

# 玄铁 C906 集成手册 (openc906)

2021 年 10 月 19 日

**Copyright 2021 T-Head Semiconductor Co., Ltd.**

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

版本	描述	日期
01	openc906 第一版发布。	2021.10.18

# 目录

<b>第一章 概述</b>	<b>1</b>
1.1 处理器简介 . . . . .	1
<b>第二章 数据包说明</b>	<b>2</b>
2.1 C906 数据包文件结构 . . . . .	2
2.2 C906 数据包使用环境 . . . . .	2
2.3 可配置选项 . . . . .	2
2.4 代码目录结构 . . . . .	3
2.5 Smart 平台仿真流程介绍 . . . . .	3
2.5.1 测试用例 . . . . .	3
2.5.2 Smart 平台 RTL 编译 . . . . .	4
2.5.3 测试用例执行 . . . . .	4
<b>第三章 Memory、ICG 替换</b>	<b>5</b>
3.1 Memory 替换 . . . . .	5
3.1.1 需作替换处理文件所在位置 . . . . .	5
3.1.2 Memory 规格 . . . . .	5
3.1.3 Memory 端口说明 . . . . .	5
3.1.4 Memory 控制与行为描述 . . . . .	6
3.1.5 拼接 Memory . . . . .	7
3.2 替换后的验证 . . . . .	7
<b>第四章 地址空间属性设置</b>	<b>9</b>
<b>第五章 集成总览</b>	<b>11</b>
5.1 命名规则 . . . . .	12
5.2 端口信号列表 . . . . .	12
<b>第六章 时钟和复位</b>	<b>21</b>
6.1 时钟概览 . . . . .	21
6.2 时钟分频 . . . . .	22
6.2.1 时钟调频 . . . . .	23
6.3 多时钟域信号同步 . . . . .	23
6.3.1 CPU 内核时钟域与系统总线时钟域 . . . . .	23

6.3.2	CPU 内核时钟域与 SYS APB 时钟域 . . . . .	23
6.3.3	SYS APB 时钟域与调试接口 TCLK 时钟域 . . . . .	23
6.4	复位信号 . . . . .	24
6.4.1	复位启动地址 . . . . .	26
6.4.2	软复位 . . . . .	26
<b>第七章</b>	<b>总线系统集成</b>	<b>27</b>
7.1	AXI 主设备接口总线概述 . . . . .	27
7.1.1	读传输 . . . . .	27
7.1.1.1	读地址通道 . . . . .	28
7.1.1.2	读数据通道 . . . . .	28
7.1.2	写传输 . . . . .	28
7.1.2.1	写地址通道 . . . . .	28
7.1.2.2	写数据通道 . . . . .	28
7.1.2.3	写响应通道 . . . . .	29
7.1.3	大小端 . . . . .	29
7.2	调试主设备接口总线概述 . . . . .	29
<b>第八章</b>	<b>中断系统集成</b>	<b>31</b>
8.1	中断处理过程简述 . . . . .	31
8.1.1	端口列表 . . . . .	31
8.1.2	中断握手时序图 . . . . .	32
<b>第九章</b>	<b>调试系统集成</b>	<b>33</b>
9.1	端口列表 . . . . .	35
<b>第十章</b>	<b>低功耗系统集成</b>	<b>37</b>
10.1	端口列表 . . . . .	37
10.2	工作模式及其转换 . . . . .	37
10.2.1	进入低功耗模式握手 . . . . .	37
10.2.2	退出低功耗模式握手 . . . . .	37
<b>第十一章</b>	<b>DFT 相关信号</b>	<b>39</b>
<b>第十二章</b>	<b>CPU 运行观测信号</b>	<b>40</b>

# 第一章 概述

## 1.1 处理器简介

玄铁 C906 处理器是嵌入式领域的 RISC-V 兼容的高能效低成本的 64 位处理器。(openc906) 采用 RV64GC 指令集架构，主要面向视频监控、智能终端、扫脸支付等领域。图 1.1 为 C906 的系统框图。

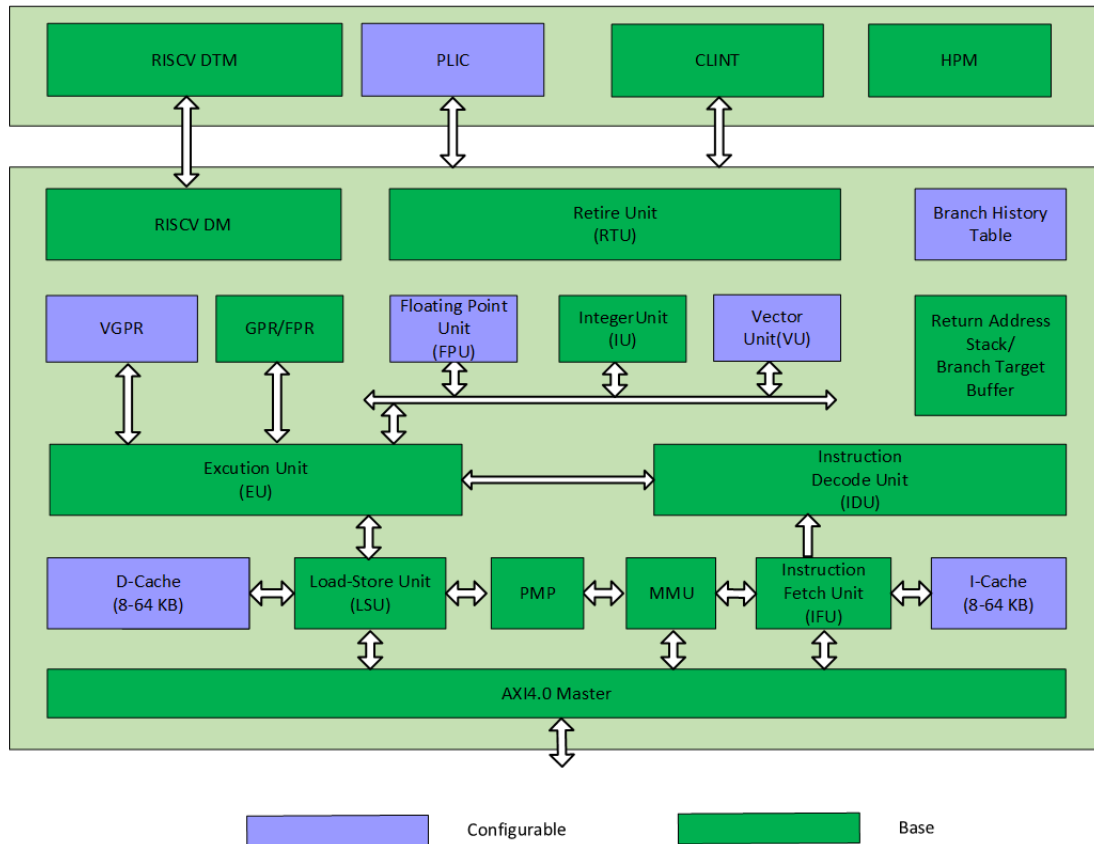


图 1.1: C906 系统框图

## 第二章 数据包说明

### 2.1 C906 数据包文件结构

玄铁 C906 的数据包由以下两部分组成：

- CPU 代码包 (C906\_RTL\_FACTORY)：固定功能配置的 C906 RTL 代码
- Smart 平台包 (smart\_run)：提供了 C906 的参考集成设计、仿真环境和测试用例，帮助客户熟悉 C906 的功能和使用方法，并辅助 C906 的集成工作。功能包括：
  - 1) C906 的参考集成设计；
  - 2) 丰富的测试用例；
  - 3) 基础的仿真环境；
  - 4) 后端综合实现的参考 SDC 约束文件；

### 2.2 C906 数据包使用环境

在使用 C906 之前，请检查运行环境是具备以下条件：

- 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平头哥玄铁处理器 RISC-V 工具链 2.0.3 及以上版本。

### 2.3 可配置选项

C906 当前的固定配置选项如 表 2.1 所示。

表 2.1: C906 当前配置

可配置单元	配置选项	openc906 当前配置
矢量单元	无/有	无
浮点单元	无/半 + 单 + 双精度浮点	无
L1 ICache	8KB/16KB/32KB/64KB	32KB
L1 DCache	8KB/16KB/32KB/64KB	32KB
PMP 表项数	8/16	8
jTLB (MMU)	128/256/512	128
BHT	8Kb/16Kb	168Kb
中断数量	32-1023	128
调试等级	最小配置/典型配置 /最大配置	典型配置
Next DM Address	下一个 DM 的基地址	32' h0
EMA 接口	无/有 (1-32 位宽)	无

## 2.4 代码目录结构

C906 CPU 的 RTL 代码都位于 `gen_rtl` 目录下:

`gen_rtl` 下面直接存放的, 是 C906 CPU core 和 RISC-V DTM+DMI 模块的 RTL 代码。其中 `openc906.v` 包含了 C906 CPU 的顶层模块和绝大多数的子模块;

`gen_rtl/INST_LIB` 目录下存放的是 C906 CPU 中涉及工艺库相关 cell 例化的全部模块, 例如例化了 `memory model` 和 `icg model` 的模块。这些模块在前端仿真中可以直接使用, 在后端流程中请参照工艺库的实际情况, 为该文件夹下的全部模块等价替换工艺库相关的 cell;

`gen_rtl/LIB` 目录下存放的是 DesignWare 的乘法器, 以及默认工艺库相关的 cell 的仿真模型, 例如 `memory model` 和 `icg model`;

