

玄铁 E902 R2S2 集成手册

版本号 01

安全级别 保密

发布日期 2020/07/30

Copyright © 2020 平头哥半导体有限公司，保留所有权利。

本文档的所有权及知识产权归属于平头哥半导体有限公司及其关联公司(下称“平头哥”)。本文档仅能分派给：(i)拥有合法雇佣关系，并需要本文档的信息的平头哥员工，或(ii)非平头哥组织但拥有合法合作关系，并且其需要本文档的信息的合作方。对于本文档，未经平头哥半导体有限公司明示同意，则不能使用该文档。在未经平头哥半导体有限公司的书面许可的情形下，不得复制本文档的任何部分，传播、转录、储存在检索系统中或翻译成任何语言或计算机语言。

商标申明

平头哥的 LOGO 和其它所有商标归平头哥半导体有限公司及其关联公司所有，未经平头哥半导体有限公司的书面同意，任何法律实体不得使用平头哥的商标或者商业标识。

注意

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

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。平头哥半导体有限公司不对任何第三方使用本文档产生的损失承担任何法律责任。

平头哥半导体有限公司 T-HEAD Semiconductor Co.,LTD

地址： 杭州市余杭区向往街 1122 号欧美金融城 (EFC) 英国中心西楼 T6

邮编： 310012

网址： www.T-head.cn

版本历史

文档版本	描述	作者	日期
1.0.0	第一次正式发布。	T-Head	2020-06-19
01	更新文档名称和版本号，勘误文档。	T-Head	2020-07-30

目录

1	概述	1
1.1	适用范围	1
1.2	处理器简介	1
1.3	可配置选项	2
1.4	处理器集成总览	2
2	端口信号总览	4
2.1	命名规则	5
2.2	端口信号列表	5
3	时钟复位集成	14
3.1	端口列表	14
3.2	时钟信号	15
3.2.1	E902 时钟域	15
3.2.2	调频操作	16
3.3	多时钟域信号同步	17
3.3.1	CPU 内核时钟域与系统总线时钟域	17
3.3.2	CPU 内核时钟域与 JTAG 时钟域	17
3.4	复位信号	17
4	总线系统集成	19
4.1	总线简介及配置信息	19
4.2	系统总线接口	21
4.3	指令总线接口	23
5	中断系统集成	26
5.1	中断处理过程简述	26
5.2	端口列表	26
5.3	中断握手时序图	27
5.4	中断嵌套	28
5.5	计时器中断	28
6	调试系统集成	29
6.1	端口列表	29
6.2	JTAG 接口	30
7	低功耗系统集成	31
7.1	端口列表	31
7.2	工作模式及其转换	31
7.3	进入低功耗模式握手	32
7.4	退出低功耗模式握手	32
8	CPU 运行观测信号集成	34
8.1	简介	34
8.2	通用观测信号	34
8.3	观测信号示例波形	36

9 地址空间属性设置	37
10 工艺映射	38
10.1 软核授权 RTL 代码结构	38
10.2 ASIC 映射	39
10.2.1 ICG 替换	39
10.2.1.1 文件所在位置	39
10.2.1.2 选择 gated cell	39
10.2.1.3 端口名字	40
10.2.1.4 更改 gated clk cell.v 文件	41
10.2.1.5 更新 filelist	41
10.2.1.6 不使用前端插的 gated cell	41
10.2.2 Memory 替换	42
10.2.2.1 文件所在位置	42
10.2.2.2 生成 memory	42
10.2.2.3 端口名字	42
10.2.2.4 RAM 的读写时序	43
10.2.2.5 修改 memory instance 文件	44
10.2.2.6 拼接 memory	45
10.2.2.7 修改 filelist	46
10.2.3 DesignWare IP	46
10.2.3.1 简介	46
10.3 FPGA 映射	47
10.3.1 DesignWare IP	47
10.3.2 修改文件列表	48
11 门级仿真	49
11.1 无复位端寄存器	49

图表目录

图表 1-1 E902 系统框图.....	1
图表 1-2 E902 可配置选项.....	2
图表 2-1 E902 CPU 系统接口总体描述.....	4
图表 2-2 信号命名规则.....	5
图表 2-3 端口信号列表.....	13
图表 3-1 时钟信号列表.....	14
图表 3-2 复位控制信号列表.....	15
图表 3-3 动态调频信号列表.....	15
图表 3-4 E902 的时钟管理方式.....	16
图表 3-5 动态调频时序图.....	16
图表 3-6 CPU 与 JTAG 时钟域之间的同步逻辑.....	17
图表 3-7 E902 复位信号产生机制.....	18
图表 4-1 E902 总线矩阵.....	19
图表 4-2 多总线接口的基本信息和可配置性.....	20
图表 4-3 指令总线对基地址和地址对齐的要求.....	21
图表 4-4 系统总线接口信号列表.....	23
图表 4-5 指令总线接口信号列表.....	25
图表 5-1 CLIC 中断控制器接口信号.....	26
图表 5-2 CPU 配置 CLIC 时中断相关信号时序简图.....	28
图表 6-1 debug 端口列表.....	30
图表 7-1 低功耗端口信号列表.....	31
图表 7-2 CPU 状态转换图.....	32
图表 7-3 进入低功耗模式握手.....	32
图表 7-4 退出低功耗模式握手.....	33
图表 8-1 通用观测信号.....	35
图表 8-2 观测信号示例波形.....	36
图表 9-1 sysmap.h 地址属性.....	37
图表 10-1 Release 文件夹内容.....	38
图表 10-2 正沿触发 gated cell.....	39
图表 10-3 TSMC40 工艺下正沿触发 gated cell 的规格表.....	40
图表 10-4 例化 gated cell.....	40
图表 10-5 gated cell 信号列表.....	40
图表 10-6 修改例化文件.....	41
图表 10-7 不使用前端插入的 gated cell.....	41
图表 10-8 例化 memory.....	42
图表 10-9 memory 信号列表.....	43
图表 10-10 RAM 的读写时序图.....	44
图表 10-11 修改 memory 例化文件.....	45
图表 10-12 需要拼接的 memory.....	45

图表 10-13 拼接 memory	46
图表 10-14 综合选项	47
图表 10-15 选择 Design ware 库	48
图表 11-1 寄存器 Q 端赋值示例	49

文档编号说明

平头哥半导体技术文档类编号采用如下所示的规则：

产品名称-产品型号-产品版本号-文档类型

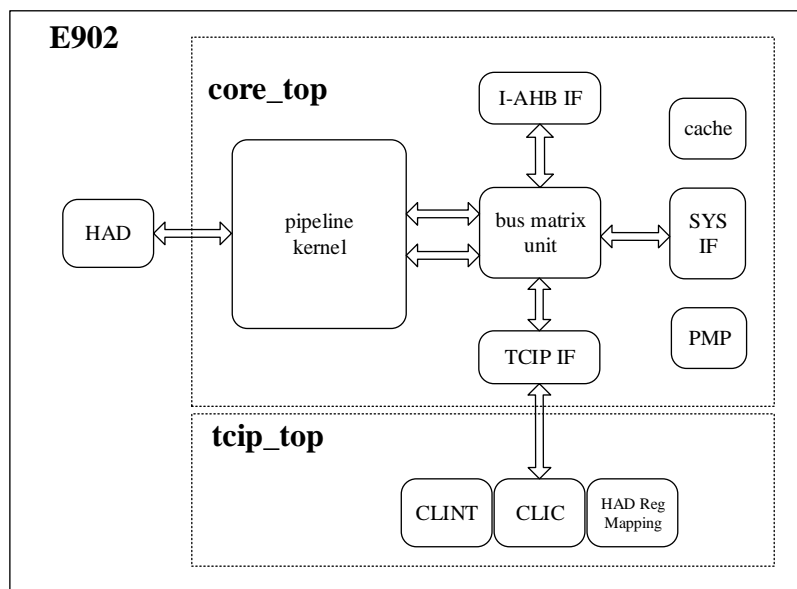
1 概述

1.1 适用范围

本文档适用于平头哥半导体有限公司的玄铁系列 CPU E902。

1.2 处理器简介

玄铁 E902 是平头哥半导体有限公司自主研发的极低功耗、极低成本嵌入式 CPU 核，以 8 位 CPU 的成本获得 32 位嵌入式 CPU 的运行效率与性能。E902 兼容 RISC-V 指令架构，采用 16/32 位混合编码系统，指令系统与流水线硬件结构精简高效，具备极低成本、极低功耗和高代码密度等优点。E902 主要针对智能卡、智能电网、低成本微控制器、无线传感网络等嵌入式应用。



图表 1-1 E902 系统框图

E902 处理器体系结构的主要特点如下：

- ✧ 32 位 RISC 处理器；
- ✧ 支持 RISC-V RV32E[M]C 指令集；
- ✧ 支持 RISC-V 32/16 位混编指令集；
- ✧ 16 个 32 位通用寄存器；
- ✧ 两级顺序执行流水线；
- ✧ 支持 RISC-V 机器模式和用户模式；

- ✧ 可配置的单周期硬件乘法器，多周期硬件除法器；
- ✧ 兼容 RISC-V CLIC 中断标准，支持中断嵌套，外部中断源数量最高可配置 240 个；
- ✧ 兼容 RISC-V PMP 内存保护标准，0/4/8/12/16 区域可配置；
- ✧ 支持 AHB-Lite 总线协议，支持指令总线，系统总线；
- ✧ 支持可配置的指令高速缓存，缓存行 16 字节，容量 2KiB/4KiB/8KiB 可配；
- ✧ 支持平头哥扩展编程模型；
- ✧ 支持复位启动地址硬件集成时可配置；
- ✧ 支持软复位操作；

1.3 可配置选项

E902 可配置选项如图表 1-2 所示。

可配置单元	配置选项	详细
指令 cache	无 /2KiB/4KiB/8KiB	可以配置为 2KiB、4KiB、8KiB。
硬件乘法器	无/有	若配置硬件乘法器，仍可配置单周期快速乘法器或者多周期（3-34）慢速乘法器。
内存保护单元	0/4/8/12/16	可以配置为 0/4/8/12/16 个表项，其中 0 表示不实现内存保护单元。
CLIC	中断源：1-240 任意可配 中断优先级有效位：2-5 位任意可配	支持 1-240 中断源任意可配，中断优先级有效位：2-5 位任意可配，对应 4-32 个优先级。

图表 1-2 E902 可配置选项

1.4 处理器集成总览

E902 的集成按照功能可以分为以下六大系统，可以按照功能逐个系统进行集成：

1) 时钟复位集成

时钟复位集成介绍了 E902 处理器中 CPU 核与调试单元（HAD）的输入时钟以及复位信号；同时介绍了 CPU 核与 HAD 以及总线系统之间信号同步逻辑。

2) 总线系统集成

总线系统集成介绍 2 条总线接口，包括：系统总线接口和指令总线接口。

3) 中断系统集成

中断系统集成介绍与 E902 处理器中断处理相关的信号以及信号握手机制。

4) 调试系统集成

调试系统集成介绍调试相关的 JTAG 接口信号和其它调试辅助信号接口。

5) 低功耗系统集成

低功耗系统集成介绍 E902 低功耗相关的接口信号以及 E902 低功耗模式与正常工作模式之间的转换关系，具体包括了：CPU 进入和退出低功耗模式时的状态转换，进入和退出低功耗模式相关信号的握手时序以及处理器处于低功耗模式被唤醒的流程。

