

# 玄铁 E906 集成手册 (opene906)

2024 年 06 月 27 日

**Copyright © 2023 Hangzhou C-SKY MicroSystems Co., Ltd. All rights reserved.**

This document is the property of Hangzhou C-SKY MicroSystems Co., Ltd. and its affiliates ("C-SKY"). This document may only be distributed to: (i) a C-SKY party having a legitimate business need for the information contained herein, or (ii) a non-C-SKY party having a legitimate business need for the information contained herein. No license, expressed or implied, under any patent, copyright or trade secret right is granted or implied by the conveyance of this document. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of Hangzhou C-SKY MicroSystems Co., Ltd.

### **Trademarks and Permissions**

The C-SKY Logo and all other trademarks indicated as such herein (including XuanTie) are trademarks of Hangzhou C-SKY MicroSystems Co., Ltd. All other products or service names are the property of their respective owners.

### **Notice**

The purchased products, services and features are stipulated by the contract made between C-SKY and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

### **杭州中天微系统有限公司 Hangzhou C-SKY MicroSystems Co., LTD**

Address: Room 201, 2/F, Building 5, No.699 Wangshang Road , Hangzhou, Zhejiang, China

Website: [www.xrvm.cn](http://www.xrvm.cn)

### **Copyright © 2023 杭州中天微系统有限公司，保留所有权利。**

本文档的所有权及知识产权归属于杭州中天微系统有限公司及其关联公司（下称“中天微”）。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中天微员工，或 (ii) 非中天微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，未经杭州中天微系统有限公司明示同意，则不能使用该文档。在未经中天微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**

中天微的 LOGO 和其它所有商标（如 XuanTie 玄铁）归杭州中天微系统有限公司及其关联公司所有，未经杭州中天微系统有限公司的书面同意，任何法律实体不得使用中天微的商标或者商业标识。

### **注意**

您购买的产品、服务或特性等应受中天微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，中天微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。杭州中天微系统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。

### **杭州中天微系统有限公司 Hangzhou C-SKY MicroSystems Co., LTD**

地址：中国浙江省杭州市网商路 699 号 5 号楼 2 楼 201 室

网址：[www.xrvm.cn](http://www.xrvm.cn)

# 版本历史

版本	描述	日期
01	opene906 第一版发布。	2021.10.18
02	更新手册模板	2024.6.27

# 文档编号

玄铁 CPU 技术文档类编号采用如下所示规则：

产品名称-产品型号-产品版本号-文档类型。

# 玄铁 E906 集成手册 (opene906)

<b>第一章 概述</b>	<b>1</b>
1.1 处理器简介	1
1.1.1 集成与实现简介	1
1.1.2 E906 数据包文件结构	3
1.1.3 E906 数据包使用环境	3
1.1.3.1 E906 的配置介绍	3
1.1.3.2 E906 RTL 代码的文件结构	4
1.1.3.3 Memory 替换	4
<b>第二章 仿真平台介绍</b>	<b>8</b>
2.1 平台架构	8
2.2 平台仿真流程介绍	9
2.2.1 测试用例	9
2.2.2 平台 RTL 编译	10
2.2.3 测试用例执行	10
2.2.4 测试用例的调试	10
<b>第三章 处理器集成总览</b>	<b>11</b>
3.1 命名规则	12
3.2 端口信号总览	13
<b>第四章 时钟集成</b>	<b>15</b>
4.1 端口列表	15
4.2 E906 时钟域	16
4.3 时钟分频	17
4.4 多时钟域信号同步	17
4.4.1 CPU 内核时钟域与系统总线时钟域	17
4.4.2 CPU 内核时钟域与调试模块时钟域	19
4.4.3 TDT DMI 模块内调试时钟域与 JTAG 接口时钟域	19
4.5 调频操作	19
<b>第五章 复位集成</b>	<b>20</b>
5.1 软复位	21

5.2	复位启动地址 . . . . .	22
<b>第六章</b>	<b>总线系统集成</b>	<b>23</b>
6.1	总线简介及配置信息 . . . . .	23
6.2	系统总线接口 . . . . .	24
6.3	指令总线接口 . . . . .	26
6.3.1	数据总线接口 . . . . .	29
<b>第七章</b>	<b>地址空间属性设置</b>	<b>32</b>
<b>第八章</b>	<b>中断系统集成</b>	<b>34</b>
8.1	中断处理过程简述 . . . . .	34
8.2	端口列表 . . . . .	34
8.3	中断握手时序图 . . . . .	35
8.4	中断嵌套 . . . . .	35
8.5	计时器中断 . . . . .	36
<b>第九章</b>	<b>调试系统集成</b>	<b>37</b>
9.1	端口列表 . . . . .	37
<b>第十章</b>	<b>低功耗系统集成</b>	<b>40</b>
10.1	端口列表 . . . . .	40
10.2	工作模式及其转换 . . . . .	40
10.3	进入低功耗模式握手 . . . . .	40
10.4	退出低功耗模式握手 . . . . .	42
10.5	浅睡眠 . . . . .	42
10.6	深睡眠 . . . . .	43
<b>第十一章</b>	<b>杂类信号集成</b>	<b>44</b>
11.1	简介 . . . . .	44
11.2	通用观测信号 . . . . .	46
11.3	观测信号示例波形 . . . . .	47
<b>第十二章</b>	<b>DFT 信号集成</b>	<b>48</b>
12.1	端口列表 . . . . .	48

# 第一章 概述

## 1.1 处理器简介

E906 是一款基于 RISC-V 指令集的高能效比嵌入式处理器，是玄铁 RISC-V MCU 产品线中的最高性能处理器产品。E906 主要面向语音、高端 MCU、轻量级 AI、导航、WiFi 等应用领域。

玄铁 CPU 向授权用户提供了包含 E906 处理器核心，调试单元，中断控制器等在内的 E906 的顶层集成接口，如 图 1.1 所示。

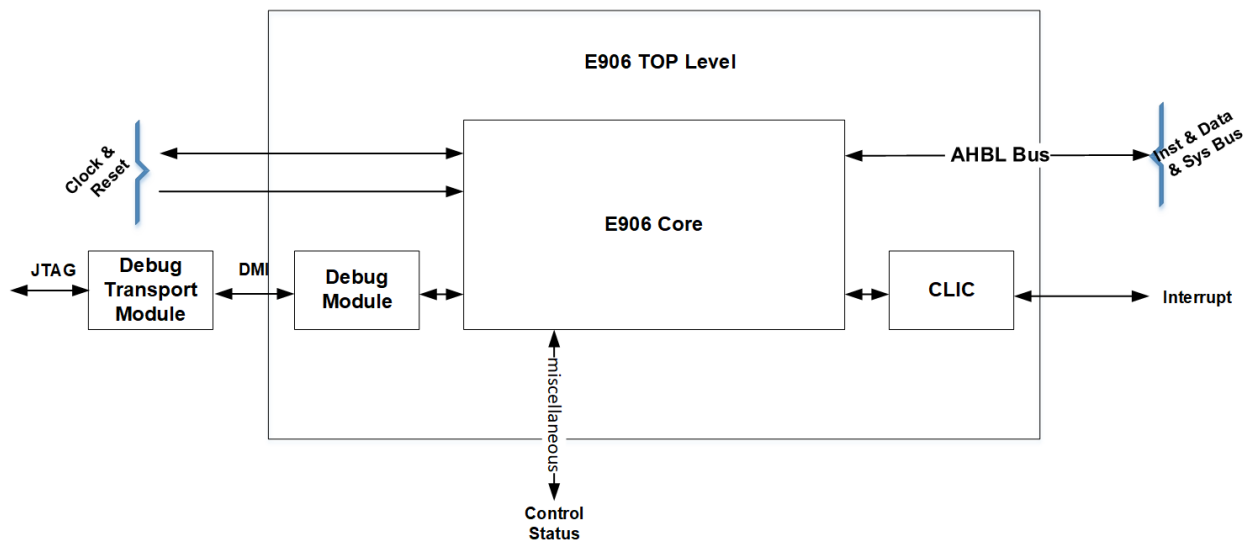


图 1.1: E906 顶层集成接口概览

用户只需对“E906 TOP Level”这一层级所显示的接口进行集成连接。

### 1.1.1 集成与实现简介

玄铁 CPU 向 opene906 的客户提供了固定配置的 E906 代码包, 因此 E906 的集成与实现可按图 1.2 进行。

E906 提供了与代码配置功能相配套的仿真验证环境，用于演示 E906 的集成和软硬件仿真流程，并且提供必要的示例程序和测试激励等，具体可参考[仿真平台介绍](#)的相关内容。

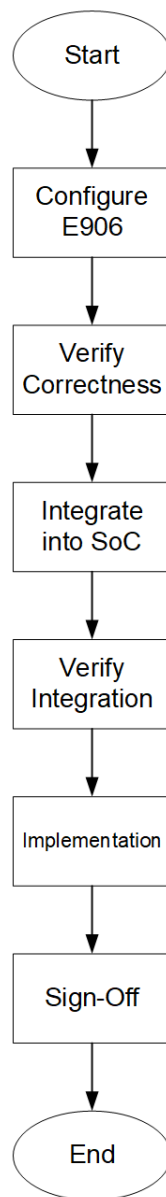


图 1.2: E906 集成与实现流程



### 1.1.2 E906 数据包文件结构

玄铁 E906 的数据包由以下两部分组成：

- CPU 代码包：固定功能配置的 E906 RTL 代码；
- Smart 平台包：提供了 E906 的参考集成设计、仿真环境和测试用例，帮助熟悉 E906 的功能和使用方法，并辅助 E906 的集成工作。功能包括：
  - 1) E906 的参考集成设计；
  - 2) 丰富的测试用例；
  - 3) 基础的软硬件仿真环境；
  - 4) E906 RTL 代码 lint 流程及 waive 文件；
  - 5) 后端综合实现的参考 SDC 约束文件。

### 1.1.3 E906 数据包使用环境

在使用 E906 数据包前需要检查运行环境是否具备以下条件：

- 环境工具依赖的 Perl 版本为 5.10.1；
- Make 版本为 3.8.1；
- EDA 仿真工具：Icarus Verilog (iverilog) 10.2, Synopsys VCS 2019.09 及以上版本，或者 Cadence irun 18.03 及以上版本；
- Smart 平台编译测试用例依赖的玄铁处理器 RISC-V 工具链版本 2.0.3 及以上。

#### 1.1.3.1 E906 的配置介绍

E906 的固定配置情况如 [表 1.1](#) 所示。

表 1.1: E906 的固定配置情况

可配置单元	配置选项	opene906 当前配置
浮点单元	无/单/单 + 双精度浮点	RV32IMAFc (单精度)
DSP 单元	无/有	NO
L1 ICache	0/2/4/8/16/32KB	16KB
L1 DCache	0/2/4/8/16/32KB	16KB
SYSMAP 软件配置	无/有	有
PMP 表项数	0/4/8/12/16	8
BHT	0/2/4/8/16Kb	8Kb
BTB	无/有	有
中断数量	16-240	128
中断优先级位宽	2-5	3
CLIC	Freq/Latency	Freq
性能监测单元	无/有	Yes
调试资源配置	最小配置/典型配置/ 最大配置	典型配置