`gen_rtl/filelists` 目录下存放的是 C906 CPU 的文件列表;

注意: `INST_LIB` 与 `LIB` 中包含的文件与模块, 可能会多于 CPU 当前配置的需要。

## 2.5 Smart 平台仿真流程介绍

### 2.5.1 测试用例

执行如下命令, 可以列出所有当前 C906 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（测试用例名可从上述 `make showcase` 命令的执行结果中挑选）

需要注意，Smart 平台依赖环境变量 `$TOOL_EXTENSION` 来选择编译测试用的 RISC-V 编译器，Smart 平台在目录 `C906_SMART_FACTORY/setup/` 下，提供了名为 `example_setup.csh` 的脚本作为参考，举例如何设置 `$TOOL_EXTENSION`。

### 2.5.2 Smart 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 E906 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` 中的提示信息。

### 2.5.3 测试用例执行

可以用以下命令，直接运行 `Testbench` 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

执行 Smart 平台提供的全部测试用例的命令为：

```
make regress
```

更多具体选项，详见 `make help` 中的提示信息。

## 第三章 Memory、ICG 替换

IP 集成时, Memory 和 ICG 模块需要按照实际使用的工艺库的要求进行替换, 包括 ASIC 集成或 FPGA 集成。本章介绍了在集成 C906 CPU IP 时, 工艺库相关的 cell 的替换方法。

### 3.1 Memory 替换

#### 3.1.1 需作替换处理文件所在位置

所有需要修改的文件都在 C906 RTL Database 下 gen\_rtl/INST\_LIB 目录, 且文件名都包含关键词 aq\_spsram\_AAAxBBB, A、B 分别代表 memory 的深度和宽度。

#### 3.1.2 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 AAAxBBB 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。

#### 3.1.3 Memory 端口说明

以特定工艺 memory 为例, 如 图 3.1 所示。

```
sprf06511_512x64 x_spsram_512x64(  
  .A      (A),  
  .D      (D),  
  .BWEN   ({32{WEN[1]}}, {32{WEN[0]}}),  
  .WEN     (&WEN),  
  .CEN     (CEN),  
  .CLK     (CLK),  
  .Q       (Q)  
);
```

图 3.1: Memory 例化

Memory port 信号列表如 表 3.1 所示。

表 3.1: 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，具体揭发要看 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 C906 顶层进行控制，C906 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 C906 顶层接口对其进行控制。

### 3.1.4 Memory 控制与行为描述

图 3.2 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。

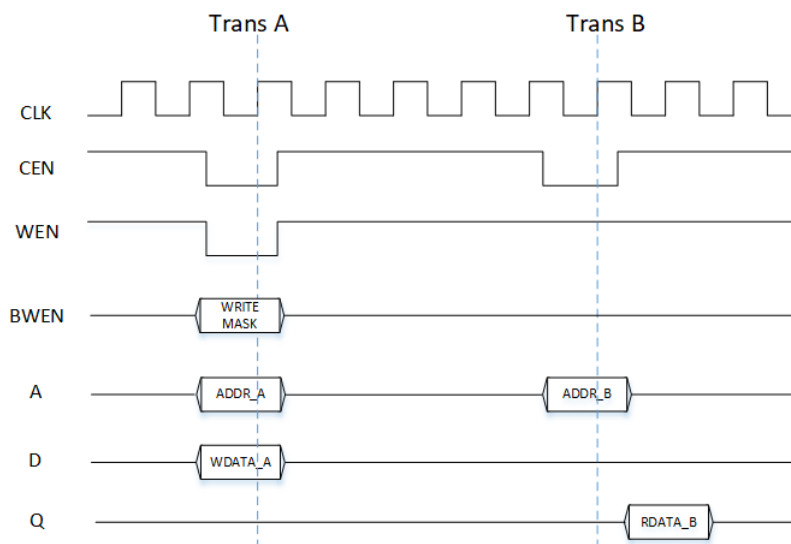


图 3.2: Memory 读写时序

### 3.1.5 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 3.3 所示。

```
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 3.3: Memory 拼接示例

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 3.4 所示。

## 3.2 替换后的验证

替换完成后，可以使用 Smart 平台提供的 MEM\_INTF 测试用例，协助检测替换结果是否正确。MEM\_INTF 需要在 smart\_run 成功生成后运行，检测步骤是：

1. cd ./smart\_run/impl/MEM\_INTF
2. 参考./smart\_run/impl/MEM\_INTF/readme，按照替换后的实际情况修改 filelist（包括 filelist 里引用的 filelist）
3. ./run\_mem\_icg\_test
4. 查看 log 信息，确保所有 cell 的替换成功

MEM\_INTF 只检测了所有 cell 的初步功能，来辅助替换过程。为了确保 cell 替换后 C906 CPU IP 的功能正确，用户仍然需要用做 LEC 检查和 SoC 级的验证完成进一步的测试。

```

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];
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 3.4: Memory 拼接示例

## 第四章 地址空间属性设置

C906 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 4.1 给出了各个内存类型对应的页面属性。页面属性的配置有两种方式:

1. 在所有不进行虚拟地址和物理地址转换的情况下: 机器模式权限或者 MMU 关闭, 地址的页面属性由 sysmap.h 决定。sysmap.h 是 C906 扩展的地址属性配置文件, 对用户开放, 用户可以根据自身需求, 定义不同地址段的页面属性, 地址区域个数上限为 8。
2. 在所有进行虚拟地址和物理地址转换的情况下: 非机器模式权限且 MMU 打开时, 地址的页面属性有两种配置方式: sysmap.h 和 C906 在 pte 中扩展的页面属性, 具体取决于 C906 扩展寄存器 mxstatus 中的 mace 位是否打开。如果 mace 打开, 地址的页面属性由对应 pte 中扩展的页面属性决定。如果 mace 关闭, 地址的页面属性由 sysmap.h 决定。

表 4.1: 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

sysmap.h 文件支持对 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_ADDRi` ( $i$  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_ADDRi`

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 strong order/non cacheable/non bufferable。每个地址空间上下边界是 4KB 对齐, 因此宏 `SYSMAP_BASE_ADDRi` 定义的是地址的高 28 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_FLAGi` ( $i$  从 0 到 7) 定义, 属性的排布如 图 4.1 所示。

4	3	2	1	0
Strong order	Cacheable	Bufferable	-	

图 4.1: `sysmap.h` 地址属性排布

# 第五章 集成总览

为了降低 C906 的硬件集成复杂度，除了总线接口部分的信号外，其他信号在平头哥的多款处理器中，均保持了稳定的延续性，以方便用户的升级换代。根据 C906 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。

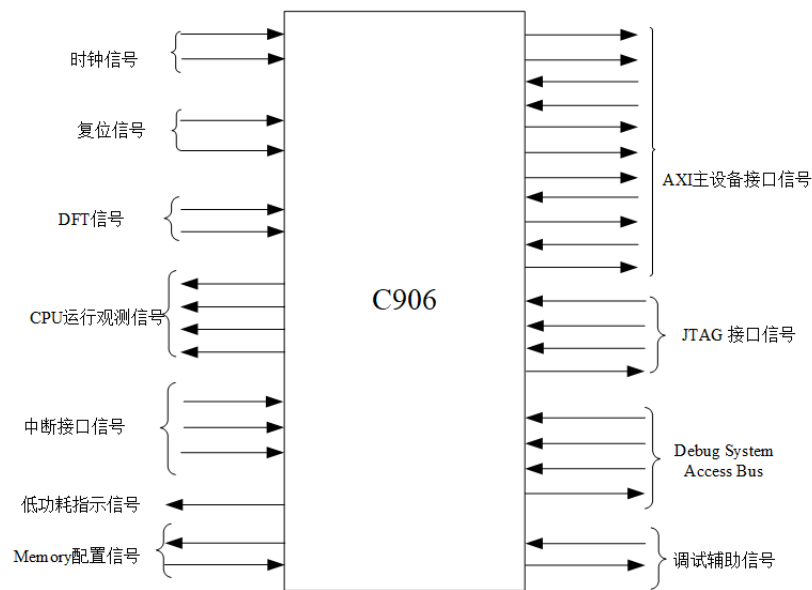


图 5.1: C906 接口框图



### 5.1 命名规则

表 5.1: 信号命名规则

信号名前缀	描述
pad_biu_*	输入信号。
pad_had_*	
pad_cpu_*	
pad_corex_*	
biu_pad_*	输出信号。
had_pad_*	
cpu_pad_*	
corex_pad_*	
*_b	低电平有效信号。
*_n	

