

# FT-2000+硬件设计指导手册

2020年6月

版本号 1.2

### 版本历史

版本号	作者	参与者	发布日期	备注
1.0			2018-12-15	
1.1			2020-02-21	修改章节 3 PCIe 布线要求
1.2			2020-06-21	增加第 5 章节, 上电时序

## 目录

1	文档介绍.....	4
2	复用引脚说明.....	4
2.1	引脚复用.....	4
2.2	LPC 功能说明.....	7
2.2.1	LPC 复用关系与信号说明.....	7
2.2.2	LPC 建议接法.....	8
2.3	软件关机、重启.....	8
2.4	配置引脚建议接法.....	8
2.5	I2C.....	11
2.5.1	SPD I2C.....	11
2.5.2	RTC I2C.....	12
3	PCIE 布线要求.....	12
4	内存布线要求.....	12
4.1	阻抗要求.....	12
4.2	交换准则.....	12
4.3	布线要求.....	13
4.3.1	间距要求.....	13
4.3.2	等长要求.....	14
4.3.3	布线叠层要求.....	14
5	上电时序.....	14

# 1 文档介绍

本文档说明了 FT-2000+平台在原理图设计、板级设计阶段需要遵循的基本规则，旨在减少用户在设计阶段的疑惑以及不确定性，增加设计可靠性。

## 2 复用引脚说明

在 FT-2000+芯片设计中，部分功能模块引脚与普通 GPIO 口存在复用关系.设计阶段需了解具体的复用关系、引脚的默认功能和特殊用途。

复用关系需通过对应寄存器来配置不同的功能，默认为 func0 功能。

### 2.1 引脚复用

表 1 引脚功能复用表

SigName	PIN	func0	func1	func2
CLK_REF_50M	AR13	clk_ref		
RESET_N	AN15	reset_n		
POR_N	AP13	por_n		
CRU_SCAN_CLK	AP15	cru_scan_clk		
CRU_CLK_SEL	AM15	cru_clk_sel		
CRU_CLK_STOP	AN14	cru_clk_stop		
CRU_SE	AM14	cru_se		
CRU_SI	AR14	cru_si		
CRU_SO	AN13	cru_so		
CRU_RST_OK	AP14	cru_rst_ok		
CRU_I2C_SCL	AM13	cru_scl		
CRU_I2C_SDA	AM13	cru_sda		
TCK	AJ15	tck		
SJTAG_TDI	AL13	sjtag_tdi		
SJTAG_TMS	AL14	sjtag_tms		
SJTAG_NTRST	AK14	sjtag_ntrst		
SJTAG_TDO	AK15	sjtag_tdo		
SJTAG_TCK	AJ13	sjtag_tck		
FORCE_MB_START	AK13	force_mb_start		
LPC_IRQ_N/GPIO_B6	BH43	hdt_mb_done_state	lpc_ext_irq_outen	peul_linkup_0

