Nuclei™ N200 系列

处理器内核简明数据手册

版权声明

联系我们

若您有任何疑问，请通过电子邮件 support@nucleisys.com 联系芯来科技。

修订历史

版本号	修订日期	修订的章节	修订的内容
1.0	2019/5/11	N/A	1. 初始版本

目录

表格清单

图片清单

1. N200 系列内核概述

Nuclei N200 系列处理器内核（简称 N200 系列内核）是由芯来科技开发的一款全国产自主可控的商用 RISC-V 处理器内核系列，主要面向极低功耗与极小面积的场景而设计，非常适合替代传统的 8051 内核或者 ARM Cortex-M 系列内核，应用于 IoT 或其他低功耗场景。

1.1. N200 系列内核特性列表

N200 系列内核的特性列表如下：

- CPU 内核（CPU Core）
  - 2 级变长流水线架构，采用一流的处理器架构设计，实现业界最高的能效比与最低的成本。
  - 可配置的分支预测器。
  - 可配置的指令预取单元，能够按顺序预取两条指令，从而隐藏指令的访存延迟。
  - 支持机器模式（Machine Mode）和可配置的用户模式（User Mode）。
- 支持指令集架构（ISA，Instruction Set Architecture）
  - N200 系列处理器核支持 32 位的 RISC-V 指令集架构，支持 RV32I/E/M/A/C/F/D 等指令子集的配置组合。
- 总线接口
  - 支持 32 比特宽的标准 AHB-Lite 系统总线接口，用于访问外部指令和数据。
  - 支持 32 比特宽的指令局部存储器（Instruction Local Memory，ILM）总线接口（支持标准的 AHB-Lite 或 SRAM 接口协议），用于连接私有的指令局部存储器。
  - 支持 32 比特宽的数据局部存储器（Data Local Memory，DLM）总线接口（支持标准的 AHB-Lite 或 SRAM 接口协议），用于连接私有的数据局部存储器。
  - 支持 32 比特宽的私有设备总线（Private Peripheral Interface，PPI），支持标准的 APB 接口协议，用于连接私有的外设。
  - 支持 32 比特宽的快速 IO 接口，用于连接快速 IO（Fast-IO，FIO）模块，譬如 GPIO 模块。
- 调试功能

- 支持标准 JTAG 接口。
- 支持 RISC-V 调试标准。
- 支持可配置数目的硬件断点（Hardware Breakpoints）。
- 支持成熟的交互式调试工具。

- 低功耗管理
  - 支持 WFI（Wait For Interrupt）与 WFE（Wait For Event）进入休眠模式。
  - 支持两级休眠模式：浅度休眠与深度休眠。
- 内核私有的计时器单元（Machine Timer，简称 TIMER）
  - 64 比特宽的实时计时器，支持产生 RISC-V 标准定义的计时器中断。
- 存储器保护单元
  - 可配置的存储器保护单元（Physical Memory Protection，PMP）。
- 增强的内核中断控制器（Enhanced Core Level Interrupt Controller，ECLIC）
  - 支持 RISC-V 标准定义的的软件中断、计时器中断和外部中断。
  - 支持可配置数目的外部中断。
  - 支持可配置数目的中断级别和优先级，支持软件动态可编程修改中断级别和中断优先级的数值。
  - 支持基于中断级别的中断嵌套。
  - 支持快速向量中断处理机制。
  - 支持快速中断咬尾机制。
- 支持 NMI（Non-Maskable Interrupt）。
- 扩展指令接口：
  - N200 系列内核提供一个可配置的扩展指令接口（Nuclei Instruction Co-unit Extension，NICE），以支持用户进行自定义指令的扩展。
- 软件开发工具：
  - N200 系列处理器核支持 RISC-V 标准的编译工具链，以及 Linux/Windows 图形化集成开发环境（Integrated Development Environment，IDE）。

1.2. N200 系列内核指令集与架构

N200 系列支持的指令集和架构详情请参见《Nuclei_N200 系列指令架构手册》。

1.3. N200 系列内核层次结构图

N200 系列内核的顶层如图 1-1 所示。Nuclei N200 系列处理器内核的组织结构主要包含如下要点：

- Core 为整个处理器内核的顶层。
- uCore 位于 Core 层次结构之下，为处理器内核的主体部分。
- 除了 uCore 之外，在 Core 层次结构之下还包含了如下主要组件：
  - DEBUG：处理 JTAG 接口和相关的调试功能。
  - ECLIC：中断控制单元。
  - TIMER：计时器单元。
  - LM Ctrl：对外部 ILM 和 DLM 接口的控制。
  - BIU：对外部 PPI 接口和 MEM 接口的控制。
  - Misc Ctrl：其他控制模块。