一般来说，没有特殊规定，C906 的输入输出信号均为高电平有效，但是注意，如果是以 “\_b” 或者 “\_n” 结尾的信号，则是低电平有效。

### 5.2 端口信号列表

端口信号列表如 表 5.2 所示。

表 5.2: 端口信号列表

信号名	I/O	初始值	时钟域	功能描述
<b>时钟复位信号集成：</b>				
时钟相关信号				
pll_core_cpucclk	I	-	-	CP U 工作时钟信号： 提供 C906 内部除调试模块外逻辑的时钟输入。
sys_apb_clk	I	-	-	CPU 内调试模块工作时钟
axim_clk_en	I	-	CPU	C906 AXI 接口信号与外部总线信号时钟同步使能信号。
复位相关信号				
pad_cpu_rst_b	I	-	CPU 异步	CPU 复位信号： 低电平有效，功能模式（非测试模式）下复位除调试模块外 C90 6 内部所有逻辑。 该信号为 pll_core_c puclk 的异步时钟信号，C906 内部实现同步处理。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
sys_apb_rst_b	I	-	SYS APB 异步	CPU 内调试模块复位信号： 低电平有效，功能模式（非测试模式）下复位 C 906 内部调试模块相关所有逻辑。 该信号为 sys_ap_b_clk 的异步时钟信号，C906 内部实现同步处理。
pad_cpu_rvba[39:0]	I	-	-	C906 复位启动地址指示信号： pad_cpu_rvba[39:1] 为有效位，要求系统集成时绑成固定值。最低位需要接 0。
tdt_dm_pad_hartreset_n	O	1' b1	SYS APB	C 906 内调试模块输出的处理器核心复位指示信号。
tdt_dm_pad_ndmreset_n	O	1' b1	SYS APB	C906 内调试模块输出的除 DM 外的全局复位指示信号。
<b>AXI 主接口总线系统集成：</b>				
biu_pad_araddr[39:0]	O	40' h0	SYS AXI	读地址总线： 40 位地址总线。
biu_pad_arburst[1:0]	O	2' h0	SYS AXI	突发传输类型指示信号： 01: INCR; 10: WRAP。
biu_pad_arcache[3:0]	O	4' h0	SYS AXI	读请求对应的 cache 属性： [3]:Other Allocate; [2]:Allocate; [1]:Modifiable; [0]:Bufferable。
biu_pad_arid[7:0]	O	8' h0	SYS AXI	读地址 ID。
biu_pad_arlen[7:0]	O	8' h0	SYS AXI	突发传输长度： 00000000: 1 拍; 00000011: 4。
biu_pad_arlock	O	1' h0	SYS AXI	读请求对应的访问方式： 0: normal access; 1: exclusive access。
biu_pad_arprot[2:0]	O	3' h0	SYS AXI	读请求的保护类型： 0   1 [2]: Data   Instruction [1]: Secure   Non-secure

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
biu_pad_arsize[2:0]	O	3' h0	SYS AXI	读请求每拍数据位宽： 000: 8bits; 001:16bits; 010:32bits; 011:64bits; 100: 128bits。
biu_pad_arvalid	O	1' h0	SYS AXI	读地址有效信号。
pad_biu_arready	O	-	SYS AXI	读地址通道 ready 信号。
biu_pad_rdata[127:0]	O	128' h0	SYS AXI	读数据总线： 128 位数据总线。
pad_biu_rid[7:0]	I	8' h0	SYS AXI	读数据 ID。
pad_biu_rresp[1:0]	I	2' h0	SYS AXI	读响应信号： 00: OKAY 01: EXOKAY 10: SLVERR 11: DECERR
pad_biu_rlast	I	-	SYS AXI	读数据最后一拍指示信号。
pad_biu_rvalid	I	-	SYS AXI	读数据有效信号。
biu_pad_rready	O	1' h1	SYS AXI	读数据通道 ready 信号。
biu_pad_awaddr[39:0]	O	40' h0	SYS AXI	写地址总线： 40 位地址总线。
biu_pad_awburst[1:0]	O	2' h0	SYS AXI	突发传输类型指示信号： 01: INCR
biu_pad_awcache[3:0]	O	4' h0	SYS AXI	写请求对应的 cache 属性： [3]:Other Allocate [2]:Allocate [1]:Modifiable [0]:Bufferable
biu_pad_awid[7:0]	O	8' h0	SYS AXI	写地址 ID。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
biu_pad_awlen[7:0]	O	8' h0	SYS AXI	突发传输长度： 00000000: 1 拍; 00000011: 4 拍。
biu_pad_awlock	O	1' h0	SYS AXI	写请求对应的访问方式： 0: normal access 1: exclusive access
biu_pad_awprot[2:0]	O	3' h0	SYS AXI	写请求的保护类型： 0   1 [2]: Data   Instruction [1]: Secure   Non-secure [0]: User   Privileged
biu_pad_awsz[2:0]	O	3' h0	SYS AXI	写请求每拍数据位宽： 000: 8bits; 001: 16bits; 010: 32bits; 011: 64bits; 100: 128bits。
biu_pad_awvalid	O	1' h0	SYS AXI	写地址有效信号。
pad_biu_awready	I	-	SYS AXI	写地址通道 ready 信号。
biu_pad_wdata[127:0]	O	128' h0	SYS AXI	写数据总线： 128 位数据总线。
biu_pad_wvalid	O	1' h0	SYS AXI	写数据有效信号。
pad_biu_wready	I	-	SYS AXI	写数据通道 ready 信号。
pad_biu_bid[7:0]	I	-	SYS AXI	写响应 ID。
pad_biu_bresp[1:0]	I	-	SYS AXI	写响应信号： 00: OKAY; 01: EXOKAY; 10: SLVERR; 11: DECERR。
pad_biu_bvalid	I	-	SYS AXI	写响应有效信号。
biu_pad_bready	I	1' h1	SYS AXI	写响应通道 ready 信号。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
中断系统集成：				
pad_cpu _apb_base[39:0]	I	-	-	C906 内部 PLIC 控制器等内存映射寄存器的基址信号；pad_cpu_apb_base[39:27] 为有效位，低 27 位需接 0。
pad_plic _int_vld[i-1:0]	I	-	CPU 异步	外部中断源的中断指示信号，表示外部中断源的中断是否有效： 1：表示中断有效； 0：表示中断无效。 该信号为 pll_core_cpucclk 的异步时钟信号，C906 内部完成同步处理
pad_plic _int_cfg[i-1:0]	I	-	-	外部中断源的中断配置信号，表示外部中断源的中断触发类型： 1：表示脉冲中断； 0：表示电平中断。 集成时该信号需要接固定值。
pad_cpu_sys_cnt[63:0]	I	-	SYS AXI	外部系统计时器的计数值，该信号需要由外部同步到 SYS 时钟域
调试系统集成				
RISC-V Debug System BUS Access（以下简称 SBA）接口信号				
tdt_dm_p ad_araddr[39:0]	O	40' h0	SYS AXI	SBA 读地址总线： 40 位地址总线
tdt_dm_p ad_arburst[1:0]	O	2' h1	SYS AXI	SBA 突发传输类型指示信号： 固定为 2' h1。
Tdt_dm_p ad_arcache[3:0]	O	4' h0	SYS AXI	SBA 读请求 Memory 属性： 固定为 4' h0。
tdt_d m_pad_arid[3:0]	O	4' h0	SYS AXI	SBA 读地址 ID： 固定为 4' h0。
tdt_dm _pad_arlen[3:0]	O	4' h0	SYS AXI	SB A 读突发传输长度，固定为 4' h0。
td t_dm_pad_arlock	O	1' h0	SYS AXI	SBA 读访问类型，固定为 2' h0。
tdt_dm_pad _arprot[2:0]	O	3' h2	SYS AXI	SB A 读访问保护类型，固定为 3' h2。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
tdt_dm_pad_ar-size[2:0]	O	3' h0	SYS AXI	SBA 读请求每拍数据位宽： 3' b000:8bits 3' b001:16bits 3' b010:32bits 3' b011:64bits 3' b100:128bits
tdt_dm_pad_arvalid	O	1' h0	SYS AXI	SBA 读地址有效信号。
tdt_dm_pad_ad_awaddr[39:0]	O	40' h0	SYS AXI	SBA 写地址总线： 40 位地址总线
tdt_dm_pad_ad_awburst[1:0]	O	2' h1	SYS AXI	SB A 写突发传输类型，固定为 2' h1。
tdt_dm_pad_ad_awcache[3:0]	O	4' h0	SYS AXI	SBA 写请求 Memory 属性，固定为 4' h0。
tdt_dm_pad_ad_awid[3:0]	O	4' h0	SYS AXI	SBA 写地址 ID： 固定为 4' h0。
tdt_dm_pad_ad_awlen[3:0]	O	4' h0	SYS AXI	SB A 写突发传输长度，固定为 4' h0。
tdt_dm_pad_ad_awlock	O	1' h0	SYS AXI	SBA 写访问类型，固定为 2' h0。
tdt_dm_pad_ad_awprot[2:0]	O	3' h2	SYS AXI	SB A 写访问保护类型，固定为 3' h2。
tdt_dm_pad_ad_awsize[2:0]	O	3' h0	SYS AXI	SBA 写请求每拍数据位宽： 3' b000:8bits 3' b001:16bits 3' b010:32bits 3' b011:64bits 3' b100:128bits
tdt_dm_pad_ad_awvalid	O	1' h0	SYS AXI	SBA 写地址有效信号。
tdt_dm_pad_bready	O	1' h1	SYS AXI	SBA 写响应 ready 信号。
tdt_dm_pad_rready	O	1' h1	SYS AXI	SBA 读数据 ready 信号。
tdt_dm_pad_ad_wdata[127:0]	O	128' h0	SYS AXI	SBA 写数据，128 位数据位宽。
tdt_dm_pad_wlast	O	1' h1	SYS AXI	SBA 写传输结束信号。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
tdt_dm_pad _wstrb[15:0]	O	16' h0	SYS AXI	SBA 写数据字节有效信号。
tdt_dm_pad_wvalid	O	1' h0	SYS AXI	SBA 写数据有效信号。
pad_tdt_dm_arready	I	-	SYS AXI	SBA 读地址 ready 信号。
pad_tdt_dm_awready	I	-	SYS AXI	SBA 写地址 ready 信号。
pad_tdt_dm_bid[3:0]	I	-	SYS AXI	SBA 写响应 ID。
pad_tdt_dm_bresp[1:0]	I	-	SYS AXI	SBA 写响应类型。
pad_tdt_dm_bvalid	I	-	SYS AXI	SBA 写响应有效。
pad_tdt_dm_rdata[127:0]	I	-	SYS AXI	SBA 读数据, 128 位宽。
pad_tdt_dm_rid[3:0]	I	-	SYS AXI	SBA 读数据 ID。
pad_tdt_dm_rlast	I	-	SYS AXI	SBA 读数据传输结束信号。
pad_tdt_dm_rresp[1:0]	I	-	SYS AXI	SBA 读响应类型。
pad_tdt_dm_rvalid	I	-	SYS AXI	SBA 读数据有效信号。
pad_tdt_dm_wready	I	-	SYS AXI	SBA 写数据 ready 信号。
RISC-V DMI APB 接口				
tdt_dmi_paddr[11:0]	I	-	SYS APB	DMI APB 传输地址信号, 12 位宽。
tdt_dmi_penable	I	-	SYS APB	DMI APB 传输使能信号。
tdt_dmi_prdata[31:0]	O	32' h0	SYS APB	DMI APB 读数据, 32 位宽。
tdt_dmi_pready	O	1' h0	SYS APB	DMI APB 传输 ready 信号。
tdt_dmi_psel	I	-	SYS APB	DMI APB Slave 选择信号。