### 1.1.3.2 E906 RTL 代码的文件结构

#### E906 RTL 代码的文件结构说明

E906 RTL 代码都在 gen\_rtl 目录下，内容如下：

cpu 文件夹：E906 顶层 RTL 模块等，如 openE906.v。

filelist 文件夹：E906 RTL 文件列表。

E906 各子模块 RTL：包含了包括 E906 绝大多数模块的 RTL 文件。

### 1.1.3.3 Memory 替换

**需作替换处理文件所在位置** 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 pa\_\*\*\_spsram\_AAAxBBBB, A、B 分别代表 memory 的深度和宽度。

**Memory 规格** 所有需要的 memory 的深度以及宽度规格可以从 INST\_LIB 下文件的 AAAxBBBB 命名中得到。

另外，我们对所需的 memory 有一个要求，需要支持是位写使能信号。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、功耗等因素。如果觉得有些 memory 过大或者过深，可以自行进行拼接，后面章节会介绍 memory 的拼接。

```
    sprf0651l_512x64 x_spsram_512x64(  
        .A      (A),  
        .D      (D),  
        .BWEN   ({32{WEN[1]}}, {32{WEN[0]}}),  
        .WEN     (&WEN),  
        .CEN     (CEN),  
        .CLK     (CLK),  
        .Q      (Q)  
    );
```

图 1.3: Memory 例化

**Memory 端口说明** 以特定工艺 memory 为例，如 图 1.3 所示。

Memory port 信号列表如 表 1.2 所示。

表 1.2: memory port 信号列表

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。
WEN*	低电平有效，位写使能信号。	RAM 位写使能信号端缩位与之后做 RAM 写使能信号。
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。

\*：这两根信号都是低电平有效，具体接法要看 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。

在某些特定工艺下，为了方便客户将 EMA，SD 等端口直连到 E906 顶层进行控制，E906 在顶层分配了一组可配置宽度的 Memory 信号：mem\_cfg\_in 和 mem\_cfg\_out。用户可根据需要对它们的存在与否以及位宽进行配置，并在 memory 例化时将 EMA 等信号连接在该组信号上实现在 E906 顶层接口对其进行控制。

**Memory 控制与行为描述** 图 1.4 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 2 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。

**拼接 Memory** 如所需 memory 规格在特定 compiler 下无法生成，可以用更小的 memory 拼接出需要的 memory。

以 2048x32 的 memory 为例如 图 1.5 所示。

假设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，用两块 1024x32 的 memory 进行拼接的示例如 图 1.6 所示。

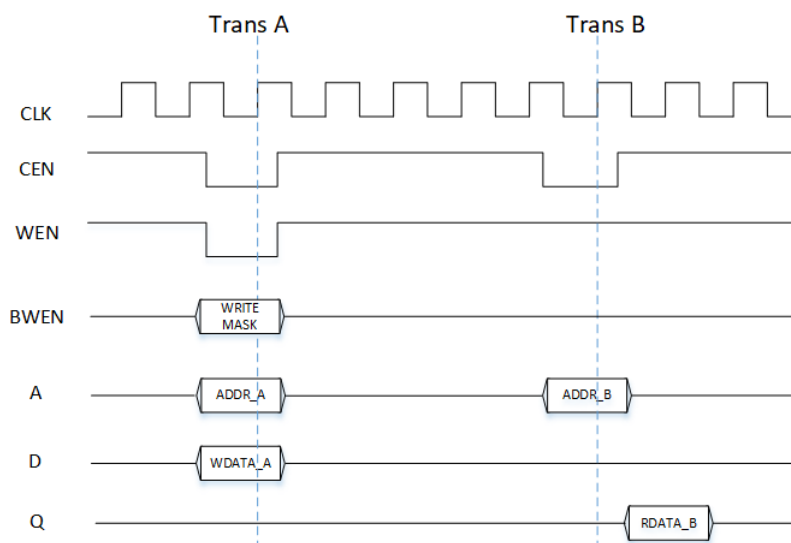


图 1.4: Memory 读写时序

```
spsram040g_2048x32 x_spsram_2048x32(
    .A      (A),
    .D      (D),
    .BWEB   ({8{WEN[3]}},{8{WEN[2]}},{8{WEN[1]}},{8{WEN[0]}}}),
    .WEB    (&WEN),
    .CEB    (CEN),
    .CLK    (CLK),
    .DELAY  (2'b00),
    .TEST   (2'b00),
    .Q      (Q)
);
```

图 1.5: Memory 拼接示例

```

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
end
assign Q[31:0] = bank_sel ? Q1[31:0] : Q0[31:0];
sprf065lp_1024x32 x_spsram_1024x32_bank0(
    .A      (A[ADDR_WIDTH-2:0]),
    .D      (D),
    .BWEB   ({ 8{WEN[3]}}, {8{WEN[2]}}, {8{WEN[1]}}, {8{WEN[0]}} },
    ),
    .WEB    (&WEN),
    .CEB    (CEN0),
    .CLK    (CLK),
    .TURBO  (1'b1),
    .RTSEL  (1'b0),
    .TSEL   (2'b01),
    .Q      (Q0)
);

sprf065lp_1024x32 x_spsram_1024x32_bank1(
    .A      (A[ADDR_WIDTH-2:0]),
    .D      (D),
    .BWEB   ({ 8{WEN[3]}}, {8{WEN[2]}}, {8{WEN[1]}}, {8{WEN[0]}} },
    ),
    .WEB    (&WEN),
    .CEB    (CEN1),
    .CLK    (CLK),
    .TURBO  (1'b1),
    .RTSEL  (1'b0),
    .TSEL   (2'b01),
    .Q      (Q1)
);

```

图 1.6: Memory 拼接示例

## 第二章 仿真平台介绍

E906 的授权使用数据包除了包含代码配置相关功能，还提供了软硬件仿真环境及特定测试激励。本章主要对该仿真平台相关组件进行介绍。

### 2.1 平台架构

鉴于 E906 顶层为 AHB-Lite 兼容的总线接口，因此仿真平台基于 AHB-Lite 和 APB 进行总线互联。图 2.1 所示为用于集成参考的 Smart 平台架构。

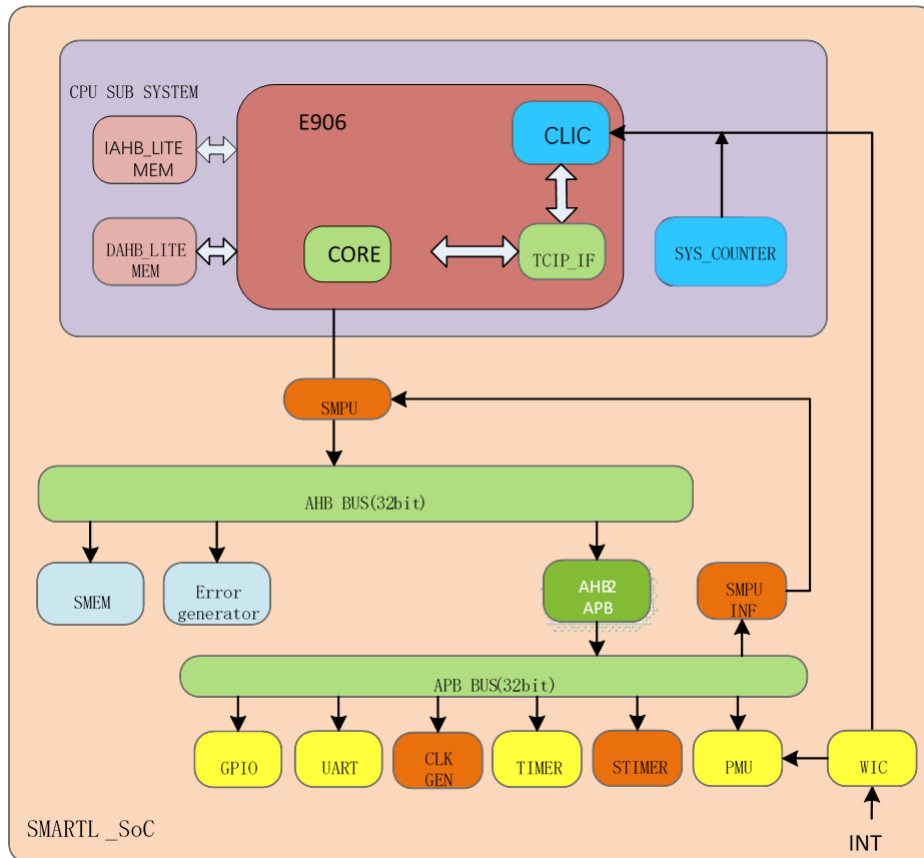


图 2.1: Smart 平台架构

平台地址空间分配如 表 2.1 所示。

表 2.1: Smart 平台地址空间分配

地址空间	映射的 IP
0x00000000-0x1FFFFFFF	本地指令存储器可用地址空间。
0x20000000-0x3FFFFFFF	本地数据存储器可用地址空间。
0x40000000-0x4FFFFFFF	APB 总线地址空间。
0x60000000-0x7FFFFFFF	系统内存可用地址空间。
0xE0000000-0xEFFFFFFF	紧耦合 IP 空间 (TCIP)。
其它	保留，读写返回错误。

外设 IP 的空间分配如 表 2.2 所示。

表 2.2: Smart 平台外设 IP 的地址空间分配

基址	IP
0x40011000	TIMER (APB Slave)
0x40015000	UART (APB Slave)
0x40016000	PMU (APB Slave)
0x40017000	CLK_GEN(APB Slave)
0x40018000	STIMER(APB Slave)
0x40019000	GPIO(APB Slave)
0x4001A000	SMPU(APB Slave)
0x40021000	NMI_WAKE_EVENT(APB Slave)

## 2.2 平台仿真流程介绍

### 2.2.1 测试用例

执行如下命令可以列出所有当前 CPU IP 配置下，Smart 提供的测试用例：

```
make showcase
```

在仿真中执行每个测试用例前，需要做测试用例的编译，命令为：

```
make buildcase CASE= 测试用例名 (测试用例名可由上述 make showcase 命令列出)
```

需要注意，Smart 平台依赖环境变量 \$TOOL\_EXTENSION 来选择编译测试用的 GNU 编译器，Smart 平台在路径 E906\_SMART\_FACTORY/setup/下，提供了名为 example\_setup.csh 的脚本作为参考，说明如何设置 \$TOOL\_EXTENSION。

### 2.2.2 平台 RTL 编译

在仿真前，需要做硬件编译，包括：1) Smart 平台提供的 CPU IP 的参考集成设计；2) Smart 平台的仿真环境。命令如下：

```
make compile
```