6) CPU 运行观测信号集成

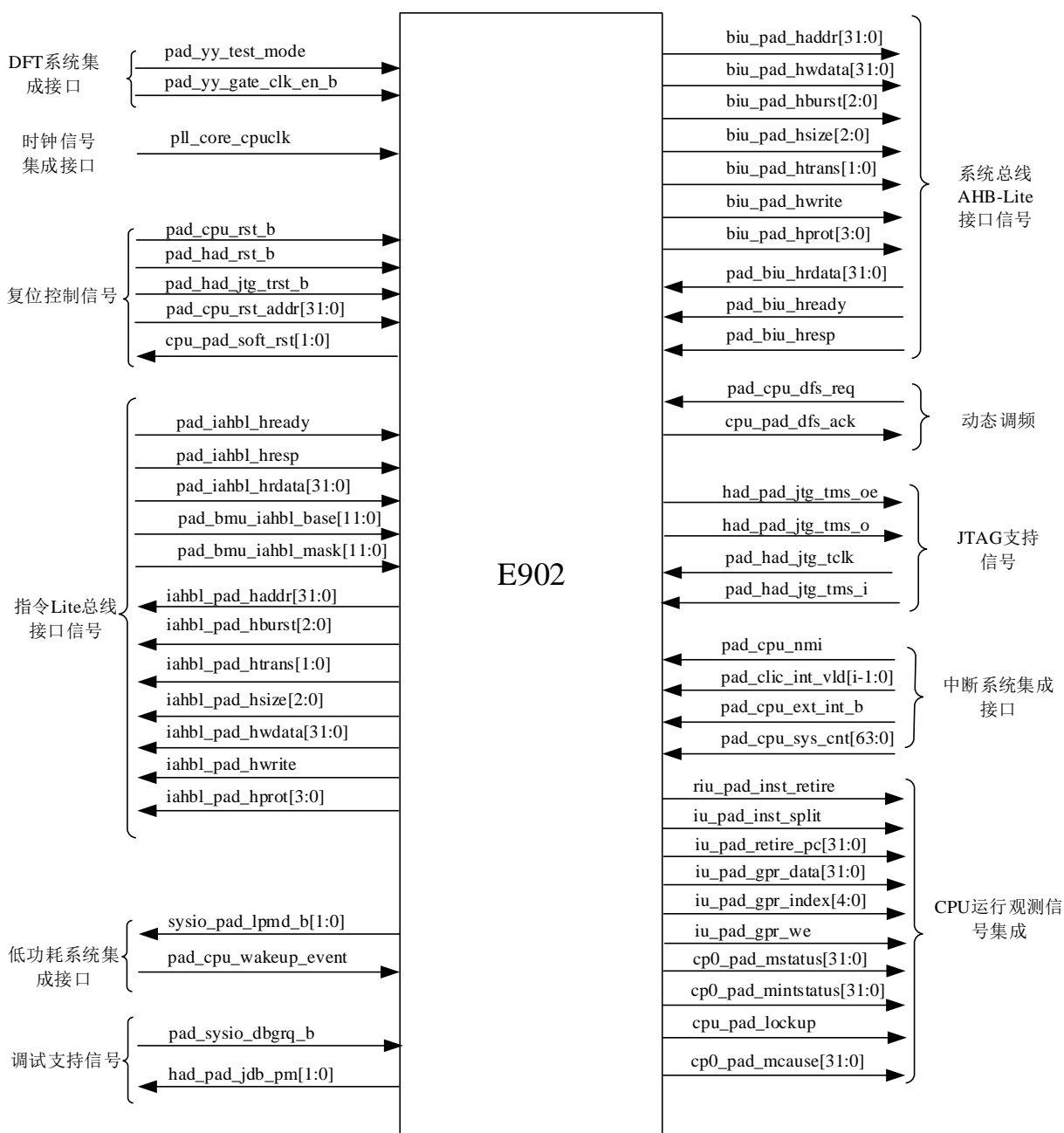
CPU 运行观测信号集成介绍 SoC 仿真阶段所需的 CPU 运行状态的观测信号。

7) 地址空间属性配置

系统设计人员需要通过设置 sysmap.h 文件完成对不同地址空间属性的设定。

2 端口信号总览

E902 处理器支持 AHB-Lite 总线接口协议。在平头哥的多款玄铁处理器中，均保持了稳定的延续性，以方便 E902 的升级换代。根据 E902 顶层端口信号的特点，划分为时钟复位信号，总线系统信号，中断系统信号，调试系统信号，低功耗系统信号，DFT 系统信号，CPU 运行观测信号等几大类。



图表 2-1 E902 CPU 系统接口总体描述

2.1 命名规则

pad_biu_* pad_bmu_* pad_clic_* pad_cpu_* pad_had_* pad_iahbl_* pad_sysio_* pad_yy_*	输入信号。
biu_pad_* cp0_pad_* had_pad_* iahbl_pad_* iu_pad_* sysio_pad_*	输出信号。
*_b	低电平有效信号。

图表 2-2 信号命名规则

一般来说，没有特殊规定，E902 的输入输出信号均为高电平有效，但是注意，如果是以“_b”结尾的信号，则是低电平有效。

2.2 端口信号列表

信号名	I/O	Reset	时 钟 域	内部 门级 数	功能描述
时钟复位信号集成：					
状态和时钟信号：					
pll_core_cpucclk	I	-	-	< 5	CPU 工作时钟信号：

					提供 CPU 工作的时钟。
pad_had_jtg_tclk	I	-	-	<5	<p>JTAG 测试时钟信号：</p> <p>JTAG 和 HAD 内部相关寄存器的时钟信号，正常工作时要求该时钟的工作频率小于 CPU 时钟频率的二分之一。</p> <p>注：该信号在测试模式（pad_yy_test_mode）下作为 E902 内所有逻辑的输入时钟。</p>
复位控制信号：					
pad_cpu_rst_b	I	-	CPU	≈7	<p>处理器复位信号：</p> <p>低电平时，复位 CPU；CPU 采用异步复位、同步释放的方式。</p> <p>注意：要求系统将 CPU 复位信号同步到 CPU 时钟域，CPU 内部不进行同步操作。</p>
pad_had_rst_b	I	-	CPU	<5	<p>调试模块复位信号：</p> <p>低电平时，复位调试模块，调试模块采用异步复位、同步释放的方式。</p> <p>注意：要求系统将 HAD 复位信号同步到 CPU 时钟域，CPU 内部不进行同步操作。</p>
pad_had_jtg_trst_b	I	-	JTAG	<5	<p>JTAG 测试复位信号：</p> <p>两线 JTAG 接口逻辑复位信号，低电平有效。该信号还用做测试模式下（pad_yy_test_mode=1）E902 内所有逻辑的复位。</p>

pad_cpu_rst_addr[31:0]	I	-	CPU	-	32 位复位启动地址，bit[1:0]需为 0
cpu_pad_soft_rst[1:0]	O	2'b00	CPU	-	<p>软复位指示信号，模式如下：</p> <p>2'b00:无复位请求</p> <p>2'b01:仅复位内核</p> <p>2'b10:复位系统</p> <p>2'b11:保留</p> <p>上述所有操作 CPU 内部均不实现复位操作，需要 SoC 设计人员依据相关信号复位相关逻辑，该信号保持一个 cpu 时钟周期有效</p>
动态调频信号：					
pad_cpu_dfs_req	I	-	CPU		系统向 CPU 发起的动态调频请求信号
cpu_pad_dfs_ack	O	-	SYS		CPU 因应系统的动态调频请求，发出的调频准备操作完成信号。系统设计人员在看到该信号为高时可以开始 CPU 的调频操作。
总线系统集成：					
pad_bmu_iahbl_base	I	-	-	<5	指令总线地址空间基地址
pad_bmu_iahbl_mask	I	-	-	<5	指令总线地址空间掩码
系统总线接口信号：					
pad_biu_hrdata[31:0]	I	-	CPU	≈ 26	读取数据信号
pad_biu_hresp	I	-	CPU	≈ 30	传输应答信号

pad_biu_hready	I	-	CPU	≈ 30	外部空闲信号
biu_pad_haddr[31:0]	O	-	CPU	≈ 25	地址信号
biu_pad_htrans[1:0]	O	-	CPU	≈ 35	传输类型信号
biu_pad_hsize[2:0]	O	-	CPU	≈ 22	传输尺寸指示信号
biu_pad_hburst[2:0]	O	-	CPU	<5	突发传输指示信号
biu_pad_hwdata[31:0]	O	-	CPU	≈ 7	写传输数据信号
biu_pad_hwrite	O	-	CPU	≈ 18	读写传输指示信号
biu_pad_hprot[3:0]	O	-	CPU	≈ 28	保护控制信号： 指示当前总线周期进行的数据传输的特性： ***0：取指令； ***1：数据访问； **0*：用户模式访问； **1*：机器模式访问； *0**：Non bufferable； *1**：bufferable； 0***：Non cacheable； 1***：cacheable。
指令总线接口信号：					
pad_iahbl_hrdata[31:0]	I	-	CPU	≈ 26	读取数据信号
pad_iahbl_hresp	I	-	CPU	≈ 30	传输应答信号
pad_iahbl_hready	I	-	CPU	≈ 30	外部空闲信号
iahbl_pad_haddr[31:0]	O	-	CPU	≈ 25	地址信号
iahbl_pad_htrans[1:0]	O	-	CPU	≈ 35	传输类型信号

iabhl_pad_hsize[2:0]	O	-	CPU	≈22	传输尺寸指示信号
iabhl_pad_hburst[2:0]	O	-	CPU	<5	突发传输指示信号
iahbl_pad_hwdata[31:0]	O	-	CPU	≈7	回写数据信号
iahbl_pad_hwrite	O	-	CPU	≈18	写传输信号
iahbl_pad_hprot[3:0]	O	-	CPU	≈28	<p>保护控制信号：</p> <p>指示当前总线周期进行的数据传输的特性：</p> <p>***0：取指令；</p> <p>***1：数据访问；</p> <p>**0*：用户模式访问；</p> <p>**1*：机器模式访问；</p> <p>*0**：Non bufferable；</p> <p>*1**：bufferable；</p> <p>0***：Non cacheable；</p> <p>1***：cacheable。</p>
中断系统集成：					
pad_clic_int_vld[i-1:0] i : 1~240	I	-	CPU	≈9	<p>CLIC 中断控制器中断源请求信号：高电平时表示该中断源发起中断申请。</p> <p>注意：1、要求系统将该信号同步到 CPU 时钟域，CLIC 内部不进行同步操作。2、E902 最多支持 240 个外部中断。</p>
pad_cpu_ext_int_b	I	-	SYS	<5	<p>CPU 中断请求信号：</p> <p>低电平时表示外部中断控制器发起中断申请。</p> <p>注：该中断为 CLINT 下外部 11 号中断。</p>

pad_cpu_sys_cnt[63:0]	I	-	-	<5	<p>定时器输入信号：</p> <p>外部 SoC 在始终不掉电的模块设计实现该计数器，该信号作为该计数器实时计数值被传入 E902 的定时器模块。CPU 将其与定时器设定的初始值比较，判断是否触发定时器中断。</p> <p>注：该信号需要由 CPU 外部同步到 CPU 时钟域</p>
pad_cpu_nmi	I	-	CPU	-	<p>不可屏蔽中断，输入脉冲信号，上升沿有效，cpu 时钟域信号。</p>
调试系统集成：					
JTAG 支持信号：					
pad_had_jtg_tclk	I	-	-	<5	<p>JTAG 测试时钟信号：</p> <p>JTAG 和 HAD 内部相关寄存器的时钟信号，正常工作时要求该时钟的工作频率小于 CPU 时钟频率的二分之一。</p>
pad_had_jtg_tms_i	I	-	JTAG	<5	<p>JTAG-2 串行数据输入信号：</p> <p>HAD 端在 JTAG 时钟信号（tclk）的上升沿对其采样，而外部调试器在 tclk 的下降沿设置该信号。</p> <p>JTAG 接口没有通信时，必须将该信号保持为高电平，同时停止 tclk。如果 tclk 信号一直有效，用户只需保持该信号为高电平状态并维持 80 个时钟周期即可同步复位 JTAG 接口逻辑，包括 TAP 状态机</p>

					等。
had_pad_jtg_tms_o	O	-	JTAG	<5	JTAG-2 串行数据输出信号： HAD 端在 tclk 的下降沿对其设置，而外部调试器在 tclk 的上升沿对其采样。
had_pad_jtg_tms_oe	O	1'b0	JTAG	<5	JTAG-2 串行数据输出有效指示信号： 要求 CPU 外部利用该信号通过一个三态门将 pad_had_jtg_tms_i 和 had_pad_jtg_tms_o 信号合为一个双向端口信号。
调试支持信号：					
pad_sysio_dbgrq_b	O	-	SYS	<5	同步调试请求信号： 可用于使 CPU 进入调试模式或使 CPU 从仅关闭时钟的低功耗状态唤醒。
had_pad_jdb_pm[1:0]	O	2'b0	SYS	<5	处理器工作模式指示信号： 表明处理器的工作模式。 00: 普通模式； 01: 低功耗模式（WFI 模式）； 10: 调试模式； 11: 保留。
低功耗系统集成：					
sysio_pad_lpmdb_b[1:0]	O	2'b11	SYS	<5	低功耗模式状态信号： 当处理器执行 wfi 指令时，sysio_pad_lpmdb_b[1:0] 被相应的改变：

					00: 深度睡眠; 01: 浅度睡眠; 10: 保留; 11: 普通模式。
pad_cpu_wakeup_event	I	-	CPU	<5	事件唤醒请求, 输入脉冲信号, 上升沿有效; 对于同一外设 IP, 集成时可选择连接在 pad_clic_int_vld 信号上利用中断唤醒 cpu, 或者连接在 pad_cpu_wakeup_event 信号上利用事件信号唤醒 cpu, 二者不可同时选择。
DFT 系统集成:					
pad_yy_test_mode	I	-	-	<5	测试模式信号: E902 的测试模式输入信号, 该信号仅在 E902 内用于选择测试模式下的时钟和复位信号。
pad_yy_gate_clk_en_b	I	-	CPU	<5	门控时钟使能信号: 当在 scan 模式下可将该信号接为 scan_enable, 其他时刻可接 0。
CPU 运行观测信号集成:					
iu_pad_inst_retire	O	-	CPU	-	指令退休有效信号。
iu_pad_inst_split	O	-	CPU	-	当前指令为拆分指令。
iu_pad_retire_pc[31:0]	O	-	CPU	-	当前退休指令程序计数器的值。
iu_pad_gpr_data[31:0]	O	-	CPU	-	当前指令退休时 GPR 回写数据。

iu_pad_gpr_index[4:0]	O	-	CPU	-	当前退休指令时 GPR 回写寄存器选择信号。
iu_pad_gpr_we	O	-	CPU	-	当前指令退休时 GPR 回写有效信号。
cp0_pad_mstatus[31:0]	O	-	CPU	-	当前 MSTATUS 寄存器的值。
cp0_pad_mintstatus[31:0]	O	-	CPU	-	当前 MINTSTATUS 寄存器的值。
cp0_pad_mcause[31:0]	O	-	CPU	-	当前 MCAUSE 寄存器的值。
cpu_pad_lockup	O	0	CPU	-	当 CPU 处于锁定状态时会将该信号置为高电平，一旦触发 CPU 锁定状态，该信号一直为 1，直到复位请求将 CPU 复位