LPC_LAD_OUTEN/PEU1_LINKUP1	AJ16	hdt_mb_fail_state	lpc_ext_lad_outen	peu1_linkup_1
INSTANCEID0	N21	instanceid_0		
INSTANCEID1	N20	instanceid_1		
SWJ_NTRST	P22	ntrst_swj		
SWJ_TDI	N22	tdi_swj		
SWJ_SWDITMS	P20	swditms_swj		
SWJ_SWDO	P21	swdo_swj		
SWJ_TCK	N19	tck_swj		
SWJ_TDO	P19	tdo_swj		
UART0_CTS_N/GPIO_A0	AM48	cru_ckobv_sel_0	gpio_porta_0	uart_0_cts_n
UART0_DCD_N/GPIO_A1	AM47	cru_ckobv_sel_1	gpio_porta_1	uart_0_dcd_n
UART0_DSR_N/GPIO_A2	AL48	cru_ckobv_sel_2	gpio_porta_2	uart_0_dsr_n
UART0_RI_N /GPIO_A3	AK47	cru_ckobv_sel_3	gpio_porta_3	uart_0_ri_n
UART0_RTS_N/GPIO_A4	AL46	cru_ckobv_sel_4	gpio_porta_4	uart_0_rts_n
UART0_DTR_N/GPIO_A5/CRU_CLK_OBV	AL47	cru_clk_obv	gpio_porta_5	uart_0_dtr_n
SPI_CSN2/GPIO_A6	AK48	spi_ext_csn2	gpio_porta_6	traceclk_out
SPI_CSN3/GPIO_A7	AK46	spi_ext_csn3	gpio_porta_7	tracectl_out
I2C0_SCL/GPIO_B0	AH47	i2c_0_scl	gpio_portb_0	tracedata_out_0
I2C0_SDA/GPIO_B1	AG47	i2c_0_sda	gpio_portb_1	tracedata_out_1
I2C1_SCL/GPIO_B2	AF47	i2c_1_scl	gpio_portb_2	tracedata_out_2
I2C1_SDA/GPIO_B3	AG46	i2c_1_sda	gpio_portb_3	tracedata_out_3
UART1_RXD/GPIO_B4	BG44	uart_1_rxd	gpio_portb_4	tracedata_out_4
UART1_TXD/GPIO_B5	BH44	uart_1_txd	gpio_portb_5	tracedata_out_5
LPC_IRQ_N/GPIO_B6	BH43	gpio_portb_6	ext_lpc_irq_n	tracedata_out_6
LPC_LAD_0/GPIO_B7	BG43	gpio_portb_7	ext_lpc_lad_0	tracedata_out_7
CRU_RST_FSM0/GPIO_C0	BH42	cru_rst_fsm_0	gpio_portc_0	tracedata_out_8
CRU_RST_FSM1/GPIO_C1	BG42	cru_rst_fsm_1	gpio_portc_1	tracedata_out_9
CRU_RST_FSM2/GPIO_C2	BG41	cru_rst_fsm_2	gpio_portc_2	tracedata_out_10
CRU_RST_FSM3/GPIO_C3	BG40	cru_rst_fsm_3	gpio_portc_3	tracedata_out_11

CRU_RST_FSM4/GPIO_C4	BG39	cru_rst_fsm_4	gpio_portc_4	tracedata_out_12
LPC_LAD_1/GPIO_C5	BH40	gpio_portc_5	ext_lpc_lad_1	tracedata_out_13
LPC_LAD_2/GPIO_C6	BH35	gpio_portc_6	ext_lpc_lad_2	tracedata_out_14
LPC_LAD_3/GPIO_C7	BH41	gpio_portc_7	ext_lpc_lad_3	tracedata_out_15
UART0_RXD /GPIO_D5	BH33	uart_0_rxd		gpio_portd_5
UART0_TXD/GPIO_D6	BG33	uart_0_txd		gpio_portd_6
UART2_RXD	BG31	uart_2_rxd		peu0_c0_clk_obs
UART2_TXD	BH32	uart_2_txd		peu0_c1_clk_obs
UART3_RXD	BH31	uart_3_rxd		peu1_c0_clk_obs
UART3_TXD	BG32	uart_3_txd		peu1_c1_clk_obs
SPI_CSNO	AJ48	spi_ext_csn0		
SPI_CSNI	AJ46	spi_ext_csn1		
SPI_SCK	AJ47	spi_ext_sck		
SPI_DO	AG48	spi_ext_so		
SPI_WP/GPIO_D7	AF48	spi_ext_wp		gpio_portd_7
SPI_DI	AH46	ext_spi_si		
CLK_LPC_33M	AH48	clk_lpc_in		
LPC_LDRQ_N/GPIO_D0	BH34	peu01_phy01_jtag_tdi	ext_lpc_ldrq_n	gpio_portd_0
LPC_LFRAME_N/GPIO_D1	BH39	peu01_phy01_jtag_tms	lpc_ext_lframe_n	gpio_portd_1
PEU01_PHY01_JTAG_TRST_N/GPIO_D2	BG34	peu01_phy01_jtag_trst_n		gpio_portd_2
PEU0_PHY0_JTAG_TCK	P29	peu0_phy0_jtag_tck		
PEU0_PHY1_JTAG_TCK	N30	peu0_phy1_jtag_tck		
PEU0_LINKUP0	N29	peu0_linkup_0		
PEU0_LINKUP1	P30	peu0_linkup_1		
PEU0_C0_CLKREQ_IN_N	N28	pad_peu0_c0_clkreq_in_n		
PEU0_C1_CLKREQ_IN_N	P28	pad_peu0_c1_clkreq_in_n		
PEU0_C2_CLKREQ_IN_N	P26	pad_peu0_c2_clkreq_in_n		
PEU0_C0_CLKREQ_OUT_N	P27	pad_peu0_c0_clkreq_out_n		
PEU0_C1_CLKREQ_OUT_N	N27	pad_peu0_c1_clkreq_out_n		