下页继续

表 5.2 – 续上页

信号名	I/O	初始值	时钟域	功能描述
tdt_dmi_pslverr	O	1' h0	SYS APB	DMI APB 传输错误指示信号。
tdt_dmi_pwdata[31:0]	I	-	SYS APB	DMI APB 写数据，32 位宽。
tdt_dmi_pwrite	II	-	SYS APB	DMI APB 读写指示信号。
调试支持信号				
core0_pad_halted	O	1' h0	CPU	CPU 处于低功耗状态指示信号。 1' b0: 1' b1:
core0_pad _lpmd_b[1:0]	O	2' h3	CPU	处理器核低功耗模式状态信号： 当处理器核执行 wfi 指令时， core0_pad_lpmd_b[1:0] 被相应的改变： 0 0: 低功耗莫斯； 11: 正常工作模式。
core0_pad_retire	O	1' h0	CPU	处理器指令退休指示信号： 0: 当前周期没有指令退休； 1: 当前周期有指令退休。
core0_pad_retire_pc[39:0]	O	-	CPU	处理器退休指令的 PC： 表明当前正在退休的指令的 PC。
Memory 访问控制信号				
mem_cfg_in[i-1:0]	I	-	-	核内 memory 访问控制输入信号。
mem_cfg_out[i-1:0]	O	-	-	核内 memory 访问控制输出信号。
DFT 相关信号				
pad_y y_dft_clk_rst_b	I	-	CPU 异步	scan mode 下 C906 内部二分频逻辑复位控制信号。
pad_yy_icg_scan_en	I	-	CPU 异步	scan mode 下 C906 内部 ICG 的控制信号。 用于控制 ICG 的 TE 端。
pad_yy_mbist_mode	I	-	CPU 异步	mbist mode 控制信号，用于将 CPU 内部复位逻辑置为 0。
pad_yy_scan_enable	I	-	CPU 异步	扫描链的使能控制信号，高电平有效。 功能模式下该信号需要接 0。
pad_yy_scan_mode	I	-	CPU 异步	scan mode 控制信号。 功能模式下该信号需要接 0。
pad_yy_scan_rst_b	I	-	CPU 异步	scan mode 下 C906 内部所有逻辑（除 C906 内部用于生成 PLIC 二分频时钟所用的 ICG）的复位控制信号。

TDT DMI 端口信号列表如 表 5.3 所示。



表 5.3: TDT DMI 端口信号列表

RISC-V Debug DTM DMI 接口信号	I/O	初始值	时钟域	功能描述
pad_tdt_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
pad_yy_scan_rst_b	I	-	-	Scan 模式下的复位控制信号
pad_tdt_dtm_jtag2_sel	I	-	TCLK	固定接 0
pad_tdt_dtm_tap_en	I	-	TCLK	用来使能 T AP 状态机, 建议固定接 1
pad_tdt_dtm_tclk	I	-	-	JTAG 接口信号
pad_tdt_dtm_tdi	I	-	TCLK	JTAG 接口信号
pad_tdt_dtm_tms_i	I	-	TCLK	JTAG 接口信号 TMS
pad_tdt_dtm_trst_b	I	-	TCLK	JTAG 接口信号
tdt_dtm_pad_tdo	O	1' h1	TCLK	JTAG 接口信号
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	可悬空处理
tdt_dtm_pad_tms_oe	O	1' h0	TCLK	可悬空处理
tdt_dmi_paddr	O	12' h0	SYS APB	DMI AP B 传输地址信号, 12 位宽
tdt_dmi_penable	O	1' h0	SYS APB	DMI APB 传输使能信号
tdt_dmi_prdata	I	-	SYS APB	DMI APB 读数据信号, 当 DTM DMI 模块接入多个调试资源时 (如 C906 和 E907), 该信号位宽为 $n \times 32$ , 其中 $n$ 是调试资源的个数 (如 1 个 C906, 1 个 E907, $n=2$ )
tdt_dmi_pready	I	-	SYS APB	DMI AP B 传输 ready 信号, 当 DTM DMI 模块接入多个调试资源时, 该信号位宽为 $n$ , 其中 $n$ 是调试资源的个数
tdt_dmi_psel	O	M' h0	SYS APB	DMI A PB 传输选择信号, 当 DTM DMI 模块接入多个调试资源时, 该信号位宽为 $n$ , 其中 $n$ 是调试资源的个数
tdt_dmi_pslverr	I	-	SYS APB	DMI A PB 传输响应信号, 当 DTM DMI 模块接入多个调试资源时, 该信号位宽为 $n$ , 其中 $n$ 是调试资源的个数
tdt_dmi_pwdata	O	32' h0	SYS APB	DMI APB 写数据信号
tdt_dmi_pwrite	O	1' h0	SYS APB	DMI APB 读写指示信号

## 第六章 时钟和复位

### 6.1 时钟概览

C906 的时钟概览如 图 6.1 所示。

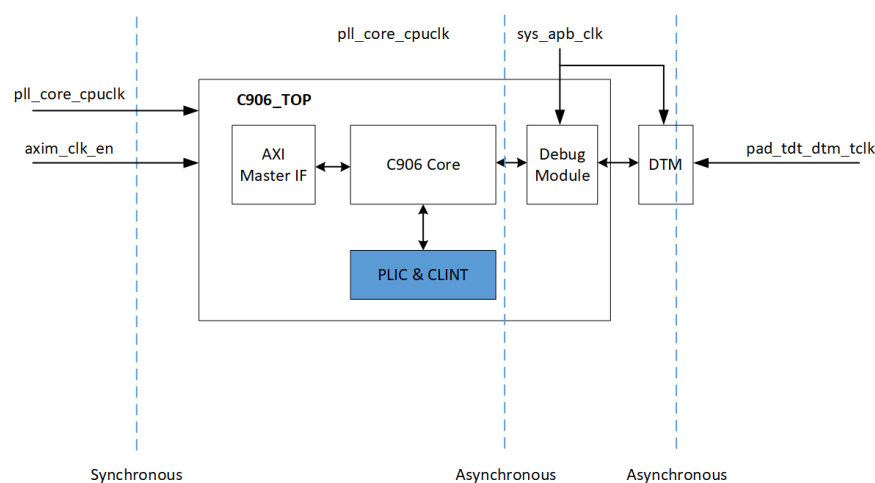


图 6.1: C906 的时钟概览

pll\_core\_cpucclk 是 C906 内除调试模块外的工作时钟，C906 支持 pll\_core\_cpucclk 是外部系统时钟的同频、2~8 倍频，要求 pll\_core\_cpucclk 与系统时钟同源同相。axim\_clk\_en 用于 C906 总线及计时器计数值等接口信号与系统时钟域进行时钟分频同步采样。

sys\_apb\_clk 是 C906 内部调试模块的工作时钟，它的频率需要比 pll\_core\_cpucclk 频率低，与 pll\_core\_cpucclk 是异步时钟。同时该时钟还被用作 DTM 模块内除 JTAG 接口逻辑外的工作时钟。

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。DTM 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 频率的两倍以上，由仿真器传递进来的 pad\_tdt\_dtm\_tclk 时钟频率最低为 100KHz。

## 6.2 时钟分频

C906 支持总线等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。axim\_clk\_en 需要由 SoC 根据分频比来产生对应的时序。由于 C906 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，C906 内部将该信号在 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。图 6.2 为 pll\_core\_cpuclock 与系统时钟频率比为 1:1~8:1 时 axim\_clk\_en 的时序图。