图表 2-3 端口信号列表

注：

1. 内部门级数是为了描述信号端口时序：如果端口是 input，该数值表示了这根信号进入处理器内部后还需要通过几级门才能到寄存器；如果端口是 output，则该数值表示该信号从 CPU 内部寄存器出来还通过了几级门才到达端口。集成人员可通过这个数字结合 SoC 连接端口的延时，预判连接后时序是否能满足设计需求。
2. 总线信号内部门级数均参照 NON FLOP_OUT 输出形式。
3. 因为总线为 NON FLOP_OUT，接口信号均为 CPU 时钟域。
4. CPU 内部 flip-flop 的门级数最坏情况大约为 53（有些特殊抗物理攻击配置配上后时序会更差一点）。
5. 内部门级数参照标准库为 scc55nll_hd_hvt，仅供参考，其它工艺库可能会存在一定误差。

3 时钟复位集成

3.1 端口列表

时钟信号：

信号名	方向	复位	时钟	功能描述
pll_core_cpucclk	I	-	-	CPU 工作时钟信号： 提供 CPU 工作的时钟。
pad_had_jtg_tclk	I	-	-	JTAG 测试时钟信号： JTAG 和 HAD 内部相关寄存器的时钟信号， 正常工作时要求该时钟的工作频率小于 CPU 时钟频率的二分之一。 注：该信号在测试模式 (pad_yy_test_mode) 下亦作为 E902 内所有逻辑的输入时钟。

图表 3-1 时钟信号列表

复位控制信号：

信号名	方向	复位	时钟	功能描述
pad_cpu_rst_b	I	-	CPU	处理器复位信号： 低电平时，复位 CPU；CPU 采用异步复位、 同步释放的方式。 注意：要求系统将 CPU 复位信号同步到 CPU 时钟域，CPU 内部不进行同步操作。
pad_had_rst_b	I	-	CPU	调试模块复位信号： 低电平时，复位调试模块，调试模块采用异 步复位、同步释放的方式。 注意：要求系统将 HAD 复位信号同步到 CPU 时钟域，CPU 内部不进行同步操作。

信号名	方向	复位	时钟	功能描述
pad_had_jtg_trst_b	I	-	TCLK	JTAG 测试复位信号： 两线 JTAG 接口逻辑复位信号，低电平有效。 该信号还用做测试模式下（pad_yy_test_mode=1）E902 内所有逻辑的复位。
cpu_pad_srst	O	0	SYS	软件复位指示信号，高电平有效。 当软件配置 CPU 内相应控制寄存器时可以使该信号维持一个系统时钟周期的高电平，系统设计人员可以用该信号对 CPU 进行复位处理。
pad_cpu_rst_addr	I	-	CPU	CPU 复位启动地址指示信号

图表 3-2 复位控制信号列表

动态调频信号：

信号名	方向	复位	时钟	功能描述
pad_cpu_dfs_req	I	0	CPU	系统向 CPU 发起的动态调频请求信号
cpu_pad_dfs_ack	O	0	SYS	CPU 因应系统的动态调频请求，发出的调频准备操作完成信号。系统设计人员在看到该信号为高时可以开始 CPU 的调频操作。

图表 3-3 动态调频信号列表

3.2 时钟信号

3.2.1 E902 时钟域

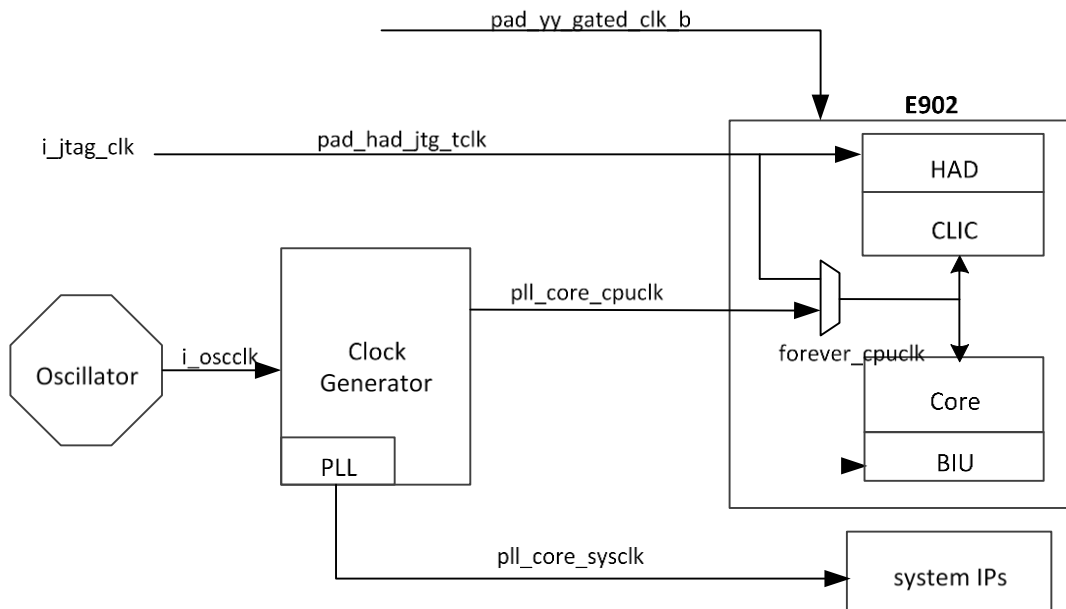
E902 CPU 有两个外部输入时钟，分别为：pll_core_cpucclk 和 pad_had_jtg_tclk。

pll_core_cpucclk 为 CPU 域时钟，用于 CPU 内部所有寄存器的时钟，包括 CPU Core、中断控制器（CLIC）、调试辅助单元（HAD）和总线接口模块（BIU）。CPU 内部设计细粒度的门控时钟单元用来降低 CPU 的动态功耗。通过处理器的执行状态控制门控时钟单元的开启和关闭。在 CPU 进入低功耗模式时，同步关闭门控时钟单元。

pad_had_jtg_clk 为 JTAG 接口逻辑时钟，用于 CPU 内部调试辅助单元（HAD）中 JTAG 状态机的运

行。另外其还作为测试模式下 E902 内部所有逻辑单元的工作时钟，E902 内部会将测试时钟（pad_had_jtg_tclk）与 pll_core_cpucclk 进行选择后产生 forever_cpucclk，作为 CPU 的工作时钟。

注意：E902 外部系统时钟与 CPUCLK 需为同频同相。

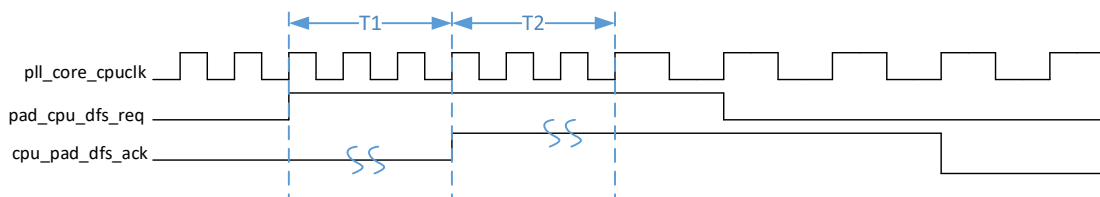


图表 3-4 E902 的时钟管理方式

图表 3-4 是 E902 的时钟管理示意图。pad_had_jtag_tclk 是 JTAG 时钟，与 CPU 时钟异步，该时钟信号属于 SoC 系统的顶层信号，需要从芯片引脚接入。时钟产生逻辑以板上晶振时钟 i_oscclk 为输入，产生系统工作时钟 pll_core_sysclk、CPU 工作时钟 pll_core_cpucclk。CPU 内部生成 forever_cpucclk，用于 CPU 核、BIU 以及 TCIP(包括 CLIC)使用。

3.2.2 调频操作

E902 支持 SoC 对处理器时钟的动态调节。SoC 设计人员通过置起 pad_cpu_dfs_req 信号通知处理器将要进行调频操作，如图表 3-5 所示，经过 T1 时间后处理器会置起 cpu_pad_dfs_ack 信号表示已经准备完成等待调节。在经过 T2 时间调频操作完成后 SoC 设计人员需将 pad_cpu_dfs_req 置 0，cpu_pad_dfs_ack 也会在两个处理器时钟周期后被置为 0，处理器继续运行。



图表 3-5 动态调频时序图

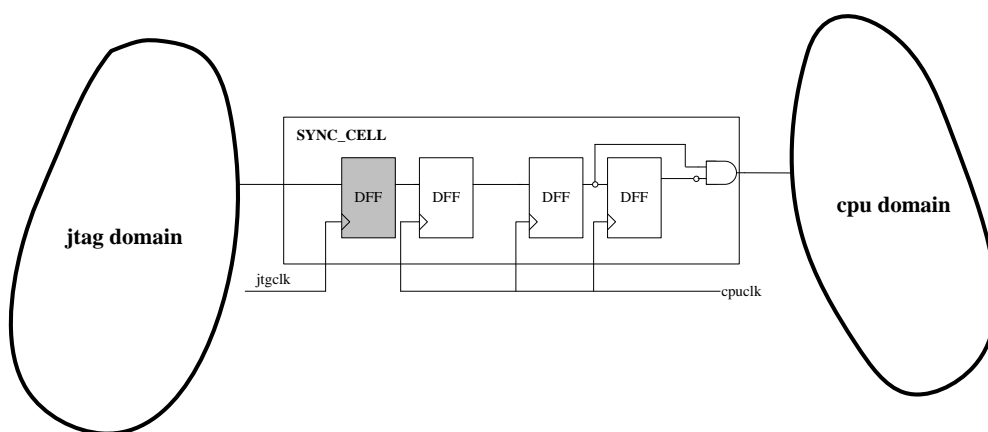
3.3 多时钟域信号同步

3.3.1 CPU 内核时钟域与系统总线时钟域

在 E902 的设计中，CPU 内核时钟域与系统总线时钟域在总线接口部分逻辑中有信号交互，要求这两个时钟在物理设计中保证同频同相。

3.3.2 CPU 内核时钟域与 JTAG 时钟域

在 E902 的设计中，CPU 内核时钟 cpuclock 信号与 HAD 内部时钟 jtclk 信号之间有信息交互，CPU 内部已经通过同步逻辑单元将两个时钟域之间的信号进行了有效同步，用户无需关心。其逻辑框图如图表 3-6 所示。



图表 3-6 CPU 与 JTAG 时钟域之间的同步逻辑

3.4 复位信号

E902 输入三根复位信号，分别是用于 CPU 核（除 HAD 之外）的复位信号 pad_cpu_rst_b、用于 HAD 中 CPU 域的复位信号 pad_had_rst_b 和用于 HAD TCLK 域的复位信号 had_pad_jtg_trst_b。