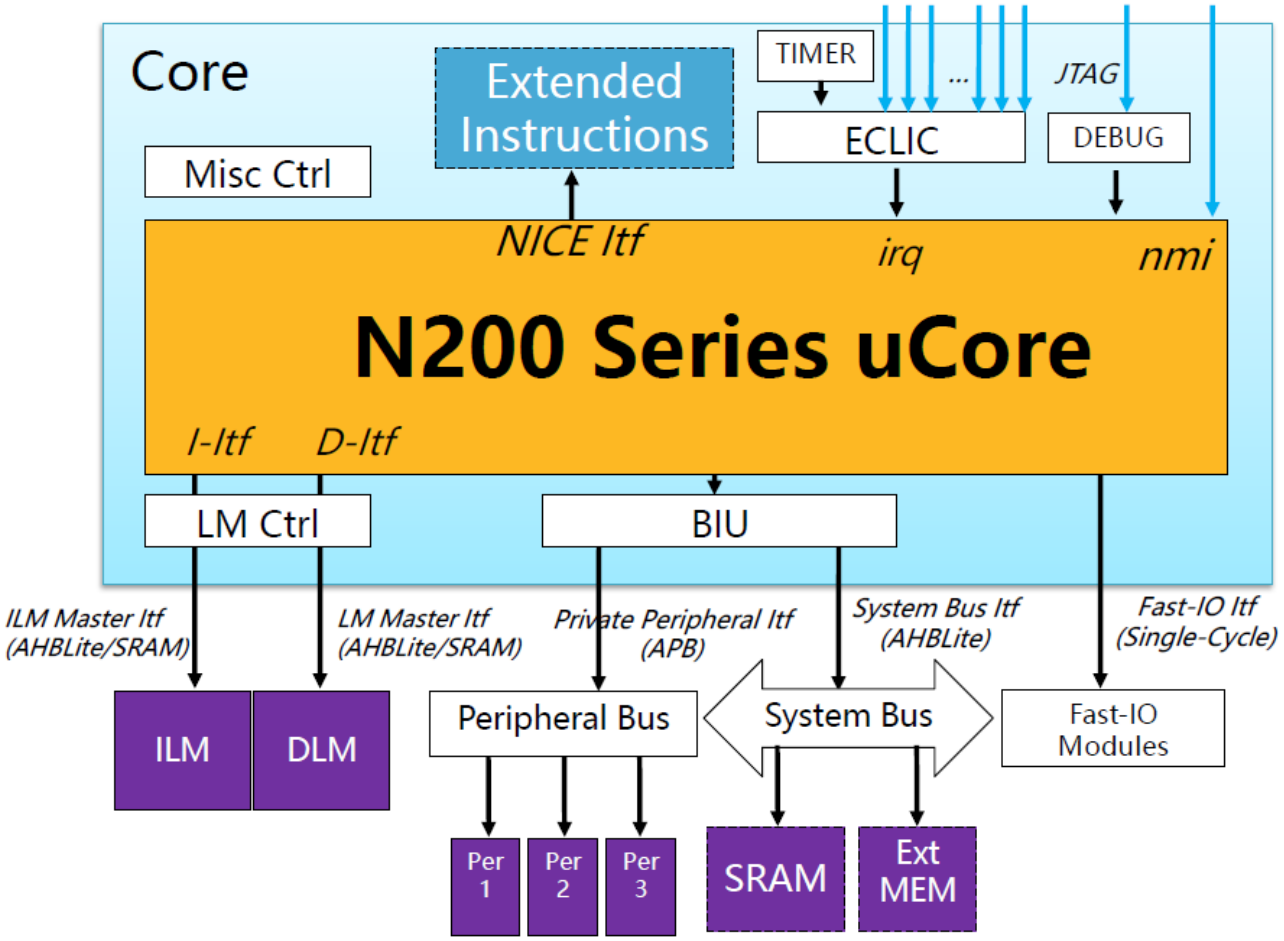

图 1-1 N200 系列内核顶层示意图

1.4. N200 系列内核系列型号介绍

N200 是一个处理器系列，包含了多款不同的具体处理器型号，不同型号的具体差别如表 1-1 所示。

表 1-1 N200 处理器型号特性介绍

	N201	N203 N203e	N205 N205e	N207 N207f N207fd
支持的 RISC-V 32 位架构指令子集	IC	IMC /EMC	IMC /EMC	IMC /IMFC /IMFDC
硬件乘法单元	无	可配置	可配置	单周期乘法器
硬件除法单元	无	多周期除法器	多周期除法器	多周期除法器

A 扩展指令子集	无	可配置	可配置	可配置
硬件单精度浮点器	无	无	无	可配置
硬件双精度浮点器	无	无	无	可配置
指令缓存（I-Cache）	无	无	无	可配置
ILM 和 DLM 接口	无	无	有	有
私有的设备接口（APB）	无	可配置	可配置	可配置
快速 IO 接口	无	可配置	可配置	可配置
User Mode 和 PMP	无	可配置	可配置	可配置
扩展指令接口(NICE)	无	可配置	可配置	可配置
时序提升	无	可配置	可配置	可配置
注意：更多详细配置信息请参见第 4 章。				

下文将分别予以简介。

1.4.1. N201 内核

N201 内核支持 RV32IC 架构，不支持硬件乘法器和除法器，是最简单一款入门级内核，用于替代传统的 8051 内核和 ARM Cortex-M0/M0+内核。