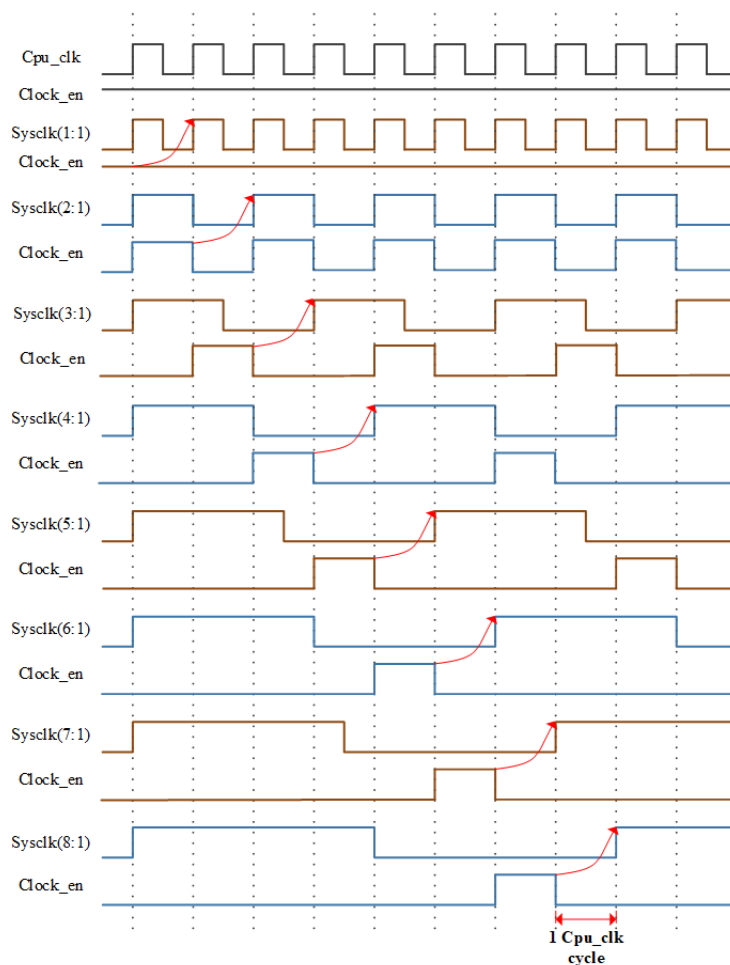


图 6.2: CPU 时钟与系统时钟域信号同步关系图

图 6.2 所示表明 axim\_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。

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 PLIC 和 CLINT 模块的寄存器翻转。外部中断信号为 pll\_core\_cpuclock 的异步时钟信号，C906 会在内部对中断信号进行同步处理。

### 6.2.1 时钟调频

C906 支持对输入时钟 `pll_core_cpucclk` 的调频操作，只要调频期间 C906 输入时钟 `pll_core_cpucclk` 不间断，则 C906 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，为了保证在调频时总线没有未完成的传输返回响应，可以在调频前通过让 C906 执行 WFI 指令的方式进入低功耗模式，保证在 `core0_pad_lpmdb[1:0]` 变为 2'b00 时 C906 发出的总线传输全部完成，也不会有新的传输发出。系统只需要在调频完成后通过如中断信号唤醒 CPU 即可。

## 6.3 多时钟域信号同步

### 6.3.1 CPU 内核时钟域与系统总线时钟域

在 C906 的设计中，CPU 时钟域与系统时钟域在顶层接口部分逻辑中有信号交互。这两个时钟要求在物理设计中，需要保证相位对齐。由于这两个信号只有频率整数倍关系，所以无需用专用的逻辑进行同步。C906 采样外部特定产生的控制信号 (`axim_clk_en`)，且 `axim_clk_en` 信号在 CPU 内部作打拍处理（图 6.3 展示的是 `axim_clk_en` 信号打拍后的时序图），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。因此从 CPU 总线到系统的信号都具有系统时钟的全周期时序延时。图 6.3 给出了 CPU 内核时钟与系统时钟比例在 4:1 的情况下的 CPU 内部信号变化与总线接口上的信号变化。

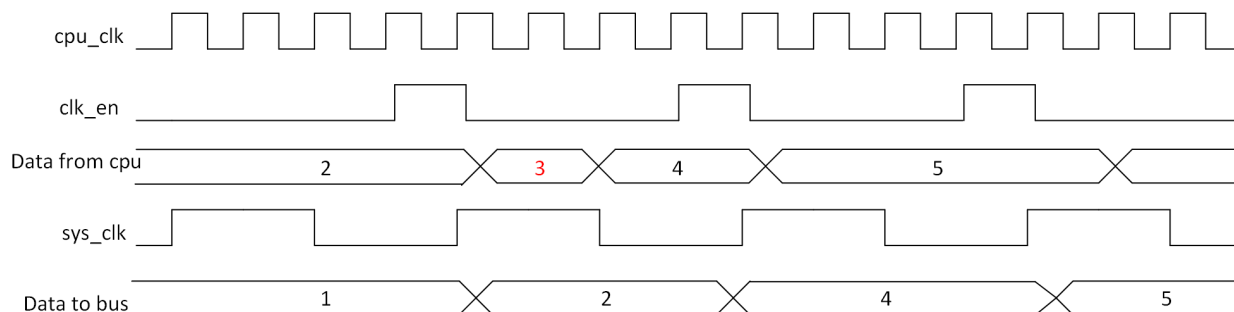


图 6.3: CPU 与系统时钟 4: 1 情况下的接口信号

### 6.3.2 CPU 内核时钟域与 SYS APB 时钟域

C906 内部除了流水线，中断控制器等逻辑单元，还有调试单元（Debug Module，以下简称 DM）用于调试模式下和核内的交互。DM 工作在 `sys_apb_clk` 时钟域下，该时钟为 `pll_core_cpucclk` 的异步时钟域。C906 内部将两个时钟域下的信号进行同步处理。其逻辑框图如图 6.4 所示。

### 6.3.3 SYS APB 时钟域与调试接口 TCLK 时钟域

调试传输模块（Debug Transport Module，以下简称 DTM）负责通过调试接口 JTAG 与上层软件交付。DTM 有两个输入时钟，`sys_apb_clk` 和 `pad_tdt_dtm_tclk`，两者为异步时钟。`sys_apb_clk` 与 C906 内

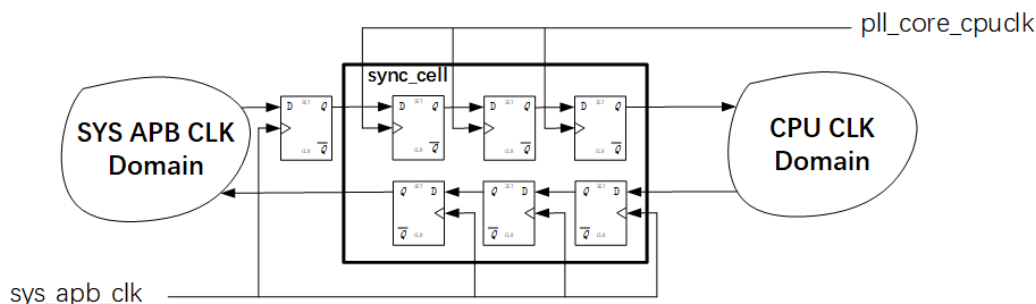


图 6.4: CPU 与 SYS APB 时钟域之间的同步逻辑

部 DM 的输入时钟为同一个，pad\_tdt\_dtm\_tclk 需要接到 SoC 的顶层接口，在调试时由外部调试器输入进来。在 DTM 内部实现了跨时钟域信号的同步处理。其逻辑框图如图 6.5 所示。

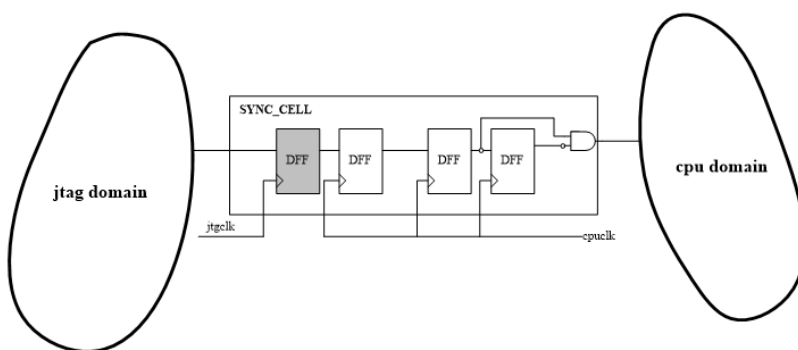


图 6.5: SYS APB 与 JTAG 时钟域之间的同步逻辑

## 6.4 复位信号

C906 输入的复位信号如下：

- pad\_cpu\_rst\_b

用于正常工作模式下 C906 内部流水线，总线接口等逻辑单元的复位控制以及用于 PLIC 二分频时钟 ICG 的复位控制。该信号在 C906 内部经过三级同步后使用。

- sys\_apb\_rst\_b

用于 C906 内部 DM 以及 DTM 模块工作在 sys\_apb\_clk 时钟域下寄存器的复位控制。该信号需要在 DM 模块外部由系统设计者同步到 sys\_apb\_clk 时钟域，然后再接入 DM 和 DTM 模块。

- pad\_yy\_scan\_rst\_b

用于 scan 模式下 C906 内部除了 PLIC 二分频时钟 ICG 外的逻辑的复位控制。也用于 scan 模式下 DTM 内除了 TAP 逻辑以外的复位控制。