默认的仿真器为 iverilog，使用 SIM 参数可以帮助选择其他仿真器，详见 make help 的提示信息。

### 2.2.3 测试用例执行

可以用以下命令，直接运行 Testbhench 编译、测试用例编译与单个测试用例的执行：

```
make runcase CASE= 测试用例名
```

更多具体选项，详见 make help 中的提示信息。

当想执行 Smart 平台提供的全部测试用例时，命令为：

```
make regress
```

当所有用例执行结束后，命令行中将显示包含所有测试用例执行结果的报告，该报告也可以在../smart\_run/tests/regress 路径下找到。

### 2.2.4 测试用例的调试

当有测试用的的行为异常时，可以在 Smart 平台里打开波形功能以便进一步分析，命令为：

```
make runcase CASE= 测试用例名 DUMP=on
```

请执行以下命令获得更多关于 Smart 环境使用的信息：

```
make help
```



# 第三章 处理器集成总览

E906 整体框图如 图 3.1 所示。

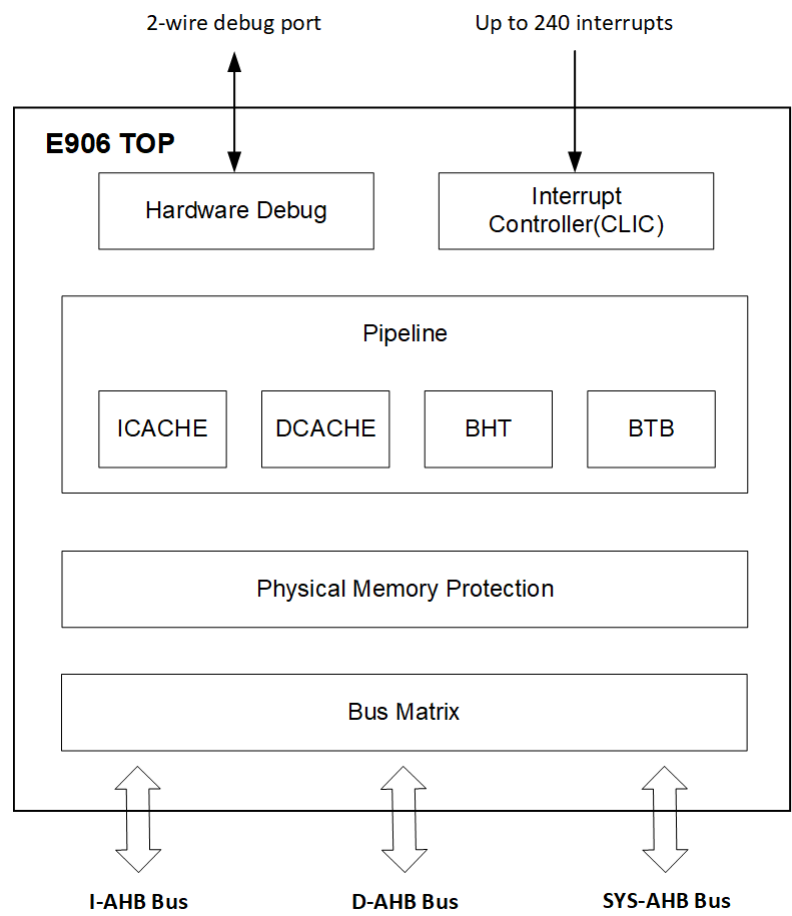


图 3.1: E906 TOP 框图

E906 的集成可按照功能划分为如下几部分进行：

1. 时钟集成

时钟集成部分介绍了 E906 处理器中 CPU 核与调试单元（HAD）的输入时钟，CPU 核与 HAD 以及总线系统之间信号同步逻辑。

2. 复位集成

复位集成部分介绍了 E906 的输入复位以及软复位相关交互信号；

3. 总线集成

总线集成部分介绍了 3 条 AHB 总线接口，包括：系统总线接口、指令总线接口和数据总线接口。

4. 中断集成

中断集成部分介绍了与 E906 处理器中断处理相关的信号以及信号握手机制。

5. 调试集成

调试集成部分介绍了两线调试接口信号和其它调试辅助信号接口。

6. 低功耗集成

低功耗集成部分介绍了 E906 低功耗相关的接口信号以及 E906 低功耗模式与正常工作模式之间的转换；

7. 杂类信号集成

杂类信号集成介绍了 SoC 仿真阶段所需的 CPU 运行状态的观测等信号。

8. DFT 信号集成

3.1 命名规则

表 3.1: 信号命名规则

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

一般来说，没有特殊规定，E906 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。

## 3.2 端口信号总览

为了降低 E906 的硬件集成复杂度，除了可配置的总线接口部分信号外，其他信号在多款玄铁处理器中，均保持了稳定的延续性，以方便 E906 的升级换代。根据 E906 顶层端口信号的特点，划分为时钟信号，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，杂类信号等几大类。

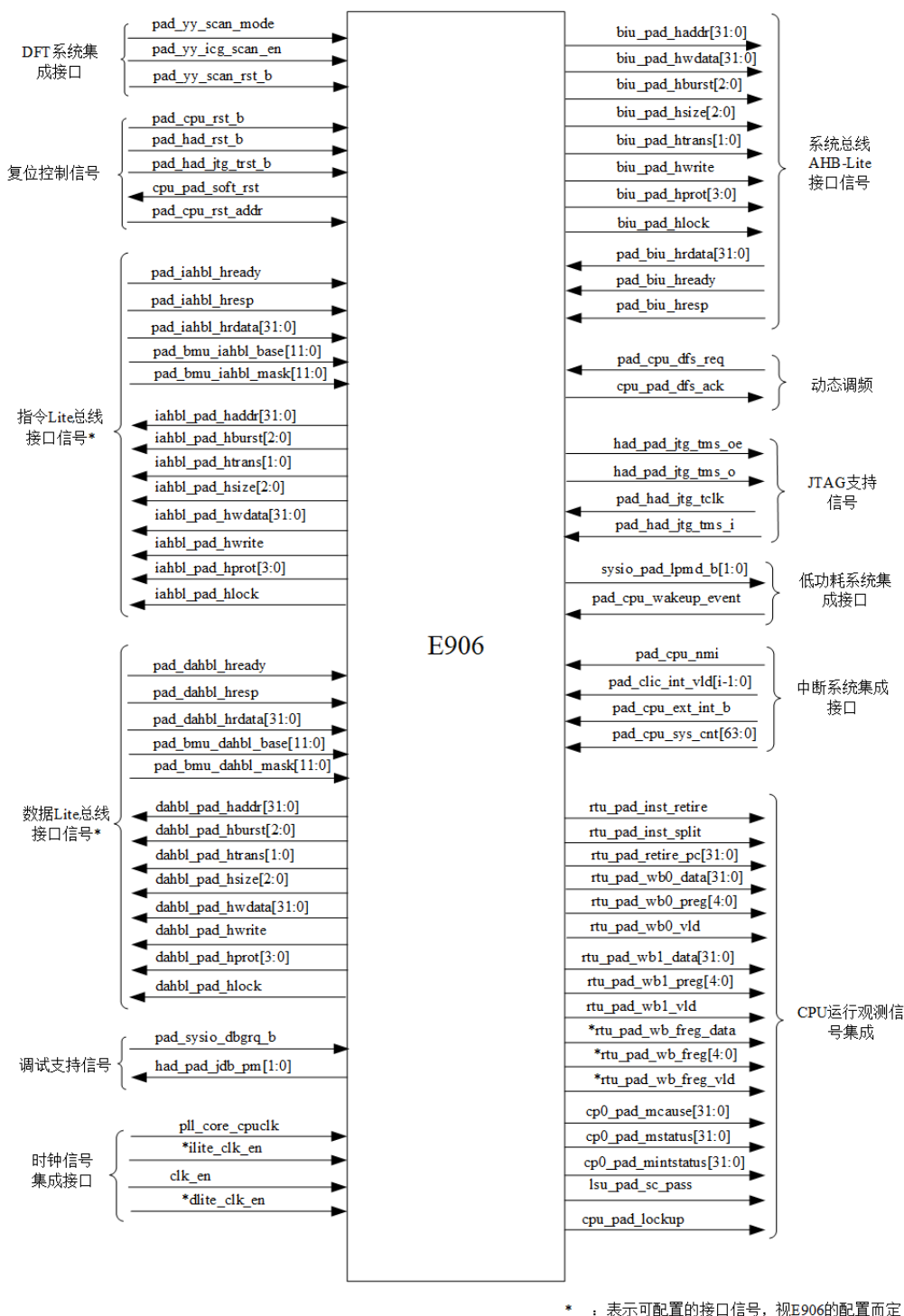


图 3.2: E906 CPU 系统接口总体描述

# 第四章 时钟集成

图 4.1 是 E906 的时钟结构。

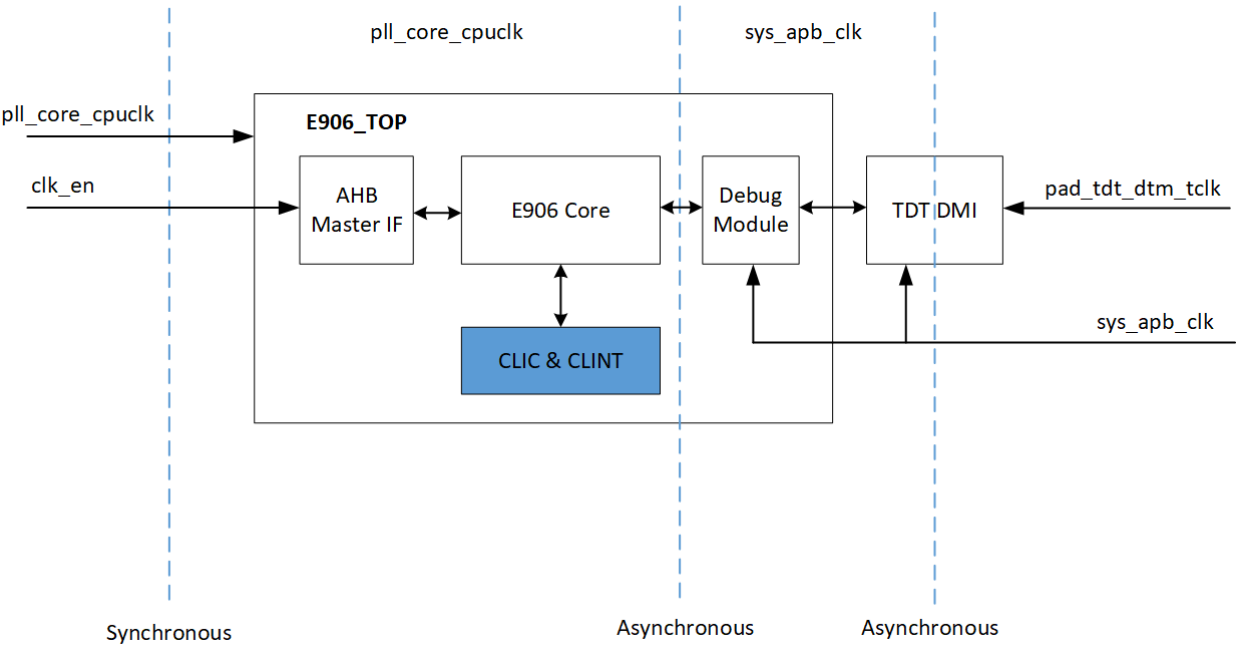


图 4.1: E906 时钟结构图

## 4.1 端口列表

时钟信号列表:

表 4.1: 时钟信号列表

信号名	方向	复位	时钟	功能描述
pll_core_cpuclock	I	-	-	CPU 工作时钟信号： 提供 CPU 工作的时钟。
sys_apb_clk	I	-	-	调试模块工作时钟： 该时钟为 pll_core_cpuclock 的异步时钟，时钟频率要小于或者等于 pll_core_cpuclock 的频率。
pad_tdt_dtm_tclk	I	-	-	调试接口逻辑时钟信号： 该时钟由调试器产生并通过芯片引脚传递进 SoC 系统，该时钟为 sys_apb_clk 的异步时钟，时钟频率要小于或者等于 sys_apb_clk 的二分之一及以下。
clk_en	I	-	-	CPU 同步时钟信号： 当为 1 时，CPU 时钟可以采样系统总线时钟的信号，或者将 CPU 时钟域的内部信号采样输出到系统总线时钟域。 注意：该信号不受低功耗影响，CPU 在低功耗模式下该信号仍需有效。

动态调频信号：

表 4.2: 动态调频信号列表

信号名	方向	复位	时钟	功能描述
pad_cpu_dfs_req	I	0	CPU	系统向 CPU 发起的调频请求信号。
cpu_pad_dfs_ack	O	0	SYS	CPU 因应系统的调频请求，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。

4.2 E906 时钟域

E906 顶层有两个输入时钟信号 pll\_core\_cpuclock 和 sys\_apb\_clk。

pll\_core\_cpuclock 是 CPU 工作时钟，E906 支持 pll\_core\_cpuclock 是系统总线时钟的同频、2~8 倍频，要求 pll\_core\_cpuclock 与系统总线时钟同源同相。clk\_en 用于 E906 系统总线及计时器计数值等接口信号与系统总线时钟域进行时钟分频同步采样。pll\_core\_cpuclock 需要与指令总线，数据总线时钟为同一时钟源。

sys\_apb\_clk 为 E906 内部调试模块（Debug Module）以及 TDT\_DMI 模块内 DTM 等寄存器的工作时钟，该时钟与 pll\_core\_cpuclock 为异步时钟。E906 内部针对跨时钟域信号做了同步处理。另外需要保证 sys\_apb\_clk 频率小于或等于 pll\_core\_cpuclock 频率。

在 E906 代码数据包里会将 E906 和调试接口模块 (tdt\_dmi\_top) 分开提供，pad\_tdt\_dtm\_tclk 是 JTAG 接口逻辑的输入时钟信号，与 sys\_apb\_clk 为异步时钟，由芯片引脚接入。在调试时由仿真器输出引

脚连接到该芯片引脚。TDT DMI 模块内部负责将该时钟域信号与 sys\_apb\_clk 时钟域信号做同步处理。为了保证调试功能的正常，需要保证 sys\_apb\_clk 频率是 pad\_tdt\_dtm\_tclk 频率的两倍及以上，由仿真器传递进来的 pad\_tdt\_dtm\_tclk 时钟频率最低为 10KHz。

## 4.3 时钟分频

E906 支持系统总线接口与外部系统时钟域的分频处理，具体分频逻辑由 SoC 完成，E906 通过 clk\_en 信号对 CPU 和外部系统总线时钟域的信号进行采样同步。clk\_en 需要由 SoC 根据分频比来产生对应的时序。由于 E906 内部会使用该信号对接口输入信号进行采样，为了后端时序收敛，E906 内部将上述 clk\_en 信号在 pll\_core\_cpuclock 时钟域下 Flop 一拍后才使用，因此 SoC 在产生该信号时序时需要提早一个 pll\_core\_cpuclock 的时钟周期。

需要注意：只有系统总线才支持使用 clk\_en 信号进行分频。指令总线和数据总线不支持分频。

除了 CPU 时钟域和外部系统时钟域的分频之外，CPU 内部会生成 pll\_core\_cpuclock 的二分频时钟，驱动 CLIC 和 CLINT 模块的寄存器翻转，该分频逻辑只有在硬件配置中断个数大于 64 个时存在。

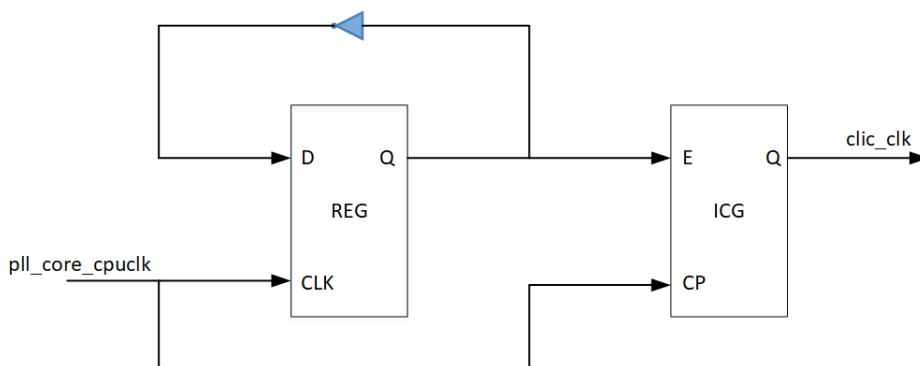


图 4.2: CLIC 二分频逻辑

## 4.4 多时钟域信号同步

### 4.4.1 CPU 内核时钟域与系统总线时钟域

在 E906 的设计中，CPU 时钟域与系统总线时钟域在顶层接口部分逻辑中有信号交互。这部分时钟要求在物理设计中，需要保证相位对齐。由于 CPU CLK 只有频率整数倍关系，所以无需用专用的逻辑进行同步。E906 采样外部特定产生的控制信号 (clk\_en)，且 clk\_en 信号在 CPU 内部作打拍处理，保证 CPU 内部到总线上的所有信号都按照系统时钟的上升沿对齐。因此 CPU 系统总线接口号都具有系统时钟的全周期时序延时。图 4.3 为 pll\_core\_cpuclock 与系统时钟频率比为 1:1~8:1 时 clk\_en 的时序图。

图 4.3 所示表明 clk\_en 在 CPU 内部被打了一拍之后才用于对顶层接口的同步采样。

指令总线和数据总线均为非寄存器输入时序，因此 CPU 外指令总线和数据总线上逻辑需工作在 CPU CLK 时钟域下。

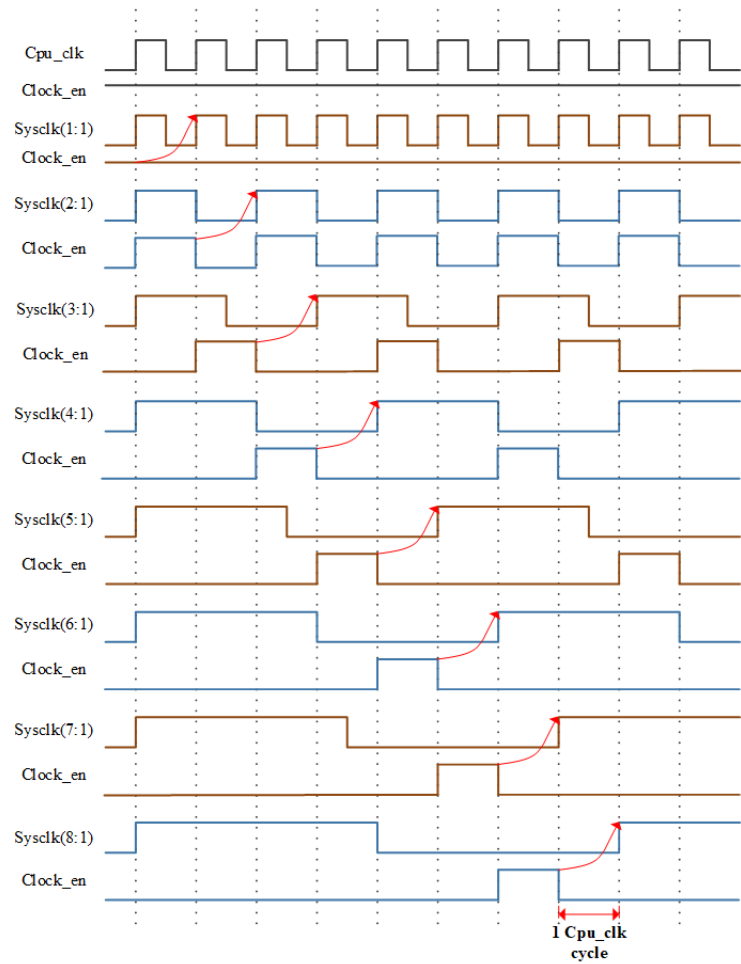


图 4.3: CPU 时钟与系统时钟域信号同步关系图



### 4.4.2 CPU 内核时钟域与调试模块时钟域

在 E906 的设计中, CPU 内核时钟 `pll_core_cpucclk` 与调试模块的 `sys_apb_clk` 时钟域间存在信号交互, 两个时钟属于异步时钟, E906 内部已经将跨时钟域信号进行了同步处理。调试模块硬件可配置的系统 AHB 总线请求在 E906 内部会先同步到 CPU CLK 时钟域, 再通过 `clk_en` 采样与 SYS CLK 时钟域信号进行交互。

### 4.4.3 TDT DMI 模块内调试时钟域与 JTAG 接口时钟域

TDT DMI 模块需要将上层软件发送的调试请求转化为针对 E906 内调试模块的寄存器访问。该模块有两个输入时钟: `sys_apb_clk` 以及 `pad_tdt_dtm_tclk`, 这两个信号为异步时钟, TDT DMI 模块内部已经将跨时钟域信号进行了同步处理。需要注意的是 `sys_apb_clk` 频率应为 `pad_tdt_dtm_tclk` 频率的两倍以上。

## 4.5 调频操作

E906 支持对输入时钟 `pll_core_cpucclk` 的调频操作, 只要调频期间 E906 输入时钟 `pll_core_cpucclk` 不间断, 则 E906 无需做任何操作, 系统保证调频逻辑正确即可。如果调频期间 `pll_core_cpucclk` 会出现间断的情况, 为了保证在调频时外设总线有未完成的传输返回响应, 可以调频前通过 `pad_cpu_dfs_req` 和 `cpu_pad_dfs_ack` 这组握手信号来实现。