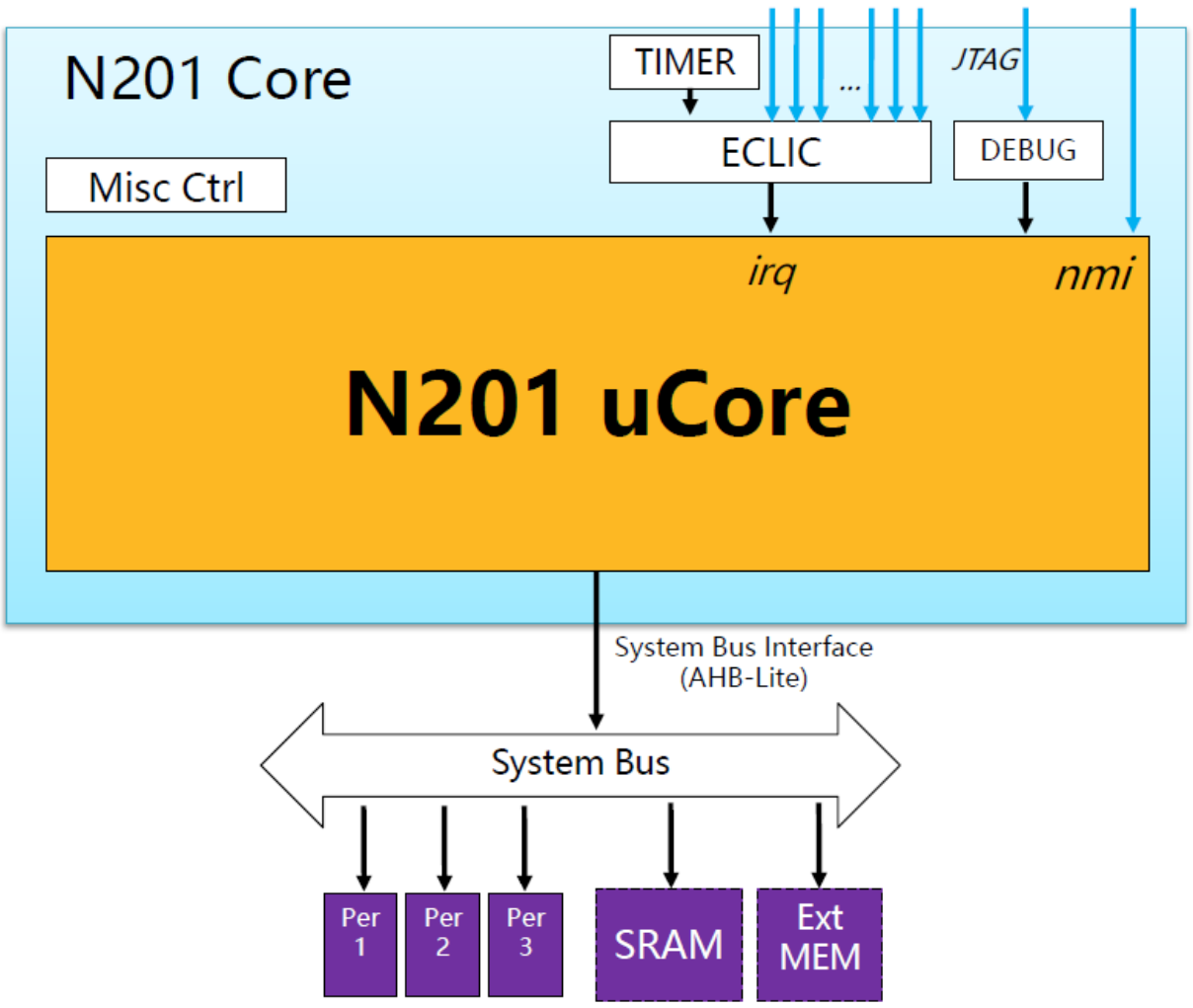

图 1-2 N201 内核顶层示意图

注意：作为一款简单易使用的入门级 RISC-V 内核，N201 内核采用固定配置（注意，并不是面积最小的配置），不支持用户进行二次配置，以达到简单稳定的效果。N201 处理器内核的固定配置特性列表如下：

- CPU 内核（CPU Core）
  - 配备静态的分支预测器
  - 不配备指令预取单元
  - 仅支持机器模式（Machine Mode Only）
  - 不支持存储器保护单元（Physical Memory Protection，PMP）
- 支持指令集架构（ISA，Instruction Set Architecture）

- RISC-V RV32IC 指令子集

- 总线接口
  - 支持一个 32 比特宽的标准 AHB-Lite 总线接口，用于访问外部指令和数据。参见第 3.4 节了解总线接口的详情。

- 调试功能
  - 支持标准 JTAG 接口，支持 2 个硬件断点（Hardware Breakpoints）

- 增强的内核中断控制器 ECLIC
  - 支持 RISC-V 标准定义的的软件中断、计时器中断、和外部中断
  - 支持 32 个外部中断
  - 支持可编程的 8 个不同中断级别
  - 支持基于中断级别的中断嵌套
  - 支持快速向量中断处理机制
  - 支持快速中断咬尾机制

- N201 内核的私有设备采用固定的地址空间分配。N201 内核的地址空间分配如表 1-2 中所示。

表 1-2 N201 内核地址空间分配

模块	基地址	偏移地址区间	描述
DEBUG	0x0000_0000	0x000~0xFFF	■ DEBUG 主要用于 JTAG 调试器使用，普通应用程序不应该使用此区间。 ■ 内核取指令或者取数据均可访问到此区间。
ECLIC	0x0C00_0000	0x0000~0xFFFF	■ ECLIC 单元的可编程寄存器地址区间。 ■ 内核取数据可访问到此区间，取指令无法访问到此区间。 ■ 有关 ECLIC 的详细介绍请参见《Nuclei_N200 系列指令架构手册》。
TIMER	0x0200_0000	0x000~0xFFF	■ TIMER 单元的可编程寄存器地址区间。 ■ 内核取数据可访问到此区间，取指令无法访问到此区间。 ■ 有关 TIMER 的详细介绍请参见《Nuclei_N200 系列指令架构手册》。
System Memory	注意： ■ N201 内核不支持 PPI、FIO、ILM、DLM 接口。 ■ 上述私有设备位于内核的内部，属于内核的私有设备，仅能够被内核访问到。处理器内核对这些私有设备的访问不会出现在 AHB-Lite 系统总线接口上。		

	■ 除了上述私有设备之外，内核访问其他的地址区间均通过 AHB-Lite 系统总线接口访问外部总线。

1.4.2. N203 内核

N203 内核支持 RV32IMC/EMC 架构，支持周期可配置的硬件乘法单元和多周期的硬件除法单元，是面积较小的内核，用于替代传统的 8051 内核和 ARM Cortex-M0/M0+内核（配置多周期乘法器）。注意：为了加以区分，支持 RV32EMC 架构的内核型号为 N203e，该架构仅仅支持 16 个通用寄存器，以达到更小的面积。

N203 内核支持机器模式（Machine Mode Only），还可配置地支持用户模式（User Mode）和存储器保护单元（Physical Memory Protection，PMP）。

N203 内核具有如下接口：

- 一个系统总线接口（System Memory Interface）供指令和数据共用，此接口为必选。
- 一个可配置的快速 IO 接口（Fast-IO Interface），此接口为可配置接口。