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

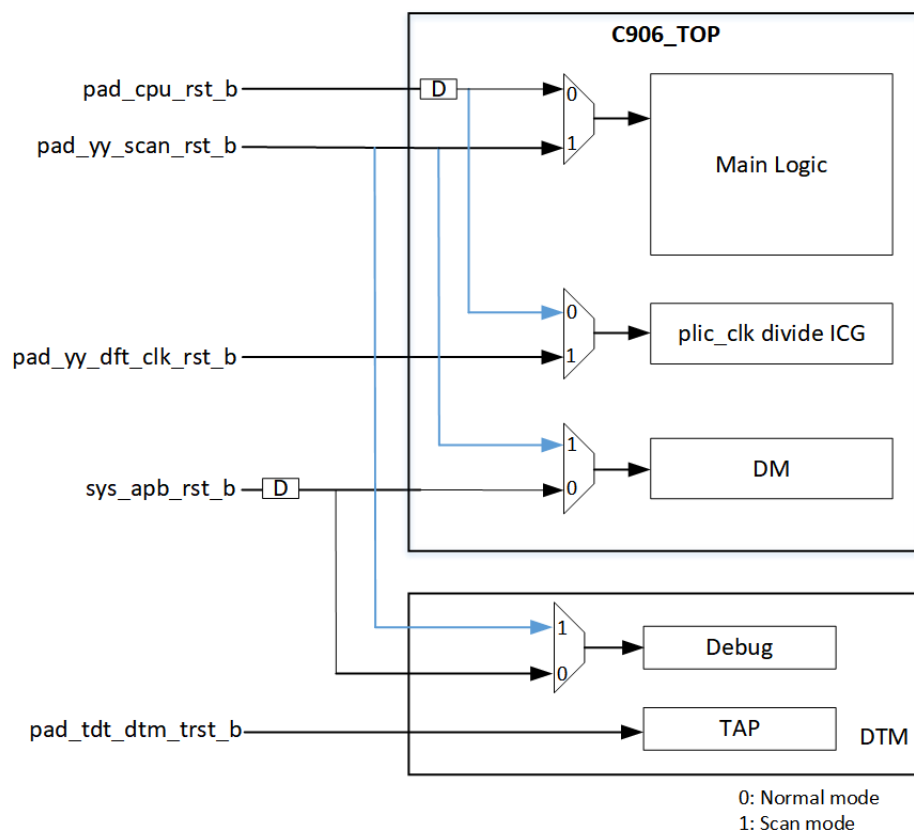


图 6.6: C906 复位信号

用于 scan 模式下 C906 内部 PLIC 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信号上。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 DTM 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。

为了在 BIST 模式下对输入到 CPU 内部的复位信号进行钳制，上述复位信号在 BIST 模式下会被旁路为 0。

### 6.4.1 复位启动地址

C906 提供一组接口信号 pad\_cpu\_rvba[39:0] 供系统配置 CPU 的复位启动基地址。系统复位时该值会被初始化到机器模式复位向量寄存器 (MRVBR) 中，该信号 pad\_cpu\_rvba[39:1] 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 C906 的复位信号 (pad\_core\_rst\_b) 释放之前保持稳定。

### 6.4.2 软复位

C906 内部 DM 模块提供了两根软复位指示信号，作为 C906 的顶层输出信号。调试软件通过设置 DM 模块内的相关寄存器来驱动这两根信号跳变，SoC 集成者使用这两根信号来实现相关模块的复位操作。

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

C906 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 C906 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。

- **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。

# 第七章 总线系统集成

## 7.1 AXI 主设备接口总线概述

AMBA 4.0 AXI 是一种高性能高频率的系统总线标准。C906 包含一个 128 比特的主设备接口，其总线协议与 AXI 标准协议兼容。

C906 主设备接口具有以下主要特征：

- 只支持以下 2 种传输方式：INCR（LEN 为 0 或者 3）和 WRAP4；
- 支持乱序完成；
- 支持关键字访问优先；
- 数据宽度为 128 位；
- 支持 CPU 到总线（cpu-to-bus）的整数倍分频（1:1, 2:1 等）；
- 支持总线响应：OKAY、SLVERR 和 DECERR；
- 总线控制和数据信号寄存器输出；
- 不支持写操作的交叉（interleaving）；

### 7.1.1 读传输

C906 AXI 总线接口传输中，在读传输时可能发起的传输类型为单次传输和突发传输两种情况。单次传输对应的基本传输类型为 INCR（长度为 1），传输宽度为字节/半字/字/双字/四字。仅在指令和数据缓存行预载入时，采用 WRAP4 和 INCR4 的传输类型。

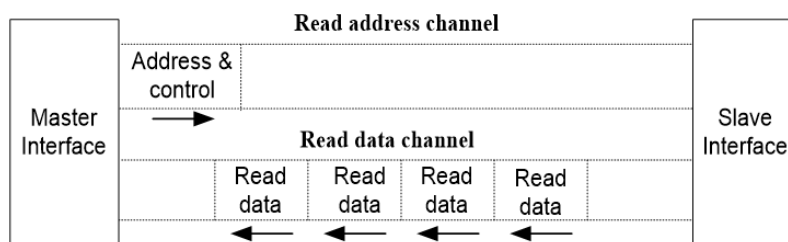


图 7.1: 读传输的通道结构



7.1.1.1 读地址通道

C906 支持的读传输 outstanding 能力为 10，单个 ID 最大 outstanding 能力为 8。详细信息如下所示。

表 7.1: 总线接口都传输属性

ARID[7:0]	0~1	4	8~15
Outstanding 能力	1	8	1
总 Outstanding 能力	10		

C906 内部读请求的延时为  $3T + \text{External\_Latency} \times \text{clk\_ratio}$ 。“T”为 pll\_core\_cpuclock 时钟周期，clk\_ratio 为 cpu\_clk 与 sys\_clk 的分频比。

7.1.1.2 读数据通道

读数据通道承载从设备返回的数据和响应两方面信息。数据通道特性包括：

- 数据总线宽度 128；
- 支持读响应类型，OKAY、SLVERR 和 DECERR。

7.1.2 写传输

C906 AXI 接口写传输请求的类型支持长度为 1 和 4 的 INCR，其中 INCR1 传输的宽度为字节/半字/字/双字/四字。不同宽度的数据传输时，对应的字节使能信号会有效。

7.1.2.1 写地址通道

写传输的地址通道包括以下特性：

- 只支持 INCR 一种传输类型；
- 传输宽度支持 8，16，32，64 和 128；
- 写传输 Outstanding 能力为 16；
- 写传输使用到的命令 ID（AWID）有 0~15 共 16 个，单个 ID 的 outstanding 能力为 1。

7.1.2.2 写数据通道

写数据通道的特性包括：

- 数据宽度 128 比特；
- 字节使能信号（byte lane strobe）指示字节有效性；
- 不支持 write interleave。

7.1.2.3 写响应通道

写响应通道用作从设备向主设备返回当前写操作的完成情况。写响应通道的特性包括：

- 支持 Bufferable 传输，返回给 C906 的写响应可不必从命令目的 slave 返回。

7.1.3 大小端

在 C906 处理器中，寄存器内部的值并没有大小端之分，只有有符号和无符号的区别。其格式均为从右至左表示逻辑低位到高位 的排布，如 图 7.2 所示。

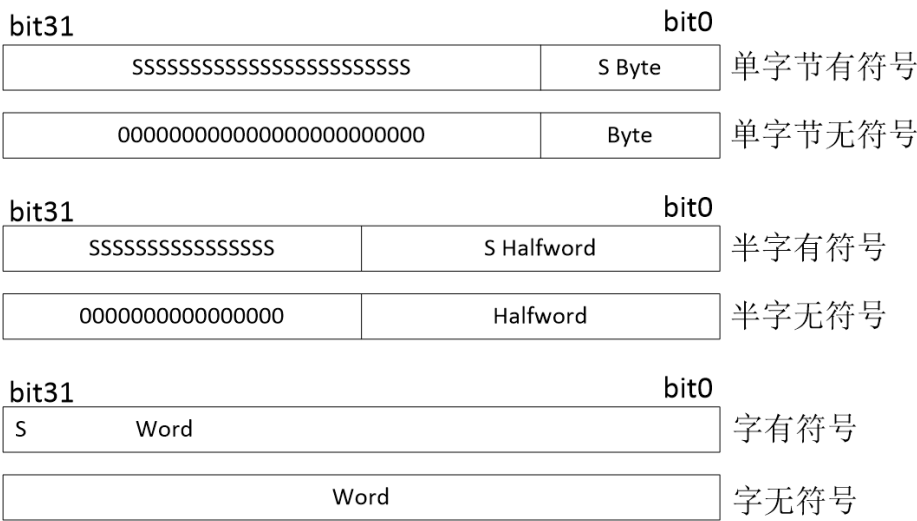


图 7.2: 数据在寄存器中的组织结构

C906 的内存存放格式为高地址字节存放至物理内存的高位，即小端，如 图 7.3 所示。

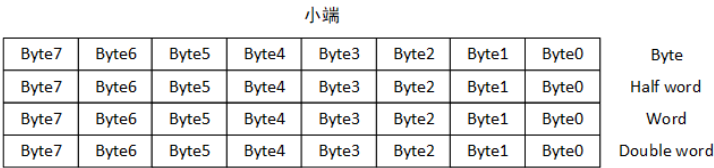


图 7.3: 数据在内存中的组织形式

7.2 调试主设备接口总线概述

C906 内部 DM 支持硬件选配一组 AXI4.0 总线接口直接访问内存空间，该请求不经过 C906 内部流水线，因此无法访问缓存在 C906 内部 CACHE 的数据。

该总线接口仅实现了部分传输请求类型，如下所述：

- 读写数据位宽为 128 位；

- 只支持 INCR0 传输；
- 每笔传输位宽为字节/半字/字/双字/四字；
- AWPROT 和 ARPROT 固定为 3' h2;