`pad_cpu_dfs_req` 信号需要在调频开始前由 SoC 置为高电平, 该信号为 CPU CLK 的异步信号, E906 内部会将其同步到 CPU CLK 时钟域。E906 内部在接收到该请求信号后会将三条 AHB-Lite 总线上未完成请求处理完毕并不再发出新的传输请求, 并将 `cpu_pad_dfs_ack` 信号置为高电平。SoC 在观测到 `cpu_pad_dfs_ack` 为高电平后开始调节 `pll_core_cpucclk` 的频率并在调节结束后将 `pad_cpu_dfs_req` 置为低电平。之所以通过这组信号来完成上述握手时序, 是避免 SoC 设计时 `pll_core_cpucclk` 频率被调节期间出现时钟短暂关闭的情况, AHB-Lite 总线未完成的请求在这种情况下会出现丢失的情况, 导致总线卡死。

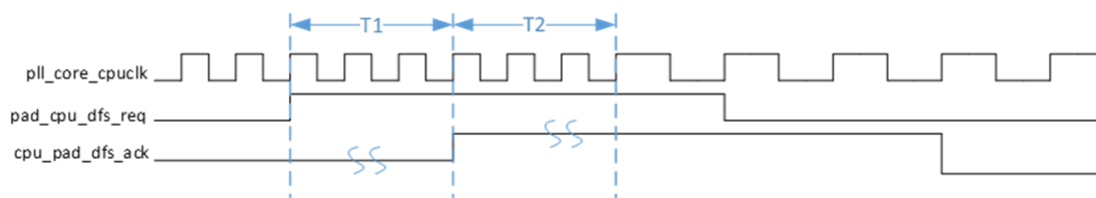


图 4.4: `pll_core_cpucclk` 频率调节握手时序图

## 第五章 复位集成

E906 支持多种复位模式，并且支持多种软复位模式定义。图 5.1 所示为 E906 的复位框图，其中 pad\_yy\_dft\_clk\_rst\_b 仅在 CLIC 模块时钟为 pll\_core\_cpucclk 时钟频率的 1/2 时才存在。

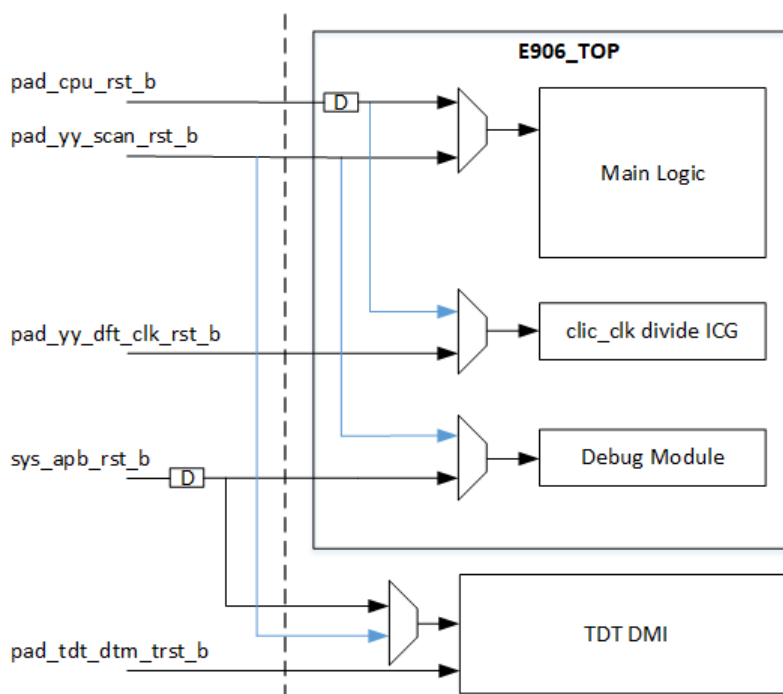


图 5.1: E906 复位示意图

E906 以及 TDT DMI 模块输入的复位信号如下：

- **pad\_cpu\_rst\_b**

用于正常工作模式下 E906 内部流水线，总线接口等逻辑单元的复位控制以及用于 CLIC 二分频时钟 ICG 的复位控制。该信号在 E906 内部经过三级同步后使用。

- **sys\_apb\_rst\_b**

用于 E906 内部调试模块 DM 以及 TDT DMI 模块工作在 sys\_apb\_clk 时钟域下寄存器的复位控制。对于 DM 和 TDT DMI 模块来说 sys\_apb\_rst\_b 是同一个复位源，对于 DM 和 TDT DMI 的复位时序相同，且是 sys\_apb\_clk 时钟域的同步信号。sys\_apb\_rst\_b 在 DM 和 TDT DMI 模块

内均打一拍后使用。该复位信号的置位操作需要早于或者与 pad\_cpu\_rst\_b 的释放操作同时，该复位信号的释放时序则没有要求。

- **pad\_yy\_scan\_rst\_b**

用于 scan 模式下 E906 内部除了 CLIC 二分频时钟 ICG 外的逻辑的复位控制。

- **pad\_yy\_dft\_clk\_rst\_b**

用于 scan 模式下 E906 内部 CLIC 和 CLINT 二分频 ICG 使能端口逻辑的复位，该复位端口需要接到上电复位端口上。之所以将该复位端口区分于 pad\_yy\_scan\_rst\_b，一是为了在上电时给上述 ICG 的始能端口一个初始值，二是为了不受 scan 下测试向量复位的影响，导致产生的二分频时钟受影响。

- **pad\_tdt\_dtm\_trst\_b**

用于 TDT DMI 模块下 JTAG 接口逻辑的复位控制，该复位信号需要接到 SoC 的顶层接口，使用时由外部调试器输入复位信号。该复位信号的置位操作需要早于或者与 sys\_apb\_rst\_b 的释放操作同时，该复位信号的释放时序则没有要求。

## 5.1 软复位

E906 提供了两组软复位指示信号供 SoC 集成者完成对 CPU 或者系统的复位操作。

- **cpu\_pad\_soft\_rst[1:0]**

该组信号为核内 MEXSTATUS 寄存器软复位寄存器域的镜像。软件通过设置该寄存器可在 cpu\_pad\_soft\_rst[1:0] 输出对应值，含义如下：2' b01: 需要复位 E906 内核，即 SoC 集成者需要设置 pad\_cpu\_rst\_b 为有效；2' b10: 需要复位整个系统，即 SoC 集成者可以进行系统复位，对于 E906 和 TDT DMI 而言需要设置 pad\_cpu\_rst\_b 和 sys\_apb\_rst\_b 有效。

- **tdt\_dm\_pad\_hartreset\_n**

E906 处理器核复位指示信号，调试软件通过设置 DM 内寄存器驱动该信号变为低电平，指示当前需要复位 E906 处理器核。SoC 集成者需要在看到该信号变为低电平时将 pad\_cpu\_rst\_b 驱动为低电平。之后调试软件通过设置 DM 内寄存器驱动该信号为高电平，指示前述复位需要释放，SoC 集成者需要在看到该信号变为高电平时解除前述对 pad\_cpu\_rst\_b 的置低操作。当不需要使用该复位信号时可悬空处理。

- **tdt\_dm\_pad\_ndmreset\_n**

除 DM 模块外的系统复位操作，调试软件通过设置 DM 内寄存器驱动该信号变为低电平，指示当前需要复位除 DM 以及影响 DM 正常访问的模块以外的 SoC 系统。该信号为低 SoC 做复位时，需要保证 sys\_apb\_rst\_b 不能为低，sys\_apb\_clk 不能被关闭，以保证调试软件可以正常访问 DM 以及 DM 内的逻辑单元不被复位。同 tdt\_dm\_pad\_hartreset\_n 一样，该信号由 DM 内的寄存器驱动，需要调试软件写 DM 内的寄存器将该信号置高，复位释放。当不需要使用该复位信号时可悬空处理。

## 5.2 复位启动地址

E906 提供一组接口信号 `pad_cpu_rst_addr[31:0]` 供系统配置 CPU 的复位启动基地址。系统复位后该值会被初始化到机器模式复位地址指示寄存器 (MRADDR) 中, 该信号 `pad_cpu_rst_addr[31:1]` 为有效位, 最低位需接为 0。在系统集成时可将该信号接为定位, 也可以寄存器形式传入, 但要满足在系统传给 E906 的复位信号 (`pad_cpu_rst_b`) 释放之前保持稳定。

# 第六章 总线系统集成

## 6.1 总线简介及配置信息

E906 实现了多总线接口，分别包括系统总线、指令总线、数据总线，以及紧耦合 IP 接口。其中指令总线、数据总线可由用户根据实际的系统需要进行配置，紧耦合 IP 接口的地址空间的高 4 位由 E906 顶层输入信号 pad\_cpu\_tcip\_base 指定，剩余地址空间对应系统总线。

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 6.1 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、数据总线以及 TCIP 接口上。

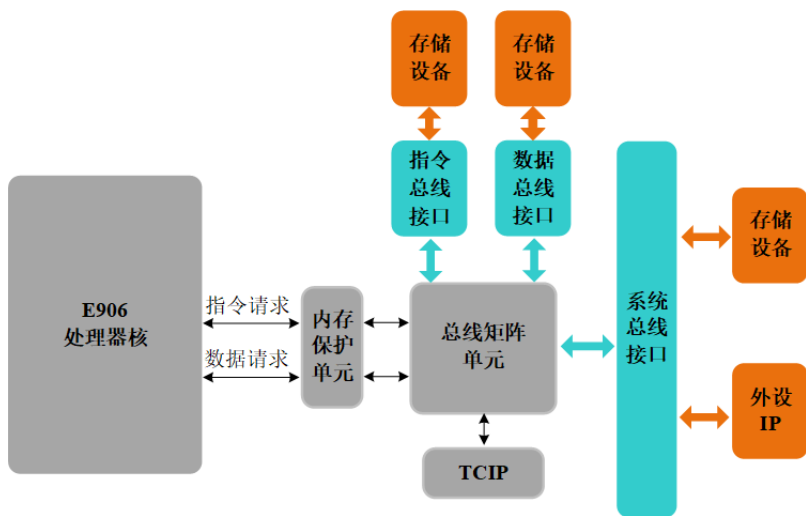


图 6.1: E906 总线矩阵

如 图 6.1 所示，E906 的指令总线和数据总线接口只允许外接 RAM，FLASH，ROM 等存储设备，不允许接外设 IP，外设 IP 只允许集成连接在 E906 的系统总线上，视系统设计需要，也可以在系统总线接口上集成连接存储设备。