参见第 3.4 节了解总线接口的详情。

N203 内核还支持可配置的扩展指令接口，更多详细配置信息请参见第 4 章。

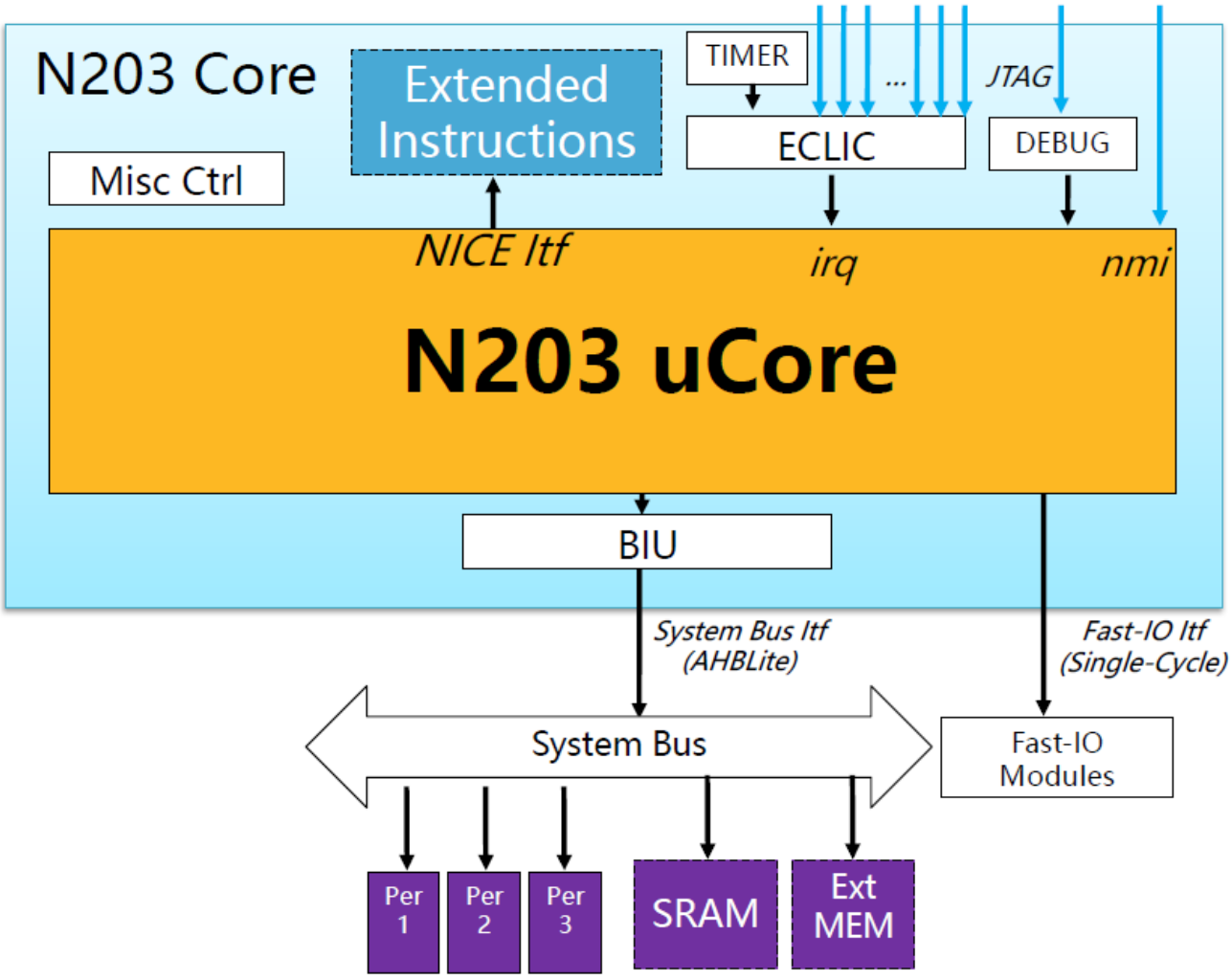

图 1-3 N203 内核顶层示意图

1.4.3. N205 内核

N205 内核支持 RV32IMC/EMC 架构，支持周期可配置的硬件乘法单元和多周期的硬件除法单元，用于替代 ARM Cortex-M0/M0+内核（配置单周期乘法器）和 Cortex-M3 内核。注意：为了加以区分，支持 RV32EMC 架构的内核型号为 N205e，该架构仅仅支持 16 个通用寄存器，以达到更小的面积。

N205 内核不仅支持机器模式（Machine Mode Only），还可配置地支持用户模式（User Mode）和存储器保护单元（Physical Memory Protection，PMP）。

N205 具有如下接口：

- 一个系统总线接口（System Memory Interface）供指令和数据共用，此接口为必选。
- 一个快速 IO 接口（Fast-IO Interface），此接口为可配置接口。
- 一个专用于指令取指的 ILM 接口，此接口为可配置接口。
- 一个专用于数据访问的 DLM 接口，此接口为可配置接口。
- 一个专用于访问外设的 PPI（Priviate Peripheral Interface）接口，此接口为可配置接口。