PEU0_C2_CLKREQ_OUT_N	N26	pad_peu0_c2_clkreq_out_n		
PEU0_PHY0_JTAG_TDO/GPIO_D3	N18	peu0_phy0_jtag_tdo	cru_error_int	gpio_portd_3
PEU0_PHY1_JTAG_TDO/GPIO_D4	P18	peu0_phy1_jtag_tdo		gpio_portd_4
PEU1_C0_CLKREQ_OUT_N	BA15	pad_peu1_c0_clkreq_out_n		
PEU1_C1_CLKREQ_OUT_N	AV15	pad_peu1_c1_clkreq_out_n		
PEU1_PHY0_JTAG_TCK	BA17	peu1_phy0_jtag_tck		
PEU1_PHY1_JTAG_TCK	AV14	peu1_phy1_jtag_tck		
PEU1_C0_CLKREQ_IN_N	BA16	pad_peu1_c0_clkreq_in_n		
PEU1_C1_CLKREQ_IN_N	AW13	pad_peu1_c1_clkreq_in_n		
ALL_PLL_LOCK	AY16	pad_all_pll_lock		
PEU1_PHY0_JTAG_TDO	AY17	peu1_phy0_jtag_tdo		
PEU1_PHY1_JTAG_TDO	AV13	peu1_phy1_jtag_tdo		

## 2.2 LPC 功能说明

由于 LPC 控制器的输入输出引脚与其他功能引脚存在复用关系，而默认的配置方式不是 LPC 功能，因此当客户需要使用 LPC 功能时，需要软件配置为 LPC 功能。

### 2.2.1 LPC 复用关系与信号说明

表 2 LPC 接口复用关系与信号说明

PIN	func0	func1	《 Intel Low Pin Count (LPC) Interface Specification Revision 1.1》对应关系及备注
BH43	gpio_portb_6	ext_lpc_irq_n	SERIRQ
AJ14	hdt_mb_done_state	lpc_ext_irq_outen	电平转换时使用，用来标明 LPC_IRQ 的方向，1: CPU 输出，0: CPU 输入
AJ16	hdt_mb_fail_state	lpc_ext_lad_outen	电平转换时使用，用来标明 LPC_LADx 的方向，1: CPU 输出，0: CPU 输入
GB43	gpio_portb_7	ext_lpc_lad_0	LAD[0]
BH40	gpio_portc_5	ext_lpc_lad_1	LAD[1]
BH35	gpio_portc_6	ext_lpc_lad_2	LAD[2]
BH41	gpio_portc_7	ext_lpc_lad_3	LAD[3]

AH48	clk_lpc_in		LCLK
BH34	peu01_phy01_jtag_tdi	ext_lpc_ldrq_n	LDRQ#
BH39	peu01_phy01_jtag_tms	lpc_ext_lframe_n	LFRAME#

## 2.2.2 LPC 建议接法

当 LPC 外设 IO 的电平与 CPU 的 1.8V LVCMOS 不兼容时，在使用 LPC 功能的时候需要进行电平转换。推荐使用 CPLD 进行电平转换，由于引入了 CPLD，处理更灵活。

lpc\_ext\_irq\_outen、lpc\_ext\_lad\_outen 信号用于电平转换时候，信号的方向控制。

时钟由外部供给 33Mhz 时钟，同一时钟分别给 CPU 和 LPC 设备。

## 2.3 软件关机、重启

CPU 采用 GPIO\_D[4:3]充当软件关机、重启功能脚，对应的 CPU PIN 如下表所示。

表 3 软关机、重启功能 PIN

PIN	func0	func1	func2	说明
N18	peu0_phy0_jtag_tdo	cru_error_int	GPIO_D[3]	
P18	peu0_phy1_jtag_tdo		GPIO_D[4]	