# 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外部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。

### 8.1.1 端口列表

表 8.1: 中断控制器接口信号

信号名	I/O	初始值	时钟域	功能描述
pad_plic_int_vld[i-1:0]	I	0	外设时钟域	外部中断源的中断指示信号，表示外部中断源的中断是否有效： 1：表示中断有效； 0：表示中断无效。  <b>注解：</b> i 表示中断源的数量。  该信号为 pll_core_cpucclk 的异步时钟信号，C906 内部完成同步处理
pad_plic_int_cfg[i-1:0]	I	-	-	外部中断源的中断配置信号，表示公有中断源的中断属性： 1：表示脉冲中断； 0：表示电平中断。 集成时需要接固定值。
pad_cpu_sys_cnt[63:0]	I	0	SYS	外部系统计时器的计数值，该信号需要由外部同步到系统时钟域。C906 内部使用打拍过的 axim_clk_en 对该信号进行采样。

**注解：** pad\_plic\_int\_vld 信号在 C906 内部会先被同步到 pll\_core\_cpucclk 的二分频时钟，然后进行采样，

仲裁和分发。在 C906 内部会使用 `axim_clk_en` 打拍后的信号去采样 `pad_cpu_sys_cnt[63:0]`。

### 8.1.2 中断握手时序图

PLIC 负责采样外部中断源的中断请求：对于电平中断，PLIC 采样中断有效信号，然后根据中断优先级和分发策略向特定中断目标发送中断请求。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，PLIC 采样完中断后，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 PLIC 发起多次中断请求，PLIC 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，PLIC 会再次触发对应中断，该中断请求在中断退出后才能够再次被 CPU 响应。

图 8.1 给出了 CPU 内核与 C906 PLIC 中断交互时序图。PLIC 工作在 `plic_clk` (C906 内部生成的 `pll_core_cpucclk` 的二分频时钟) 下，PLIC 会将传递进来的中断有效信号 `pad_plic_int_vld` 信号在 `plic_clk` 时钟下同步两个周期。

采样到中断信号 `pad_plic_int_vld` 时，PLIC 会先将其同步到 PLIC 时钟域 (`pad_plic_int_vld_sync2`)。PLIC 在 `plic_clk` 时钟上升沿向 CPU 发起中断请求。CPU 需要将中断请求信号从 PLIC 时钟域同步到 CPU 时钟域，然后响应中断进入中断服务程序。在中断服务程序中，需要清掉外部中断源请求，即拉低 `pad_plic_int_vld`。

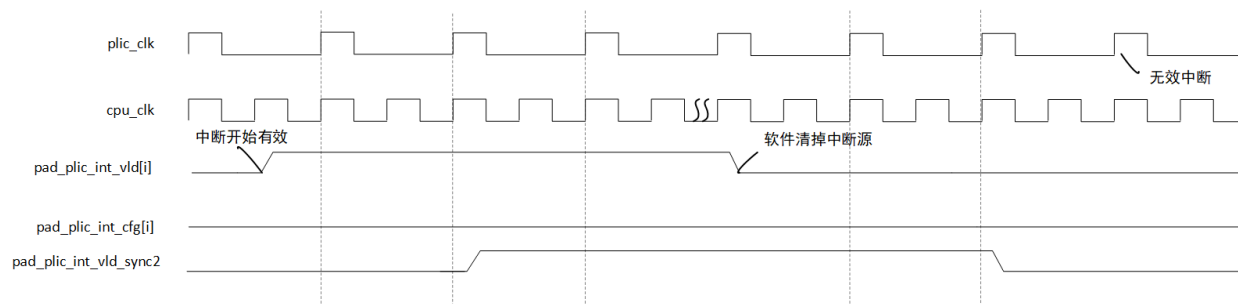


图 8.1: 中断相关信号时序简图

## 第九章 调试系统集成

C906 设计实现了 RISC-V Debug 标准定义的调试系统，支持复用 DTM 模块进行异构多核调试连接。调试系统如下所示：

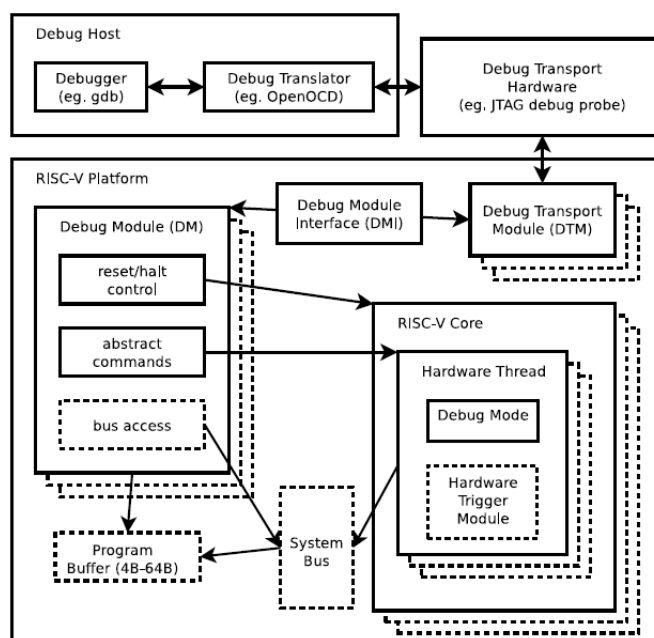


图 9.1: 调试系统框图

C906 发布代码分两部分，DTM，DM 与 C906。DTM 和 DM 以 DMI (Debug Module Interface，简称 DMI) 连接，DMI 以标准 APB 总线实现。多核异构调试时，每个 RISC-V Core 所连接的 DM 模块作为一个 Slave 设备挂载在 DMI 接口上。上层调试软件通过识别第 0 个 Slave 设备 DM 中的寄存器来获取 Slave1 的访问偏移量，并通过 APB 总线访问。如下图所示：

C906 可分为三种等级的调试资源配置，如下所示：

- **最小配置** Program Buffer 大小为 1；1 个硬件断点；
- **典型配置** Program Buffer 大小为 2；3 个硬件断点和观察点；8 个表项的 PCFIFO；
- **最大配置** Program Buffer 大小为 2；8 个硬件断点和观察点，且可以设置触发链；16 个表项的 PCFIFO；调试 AXI 主设备接口；

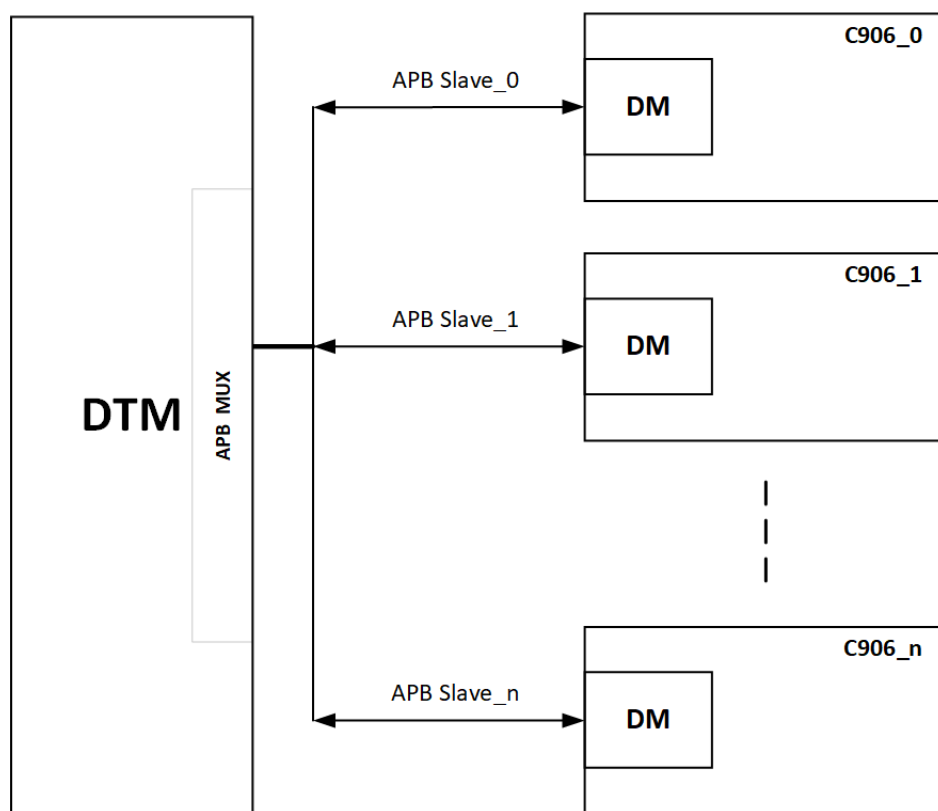


图 9.2: 多核调试系统框图

## 9.1 端口列表

表 9.1: DTM 模块端口列表

信号名	I/O	Reset	时钟域	描述
pad_tdt_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
pad_yy_scan_rst_b	I	-	-	Scan 模式下的复位控制信号
pad_tdt_dtm_jtag2_sel	I	-	TCLK	固定接 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	-	-	JTAG 接口信号
pad_tdt_dtm_tdi	I	-	TCLK	JTAG 接口信号
pad_tdt_dtm_tms_i	I	-	TCLK	JTAG 接口信号 TMS
pad_tdt_dtm_trst_b	I	-	TCLK	JTAG 接口信号
tdt_dtm_pad_tdo	O	1' h1	TCLK	JTAG 接口信号
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	可悬空处理
tdt_dtm_pad_tms_oe	O	1' h0	TCLK	可悬空处理
tdt_dmi_paddr[11:0]	I	-	SYS APB CLK	DMI APB 传输地址信号, 12 位宽。
tdt_dmi_penable	I	-	SYS APB CLK	DMI APB 传输使能信号
tdt_dmi_p_rdata[31:0]	O	32' h0	SYS APB CLK	DMI APB 读数据, 32 位宽。
tdt_dmi_pready	O	1' h0	SYS APB CLK	DMI APB 传输 ready 信号。
tdt_dmi_psel	I	-	SYS APB CLK	DMI APB Slave 选择信号。
tdt_dmi_pslverr	O	1' h0	SYS APB CLK	DMI APB 传输错误指示信号。
tdt_dmi_p_wdata[31:0]	I	-	SYS APB CLK	DMI APB 写数据, 32 位宽。
tdt_dmi_pwrite	I	-	SYS APB CLK	DMI APB 读写指示信号。