参见第 3.4 节了解总线接口的详情。

N205 内核还支持可配置的扩展指令接口，更多详细配置信息请参见第 4 章。

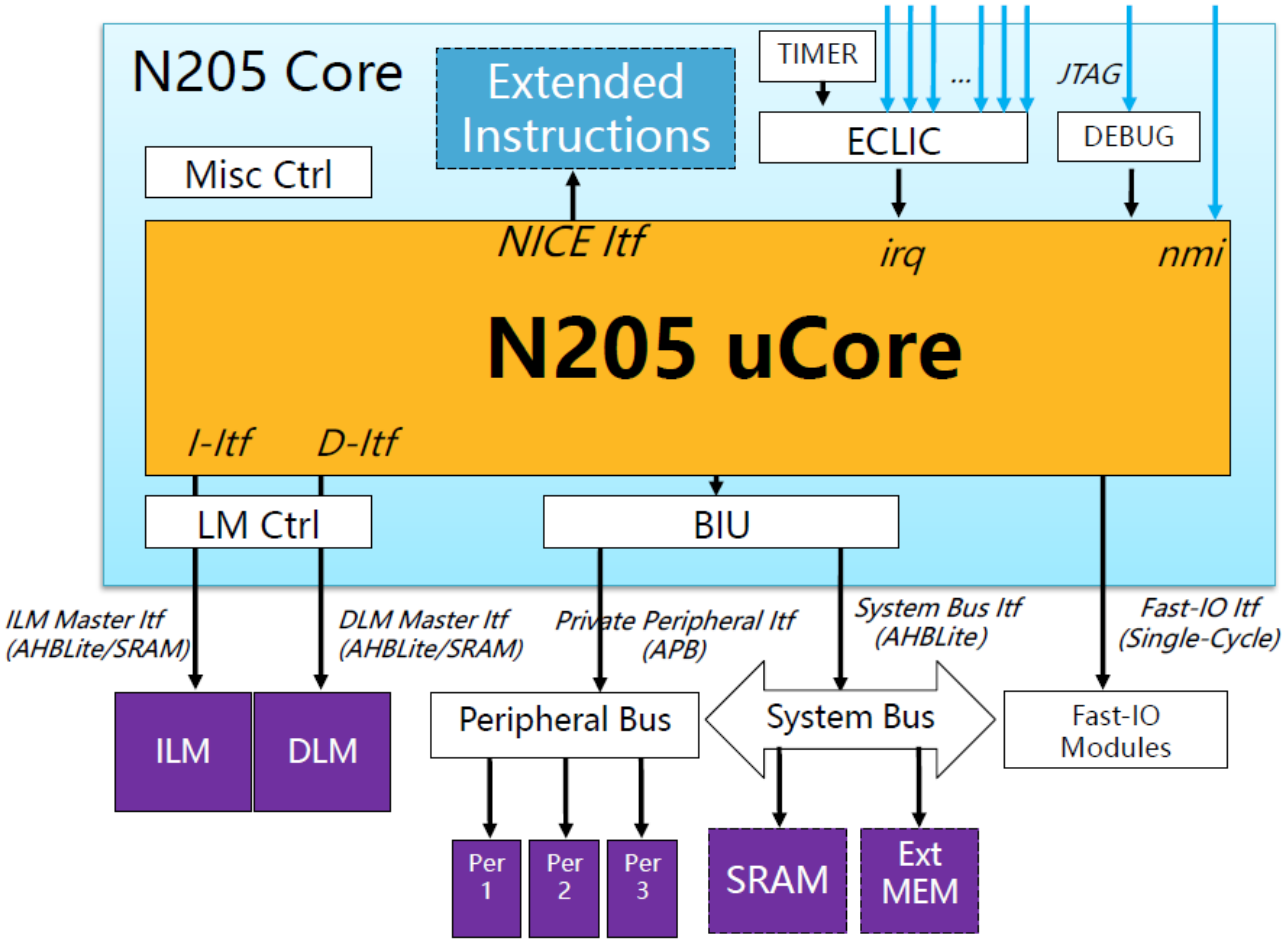

图 1-4 N205 内核顶层示意图

1.4.4. N207 内核

N207 内核支持 RV32IMAC 架构，使用单周期的硬件乘法单元和多周期的硬件除法单元，用于替代 ARM Cortex-M0/M0+内核（配置单周期乘法器）和 Cortex-M3 内核。

N207 内核不仅支持机器模式（Machine Mode Only），还可配置地支持用户模式（User Mode）和存储器保护单元（Physical Memory Protection，PMP）。

N207 具有如下接口：

- 一个系统总线接口（System Memory Interface）供指令和数据共用，此接口为必选。
- 一个快速 IO 接口（Fast-IO Interface），此接口为可配置接口。
- 一个专用于指令取指的 ILM 接口，此接口为可配置接口。
- 一个专用于数据访问的 DLM 接口，此接口为可配置接口。
- 一个专用于访问外设的 PPI（Priviate Peripheral Interface）接口，此接口为可配置接口。

参见第 3.4 节了解总线接口的详情。

N207 还具有如下额外配置特性：

- 可配置指令缓存（Instruction Cache）。指令缓存的总大小容量可配置，使用两路组相连（2-ways Associative）结构，Cache Line Size 为 32 Bytes。
- 配置硬件单精度浮点运算单元（内核型号为 N207f），配置了单精度浮点运算单元时则支持 RV32IMAFC 架构。
- 配置硬件单/双精度浮点运算单元（内核型号为 N207fd），配置了单/双精度运算单元时则支持 RV32IMAFDC 架构。