在测试模式下（pad_yy_test_mode=1），复位信号全部选择 pad_had_jtg_trst_b。

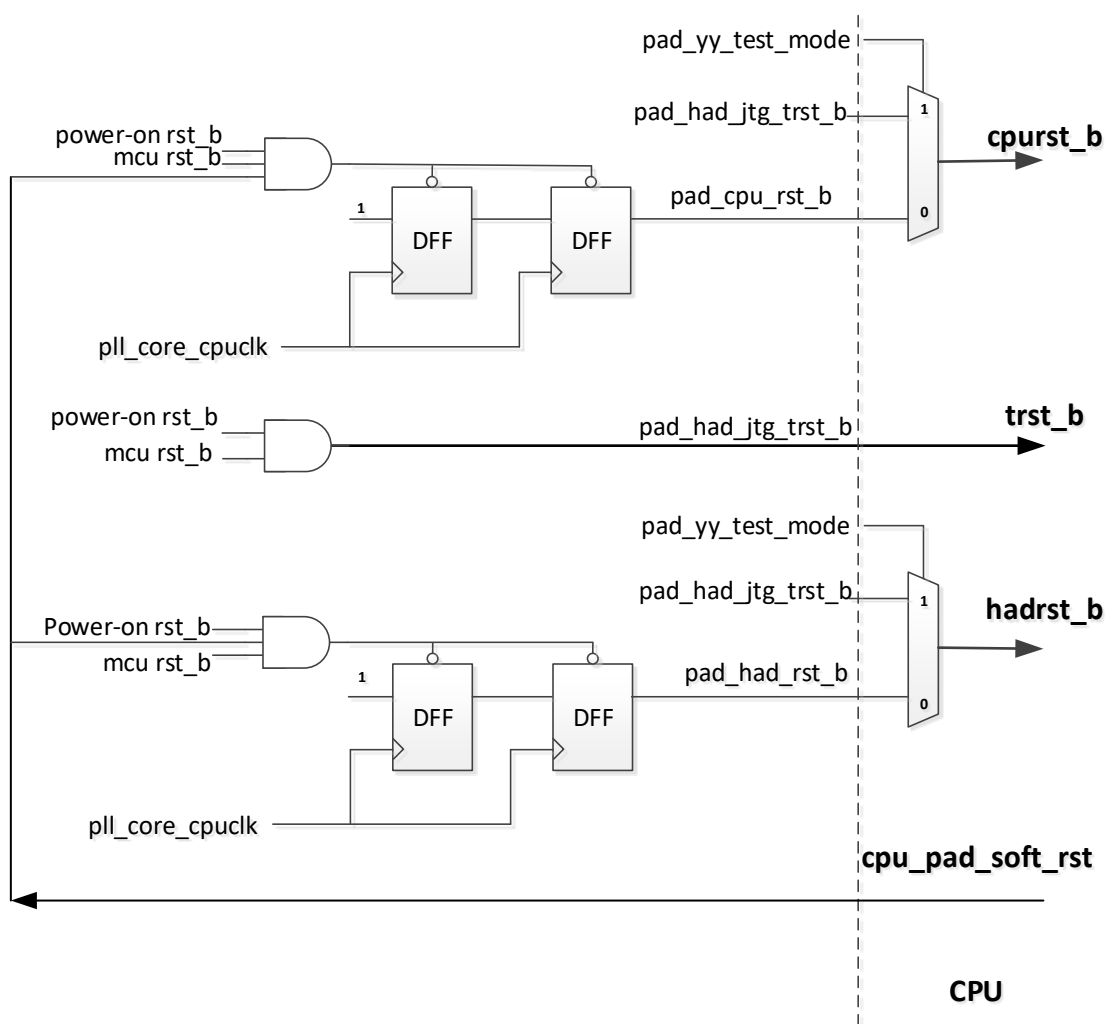
为了有效避免复位过程中亚稳态的出现，E902 采用异步复位、同步释放的方式进行系统的复位。因此需要各个复位信号同步到相对应的时钟域中。复位信号的同步工作由系统完成，CPU 自身仅对复位信号进行选择。下图给出了 E902 的复位信号的一个示例。

- ✧ **pad_cpu_rst_b**: 信号由系统上电复位，mcu 复位（通常为整个 mcu 的输入 port 端口）以及 nreset（在线仿真器发出的 cpu 复位信号，用户可以根据实际情况选择是否实现 nreset）经过两级工作在 pll_core_cpuclock 时钟下的同步寄存器得到。

- ✧ **pad_had_rst_b**: 信号由系统上电复位和 mcu 复位经过两级工作在 pll_core_cpuck 时钟下的同步寄存器得到。
- ✧ **pad_had_jtg_trst_b**: 由系统上电复位, mcu 复位以及 jtag 复位 (在线仿真器发出的 jtag 复位信号, 在 2 线 jtag 调试配置下用户可以根据实际情况选择是否实现该 jtag 复位, 因为状态机可以实现自复位) 得到, jtag 复位信号由在线仿真器完成时钟域同步工作; 另外, 在测试模式下, pad_had_jtag_trst_b 为 cpu 全局复位信号, 测试机通过上述 MCU 复位端口对复位信号进行控制。

另外, E902 还实现了同步复位的功能 (cpu_pad_srst), 软件可以通过设置相关控制寄存器来驱动 cpu_pad_srst 信号有效一个系统时钟周期, 系统设计人员可以使用该信号反过来驱动上述三个复位端口, 完成 E902 的复位操作。

另外, E902 支持通过集成信号指定复位启动地址, 通过 pad_cpu_rst_addr[31:0]传入, 该信号的低两位需要为 0。



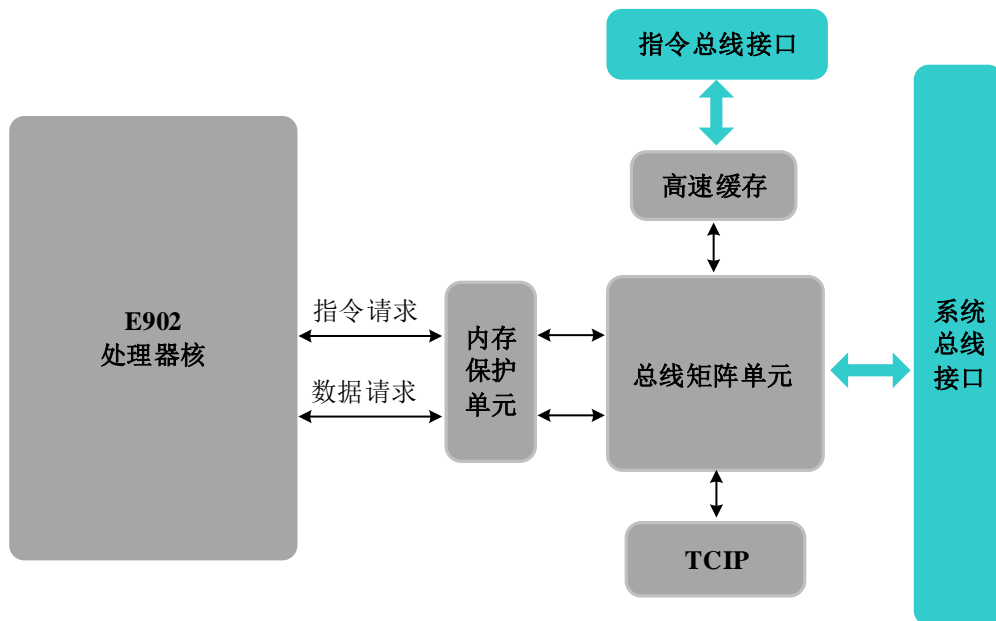
图表 3-7 E902 复位信号产生机制

4 总线系统集成

4.1 总线简介及配置信息

E902 实现了多总线接口，分别包括系统总线、指令总线和紧耦合 IP 接口。其中指令总线可由用户根据实际的系统需要进行配置。紧耦合 IP 接口的地址空间固定为 0xE0000000~0xEFFFFFFF，用于 clint、cllic 等 IP 的访问。当总线矩阵接收访问紧耦合 IP 的地址的请求时，该请求将直接在核内处理，不影响指令总线和系统总线。

总线矩阵为处理器内部请求访问外部总线接口提供了互联功能。总线矩阵与 CPU 内部请求及总线接口的连接关系如图表 4-1 所示。总线矩阵根据内存访问的地址仲裁总线接口类型，将处理器内部访问分发到系统总线、指令总线以及 TCIP 接口上。



图表 4-1 E902 总线矩阵

处理器内部的取指访问和数据访问拥有相同的总线访问权限，可以访问所有总线接口。为了解决同一时钟周期取指访问和数据访问竞争同一总线接口的问题，总线矩阵也负责请求的优先级判断。当取指请求和数据请求竞争同一总线接口时，数据请求拥有更高的优先级。

E902 多总线接口的基本信息如图表 4-2 多总线接口的基本信息和可配置性所示。

总线接口	总线协议	时序方式
------	------	------

系统总线	AHB-Lite	直接输出
指令总线	AHB-Lite	直接输出

图表 4-2 多总线接口的基本信息和可配置性

另外，E902 通过提供一组接口信号（pad_bmu_iahbl_base 和 pad_bmu_iahbl_mask），支持指令总线基地址和空间大小硬件集成时可配置。其中，pad_bmu_iahbl_base 指定了指令总线的基地址；pad_bmu_iahbl_mask 指定了不同地址空间下对地址对齐的需求。

指令总线的地址空间 1MB 到 4GB 可配置，例如设置指令总线地址空间大小为 8M，pad_bmu_iahbl_base[2:0]必须为 3'b0，pad_bmu_iahbl_mask[11:0]必须为 12'b1111 1111 1000。不同大小的地址空间具体要求见下表。

地址空间大小	对 pad_bmu_iahbl_base 的要求	对 pad_bmu_iahbl_mask 的要求
1MB	没有要求	bit[11:0]=12'b1111 1111 1111
2MB	bit[0] =0	bit[11:0]=12'b1111 1111 1110
4MB	bit[1:0] =2'b0	bit[11:0]=12'b1111 1111 1100
8MB	bit[2:0] =3'b0	bit[11:0]=12'b1111 1111 1000
16MB	bit[3:0] =4'b0	bit[11:0]=12'b1111 1111 0000
32MB	bit[4:0] =5'b0	bit[11:0]=12'b1111 1110 0000
64MB	bit[5:0] =6'b0	bit[11:0]=12'b1111 1100 0000
128MB	bit[6:0] =7'b0	bit[11:0]=12'b1111 1000 0000
256MB	bit[7:0] =8'b0	bit[11:0]=12'b1111 0000 0000
512MB	bit[8:0] =9'b0	bit[11:0]=12'b1110 0000 0000
1GB	bit[9:0] =10'b0	bit[11:0]=12'b1100 0000 0000
2GB	bit[10:0] =11'b0	bit[11:0]=12'b1000 0000 0000

4GB	bit[11:0] = 12'b0	bit[11:0] = 12'b0000 0000 0000
-----	-------------------	--------------------------------

图表 4-3 指令总线对基地址和地址对齐的要求

E902 指令总线地址空间在集成分配时需保证与紧耦合 IP 的地址空间不重叠, 否则在处理器内部请求地址命中该重叠空间时行为不可预期。

4.2 系统总线接口

E902 的系统总线接口支持 AMBA3.0 AHB-Lite 协议 (请参考 AMBA 3.0 规格说明—AMBA3 AHB-Lite Protocol Specification Rev 1.0)。考虑到 E902 的应用领域及成本, 系统总线接口只实现了 AHB-Lite 协议中的部分内容。

在 AHB-Lite 协议下, 作为主设备, 总线接口支持的传输类型为:

- ✧ HBURST 只支持 SINGLE 传输, 其它突发类型均不支持;
- ✧ HTRANS 只支持 IDLE 和 NONSEQ, 其它传输类型均不支持;
- ✧ HSIZE 支持字、字节和半字传输, 其它传输大小不支持;
- ✧ HWRITE 支持读和写操作。

在 AHB-Lite 协议下, 总线接口接受从设备的响应类型为:

- ✧ HREADY 支持 Ready 和 Not Ready;
- ✧ HRESP 支持 OKAY 和 ERROR, 其它响应类型不支持。

信号名	I/O	Reset	定义
系统总线接口信号:			
biu_pad_haddr[31:0]	O	-	地址总线: 32 位地址总线。
biu_pad_hwdata[31:0]	O	-	写数据总线: 32 位写数据总线。
biu_pad_hburst[2:0]	O	-	突发传输指示信号: 指示传输是一次突发传输的一部分: 000: SINGLE;

			001: INCR; 010: WRAP4。 E902 仅支持 SINGLE 传输。
biu_pad_hsize[2:0]	0	-	传输宽度指示信号: 指示传输的数据宽度: 000: byte; 001: halfword; 010: word。
biu_pad_htrans[1:0]	0	00	传输类型表示信号: 指示当前总线周期的传输类型: 00: IDLE; 01: BUSY; 10: NONSEQ; 11: SEQ。 E902 中仅支持 NONSEQ 和 IDLE 两种传输类型。
biu_pad_hwrite	0	0	读写表示信号: 指示当前 CPU 是读取总线数据还是写总线: 1: 指示是写总线传输; 0: 指示是读总线传输。
biu_pad_hprot[3:0]	0	-	保护控制信号: 指示当前总线周期进行的数据传输的特性: ***0: 取指令; ***1: 数据访问; **0*: 用户模式访问; **1*: 机器模式访问; *0**: Non bufferable;

			*1**：bufferable; 0***：Non cacheable; 1***：cacheable。
pad_biu_hrdata[31:0]	1	-	读数据总线： 32 位读数据总线。
pad_biu_hready	1	-	传输完成指示信号： 有效时指示当前传输已完成，CPU 将总线置于待命状态。
pad_biu_hresp	1	-	传输应答信号： 传输应答： 0：OKAY; 1：ERROR; E902 中仅支持 OKAY 和 ERROR 两种响应

图表 4-4 系统总线接口信号列表

4.3 指令总线接口

E902 的指令总线只支持 AMBA3.0 AHB-LITE 协议，可参考 AMBA 3.0 规格说明- AMBA3 AHB-Lite Protocol Specification Rev 1.0。

E902 指令总线接口只实现了 AHB-Lite 协议中的部分内容。在 AHB-Lite 协议下，作为主设备，总线接口支持的传输类型为：

- ✧ HBURST 只支持 SINGLE 传输，其它突发类型均不支持；
- ✧ HTRANS 只支持 IDLE 和 NONSEQ，其它传输类型均不支持；
- ✧ HSIZE 支持字、字节和半字传输，其它传输大小不支持；
- ✧ HWRITE 支持读和写操作。