表 9.2: 其他调试端口列表

信号名	I/O	Reset	时钟域	描述
core0_pad_halted	O	1' h0	SYS AXI CLK	指示 C906 当前处于调试模式，高电平有效。
pad_tdt_dm_core_unavail	I	1' h0	SYS APB CLK	指示当前 CPU 不可以被调试，包括 CPU 正在被复位，时钟未开启等状态。该信号会体现在 DM 内的寄存器，调试软件读取该寄存器得到 CPU 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: 低功耗端口列表

信号名	I/O	初始值	时钟域	功能描述
core0_pad_lpm�_b[1:0]	O	2' b11	SYS	处理器核低功耗模式状态信号： 当处理器核执行 wfi 指令时 core0_pad_lpm�_b[1:0] 被相应的改变： 00: 低功耗模式； 11: 正常工作模式。

## 10.2 工作模式及其转换

C906 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，其中低功耗工作模式下 CPU 会关闭内部 ICG，停止取指令和执行指令，并通过 CPU 顶层指示信号传递到核外供 SoC 查询处理。

### 10.2.1 进入低功耗模式握手

CPU 通过执行低功耗指令 WFI 进入低功耗模式，并且将顶层输出信号 core0\_pad\_lpm�\_b[1:0] 置为 2' b00。SoC 可根据低功耗场景设计需求决定是否采取关闭 CPU 的全局时钟 pll\_core\_cpuclock 或者降低该时钟的频率。但需要注意的是 C906 内部绝大多数逻辑均工作在 pll\_core\_cpuclock 时钟域下，在关闭 pll\_core\_cpuclock 的低功耗模式下要想再次唤醒 CPU，需要先将时钟 pll\_core\_cpuclock 恢复正常。

### 10.2.2 退出低功耗模式握手

中断请求/调试请求/复位请求可以唤醒处于低功耗模式下的处理器。当 C906 输入时钟没有被关闭时，C906 采样到的有效中断信号可以将处于低功耗模式下的处理器唤醒。如果全局中断使能寄存器没有开启的话，被唤醒的 CPU 并不会响应中断，而是执行 WFI 指令的后续指令。

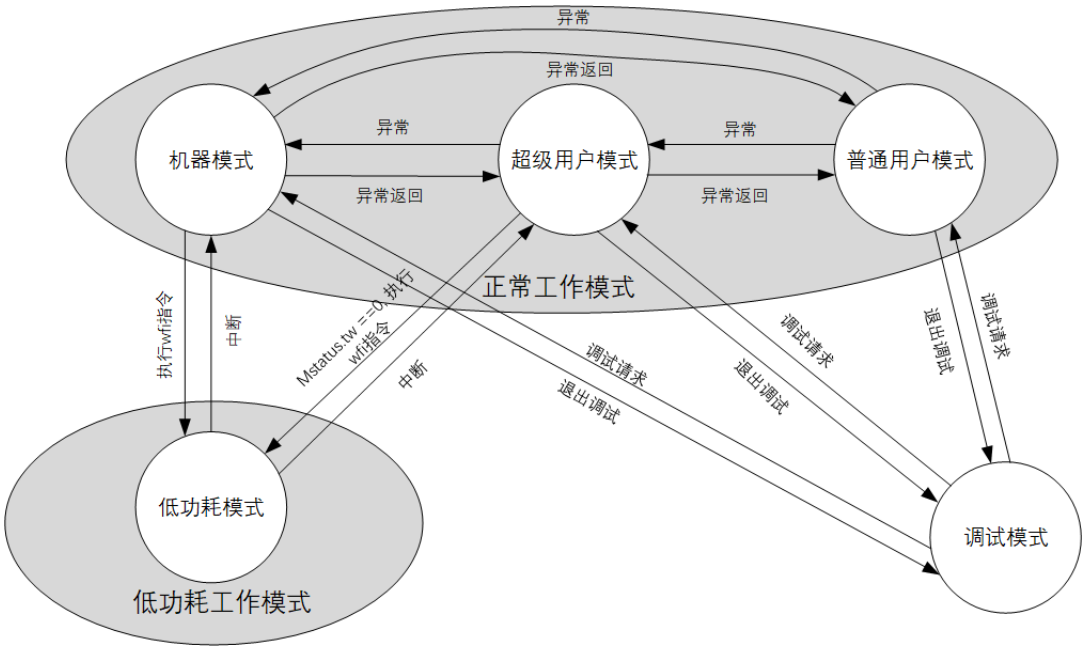


图 10.1: CPU 状态转换图

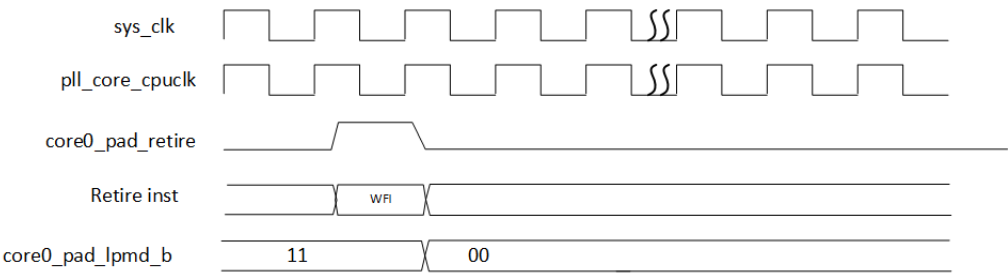


图 10.2: 进入低功耗握手

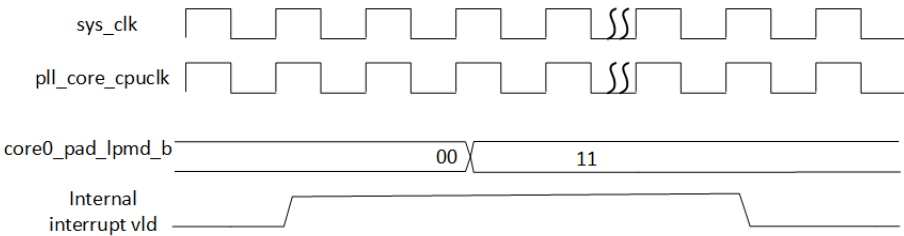


图 10.3: 退出低功耗握手

## 第十一章 DFT 相关信号

C906 内部预设计了 DFT 相关信号，列表如 表 11.1 所示。

表 11.1: C906 DFT 信号列表

信号名	I/O	复位值	时钟域	描述
pad_yy_scan_enable	I	-	-	扫描使能: 扫描链的使能控制信号，高电平有效。 不用该信号时，需要接 0。
pad_yy_scan_mode	I	-	-	scan mode 控制信号： 不用该信号时，需要接 0。
pad_yy_icg_scan_en	I	-	-	scan mode 下 C906 内部 ICG 的控制信号： 用于控制 ICG 的 TE 端。
pad_yy_scan_rst_b	I	-	-	scan mode 下 C906 内部所有逻辑的复位控制信号。
pad_yy_dft_clk_rst_b	I	-	-	scan mode 下 C906 内部二分频逻辑复位控制信号该复位信号可以接到上电复位端，DFT 模式下需要保持高电平。
pad_yy_mbist_mode	I	-	-	mbist mode 控制信号，用于将 CPU 内部复位逻辑置为 0。

在 DTM DMI 模块内部也预设计了 DFT 相关信号，如 表 11.2 所示。

表 11.2: DMI 模块 DFT 信号列表

信号名	I/O	复位值	时钟域	描述
pad_yy_scan_mode	I	-	-	scan mode 控制信号： 不用该信号时，需要接 0。
pad_tdt_icg_scan_en	I	-	-	scan mode 下 DTM 内部 ICG 的控制信号： 用于控制 ICG 的 TE 端。
pad_yy_scan_rst_b	I	-	-	scan mode 下 DTM 内部逻辑的复位控制信号。

## 第十二章 CPU 运行观测信号

CPU 运行观测信号是将处理器内部的指令退休信息输出到顶层供 SoC 来调试分析，SoC 设计可选择性的将这些信号锁存到外部寄存器中以方便调试，如果不用可悬空。

表 12.1: CPU 运行观测信号

信号名	I/O	初始值	时钟域	功能描述
core0_pad_retire	O	1' b0	CPU	处理器指令退休指示信号： 0：当前周期没有指令退休； 1：当前周期有指令退休。
core0_pad_retire_pc[39:0]	O	-	CPU	处理器退休指令的 PC： 表明当前正在退休的指令的 PC。