N207 内核还支持可配置的扩展指令接口，更多详细配置信息请参见第 4 章。

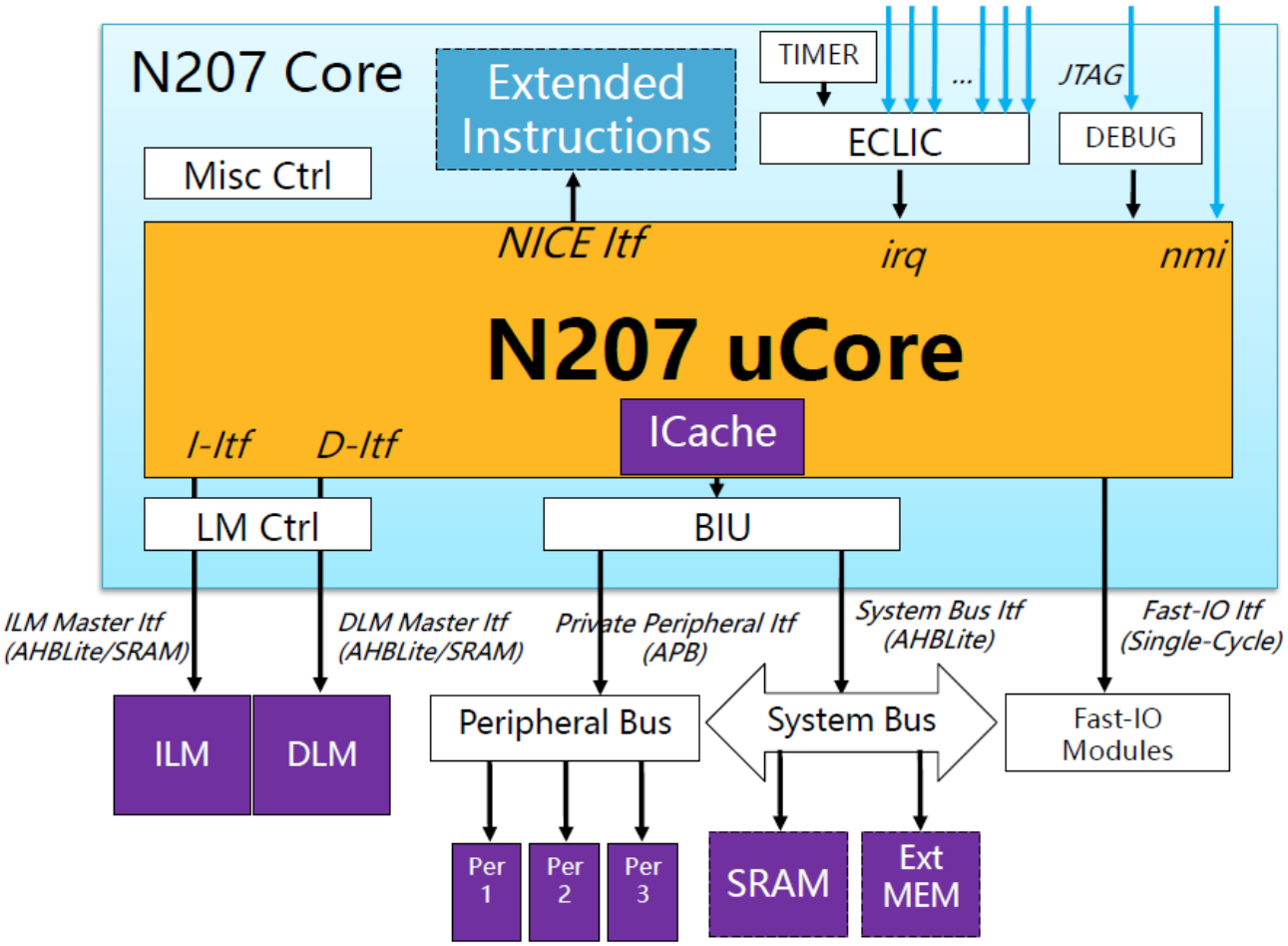

图 1-5 N207 内核顶层示意图

2. N200 系列内核功能简介

2.1. N200 系列内核时钟域介绍

N200 系列内核的时钟域划分如图 2-1 中所示，整个处理器内核分为两个彼此异步的时钟域：

- 工作时钟域，由输入的时钟 core_clk 和 core_clk_aon 驱动处理器内核的绝大部分功能逻辑。注意：
  - core_clk 和 core_clk_aon 为来自于同一个时钟源的同频同相时钟。
  - core_clk 为主工作时钟，驱动处理器内核内部的主要工作逻辑，并且可以在系统层面上被全局门控。
  - core_clk_aon 为常开时钟，驱动内核中的常开（Always-On）逻辑，主要包括 ECLIC、TIMER 以及 DEBUG。有关 ECLIC 和 TIMER 的详情请参见《Nuclei_N200 系列指令架构手册》。
- JTAG 时钟域，由输入的信号 jtag_TCK 驱动处理器内核的 JTAG 调试相关逻辑。