系统启动后，GPIO\_D[4:3]会置为{1, 1}，然后 CLPD 通过判断以下真值表进行相关操作。

表 4 软关机、重启真值表

GPIO_D[4]	GPIO_D[3]	说明
0	0	保留
0	1	断电
1	0	重启
1	1	无操作

注：为保证软件能进行正常的关机与重启，需将 GPIO\_D[4:3]接入 CPLD 或者主板独立控制器，并配合相关逻辑代码实现。

由于 GPIO\_D[4:3]属于复用引脚，且在启动初始化时期，输出状态不确定。必须 CPU 上电复位 5 秒后，外部控制器再判断其值，才为有效值。

## 2.4 配置引脚建议接法

为保证 CPU 能进入正常工作模式，部分引脚需按建议接法连接。若不按此连接，可能会导致 CPU 无法正常工作。建议接法如表 5 所示。

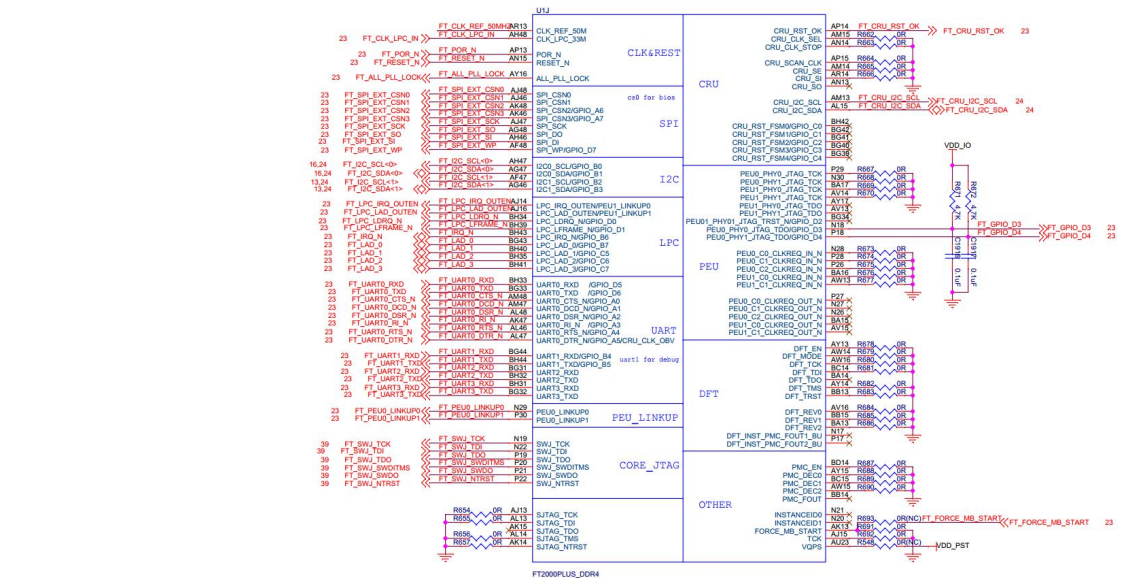


表 5 配置引脚建议接法

NAME	PIN_Number	功能模式接法
CLK_LPC_33M	AH48	33MHz clock
CLK_REF_50M	AR13	50MHz clock
ALL_PLL_LOCK	AY16	CPLD & testpoint
POR_N	AP13	CPLD
RESET_N	AN15	CPLD
CRU_CLK_OBV	AL47	testpoint
CRU_CLK_SEL	AM15	低电平
CRU_CLK_STOP	AN14	低电平
CRU_RST_OK	AP14	CPLD & testpoint
CRU_SCAN_CLK	AP15	低电平
CRU_SE	AM14	低电平
CRU_SI	AR14	低电平
CRU_SO	AN13	悬空
DFT_EN	AY13	低电平
DFT_INST_PMC_FOUT1_BU	N17	悬空
DFT_INST_PMC_FOUT2_BU	P17	悬空
DFT_MODE	AW14	低电平
DFT_REVO	AV16	低电平
DFT_REV1	BB15	低电平
DFT_REV2	BA13	低电平
DFT_TCK	AW16	低电平
DFT_TDI	BC14	低电平
DFT_TDO	BA14	悬空
DFT_TMS	AY14	低电平
DFT_TRST	BB13	低电平
FORCE_MB_START	AK13	低电平
PEU0_C0_CLKREQ_IN_N	N28	低电平
PEU0_C0_CLKREQ_OUT_N	P27	悬空
