情况：

- 1、外部显示设备无法提供 EDID 信息；
- 2、无法采用标准的获取方式获取 EDID。

4.4 热插拔探测方案

龙芯 7A1000 内置显卡支持轮询和中断两种热插拔方式。不同的热插拔方案需要配合不同的硬件连接方案。如图 5 所示，采用软件轮询方式时，硬件连接选择方案 A。采用中断方式时，硬件连接选择方案 B。

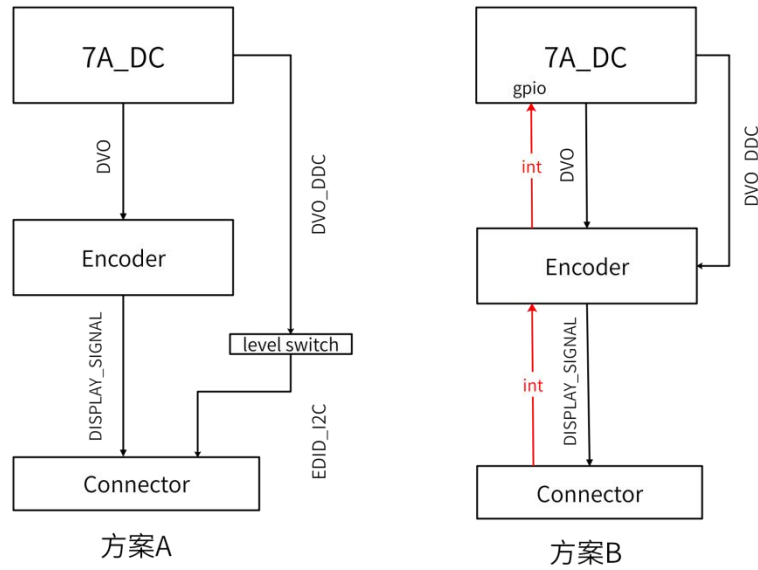


图 5 中断连接方案

无论选择那一种方案都需要在生成 VBIOS 固件时进行配置，如不支持热插拔功能，需要在 VBIOS 中配置关闭热插拔探测功能，关闭热插拔探测情况下，显示接口强制输出视频信号，在这种模式下，驱动不获取外部显示设备的 EDID 信息，仅能将支持 EDID 预存储在 VBIOS 中。

4.4.1 轮询

采用软件轮询方式支持热插拔功能，通过判定是否获取到有效的 EDID 信息来判定显示器是否连接，从而实现热插拔探测功能。如果获取到正确的 EDID 信息则表示显示器连接，否则表示显示器断开连接。通常当连接器不支持热插拔中断时，例如 VGA 接口，我们推荐采用软件轮询方案，需要按照 4.2 节所述方案 A 设计硬件，保证软件能够正确读取到 EDID 信息，通过轮询方式实现热插拔探测，探测间隔固定为 10S。

4.4.2 中断

采用中断方式实现热插拔中断功能，采用这种方案时需要编码器芯片支持热插拔中断功能，在硬件连线时需要将连接器的中断管脚连接到编码芯片的中断输入，将编码器的中断输出连接到 7A1000 桥片的 SPI_CS_{n2}/SPI_CS_{n3}(复用 GPIO 功能，参见《龙芯 7A1000 桥片数据手册》11.2 节)。中断的触发方式为电平触发，由驱动处理上报热插拔中断事件。

hpd_int0	SPI_CS _{n2} 默认下拉
hpd_int1	SPI_CS _{n3} 默认下拉

表 3 热插拔中断管脚

4.5 背光控制

使用 7A1000 内置显卡方案时，背光的硬件设计约束为：显示屏背光开关控制必须使用 7A1000 桥片的 GPIO46 和 GPIO47，显示屏背光的亮度控制必须使用 7A1000 桥片上的 PWM3(GPIO07)，如表 4 所示，否则背光功能可能无法正常工作。

LCD_EN(LCD backlight enable)	CLKSEL0(GPIO46)默认下拉
LCD_VDD_EN(LCD power enable)	CLKSEL1(GPIO47)默认下拉
LCD_PWM(LCD backlight PWM)	PWM3(GPIO07)

表 4 背光控制硬件连接关系

5. VBIOS 存储约定

VBIOS 推荐存放在 7A 桥片上独立的 SPI ROM 中，同时 BIOS 固件中会包含一个默认配置的 VBIOS，固件会优先从 7A 桥片中的 SPI ROM 中读取 VBIOS 传递给内核。我们推荐 VBIOS 存放在 7A 桥片上独立的 SPI ROM 中，方便独立更新 VBIOS。

6. 编码器芯片的选型

为减少和规范编码器芯片的适配工作，编码器芯片选型时，建议和龙芯公司取得联系，方便龙芯公司给出更专业的建议。