上述两个时钟域之间完全异步，在处理器内核的内部实现中已经进行了异步跨时钟域的处理。

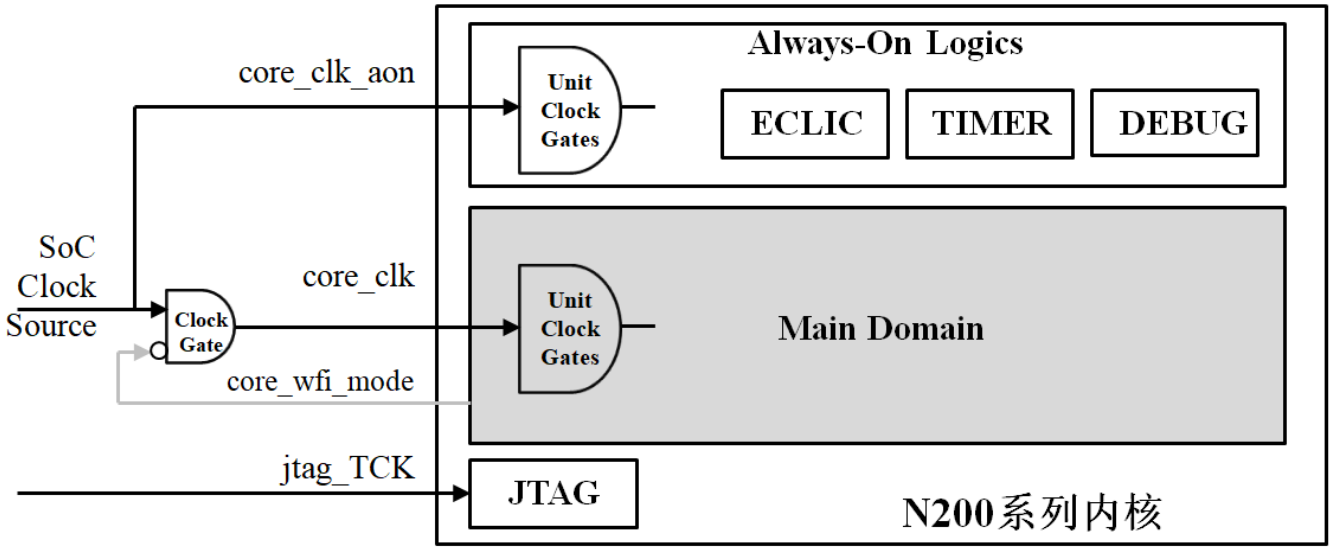

图 2-1 N200 内核时钟域示意图

2.2. N200 系列内核电源域介绍

N200 系列内核的内部并没有划分电源域，SoC 系统集成者可以根据 N200 系列内核的层次结构自行划分电源域和进行跨电源域处理。

2.3. N200 系列内核接口简介

Nuclei N200 系列处理器内核包含如下几类接口：

- 时钟和复位接口
- 调试接口
- 外部中断接口
- 总线接口
- 扩展指令接口
- 其他功能接口

请参见第 3.4 节了解接口的详细信息。

2.4. N200 系列内核地址空间分配

N200 系列内核的地址空间分配如

表 2-1 中所示。注意：

- ILM 和 DLM 地址区间可以彼此重叠，N200 系列内核的指令访问通路和数据访问通路各自独立：
  - 取指令无法通过 DLM 主接口访问外部 DLM。
  - 如果配置了选项 N20<x>_CFG _LSU_ACCESS_ILM，则数据访问可以通过 ILM 主接口访问外部 ILM（在此配置下如果 ILM 和 DLM 的地址区间配置得重叠，则数据优先访问 ILM 空间），否则无法通过 ILM 主接口访问外部 ILM。更多详细配置信息请参

见第 4 章。

- “ILM 和 DLM 地址区间”与“DEBUG、TIMER、ECLIC、FIO 或 PPI 地址区间”不应该彼此重叠，否则属于配置错误。
- DEBUG、TIMER、ECLIC、FIO 或 PPI 地址区间不应该彼此重叠，否则属于配置错误。

表 2-1 N200 内核地址空间分配

模块	基地址	偏移地址区间	描述
DEBUG	可配置	0x000~ 0xFFF	■ DEBUG 单元的地址空间 ■ 注意： 1. DEBUG 主要用于调试器使用，普通软件程序不应该使用此区间。 2. DEBUG 单元位于 Core 内部，属于私有于 Core 的外设。
ECLIC	可配置	0x0000 ~ 0xFFFF	■ ECLIC 单元寄存器地址空间。 ■ 注意：ECLIC 单元位于 Core 内部，属于私有于 Core 的外设。有关 ECLIC 的详细介绍请参见《Nuclei_N200 系列指令架构手册》。
TIMER	可配置	0x00 ~ 0xFF	■ TIMER 单元寄存器地址空间。 ■ 注意： TIMER 单元位于 Core 内部，属于私有于 Core 的外设。有关 TIMER 的详细介绍请参见《Nuclei_N200 系列指令架构手册》。
ILM	可配置	取决于 ILM 配置大小	■ ILM 主接口的地址空间。 ■ 注意：此地址区间只有在配置了 ILM 接口时才会存在。
DLM	可配置	取决于 DLM 配置大小	■ DLM 主接口的地址空间。 ■ 注意:此地址区间只有在配置了 DLM 接口时才会存在。
FIO	可配置	取决于 FIO 区间的配置大小	■ FIO 接口的地址空间。 ■ 注意：此地址区间只有在配置了 FIO 接口时才会存在。
PPI	可配置	取决于 PPI 区间的配置大小	■ PPI 接口的地址空间。 ■ 注意：此地址区间只有在配置了 PPI 接口时才会存在。
MEM	N/A	N/A	■ 排除了以上任何地址空间的其他地址空间均可归于此 MEM（System Memory Interface）地址空间。
注：有关可配置参数的详细信息请参见本文档第 4 章。			

2.5. N200 系列内核的特权模式

N200 系列内核支持两个特权模式（Privilege Modes）：机器模式（Machine Mode）是必须的模式，用户模式（User Mode）是可配置的模式。有关 Privilege Modes 的详情请参见《Nuclei_N200 系列指令架构手册》。

2.6. N200 系列内核的存储器资源

N200 系列内核支持如下类型的存储器资源：

- ILM：
  - N200 系列内核如果配置了指令局部存储器（Instructon Local Memory，ILM）接口，则支持通过专有的 AHB-Lite 总线或者 SRAM 接口访问 ILM。
  - ILM 的大小可以配置，可配置参数的详细信息请参见本文档第 4 章。
  - ILM 接口有独立的地址区间，用户可以配置具体的基地址，请参见第 2.4 节了解其详情。
  - ILM 由 SoC 系统集成者自行实现，一般可以是用于存放指令的片上 SRAM 或者片上 Flash。如果使用 AHB-Lite 接口，要达到最佳性能，ILM 应该遵循 AHB 协议规则，在收到地址后的下一个周期返回指令。
- DLM：
  - N200 系列内核如果配置了数据局部存储器（Data Local Memory，DLM）接口，则支持通过专有的 AHB-Lite 总线或则 SRAM 接口访问 DLM。
  - DLM 的大小可以配置，可配置参数的详细信息请参见本文档第 4 章。
  - DLM 接口有独立的地址区间，用户可以配置具体的基地址，请参见第 2.4 节了解其详情。
  - DLM 由 SoC 系统集成者自行实现，一般可以是用于存放数据的片上 SRAM。如果使用 AHB-Lite 接口，要达到最佳性能，DLM 应该遵循 AHB 协议规则，在收到地址后

的下一个周期返回数据。

- Cache：
  - N200 系列内核如果配置了 I-Cache（指令缓存），则会将来自于 MEM（System Memory Interface）地址空间取回的指令进行缓存。
  - 注意：来自于 ILM 地址空间的取指令操作不会被缓存至 I-Cache 中。
  - Cache 使用两路组相连（2-ways Associative）结构，Cache Line Size 为 32 Bytes，指令缓存的总大小容量可配置，可配置参数的详细信息请参见本文档第 4 章。