在 AHB-Lite 协议下，总线接口接受从设备的响应类型为：

- ✧ HREADY 支持 Ready 和 Not Ready；
- ✧ HRESP 支持 OKAY 和 ERROR，其它响应类型不支持。

信号名	I/O	Reset	定义
指令 AHB-Lite 接口			
iahbl_pad_haddr[31:0]	O	-	地址总线： 32 位地址总线。
iahbl_pad_hburst[2:0]	O	-	突发传输指示信号： 指示传输是一次突发传输的一部分： 000: SINGLE; 001: INCR; 010: WRAP4。 E902 仅支持 SINGLE 传输。
iahbl_pad_hprot[3:0]	O	-	保护控制信号： 指示当前总线周期进行的数据传输的特性： ***0: 取指令; ***1: 数据访问; **0*: 用户模式访问; **1*: 机器模式访问; *0**: Non bufferable; *1**: bufferable; 0***: Non cacheable; 1***: cacheable。
iahbl_pad_hsize[2:0]	O	-	传输宽度指示信号： 指示传输的数据宽度： 000: byte; 001: halfword; 010: word。
iahbl_pad_htrans[1:0]	O	00	传输类型表示信号： 指示当前总线周期的传输类型：

			00: IDLE; 01: BUSY; 10: NONSEQ; 11: SEQ。 E902 仅支持 IDLE 和 NONSEQ 两种传输类型。
iahbl_pad_hwddata[31:0]	O	-	写数据总线: 32 位写数据总线。
iahbl_pad_hwrite	O	0	读写表示信号: 指示当前 CPU 是读取总线数据还是写总线: 1: 指示是写总线传输; 0: 指示是读总线传输。
pad_iahbl_hready	I	-	传输完成指示信号: 有效时指示当前传输已完成, CPU 将总线置于待命状态。
pad_iahbl_hresp	I	-	传输应答信号: 传输应答: 0: OKAY; 1: ERROR。 E902 中仅支持 OKAY 和 ERROR 两种响应
pad_bmu_iahbl_base[11:0]	I	-	IAHB-Lite 基址控制信号, 上电复位之后需固定
pad_bmu_iahbl_mask[11:0]	I	-	IAHB-Lite 地址对齐控制信号, 上电复位之后需固定

图表 4-5 指令总线接口信号列表

5 中断系统集成

5.1 中断处理过程简述

中断处理是指处理器在接受到中断请求后从正常的程序处理转而响应中断处理，执行特定的中断处理程序。被中断的指令将正常退休，并在退休时响应中断请求。CPU 会保存当前的指令运行状态，将下一条指令作为中断返回的指令入口，并在退出中断服务程序时恢复之前的状态。

5.2 端口列表

中断控制器的接口信号主要用于 CLIC 接收并采样中断源的中断请求信号。

信号名	I/O	Reset	定义
中断源信号：			
pad_clic_int_vld[i-1:0]* i: 1-240	I	0	中断请求信号。 高电平时表示中断请求有效，bit[0]对应的中断号为 16，bit[i]对应的中断号为 16+i，i 最大为 240。
pad_cpu_ext_int_b	I	1	CLINT 11 号中断请求信号，为了在 CLIC 配置时兼容 CLINT 中断而存在。 低电平时表示中断请求有效。
pad_cpu_nmi	I	0	不可屏蔽中断请求，脉冲信号，上升沿有效。
时钟信号：			
pll_core_cpucclk	I	-	CLIC 的工作及采样时钟

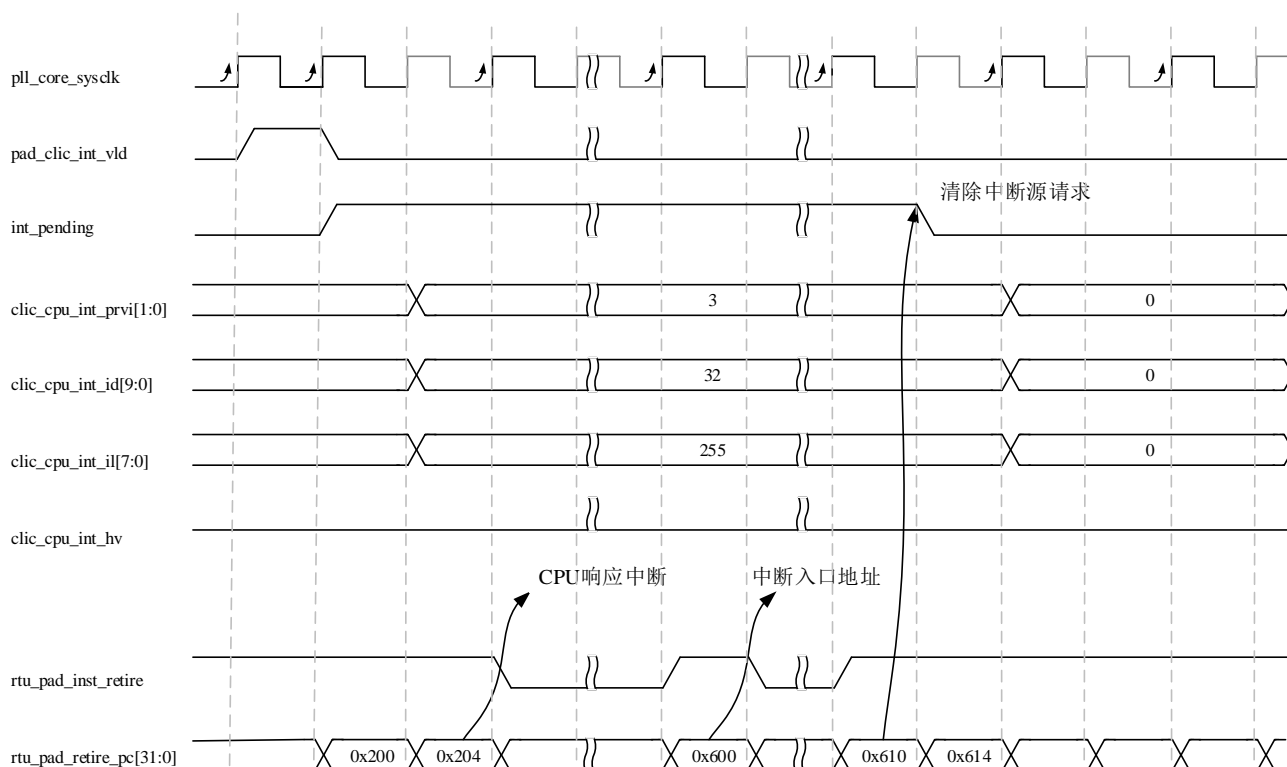
图表 5-1 CLIC 中断控制器接口信号

注意：CLIC 低 16 号中断中除 3 号、7 号及 11 号外，其余均为保留中断，中断有效信号均硬件绑 0；3 号中断为软件中断，需要通过写对应 CLINT 寄存器置位；7 号中断为 Timer 中断，其中断有效信号由 CLINT 中控制寄存器与外部计数器（pad_cpu_sys_cnt）的比较结果产生；11 号中断为外部中断，直接与 CLINT 中机器模式外部中断相连，对应中断源为 pad_cpu_ext_int_b。

5.3 中断握手时序图

当 CPU 配置 CLIC 中断控制器且处于 CLIC 模式时，由 CLIC 负责采样外部中断源的中断请求：对于电平中断，CLIC 中断控制器采样到中断有效信号的高电平后设置对应中断进入等待状态，电平中断要求中断服务程序中清除外设的中断源有效信号，否则当中断退出时会重新发起中断请求。外设可以根据这一特点，常置中断信号直到不再需要中断处理程序处理；对于脉冲中断，CLIC 中断控制器采样中断有效信号的上升沿，设置对应中断进入等待状态。然后根据中断的优先级向 CPU 发送中断请求，在 CPU 响应该脉冲中断请求前，若脉冲中断源向 CLIC 中断控制器发起多次中断请求，CLIC 中断控制器只会记录一次中断请求。在 CPU 响应该脉冲中断请求后，若脉冲中断源再次向中断控制器发起请求，CLIC 中断控制器会再次触发对应中断进入等待状态，该处于等待状态的中断请求在中断退出后才能够再次被 CPU 响应。脉冲中断和电平中断的区分由 CLIC 每个中断源中的 clicintattr 控制寄存器的 trig 域决定，当 trig[0]为 0 时，代表为电平中断。当 trig[0]为 1 时，trig[1]为 0 代表上升沿中断，trig[1]为 1 代表下降沿中断。

图表 5-2 给出了 CPU 内核与 CLIC 中断交互时序图。当 CLIC 采样到中断信号时，CLIC 设置中断等待状态位 int_pending，CLIC 根据中断优先级进行仲裁，向 CPU 发出中断的相关信息 (clic_cpu_int_priv、clic_cpu_int_il、clic_cpu_int_id、clic_cpu_int_hv)；CPU 响应中断进入中断服务程序。注意：CPU 在中断响应和中断退出时的行为兼容 RISC-V 中 CLIC 控制器的定义。对于电平中断，CPU 不会向 CLIC 中断控制器返回任何响应和退出信号，需要软件在退出中断服务程序前清除外部 IP 的中断源有效信息；对于 Vector 模式的脉冲中断，CPU 在响应中断以后，会清除对应的中断等待状态位；对于非 Vector 模式的脉冲中断，CPU 在执行有效的读 mnxti 寄存器操作后，会清除对应的中断等待位。



图表 5-2 CPU 配置 CLIC 时中断相关信号时序简图

5.4 中断嵌套

配备了 CLIC 中断控制器的 CPU 支持中断嵌套功能，CPU 记录当前所处中断等级，并根据等待响应中断的中断等级判定新来的中断是否要能打断当前正在处理的中断，从而提高中断响应实时性。具体参考《玄铁 E902 R2S2 用户手册》。

5.5 计时器中断

CLINT 可用于生成机器模式计时器中断。该计时器中断需要搭配 E902 外部由系统设计实现的 64 位计数器使用。该计数器需要工作在 always-on 的电压域，复位后在每个时钟周期进行计数。该计数器软件不可写，仅能通过复位清零。在 E902 集成时需要将该 64 位系统计数器的值通过 pad_cpu_sys_cnt[63:0] 信号传入 E902 内部，高 32 位的值可通过 E902 设计实现的 MTIMEHI[31:0] 寄存器读取，低 32 位的值可通过 MTIMELO[31:0] 寄存器读取。pad_cpu_sys_cnt[63:0] 在 E902 内部会由 clk_en 进行采样，系统设计人员需要将该信号在 E902 外部完成同步到 pll_core_cpuclock 时钟域的操作。

6 调试系统集成

6.1 端口列表

信号名	I/O	Reset	描述
pad_had_jtg_trst_b	I	-	JTAG 测试复位信号： JTAG 复位信号，下跳变可以初始化和 JTAG 接口相关的触发器，正常工作情况下该信号需置成高电平。
pad_had_jtg_tclk	I	-	JTAG 测试时钟信号： JTAG 和 HAD 内部相关触发器的时钟信号，要求其频率是 cpuclock 频率的一半以下。
pad_had_jtg_tms_i	I	-	JTAG 数据输入信号： JTAG 串行输入端口，包括控制命令，数据，输入顺序由低位到高位。 注：此信号出现于实现 2 线制 HAD 的处理器中。
had_pad_jtg_tms_o	O	-	JTAG 数据输出信号： JTAG 串行数据输出端口，输出顺序由低位到高位。 注：此信号出现于实现 2 线制 HAD 的处理器中。
had_pad_jtg_tms_oe	O	-	JTAG 数据输出有效信号： 注：此信号出现于实现 2 线制 HAD 的处理器中。
调试支持信号：			
pad_sysio_dbgrq_b	I	-	外部调试请求信号： 可用于使 CPU 进入调试模式或使 CPU 从仅关

			闭时钟的低功耗状态唤醒。 不用该信号时，需要接 1。
had_pad_jdb_pm[1:0]	0	00	处理器工作模式指示信号： 这些输出信号表明处理器的工作模式,异步于 pad_had_jtg_tclk。 00: normal 模式; 01: 低功耗模式; 10: 调试模式; 11: 保留。

图表 6-1 debug 端口列表

6.2 JTAG 接口

pad_had_jtg_tap_en 信号为调试接口中 TAP 状态机的使能信号，维持该信号为高电平至少一个 JTAG 时钟周期可以使能调试接口中的 TAP 状态机。如果该信号在 CPU 上电之后一直无效，那么使用同步方式（如设置调试接口寄存器 HCR 中的 DR 位，断点等）使 CPU 进入调试状态时可能无法调试程序。

在 TAP 状态机启动之后，had_pad_jtg_tap_on 信号将拉高。

pad_had_jtg_tclk 信号为 JTAG 时钟信号。该信号为外部输入信号，一般为调试器产生的时钟信号，要保证该时钟信号的频率低于 CPU 时钟信号的频率 1/2 才能保证调试模块与 CPU 核之间的正常工作。

pad_had_jtg_trst_b 信号为 JTAG 复位信号，可以复位 TAP 状态机以及其他相关控制信号。

pad_had_jtg_tms_i 信号为 2 线制 JTAG 串行数据输入信号，调试接口在 JTAG 时钟信号 TCLK 的上升沿对其采样，而外部调试器在 JTAG 时钟的下降沿设置该信号；