PEU0_C1_CLKREQ_IN_N	P28	低电平
PEU0_C1_CLKREQ_OUT_N	N27	悬空
PEU0_C2_CLKREQ_IN_N	P26	低电平
PEU0_C2_CLKREQ_OUT_N	N26	悬空
PEU0_PHY0_JTAG_TCK	P29	低电平
PEU0_PHY0_JTAG_TDO/GPIO_D3	N18	CPLD
PEU0_PHY1_JTAG_TCK	N30	低电平
PEU0_PHY0_JTAG_TDO/GPIO_D4	P18	CPLD
PEU1_C0_CLKREQ_IN_N	BA16	低电平
PEU1_C0_CLKREQ_OUT_N	BA15	悬空
PEU1_C1_CLKREQ_IN_N	AW13	低电平
PEU1_C1_CLKREQ_OUT_N	AV15	悬空
PEU1_PHY0_JTAG_TCK	BA17	低电平

PEU1_PHY0_JTAG_TDO	AY17	悬空
PEU1_PHY1_JTAG_TCK	AV14	低电平
PEU1_PHY1_JTAG_TDO	AV13	悬空
PMC_DEC0	AY15	低电平
PMC_DEC1	BC15	低电平
PMC_DEC2	AW15	低电平
PMC_EN	BD14	低电平
PMC_FOUT	BB14	悬空
SJTAG_NTRST	AK14	低电平
SJTAG_TCK	AJ13	低电平
SJTAG_TDI	AL13	低电平
SJTAG_TDO	AK15	悬空
SJTAG_TMS	AL14	低电平
SWJ_SWDITMS	P20	座子
SWJ_SWDO	P21	悬空
TCK	AJ15	低电平
SWJ_TCK	N19	JTAG 调试座
SWJ_TDI	N22	
SWJ_TDO	P19	
SWJ_NTRST	P22	

参考设计如下：



## 2.5 I2C

### 2.5.1 SPD I2C

CPU 初始化内存时，会通过 CPU I2C 读取 DIMM 条的 SPD 信息。为了保证 BIOS/OS 的兼容性，请按照如下表对各内存通道 I2C 地址进行分配。

表 6 SPD I2C 地址分配

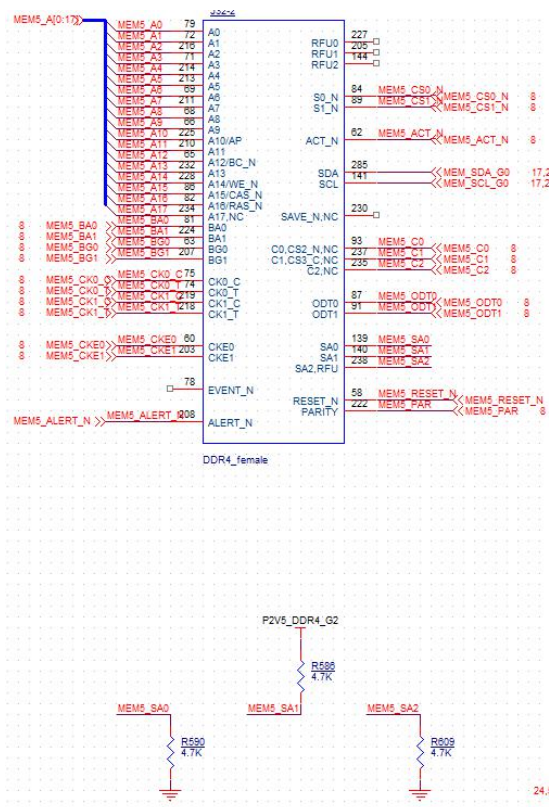
I2C 接口	内存通道	I2C 地址
I2C_0	LMU0	000
	LMU4	001
	LMU5	010
	LMU7	011
I2C_1	LMU1	000
	LMU2	001
	LMU3	010
	LMU6	011