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。

E906 多总线接口的基本信息和可配置性如 表 6.1 所示。

表 6.1: 多总线接口的基本信息和可配置性

总线接口	总线协议	时序方式
系统总线	AHB-Lite	寄存器输出
指令总线	AHB-Lite	直接输出
数据总线	AHB-Lite	直接输出

另外，E906 通过提供两组接口信号（pad\_bmu\_iahbl\_base 和 pad\_bmu\_iahbl\_mask，pad\_bmu\_dahbl\_base 和 pad\_bmu\_dahbl\_mask），分别支持指令总线和数据总线基地址和空间大小硬件集成时可配置。其中，pad\_bmu\_iahbl\_base 和 pad\_bmu\_dahbl\_base 指定了指令总线和数据总线的基地址；pad\_bmu\_iahbl\_mask 和 pad\_bmu\_dahbl\_mask 指定了不同地址空间下对地址对齐的需求。

指令总线和数据总线的地址空间 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 表 6.2，数据总线配置同理。

表 6.2: 指令总线对基地址和地址对齐的要求

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

E906 各总线地址空间在集成时分配时需保证各条总线连接的地址空间不重叠，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。因此，在 E906 配置实现多条总线的情况下，每条总线的地址空间大小都不应设置为 4GB。

## 6.2 系统总线接口

E906 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 E906 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。

表 6.3: 系统总线接口信号列表

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。 E906 仅支持 SINGLE 传输。
biu_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。
biu_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。 E906 中仅支持 NONSEQ 和 IDLE 两种传输类型。

下页继续

表 6.3 – 续上页

信号名	I/O	Reset	定义
biu_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。
biu_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。
biu_pad_hlock	O	0	锁定总线信号： 当该信号为高时表示 E906 发出的请求需要独占总线，直到该信号重新变为 0 之前系统中其他 master 不得占有总线。

表 6.4: 系统总线接口信号列表

信号名	I/O	Reset	定义
pad_biu_hrdata[31:0]	I	-	读数据总线： 32 位读数据总线。
pad_biu_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_biu_hresp	I	-	传输应答信号： 传输应答： 0：OKAY； 1：ERROR； E906 中仅支持 OKAY 和 ERROR 两种响应。

6.3 指令总线接口

E906 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。

E906 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 和 WRAP8 传输，其它突发类型均不支持；
- HTRANS 支持 IDLE/NONSEQ/SEQ 传输；
- 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。

表 6.5: 指令总线接口信号列表

信号名	I/O	Reset	定义
<b>指令 AHB-Lite 接口</b>			
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; 100: WRAP8。
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_hlock	O	-	锁定总线信号： 当该信号为高时表示 E906 发出的请求需要独占总线，直到该信号重新变为 0 之前系统中其他 master 不得占有总线。
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。
iahbl_pad_hwdata[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。

表 6.6: 指令总线接口信号列表

信号名	I/O	Reset	定义
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。 E906 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。

6.3.1 数据总线接口

E906 的数据总线只支持 AMBA3.0 AHB-LITE 协议，可参考 AMBA 3.0 规格说明- AMBA3 AHB-Lite Protocol Specification Rev 1.0。

E906 数据总线接口只实现了 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/WRAP8 传输，其它突发类型均不支持；
- HTRANS 支持 IDLE/NONSEQ/SEQ 传输；
- 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。

表 6.7: 数据总线接口信号列表

信号名	I/O	Reset	定义
<b>指令 AHB-Lite 接口</b>			
dahbl_pad_haddr[31:0]	O	-	地址总线： 32 位地址总线。
dahbl_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; 100: WRAP8;
dahbl_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。
dahbl_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。
dahbl_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。
dahbl_pad_hlock	O	-	锁定总线信号： 当该信号为高时表示 E906 发出的请求需要独占总线，直到该信号重新变为 0 之前系统中其他 master 不得占有总线。
dahbl_pad_hwdata[31:0]	O	-	写数据总线： 32 位写数据总线。
dahbl_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。

表 6.8: 数据总线接口信号列表

信号名	I/O	Reset	定义
pad_dahbl_hrdata[31:0]	I	-	读数据总线： 32 位读数据总线。
pad_dahbl_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_dahbl_hresp	I	-	传输应答信号： 传输应答： 0：OKAY； 1：ERROR。 E906 中仅支持 OKAY 和 ERROR 两种响应。
pad_bmu_dahbl_base[11:0]	I	-	DAHB-Lite 基址控制信号，上电复位之后需固定。
pad_bmu_dahbl_mask[11:0]	I	-	DAHB-Lite 地址对齐控制信号，上电复位之后需固定。

## 第七章 地址空间属性设置

E906 支持两种内存类型, 分别是内存 (memory) 和外设 (device) (由 SO 位区分)。其中, memory 类型根据是否可高缓 (Cacheable, C) 进一步分为可高缓内存 (cacheable memory) 和不可高缓内存 (non-cacheable memory)。device 类型的特点为必须按序执行, 因此 device 一定带有不可高缓的属性。device 根据是否可缓存 (Bufferable, B) 分为可缓存外设 (bufferable device) 和不可缓存外设 (non-bufferable device) (bufferable 表示 slave 允许在某个中间节点快速返回写完成; 反之, non-bufferable 表示 slave 只有在最终设备真正写完成后才返回写响应)。

表 7.1 给出了各个内存类型对应的属性。属性的配置有两种方式:

1. E906 顶层保留 8 组输入信号, 分别是 `pad_cpu_sysmap_addri` 和 `pad_cpu_sysmap_addri_attr(0 ≤ i ≤ 7)`。这 8 组信号指定了 CPU 复位启动后所访问的内存空间的属性, 集成时需要将其接为常值。
2. E906 支持硬件可配置的 8 组寄存器, 用于指定 CPU 所访问内存空间的属性。在硬件配置这组寄存器时, 它们的地址空间固定映射到 `0xEFFF_F000` 到 `0xEFFF_FFFF` (共 4KB), SoC 不可再次将该地址空间分配给其它外设。但因 E906 内部 TCIP 地址空间由 SoC 指定了基址的高 4 位, 低 28 位固定接 0, 且它所占用的 256MB 地址空的高 4KB 为保留空间, 因此 `pad_cpu_tcip_base[31:0]` 可设为 `0xE000_0000`, 并不与该地址空间属性寄存器冲突。

当 E906 内部硬件配置了 8 组属性配置寄存器时, 该 8 组地址属性配置寄存器的复位值对应为 SoC 集成接入的 `pad_cpu_sysmap_addri` 和 `pad_cpu_sysmap_addri_attr(0 ≤ i ≤ 7)` 信号值。SoC 集成者需将 `0xEFFF_F000` 到 `0xEFFF_FFFF` 地址空间设为不可缓存外设的属性。在 CPU 复位启动后, 软件可通过配置该 8 组地址属性寄存器动态调整内存空间属性, 但是 `0xEFFF_F000` 到 `0xEFFF_FFFF` 需要固定设置为不可缓存外设。需要注意的是第  $i+1$  个地址空间地址寄存器值要大于第  $i$  个地址空间地址寄存器的值, 无论是通过 SoC 集成时指定还是在 CPU 复位释放后由软件配置指定。

表 7.1: 内存类型分类

内存类型	SO	C	B
可高缓内存	0	1	1
不可高缓内存	0	0	1
可缓存外设	1	0	1
不可缓存外设	1	0	0

E906 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) 由寄存器

SYSMAP\_ADDR<sub>i</sub> (i 从 0 到 7) (当 CPU 内部没有配置该 8 组硬件寄存器时, 参照 pad\_cpu\_sysmap\_addr<sub>i</sub> 的值) 定义, 地址下限 (包含) 由 SYSMAP\_ADDR(i-1) 定义, 具体为:

$$\text{SYSMAP\_ADDR}(i-1) \leq \text{第 } i \text{ 个地址空间地址} < \text{SYSMAP\_ADDR}_i$$

第 0 个地址空间下限是 0x0, 内存地址不在设定的 8 个地址区间的地址属性默认为 SO, Non-Cacheable, Non-Bufferable。每个地址空间上下边界是 4KB 对齐, 因此寄存器 SYSMAP\_ADDR<sub>i</sub> 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寄存器 SYSMAP\_FLAG<sub>i</sub> (i 从 0 到 7) 定义, 属性的排布如 图 7.1 所示。

31					5	4	3	2	1	0
-						Strong Order	Cacheable	Bufferable	-	-

图 7.1: 内存地址属性排布

## 第八章 中断系统集成

### 8.1 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。

### 8.2 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。

表 8.1: CLIC 中断控制器接口信号

信号名	I/O	Reset	定义
<b>中断源信号:</b>			
pad_clic_int_vld[i-1:0]* i: 1-240	I	-	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。 该信号需要同步到 CPU 时钟域后再传入 E906。
pad_cpu_ext_int_b	I	-	CLINT11 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	-	NMI 中断请求, 上升沿有效, 该信号需要先同步到系统时钟再传入 E906, CPU 内部会用 clk_en 进行采样, 需要维持至少一个系统时钟周期。
pad_cpu_sys_cnt[63:0]	I	-	系统计时的当前计数值, 该信号需要先同步到系统时钟域再传入 E906。
pad_cpu_tcip_base[31:0]	I	-	CLIC 和 CLINT 的地址空间基地址信号。集成时接常值, 低 28 位需接 0, 高 4 位可以接任意常数, 共指定 256MB 的空间。



**注解：** 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。

关于 pad\_cpu\_tcip\_base[31:0]，需要注意：(1) E906 的复位启动地址不能落在该 256MB 空间内；(2) E906 支持硬件可配的 sysmap 寄存器，地址范围固定为：0xEFFF\_F000~0xEFFF\_FFFF (共 4KB)。如果 pad\_cpu\_tcip\_base == 0xE000\_0000，则 CLIC/CLINT 空间的最高 4KB 用于访问 sysmap 寄存器。如果 pad\_cpu\_tcip\_base != 0xE000\_0000，则 CLIC/CLINT 空间的最高 4KB 是保留空间。强烈建议客户把 sysmap 寄存器空间和 CLIC/CLINT 空间都配置为“不可缓存外设”。

## 8.3 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 8.1 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、ciic\_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。