该信号在空闲时必须保持为高电平，同时空闲时时钟信号需要停止翻转。用户可以利用该信号同步复位 HAD 逻辑：在实现 2 线制 JTAG 接口的调试模块中，如果 TCLK 时钟信号一直有效，用户只需保持该信号为高电平状态并维持 80 个时钟周期即可同步复位 JTAG 接口控制逻辑，即调试模块的 TAP 状态机回到 RESET 态，同时调试模块寄存器 HACR 复位到 0x82（指向 ID 寄存器）。

had_pad_jtg_tms_o 信号为 2 线制 JTAG 串行数据输出信号，调试接口在 JTAG 时钟信号 TCLK 的下降沿对其设置，而外部调试器在 JTAG 时钟的上升沿对其采样；

had_pad_jtg_tms_oe 信号为 had_pad_jtg_tms_o 信号有效指示信号。CPU 外部应利用该信号将 pad_had_jtg_tms_i 和 had_pad_jtg_tms_o 信号合为一个双向端口信号。

7 低功耗系统集成

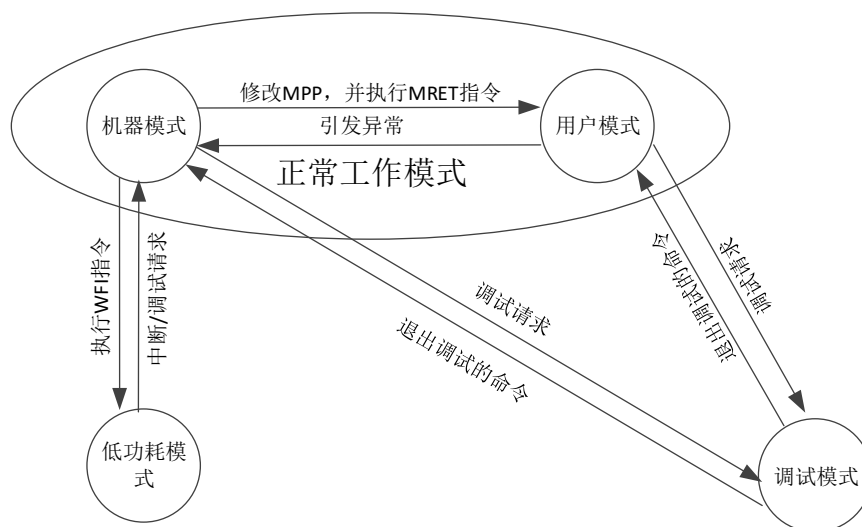
7.1 端口列表

信号名	I/O	Reset	功能描述
sysio_pad_lpm�_b[1:0]	O	11	低功耗模式状态信号： 当处理器执行 WFI 指令时，sysio_pad_lpm�_b[1:0]被相应的改变： 00：深度睡眠； 01：浅度睡眠； 10：保留； 11：正常工作模式。
pad_cpu_wakeup_event	I	-	事件唤醒请求，脉冲信号，上升沿有效。该信号需要在 CPU 外面完成到 CPU 时钟域的同步

图表 7-1 低功耗端口信号列表

7.2 工作模式及其转换

E902 总共有三类工作模式：正常运行模式、低功耗工作模式和调试模式，如图表 7-2。若支持用户模式，则正常运行模式包括机器模式和用户模式，否则只存在机器模式。

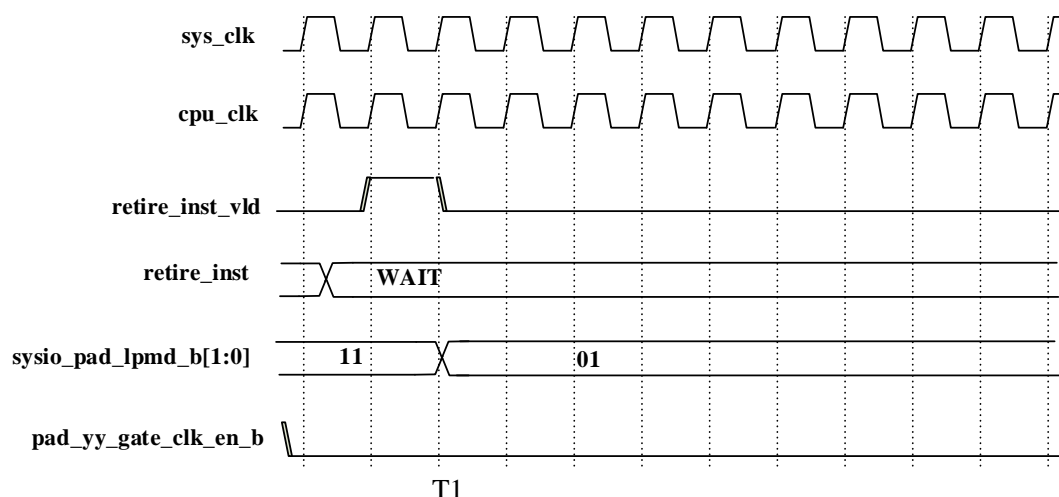


图表 7-2 CPU 状态转换图

7.3 进入低功耗模式握手

CPU 可通过执行低功耗指令（WFI）进入低功耗模式。CPU 执行低功耗指令之后，会关闭内部绝大部分寄存器时钟，除开少量始终采样的寄存器，CPU 内部的流水线停止运行，不再取指令和执行指令，同时还会修改 sysio_pad_lpmdb[1:0]，通知 SoC 进行低功耗操作，之后就一直处于等待状态直到被唤醒。要进入低功耗模式就需要对门控时钟进行全局使能，即 pad_yy_gate_clk_en_b 设置为 0。

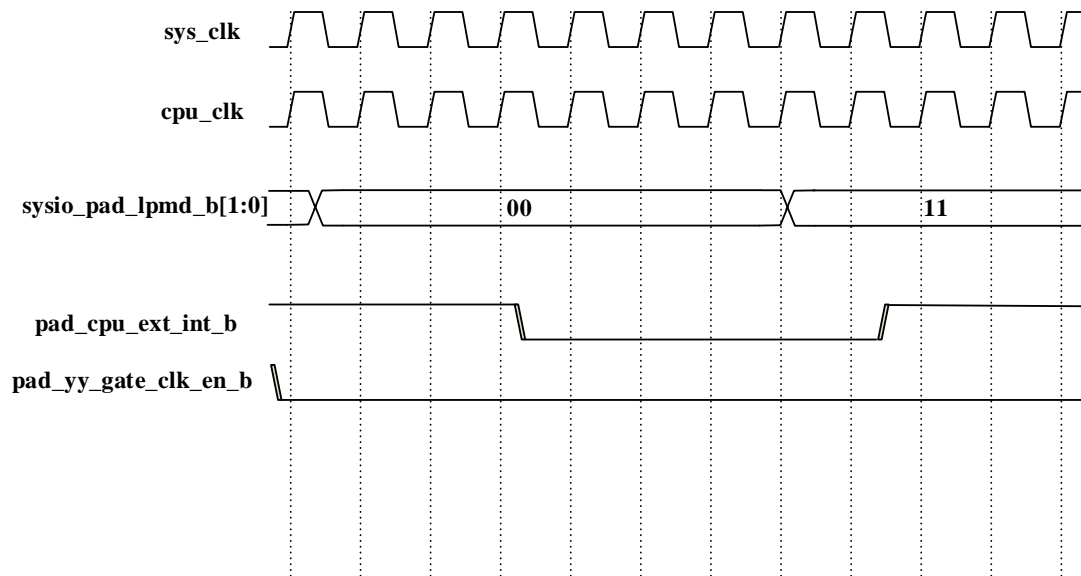
SoC 在集成时需要把 sysio_pad_lpmdb[1:0]作为切换到低功耗模式的控制信号，由它来指示 SoC 进入低功耗场景。



图表 7-3 进入低功耗模式握手

7.4 退出低功耗模式握手

退出低功耗模式由系统中断实现。当处理器接受到中断，且局部中断使能时，处理器会被唤醒。处理器被唤醒后，如果全局中断使能打开，则进入中断服务程序开始执行，否则，唤醒之后 CPU 继续执行低功耗指令（WFI）之后的指令。外部模块可以通过查询 sysio_pad_lpmdb[1:0]来查询 CPU 状态。图表 7-4 所示为通过 SoC 驱动 11 号中断的来源 pad_cpu_ext_int_b 来唤醒 CPU，亦可通过驱动 pad_clic_int_vld[i-1:0]这组信号集成连接的中断来唤醒 CPU。NMI 和调试请求，外部事件请求也可用来唤醒 CPU。



图表 7-4 退出低功耗模式握手

8 CPU 运行观测信号集成

8.1 简介

CPU 运行观测信号是提供给 SoC 集成设计人员观测所设计，集成时可悬空处理。SoC 设计人员可以通过监测这些信号来获知 CPU 的指令回写信息和寄存器的相关值。

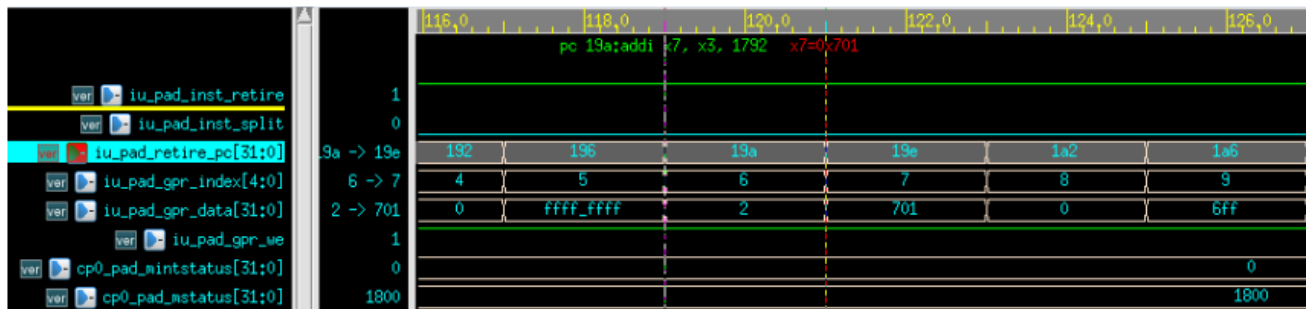
8.2 通用观测信号

信号名	I/O	Reset	定义
iu_pad_inst_retire	O	0	指令退休指示信号： 0：当前周期没有指令退休； 1：当前周期有指令退休。
iu_pad_retire_pc[31:0]	O	-	退休指令的 PC：表明当前正在退休的指令的 PC。
iu_pad_inst_split	O	0	指令类型指示信号： 0：当前正在退休的指令不是拆分指令； 1：当前正在退休的指令是拆分指令。
iu_pad_gpr_we	O	0	通用寄存器回写指示信号： 0：当前周期不回写通用寄存器； 1：当前周期回写通用寄存器。
iu_pad_gpr_index[4:0]	O	-	寄存器索引： 表示当前回写的通用寄存器的索引。
iu_pad_gpr_data[31:0]	O	-	寄存器回写总线： 表示当前回写通用寄存器的值。
cp0_pad_mstatus[31:0]	O	-	表示当前 mstatus 值。
cp0_pad_mintstatus[31:0]	O	-	表示处理器当前中断等级。

Cp0_pad_mcause[31:0]	O	-	表示当前 MCAUSE 寄存器的值。
cpu_pad_lockup	O	0	当 CPU 处于 lockup 状态时会将该信号置为高电平。一旦触发 CPU 锁定状态，该信号一直为 1，当 CPU 进入调试模式或者响应 nmi 请求将该状态打断时，该信号为低电平。

图表 8-1 通用观测信号

8.3 观测信号示例波形



图表 8-2 观测信号示例波形

如图表 8-2 所示，左侧光标所指位置 `iu_pad_retire_pc[31:0]`，当前退休 pc 为 0x19a，其指令为 `addi x7, x3, 1792`，其计算结果为 x7 的值为 0x701，在右侧光标位置可看到 `iu_pad_gpr_index[4:0]` 为 7，表示回写目标寄存器标号为 7，回写值 `iu_pad_gpr_data[31:0]` 为 0x701。由于 E902 指令退休位于流水线 EX 级，而寄存器回写操作位于 EX 级的后一个时钟周期，所以从观察到退休至观察到回写值，差了一级流水线。

9 地址空间属性设置

玄铁 E902 支持两种内存属性的配置，分别为是否可高缓（cacheable 或者 non-cacheable）的内存以及是否可暂存的内存（bufferable 或者 non-bufferable）。对于 cacheable 的内存地址空间，E902 从该区域访问得到的数据可以缓存在 E902 内部的指令或者数据 cache 中。而对于 bufferable 的内存地址空间，E902 在对该地址空间进行写操作时，总线写响应（HRESP）可由总线通路的任一节点返回，而不必等待 E902 访问的 slave 上写操作完成后再返回。反之，对于 non-bufferable 的内存地址空间，E902 对该地址空间的写操作的响应必须在 E902 访问的 slave 上写操作完成后由该 slave 返回给 E902。

全空间内存属性的配置在平头哥自定义的 sysmap.h 文件完成。sysmap.h 文件会随着 E902 代码一同交付给客户，客户需要在系统设计阶段完成对内存空间属性的设置，并将该文件放入项目文件列表中，一经设定，软件无法修改。

sysmap.h 支持对 8 个内存地址空间的属性设定，第 i (i 从 0 到 7) 个地址空间地址上限（不包含）由宏 `SYSMAP_BASE_ADDR i` (i 从 0 到 7) 定义，地址下限（包含）由 `SYSMAP_BASE_ADDR $i-1$` 定义，具体为 `SYSMAP_BASE_ADDR $i-1$ ≤ 第 i 个地址空间地址 < SYSMAP_BASE_ADDR i` 。第 0 个地址空间下限是 0x0，内存地址不在 sysmap.h 文件设定的 8 个地址区间的地址属性默认为 cacheable 和 bufferable。每个地址空间上下边界是 4KiB 对齐，因此宏 `SYSMAP_BASE_ADDR i` 定义的是地址的高 20 位。