注：在设计时，必须按照以上地址分配进行设计

I2C\_0 读 MEM0、MEM4、MEM5、MEM7 的 SPD

I2C\_1 读 MEM1、MEM2、MEM3、MEM6 的 SPD

例如：如下图所示{MEM5\_SA0, MEM5\_SA1, MEM5\_SA2}={L, H, L}



## 2.5.2 RTC I2C

为了更好的兼容性，外接 RTC 模块，请连接 I2C\_0。

# 3 PCIE 布线要求

差分数据信号阻抗： $85\ \Omega \pm 10\%$

- 建议 TX 差分对上的 AC 耦合电容摆放在主板上，RX 上的 AC 耦合电容摆放在相应的 PCIE 设备上。
- 从输出端到接收端之间，PCIE 布线采用交流耦合的方式。耦合电容采用封装不大于 0402，容值在 176-265nF 之间的电容。
- 差分对 P 和 N 布线上电容应该对称摆放。
- 差分对两线有完全相同的走线方式，同信号层。
- 从布线长度考虑，差分对 P 和 N 两条布线总长度之差不能超过 0.1mm。

# 4 内存布线要求

## 4.1 阻抗要求

内存走线分为单端线和差分线，设计阻抗要求如下：

- 单端线阻抗： $45\ \text{ohm} (\pm 10\%)$ ；
- 差分线阻抗： $75\ \text{ohm} (\pm 10\%)$ ；

## 4.2 交换准则

若按照 8bit 划分为一个 slice，一个通道 72bit 可划分为 9 个 slice。如表 7 所示，每组 slice 内有 12 个信号，以下准则是基于该划分。

### ➤ X8、X16 内存交换准则

- DQ 交换：在进行 X8 兼容内存的设计时，slice 内部的 8 个 DQ 可以自由互换。
- Slice 交换： slice[0:7] 可以进行 slice 之间交换。Slice[8] 为 ECC 用途，不能与其它 slice 交换。若用户需要使用 ECC 的内存条，slice[8] 必须与 DIMM 条上的 slice [8] 进行连接。若确认无需 ECC 功能，可以将 slice [8] 信号浮空即可。

### ➤ X4、X8、X16 内存交换准则

- DQ 交换：每个 slice 可以再细拆分为低 4 位和高 4 位两组，分别为 slice[n]\_L 与 slice[n]\_H，如表 1 所示。slice[n]\_L 内的 DQ 之间可以互相交换，slice[n]\_H 内的 DQ 之间可以互相交换。但不能 slice[n]\_L 内的 DQ 与 slice[n]\_H 内的 DQ 进行交换。
- Slice 交换： slice[0:7] 可以进行 slice 之间交换。Slice[8] 为 ECC 用途，不能

与其它 slice 交换。若用户需要使用 ECC 的内存条，slice[8]必须与 DIMM 条上的 slice [8]进行连接。若确认无需 ECC 功能，可以将 slice [8]信号浮空即可。

- 由上可见，“X8、X16 兼容内存交换准则”相对宽松和灵活，但不一定能兼容 X4，最终需要采用何种准则，需依据用户具体需求而定。用户在不不确定未来可能采用何种内存类型时候，建议“X4、X8、X16 内存交换准则”获取最好的兼容性。