## 8.4 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 E906 R1S3 用户手册》。

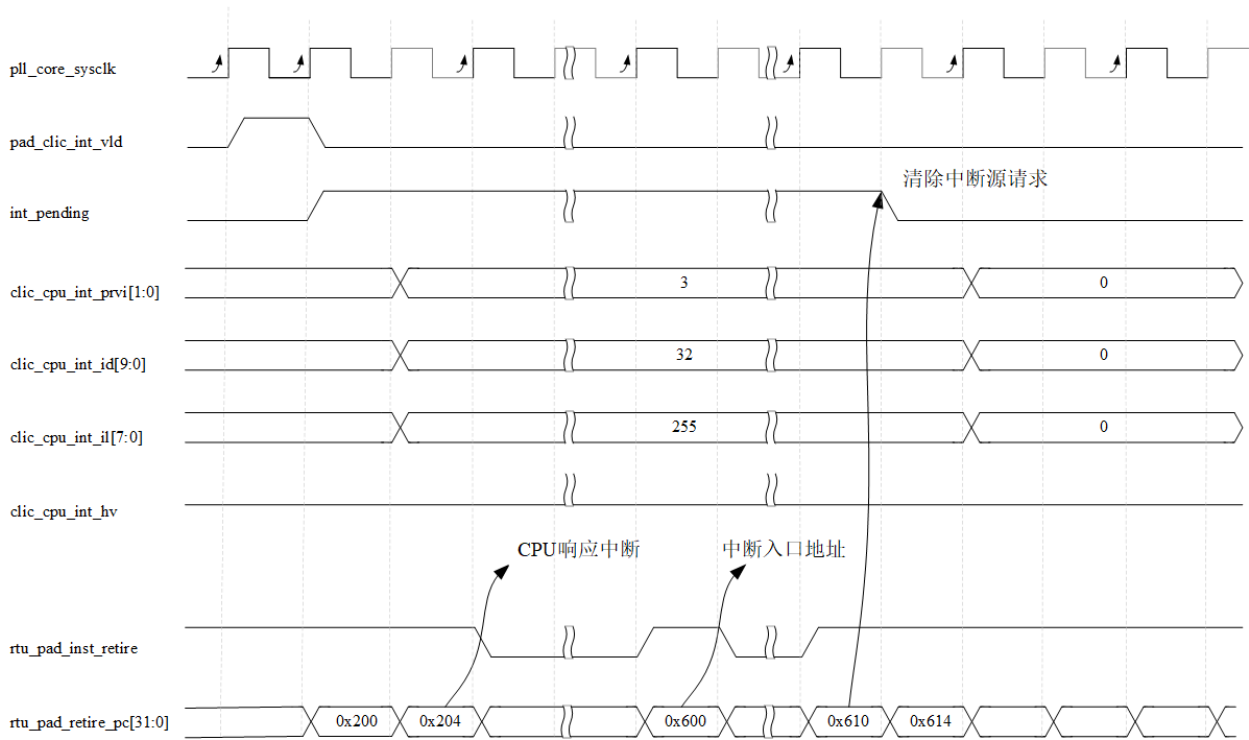


图 8.1: CPU 配置 CLIC 时中断相关信号时序简图

### 8.5 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 E906 外部由系统设计实现的 64 位计数器使用。该系统计数器需要工作在 always-on 的电压域，复位后在每个时钟周期进行计数。该系统计数器软件不可写，仅能通过复位清零。在 E906 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 E906 内部，高 32 位的值可通过 E906 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 E906 内部会由 clk\_en 进行采样，系统设计人员需要将该信号在 E906 外部完成同步到系统时钟域的操作。

## 第九章 调试系统集成

E906 设计实现了 RISC-V Debug 标准定义的调试系统，支持复用 DTM (Debug Transport Module) 模块进行异构多核调试连接。调试系统如下所示：

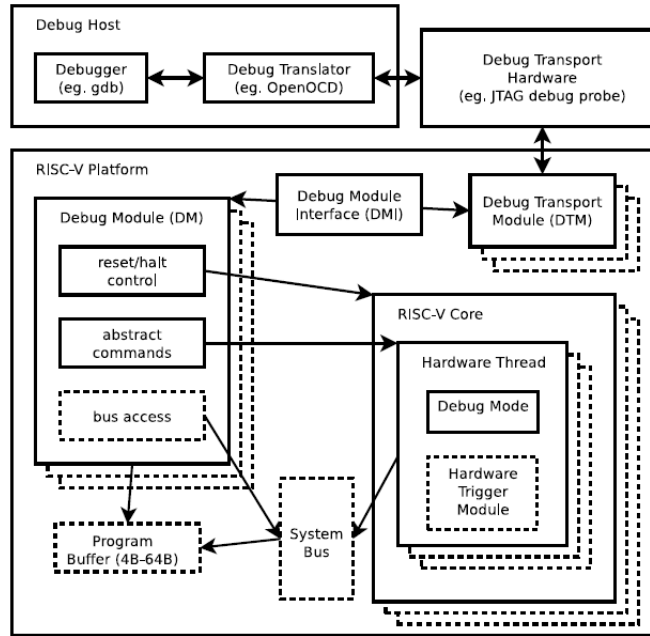


图 9.1: 调试系统框图

E906 发布代码分为两个顶层文件：E906 CPU top (pa\_cpu\_top\_merged.v) 与 Debug infrastructure top (tdt\_dmi\_top.v)。两个模块以标准 APB 总线连接。tdt\_dmi\_top.v 包含一个 APB master 接口，pa\_cpu\_top\_merged.v 包含一个 APB slave 接口，集成时把二者对应连接即可。

多核异构调试时，tdt\_dmi\_top.v 包含多个 APB master 接口，每个 APB master 接口上可以连接一个 RISC-V Core 作为 APB slave。上层调试软件通过识别第  $n$  个 Slave 设备 DM 中的寄存器来获取第  $(n+1)$  个 Slave 的访问偏移量，并通过 APB 总线访问。需要注意的是，E906 暂不支持多核异构调试。

### 9.1 端口列表

表 9.1: tdt\_dmi\_top 模块端口列表

信号名	I/O	Reset	时钟域	描述
pad_icg_scan_en	I	-	-	Scan 模式下用来控制 DTM 模块内部 ICG 的 TE 端, 功能模式下接 0
pad_yy_scan_mode	I	-	-	scan 模式选择信号, 功能模式下接 0
pad_yy_scan_rst_b	I	-	-	scan 模式下的复位信号
pad_tdt_dtm_jtag2_sel	I	-	TCLK	当选择玄铁两线调试接口时该信号线接 1, 否则需要接 0。
pad_tdt_dtm_tap_en	I	-	TCLK	用来使能 TAP 状态机, 建议固定接 1
pad_tdt_dtm_tclk	I	-	-	调试接口信号
pad_tdt_dtm_tdi	I	-	TCLK	5 线 JTAG 接口信号, 当选用两线调试接口时该信号可固定接 1。
pad_tdt_dtm_tms_i	I	-	TCLK	调试接口信号 TMS, 在两线调试接口下为输入的模式和数据信号。
pad_tdt_dtm_trst_b	I	-	TCLK	调试接口信号, 用于复位 DTM 中跟调试接口相关的逻辑资源。该信号可以不接到芯片引脚上。
tdt_dtm_pad_tdo	O	1' h1	TCLK	5 线 JTAG 接口信号。当选用两线调试接口时该信号可悬空处理。
tdt_dtm_pad_tdo_en	O	1' h0	TCLK	用来指示 TDO 有效状态: 该信号为 1 时 TDO 输出有效
tdt_dtm_pad_tms_o	O	1' h1	TCLK	两线调试接口数据的输出信号。
tdt_dtm_pad_tms_oe	O	1' h0	TCLK	两线调试接口数据输出有效指示信号。SoC 设计者可使用该信号将上述 pad_tdt_dtm_tms_i 和 tdt_dtm_pad_tms_o 信号组合成 inout 信号。
tdt_dmi_paddr[11:0]	O	-	SYS APB CLK	DMI APB 传输地址信号, 12 位宽。
tdt_dmi_penable	O	-	SYS APB CLK	DMI APB 传输使能信号
tdt_dmi_prdata[31:0]	I	32' h0	SYS APB CLK	DMI APB 读数据, 32 位宽。
tdt_dmi_pready	I	1' h0	SYS APB CLK	DMI APB 传输 ready 信号。
tdt_dmi_psel	O	-	SYS APB CLK	DMI APB Slave 选择信号。
tdt_dmi_pslverr	I	1' h0	SYS APB CLK	DMI APB 传输错误指示信号。
tdt_dmi_pwdata[31:0]	O	-	SYS APB CLK	DMI APB 写数据, 32 位宽。

下页继续

表 9.1 – 续上页

信号名	I/O	Reset	时钟域	描述
tdt_dmi_pwrite	O	-	SYS APB CLK	DMI APB 读写指示信号。

表 9.2: E906 CPU top 模块端口列表

信号名	I/O	Reset	时钟域	描述
APB 信号	I or O	-	CPU	与 tdt_dmi_top 一一对应
pad_tdt_dm_core_unavail		-	CPU	由 SoC 驱动，表示处理器不可调试，高电平有效。
cpu_pad_halted	O	1' h0	CPU	表示处理器已进入调试模式，高电平有效。

当处理器处于复位或者掉电等不可调试的状态时，SoC 可以把 pad\_tdt\_dm\_core\_unavail 拉高。这个信号的值会反映在调试寄存器里，调试软件读取该寄存器，即可知道 E906 当前处于不可调试的状态。

# 第十章 低功耗系统集成

## 10.1 端口列表

表 10.1: 低功耗端口信号列表

信号名	I/O	Reset	功能描述
sysio_pad_lpmd_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_lpmd_b[1:0] 被相应的改变： 00：深睡眠； 01：浅睡眠； 10：保留； 11：正常工作。
pad_cpu_wakeup_event	I	-	低功耗模式下事件唤醒请求，上升沿有效。 该信号会在 E906 内部使用 clk_en 采样后使用，需要维持至少一个系统时钟周期。

## 10.2 工作模式及其转换

E906 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 10.1 所示。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。

## 10.3 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\_lpmd\_b[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\_icg\_scan\_en 设置为 0。

SoC 在集成时需要把 sysio\_pad\_lpmd\_b[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。具体如 图 10.2 所示。

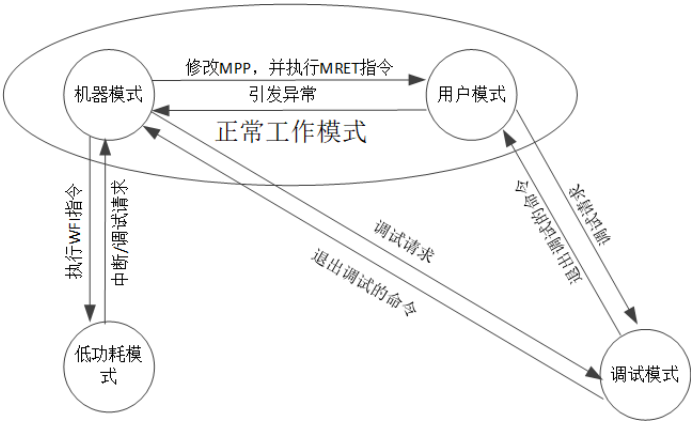


图 10.1: CPU 状态转换图

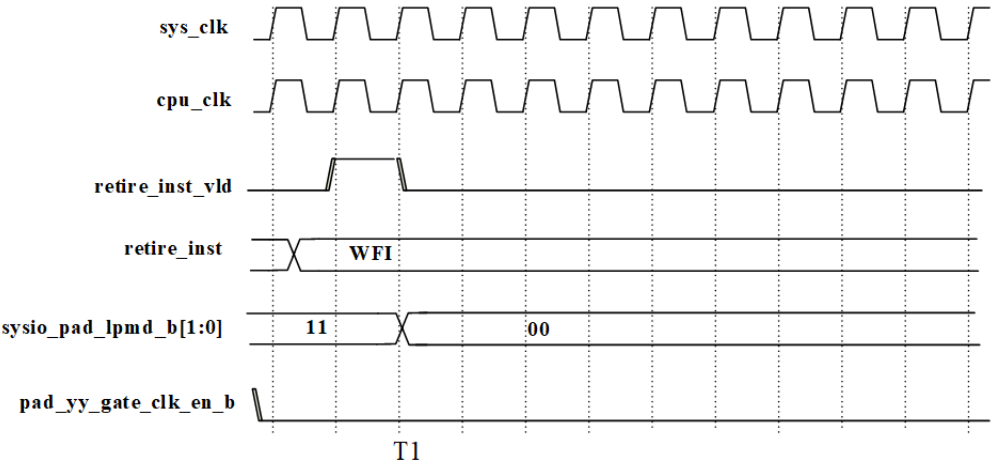


图 10.2: 进入低功耗模式握手

## 10.4 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_lpmc_b[1:0]` 来查询 CPU 状态。图 10.3 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_clc_int_vld[i-1:0]` 这组信号集成连接的中断来唤醒 CPU。