落在第 i (i 从 0 到 7) 个地址空间内的地址的属性由宏 `SYSMAP_FLAG i` (i 从 0 到 7) 定义，具体如图表 9-1 所示。

4	3	2	1	0
Reserved	cacheable	bufferable	Reserved	

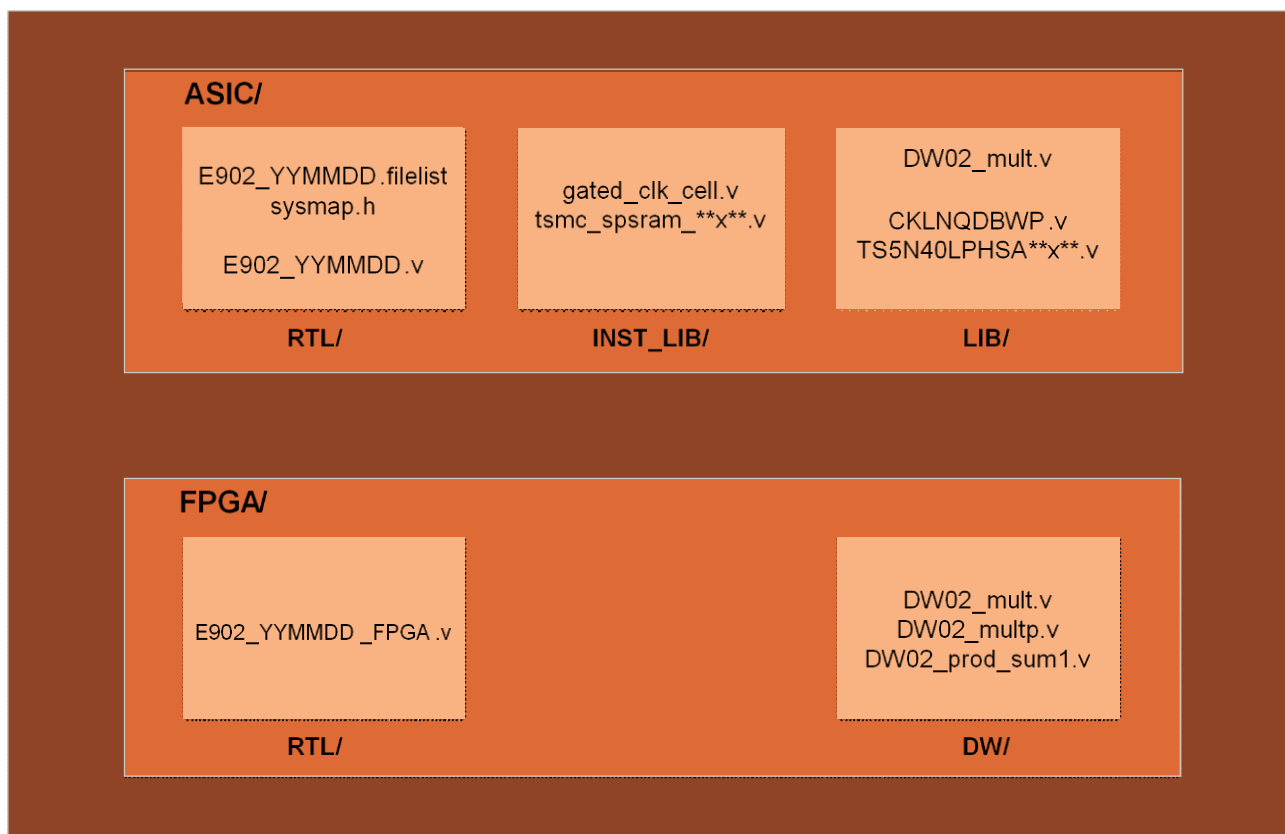
图表 9-1 sysmap.h 地址属性

需要说明的是，该文件仅在硬件配置了 CACHE 时存在。

10 工艺映射

10.1 软核授权 RTL 代码结构

平头哥软核授权客户都可以得到如下文件夹，内部结构如下：



图表 10-1 Release 文件夹内容

ASIC/RTL 目录：包含了内核代码 E902_YMMDD.v（YMMDD 表示日期，比如 20191212），地址空间属性配置文件 sysmap.h，E902 硬件配置定义文件 cpu_cfg.h 以及一个 file list 文件 E902_YMMDD.filelist，里面包含所有的.v 和.h 文件，综合时需要把该文件列表加载进去。

ASIC/INST_LIB 目录：所有 memory 例化文件和 gated cell 例化文件。

ASIC/LIB 目录：所有的 memory，gated cell 和 design ware 仿真模型。

FPGA/DW 目录：FPGA 平台下仿真的 design ware 文件。

FPGA/RTL 目录：FPGA 平台下 E902 可综合文件 E902_YMMDD_FPGA.v，包含了 CPU，替换后时钟

常开的 gated_cell，以及可 FPGA 平台综合的 memory 模型。

10.2 ASIC 映射

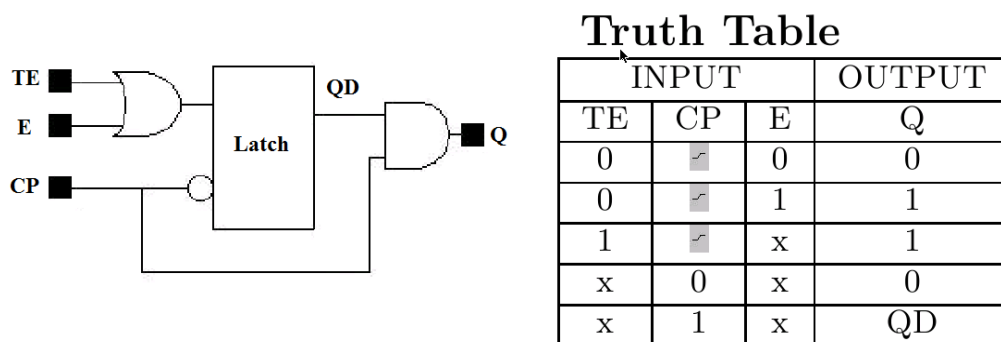
10.2.1 ICG 替换

10.2.1.1 文件所在位置

所需要修改的文件位于 INST_LIB 目录, 文件名为 gated_clk_cell.v, 该文件里例化了特定工艺下的 gated cell。

10.2.1.2 选择 gated cell

平头哥所用的 gated cell 都是上升沿有效，其逻辑图和真值表如下：



图表 10-2 正沿触发 gated cell

一般工艺库都会有两类 gated cell，一个是正沿触发而另一类是负沿触发的，注意需要选择和以上真值表一样的正沿的 gated cell。

一般工艺库里同一类有不同规格的 gated cell，这就需要用户根据自己的需求进行选择，平衡速度和功耗等因素，如下图是 TSMC40 工艺下正沿触发 gated cell 的规格：

Cell Information

(Characterization Condition: Process=Slow-Slow, Voltage=0.99v, Temp=125degreeC)
PG Pin=VDD

Cell Name	Gate Count	Width(um)	Leakage(nW)		
			Min.	Ave.	Max.
CKLNQD12BWP	12	5.04	5.3373	6.13635	6.71089
CKLNQD16BWP	15	6.3	7.94464	8.53875	9.02614
CKLNQD1BWP	5.5	2.66	1.57111	1.89138	2.45396
CKLNQD20BWP	18	7.28	8.58731	9.84222	10.74234
CKLNQD24BWP	20	7.98	9.50224	11.18802	12.29832
CKLNQD2BWP	6	2.8	2.06619	2.29296	2.62804
CKLNQD3BWP	6.5	3.08	2.50378	2.70117	2.92192
CKLNQD4BWP	7	3.22	2.71468	3.03671	3.26009
CKLNQD6BWP	9	3.92	3.91617	4.19528	4.50158
CKLNQD8BWP	10	4.34	4.35861	4.89274	5.3282

图表 10-3 TSMC40 工艺下正沿触发 gated cell 的规格表

10.2.1.3 端口名字

gated_clk_cell.v 文件中例化了具体工艺库的 gated cell，如下图所示。

```
CKLNQD1BWP x_gated_clk_cell(
    .CP(clk_in),
    .TE(SE),
    .E(clk_en_bf_latch),
    .Q(clk_out)
);
```

图表 10-4 例化 gated cell

信号线名	功能	连接
clk_in	Clock 输入	Clock input
SE	pad_yy_gated_clk_en_b	Test clock enable
clk_en_bf_latch	Clock function 使能信号	Function Clock enable
clk_out	Clock 输出	Clock output

图表 10-5 gated cell 信号列表

10.2.1.4 更改 gated_clk_cell.v 文件

选定 gated cell 之后修改 gated_clk_cell.v 文件，如图表 10-6 所示。

```
5 //CKLNQD1BWP x_gated_clk_cell(  
6 //      .CP(clk_in),  
7 //      .TE(SE),  
8 //      .E(clk_en_bf_latch),  
9 //      .Q(clk_out)  
10 //);  
11  
12  
13 HVT_CLKLANQH8V8 x_gated_clk_cell(  
14      .CP(clk_in),  
15      .TE(SE),  
16      .E(clk_en_bf_latch),  
17      .Q(clk_out)  
18 );
```

图表 10-6 修改例化文件

将原 gated cell 替换掉，接上新选的 gated cell。

10.2.1.5 更新 filelist

gated cell 替换完后需要将新的 gated cell 仿真模型文件准备好，可以放入 LIB 文件里也可以放到用户自己的文件里，重要的是需要修改 filelist，删除之前调用的 model 加入用来替换的文件路径。

10.2.1.6 不使用前端插的 gated cell

后端工具支持自动插 gated cell，如果客户不希望使用平头哥前端手动插的 gated cell，可以简单注释掉 gated_clk_cell.v 里的 instance gated cell 部分代码然后加上一行代码，如下图所示。

```
5 //CKLNQD1BWP x_gated_clk_cell(  
6 //      .CP(clk_in),  
7 //      .TE(SE),  
8 //      .E(clk_en_bf_latch),  
9 //      .Q(clk_out)  
10 //);  
11  
12 assign clk_out = clk_in;
```

图表 10-7 不使用前端插入的 gated cell

这样前端插的 gated cell 都没了，客户可以按照自己的意愿插入 gated cell。

10.2.2 Memory 替换

在 ASIC/RTL/E902_xxx.filelist 文件中包含了全部指令 Cache 和数据 Cache 大小所需要的 memory 类型，用户可将该文件列表中所有的 memory 文件进行替换，以便对 Cache 大小进行定义时的使用。

10.2.2.1 文件所在位置

所有需要修改的文件都在 INST_LIB 目录，且文件名都包含关键词 spsram_AAAxBBB，A、B 分别代表 memory 的深度和宽度。

10.2.2.2 生成 memory

所有需要的 memory 信号都可以从 INST_LIB 里看出 (AAAxBBB)，在生成时我们只对一个选项有要求，就是所有的 memory 都需要支持位写使能信号。其他的选项用户可以根据自己的需求选择，包括 memory 的形状、时序、面积、功耗等。如果有些 memory 过大或者过深，可以自行进行拼接 (详见 10.2.2.6)。

10.2.2.3 端口名字

该文件例化了具体工艺的 memory 库 model，如下图所示。

```
TS5N40LPHSA128X42M2S x_pa_spsram_tsmc40_128x42(  
  .A      (A),  
  .D      (D),  
  .BWEB   (WEN),  
  .WEB    (GWEN),  
  .CEB    (CEN),  
  .CLK    (CLK),  
  .Q      (Q),  
  .PD     (1'b0),  
  .RTSEL  (1'b0),  
  .TSEL   (2'b01),  
  .TURBO  (1'b1)  
);
```