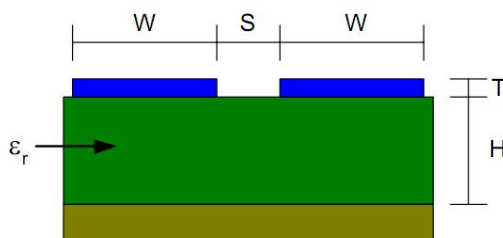
表 7 Slice 分组表

Slice[n]	Slice[n]_L	Slice[n]_H	备注
0	S0_DQ[0:3]、DQS0_C、DQS0_T	S0_DQ[4:7]、DQS9_C、DQS9_T	数据
1	S1_DQ[0:3]、DQS1_C、DQS1_T	S1_DQ[4:7]、DQS10_C、DQS10_T	数据
2	S2_DQ[0:3]、DQS2_C、DQS2_T	S2_DQ[4:7]、DQS11_C、DQS11_T	数据
3	S3_DQ[0:3]、DQS3_C、DQS3_T	S3_DQ[4:7]、DQS12_C、DQS12_T	数据
4	S4_DQ[0:3]、DQS4_C、DQS4_T	S4_DQ[4:7]、DQS13_C、DQS13_T	数据
5	S5_DQ[0:3]、DQS5_C、DQS5_T	S5_DQ[4:7]、DQS14_C、DQS14_T	数据
6	S6_DQ[0:3]、DQS6_C、DQS6_T	S6_DQ[4:7]、DQS15_C、DQS15_T	数据
7	S7_DQ[0:3]、DQS7_C、DQS7_T	S7_DQ[4:7]、DQS16_C、DQS16_T	数据
8	S8_DQ[0:3]、DQS8_C、DQS8_T	S8_DQ[4:7]、DQS17_C、DQS17_T	ECC

## 4.3 布线要求

内存布线在满足高速信号完整性基本要求的基础上，有如下的布线要求。

### 4.3.1 间距要求



如上图所示，W 为走线宽度，S 为线间距，H 为线与参考平面的距离（带状线为走线与最近的参考平面距离）、 $\epsilon_r$  为介电常数，T 为铜厚。

- 带状线，同一 slice，slice 内部之间的布线间距  $S \geq 3xH$ ，长距离的平行走线因该尽可能的拉开间距。
- 带状线，不同 slice 之间，或与 Command/Address/Control 的间距不小于  $4xH$ 。
- 避免走微带线，如果非得走微带线，请保持走线尽量短，同时  $S \geq 4xH$ 。长距离的平行走线因该尽可能的拉开间距。
- 以上间距规则适应的是主板的主体布线。CPU 附近扇出区域，可能违反规则，若长度过长，建议以仿真为准。

### 4.3.2 等长要求

- 同一组 slice（见表 7 Slice 分组表）内 DQ/DQS 相差±10mil。
- MEMx\_CKx\_C、MEMx\_CKx\_T 之间按照±5mil 等长。
- MEMx\_DQSx\_T、MEMx\_DQSx\_C 之间按照±5mil 等长。
- Command/Address/Control 相对 Clock 长度相差±20mil 内。
- RDIMM、UDIMM、SODIMM 内存，DQ/DQS/Command/Address/Control/Clock 在主板上布线建议尽量短，建议控制在 6000mil 内。

### 4.3.3 布线叠层要求

- 同一组 slice（见表 7 Slice 分组表）必须走同一信号层。
- 同一个通道建议走微带线或者带状线，便于生产工艺控制。
- Command/Address/Control 分两层布线的情况下，两层建议同为带状线或微带线。

## 5 上电时序

表 8 上电复位信号列表

信号名	说明
PWR_BTN_EN	上电按钮，低电平有效
ATX_EN	ATX 电源加电信号，低电平有效
ATX_GD	ATX 电源加电完成信号
VDDIO_EN	1.8V IO 电源加电信号，低电平有效
VDDIO_GD	1.8V IO 电源加电信号加电完成信号
VDD_CORE_EN	CPU 核心电源加电信号
VDD_CORE_GD	CPU 核心电源加电完成信号
VDDA_PCIE_EN	PCIE 核心电源加电信号，低电平有效
VDDA_PCIE_GD	PCIE 核心电源加电完成信号
VDD_MCU_EN	1.2V DDR 电源加电信号，低电平有效
VDD_MCU_GD	1.2V DDR 电源加电完成信号
RESET_N	热复位信号，低有效
POR_N	上电复位信号，低有效
PCIE_SLOT_RST_N	所有 PCIE 插槽端的复位信号，信号数目根据 PCIE 设备而定。