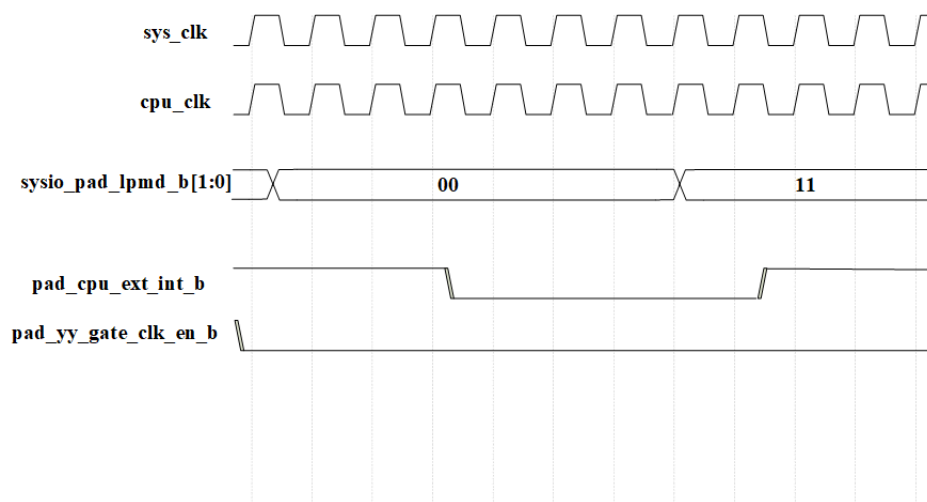


图 10.3: 退出低功耗模式握手

E906 支持外部事件信号 (`pad_cpu_wakeup_event`) 唤醒处于低功耗模式下的 CPU。通过该方式唤醒 CPU 后，CPU 不会响应中断，而是继续执行 WFI 指令后面的指令。用户在集成时不要将来自同一外设 IP 的请求既接入到 CLIC 中断控制器上又接到该信号上。

另外，调试请求和 NMI 请求都可以唤醒 CPU。

## 10.5 浅睡眠

E906 在低功耗模式下定义了深浅睡眠两种模式，通过 `sysio_pad_lpmc_b[1:0]` 信号来指示，当 `sysio_pad_lpmc_b[1:0]==2'b1` 时，表示 CPU 进入浅睡眠低功耗模式，该模式可由软件配置 E906 内部的 CSR 指定。

在浅睡眠模式下，`pll_core_cpucclk` 作用在 E906 内部绝大多数寄存器上的时钟都会被关闭，只保留跟唤醒逻辑相关的寄存器时钟开启。另外 `pad_had_jtgc_tclk` 作用的寄存器时钟不会被关闭，以便在低功耗模式下调试请求可以唤醒 CPU。

在浅睡眠模式下，系统设计者可以将 E906 进行降频，进一步降低 E906 的功耗。在这种情况下如果想通过外部调试器操作两线调试接口唤醒 CPU，需要保证 CPU 频率是调试器输出 `TCLK` 频率的两倍及以上，调试器最低输出频率为 10KHz。



## 10.6 深睡眠

E906 支持通过执行 WFI 指令进入深睡眠模式，通过 E906 顶层输出信号 `sysio_pad_lpm�_b[1:0]` 信号来指示，当 `sysio_pad_lpm�_b[1:0]==2'b0` 时，表示 CPU 进入深睡眠低功耗模式，该模式可由软件配置 E906 内部的 CSR 指定。

在 E906 进入深睡眠低功耗模式时，CPU 内部相比于浅睡眠模式对于时钟的处理完全相同。系统设计者可以根据该指示信号对 CPU 采取进一步的降低功耗的策略，比如降压，关闭 E906 输入时钟 (`pll_core_cpuclock`) 来消除 E906 内部动态功耗的耗费。

在 E906 输入时钟被关闭的情况下，如果想唤醒 CPU，需要保持唤醒信号，并先将 E906 的输入时钟恢复到正常，这样 CPU 才会从低功耗模式被唤醒。系统设计者可以在 CPU 外部设计针对 E906 的功耗管理模块，并负责采样唤醒请求。当 E906 进入深睡眠低功耗模式时，关闭 E906 的输入时钟。在 E906 处于低功耗模式下去采样 CPU 的唤醒请求信号并传递给 CPU，一旦采样到有效信号，可以先开启 E906 的输入时钟，然后其传递给 CPU 的唤醒请求信号会将 CPU 唤醒。该模块在观测到 CPU 从低功耗模式被唤醒后可以清除其内部的采样寄存器。

# 第十一章 杂类信号集成

## 11.1 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。



## 11.2 通用观测信号

表 11.1: 通用观测信号

信号名	I/O	Reset	定义
rtu_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。
rtu_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。
rtu_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。
rtu_pad_wb0_vld	O	0	通用寄存器回写指示信号 0: 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。
rtu_pad_wb0_preg[4:0]	O	-	寄存器索引 0: 表示当前回写的通用寄存器的索引。
rtu_pad_wb0_data[31:0]	O	-	寄存器回写总线数据 0: 表示当前回写通用寄存器的值。
rtu_pad_wb1_vld	O	0	通用寄存器回写指示信号 1: 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。
rtu_pad_wb1_preg[4:0]	O	-	寄存器索引 1: 表示当前回写的通用寄存器的索引。
rtu_pad_wb1_data[31:0]	O	-	寄存器回写总线数据 1: 表示当前回写通用寄存器的值。
rtu_pad_wb_freq_vld	O	0	通用寄存器回写指示信号 1: 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。
rtu_pad_wb_freq[4:0]	O	-	寄存器索引 1: 表示当前回写的通用寄存器的索引。
rtu_pad_wb_freq_data[i:0] i=31 或 63	O	-	寄存器回写总线数据 1: 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_mcause[31:0]	O	-	表示当前 mcause 值。
cp0_pad_mintstatus[31:0]	O	-	表示处理器当前中断等级, 没有配置 CLIC 时, 该信号不存在。 具体参考《玄铁 E906R1S3 用户手册》。
lsu_pad_sc_pass	O	-	CPU 成功执行需要 LOCK 总线的写操作请求。
cpu_pad_lockup	O	0	高有效, 表明 CPU 正在被锁定, 建议系统设计者用其将 CPU 进行复位处理。

11.3 观测信号示例波形

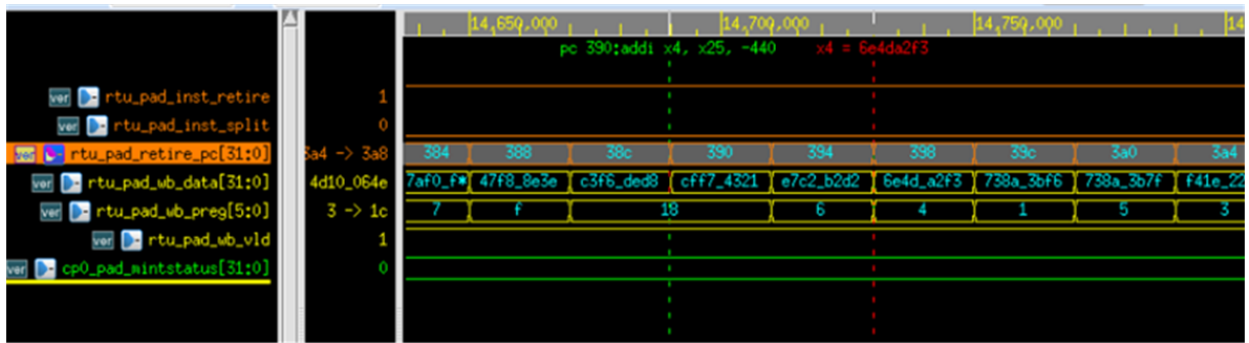


图 11.1: 观测信号示例波形

图 11.1 所示，左侧光标所指位置 `rtu_pad_retire_pc[31:0]`，当前退休 pc 为 `0x390`，其指令为 `addi x4, x25, -440`，其计算结果为 `x4` 的值为 `0x6e4d_a2f3`，在右侧光标位置可看到 `rtu_pad_wb_preg[5:0]` 为 `4`，表示回写目标寄存器标号为 `4`，回写值 `rtu_pad_wb_data[31:0]` 为 `0x6e4d_a2f3`。注意由于 E906 指令退休位于流水线 EX1 级，而寄存器回写操作位于流水线 WB 级，所以从观察到退休至观察到回写值，差了两级流水线。

## 第十二章 DFT 信号集成

### 12.1 端口列表

表 12.1: DFT 相关信号列表

信号名	I/O	Reset	功能描述
pad_yy_scan_enable	I	NA	scan 模式指示信号： 该信号在 E906 处于功能模式下需置为 0。
pad_yy_scan_mode	I	NA	scan 模式指示信号： 该信号在 E906 处于功能模式下需置为 0。
pad_yy_icg_scan_en	I	NA	该信号为 scan_enable 信号，用于 E906 内部 ICG TE 端口的使能，在功能模式下该信号需接 0，在 SCAN 模式下，该信号需接到 scan_enable 信号。
pad_tdt_icg_scan_en	I	NA	scan mode 下 dmi 模块内部 ICG 的控制信号： 用于控制 ICG 的 TE 端。
pad_yy_scan_rst_b	I	NA	scan 模式下的复位信号，低有效。功能模式下应接为 1。
pad_yy_dft_clk_rst_b	I	NA	scan mode 下 E906 内部 CLIC 二分频逻辑复位控制信号，该复位信号只需接到上电复位端，scan 模式下需要保持高电平。