图表 10-8 例化 memory

信号线名	功能	连接
A	地址线	RAM 的地址线端口。
D	写入数据线	RAM 的数据输入端。
BWEN*	低电平有效，位写使能信号	RAM 位写使能信号端。

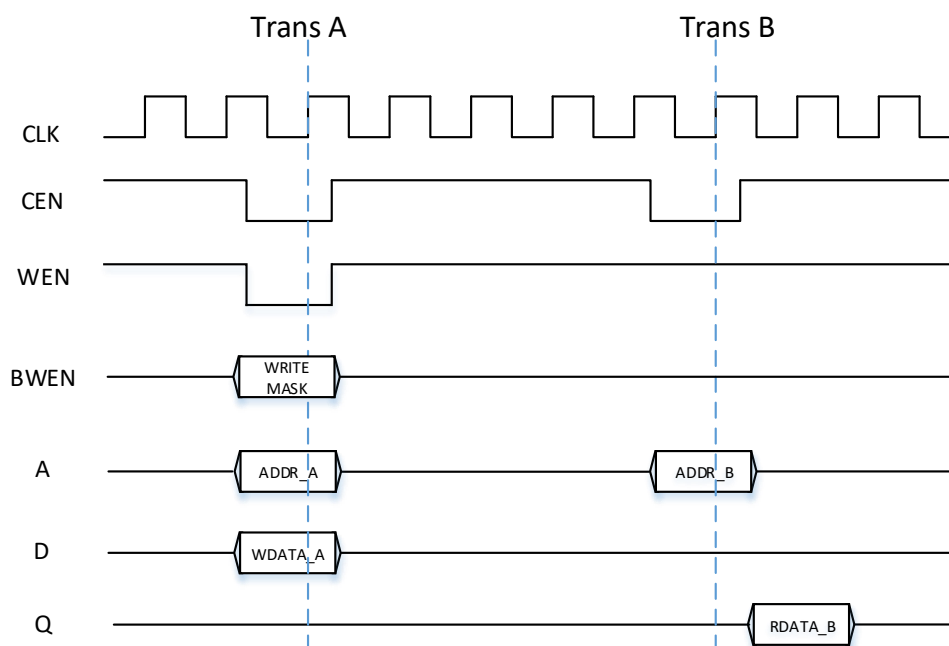
WEB*	低电平有效，写使能信号	RAM 写使能信号端。
CEN*	低电平有效，位片选信号	RAM 使能信号。
CLK	时钟	RAM 时钟端。
Q	输出数据线	RAM 的输出。
PD	高电平有效，Power Down 信号	RAM Power Down Mode 信号： 默认值：1'b0。
RTSEL	Debug Timing 设置端口	RAM Timing 设置端： 默认值：1'b0。
TSEL	Debug Read Timing 设置端口	RAM Timing 设置端： 默认值：2'b01。
TURBO	Debug Read Timing 设置端口	RAM Timing 设置端： 默认值：1'b1。

图表 10-9 memory 信号列表

*这三根信号都是低电平有效，具体要看 RAM 的相应信号是否也是低电平有效，另外，不同 vender 提供 ram 可能还有一些其他的控制 port，针对这些 port 用户需要根据自己的需要按照 RAM vender 的用户手册进行处理。

10.2.2.4 RAM 的读写时序

图表 10-10 是一个 RAM 的读写时序图，Trans A 是一个写请求，Trans B 是一个读请求。写部分数据是 BWEN 控制的，参考图表 10-8 例化 memory 的具体连接方法。Trans B 时钟上升沿采到到输入读请求，下一个 cycle 将数据输出到 Q 端。



图表 10-10 RAM 的读写时序图

10.2.2.5 修改 memory instance 文件

修改 memory，将 RAM 库文件替换掉，图表 10-11 就将上一节中的 memory 换成了另一种 tsmc Compiler 生成的 memory，多出的管脚参考 vender 提供的手册后 tie 上具体的值。再将原来 instance memory 的代码注释或者删除。

```
// TS5N40LPHSA128X42M2S x_pa_spsram_tsmc40_128x42(
// .A      (A),
// .D      (D),
// .BWEB    (WEN),
// .WEB     (GWEN),
// .CEB     (CEN),
// .CLK     (CLK),
// .Q       (Q),
// .PD      (1'b0),
// .RTSEL   (1'b0),
// .TSEL    (2'b01),
// .TURBO   (1'b1)
// );

TS1N40LPHSB128x42M4F x_pa_spsram_tsmc40_128x42(
    .A      (A),
    .D      (D),
    .BWEB    (WEN),
    .WEB     (GWEN),
    .CEB     (CEN),
    .CLK     (CLK),
    .Q       (Q),
    .PD      (1'b0)
);
```

图表 10-11 修改 memory 例化文件

10.2.2.6 拼接 memory

如果有些 memory 规格生成不出来，或者客户对该规格 memory 的形状或者其他一些特性不满意，用户可以用更小的 memory 拼接出需要的 memory，比如一块 2048x32 的 memory：

```
TS1N40LPHSB2048X32M4F x_pa_spsram_tsmc40_2048x32(
    .A      (A),
    .D      (D),
    .BWEB    (WEN),
    .WEB     (GWEN),
    .CEB     (CEN),
    .CLK     (CLK),
    .Q       (Q),
    .PD      (1'b0)
);
```

图表 10-12 需要拼接的 memory

某个工艺下并不支持该规格的 memory，就需要用其他规格 memory 进行拼接，下面是用两块 1024x32 的 memory 进行拼接的例子，当然也可以 2048x16 的拼接，那是另外一种接法。

```

assign CEN0 = CEN | A[ADDR_WIDTH-1];
assign CEN1 = CEN | ~A[ADDR_WIDTH-1];

always@(posedge CLK)
begin
    if(!CEN)
    begin
        bank_sel <= A[ADDR_WIDTH-1];
    end
    else
    begin
        bank_sel <= bank_sel;
    end
end
assign Q[31:0] = bank_sel ? Q1[31:0] : Q0[31:0];

    TS1N40LPHSB1024X32M4F x_pa_spsram_tsmc40_2048x32_bank0(
        .A      (A),
        .D      (D),
        .BWEB    (WEN),
        .WEB     (GWEN),
        .CEB     (CEN0),
        .CLK     (CLK),
        .Q       (Q0),
        .PD      (1'b0)
    );

    TS1N40LPHSB1024X32M4F x_pa_spsram_tsmc40_2048x32_bank1(
        .A      (A),
        .D      (D),
        .BWEB    (WEN),
        .WEB     (GWEN),
        .CEB     (CEN1),
        .CLK     (CLK),
        .Q       (Q1),
        .PD      (1'b0)
    );

```

图表 10-13 拼接 memory

10.2.2.7 修改 filelist

Memory 替换完后需要将，新的 memory 仿真模型文件和库文件都准备好，可以放入 LIB 文件里也可以放到用户自己的文件里，重要的是需要修改 filelist，删除以前调用的 mode 加入现在调用的文件路径。

10.2.3 DesignWare IP

10.2.3.1 简介

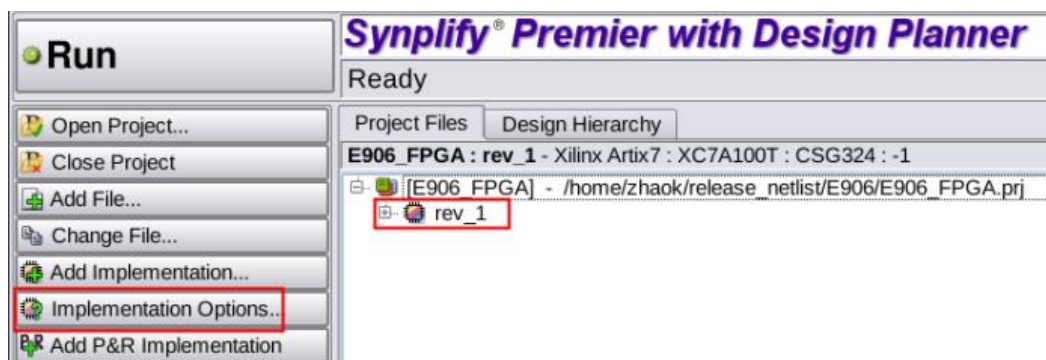
有些配置下的 E902 里使用了 Desingn ware IP，指代需要用到的库里的加法器乘法器或是乘法累加器，与工艺无关，用户不需要替换。

10.3 FPGA 映射

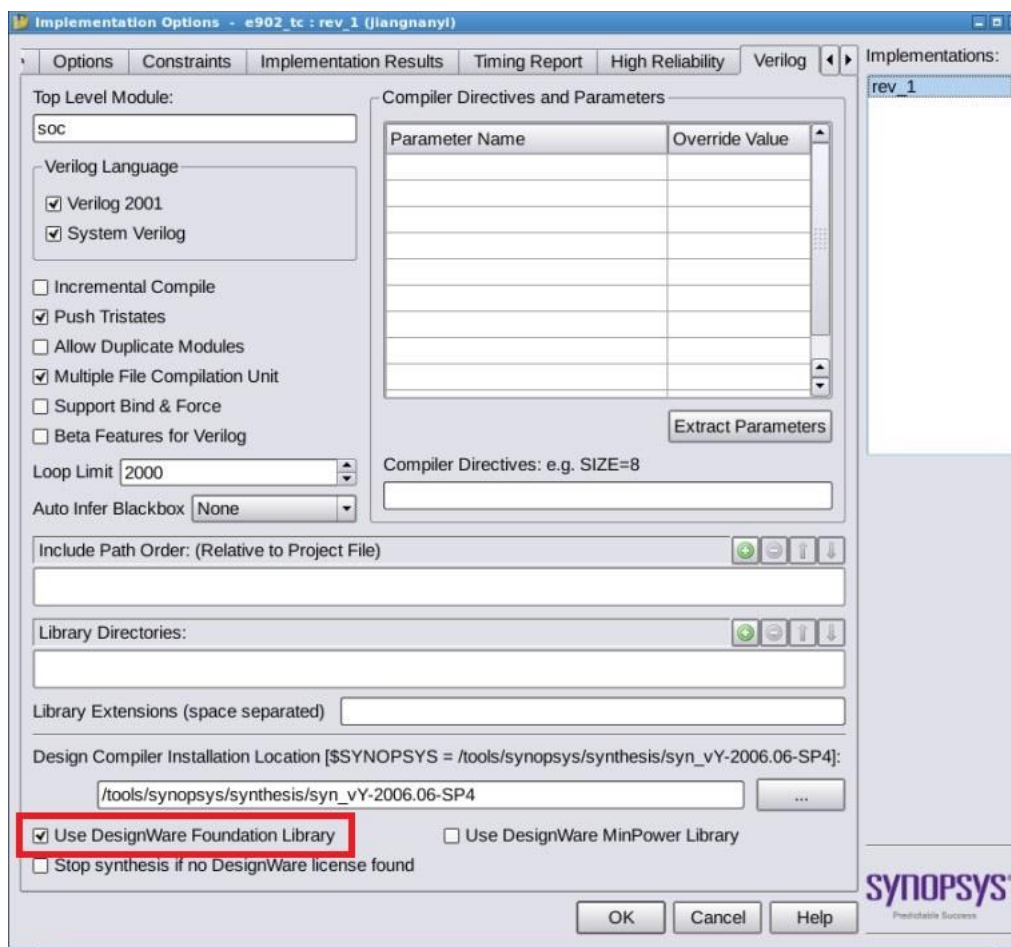
FPGA 的映射过程如下：

10.3.1 DesignWare IP

客户在使用 FPGA 综合时，如果使用 Synplify 软件，只需要在综合选项中选择如下图所示的选项，就能把代码中实例化的 design ware 综合进去。



图表 10-14 综合选项



图表 10-15 选择 Design ware 库

为了方便不同 FPGA 厂商的工具软件都可以对其进行综合，我们还提供了 FPGA 下可综合的代码版本，位于 FPGA/DW 目录下。只需将 E902_YMMDD.filelist 里面包含的 ASIC/LIB 目录下的 designware 替换为 FPGA_SIM/DW 目录下同名的文件即可。

10.3.2 修改文件列表

FPGA 平台综合时需要将 ASIC/RTL 目录下面的 E902_YMMDD.filelist 文件中所包含的文件替换为 FPGA 平台可综合的文件，具体为 FPGA/RTL 目录下面的 E902_YMMDD_FPGA.filelist, FPGA/DW 目录下面的 DesignWare 乘法器以及 ASIC/RTL 目录下面的 sysmap.h 文件，cpu_cfg.h 文件。

11 门级仿真

11.1 无复位端寄存器

为了节省面积，数据通路上的寄存器通常不含复位端。在门级仿真中为了防止这类寄存器的初始 X 态传播开来，可以对其进行初始化操作。这里，我们介绍一种修改标准工艺库的方法。不同工艺库下都会有门级仿真中用到的各种 standard cell 的模型描述文件，以 TSMC 为例，我们只需修改 standard cell 中寄存器相关的 cell，如图表 11-1 所示。

```
`celldefine
module DFF1BWP (D, CP, Q, QN);
  input D, CP;
  output Q, QN;
  reg notifier;
  `ifdef NTC
    wire D_d, CP_d;
    pullup (CDN);
    pullup (SDN);
    tsmc_dff (Q_buf, D_d, CP_d, CDN, SDN, notifier);
    initial begin $deposit(Q_buf,1'b0); end
    buf (Q, Q_buf);
    not (QN, Q_buf);
  `else
    pullup (CDN);
    pullup (SDN);
    tsmc_dff (Q_buf, D, CP, CDN, SDN, notifier);
    initial begin $deposit(Q_buf,1'b0); end
    buf (Q, Q_buf);
    not (QN, Q_buf);
  `endif
endmodule
```

图表 11-1 寄存器 Q 端赋值示例

对于 DFF1BWP 这个不含置位端和复位端的寄存器来说，可以通过调用 VCS 的 deposit 这个 task 来将 tsmc_dff 的 Q_buf 端口值在仿真初始时刻无条件赋值为 0。因为该操作只会在仿真初始时刻发挥作用，后续仿真过程中若有违例或者 X 态，不会影响其传播，因此可以选择将所有类型的寄存器模型都采用这种方式进行赋值操作。