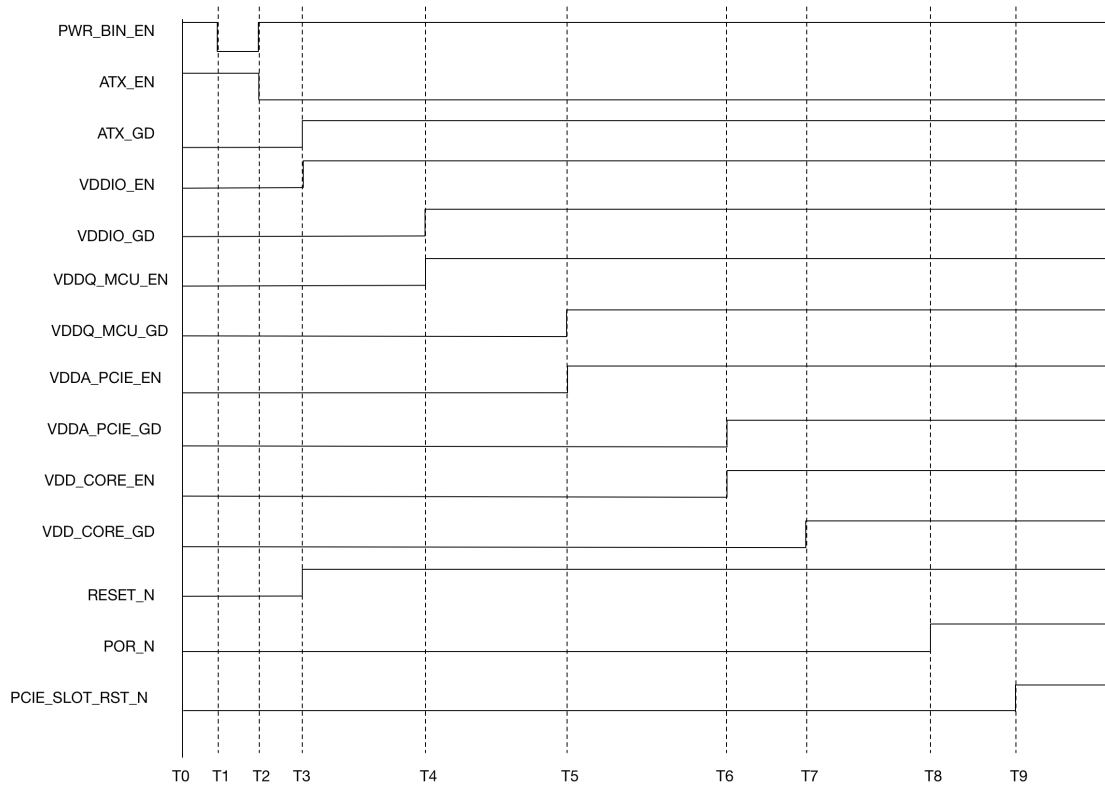


图 5-1 上电复位参考时序图

时序说明:

表 9 上电复位参考时序说明

时刻	说明
T0	作为上电复位控制器的 CPLD 或 FPGA 加电并完成复位
T1	加电按钮按下
T2	加电按钮弹起，ATX 电源加电
T3	收到 ATX 加电完成信号，给 1.8V IO 电源加电，可以视需要增加延迟；释放 RESET_N（热复位）信号
T4	收到 1.8V IO 电源加电完成信号，给 VDDQ_MCU 1.2V 电源加电
T5	收到 VDDQ_MCU 1.2V 加电完成信号，给 PCIE 0.85V 核心电源加电
T6	收到 PCIE 0.85V 加电完成信号，给 VDD_CORE 电源上电
T7	收到 VDD_CORE 上电完成信号
T8	相对 T7 时刻延时 40ms（可调整），释放 POR_N（上电复位）信号
T9	相对 T8 时刻延时 100ms，拉高 PCIE_SLOT_RST_N，完成 PCIE 设备复位