2.7. N200 系列内核的私有设备

如图 1-1 中所示，N200 系列内核的 Core 层次结构之下，除了 uCore 之外，还包含了如下私有设备：

- DEBUG：处理 JTAG 接口和相关的调试功能。
- ECLIC：内核中断控制单元。
- TIMER：内核私有计时器单元。

上述设备属于处理器内核私有，使用存储器地址寻址方式进行访问，有关其具体的地址区间分配请参见第 2.4 节。

2.8. N200 系列内核的物理存储保护

由于 N200 系列处理器内核是面向微控制器领域的低功耗内核，其不支持虚拟地址管理单元（Memory Management Unit, MMU），因此所有的地址访问操作都是使用的物理地址。为了根据不同的物理地址区间和不同的 Privilege Mode 进行权限隔离和保护，N200 系列处理器内核（可配置地）支持 PMP（Physical Memory Protection）单元。有关 PMP 的详细介绍，请参见《Nuclei_N200 系列指令架构手册》。

2.9. N200 系列内核的调试机制

N200 系列内核支持标准的 JTAG 调试接口，以及成熟的交互式调试工具 GDB。有关调试工具详情请参见《Nuclei_N200 系列 IDE 使用说明》和《Nuclei_N200 系列 SDK 使用说明》。注意：

- N200 系列内核支持的硬件断点（Hardware Breakpoint）数目可配置。硬件断点主要用于向只读区间（譬如 Flash）设置断点。

N200 系列内核定义了一根输入信号，i_dbg_stop 可以通过其输入信号的值来进行控制：

- 如果 i_dbg_stop 信号的值为 1，则处理器内核的调试功能被关闭。
- 如果 i_dbg_stop 信号的值为 0，则处理器内核的调试功能正常工作。

2.10. N200 系列内核的中断和异常机制

有关 N200 系列的中断和异常机制详细介绍，请参见《Nuclei_N200 系列指令架构手册》。

2.11. N200 系列内核的 NMI 机制

NMI（Non-Maskable Interrupt）是处理器内核的一根特殊的输入信号，往往用于指示系统层面的紧急错误（譬如外部的硬件故障等）。在遇到 NMI 之后，处理器内核应该立即中止执行当前的程序，转而去处理该 NMI 错误。有关 N200 系列的 NMI 机制详细介绍，请参见《Nuclei_N200 系列指令架构手册》。

2.12. N200 系列内核的 CSR 寄存器

RISC-V 的架构中定义了一些控制和状态寄存器（Control and Status Register，CSR），用于配置或记录一些运行的状态。CSR 寄存器是处理器核内部的寄存器，使用其专有的 12 位地址编码空间。详情请参见《Nuclei_N200 系列指令架构手册》了解其详情。

2.13. N200 系列内核的性能计数器

RISC-V 架构定义了如下两种性能计数器：

- 时钟计数器（Cycle Counter）：
  - 一个 64 位宽的时钟周期计数器，用于反映处理器执行了多少个时钟周期。只要处理器处于执行状态时，此计数器便会不断自增计数。
  - CSR 寄存器 mcycle 反映了该计数器低 32 位的值，CSR 寄存器 mcycleh 寄存器反映了该计数器高 32 位的值。有关 mcycle 和 mcycleh 的详情请参见《Nuclei_N200 系列指令架构手册》。
- 指令完成计数器（Instruction Retirement Counter）：
  - RISC-V 架构定义了一个 64 位宽的指令完成计数器，用于反映处理器成功执行了多少条指令。只要处理器每成功执行完成一条指令，此计数器便会自增计数。
  - CSR 寄存器 minstret 反映了该计数器低 32 位的值，CSR 寄存器 minstreth 反映了该计数器高 32 位的值。有关 minstret 和 minstreth 的详情请参见《Nuclei_N200 系列指令架构手册》。

时钟计数器（Cycle Counter）和指令完成计数器（Instruction Retirement Counter）通常用于测量性能。

默认情况下，计数器在内核复位后的值为 0，然后一直不断的自增计数。由于考虑到计数器计数会消耗某些动态功耗，因此在 N200 系列处理器内核的实现中，在自定义的 CSR 寄存器 mcountinhibit 中额外增加了若干位控制域，软件可以配置相应的控制域分别将不同的计数器关停，从而在不需要使用它们之时停止计数以达到省电的作用。

有关 CSR 寄存器 mcountinhibit 的详情请参见《Nuclei_N200 系列指令架构手册》。

2.14. N200 系列内核的计时器单元

RISC-V 架构定义了一个 64 位的计时器（Timer Counter），该计时器按照系统的低速实时时钟（Real Time Clock）频率进行计时。该计时器的值实时反映在 mtime 寄存器中。RISC-V 架构还

定义了一个 64 位的 mtimecmp 寄存器，该寄存器作为计时器的比较值，假设计时器的值 mtime 大于或者等于 mtimecmp 的值，则产生计时器中断。

注意：RISC-V 架构没有将 mtime 和 mtimecmp 寄存器定义为 CSR 寄存器，而是定义为存储器地址映射（Memory Address Mapped）的系统寄存器，具体的存储器映射地址 RISC-V 架构并没有规定，而是交由内核设计者自行实现。在 N200 系列处理器内核的实现中，mtime/mtimecmp 均由 TIMER 单元实现，有关 N200 系列内核的 TIMER 单元详情请参见《Nuclei_N200 系列指令架构手册》。

2.14.1. 调试模式时的计时器行为

当 N200 系列内核在处于调试模式时会偶尔执行一些调试器（Debugger）设定的代码（DEBUG 单元中，对于用户透明不可见）以支持调试器的功能。如果在执行这些调试器设定的代码时计时器仍然计数，则无法真实反映被调试的程序的真实行为。因此 N200 系列内核在执行调试器设定的代码时，计时器会自动停止计数。

2.14.2. 正常模式时的计时器行为

默认情况下，计时器在内核复位后的值为 0，然后一直不断的自增计时。由于考虑到计时器计数会消耗某些动态功耗，因此在 N200 系列处理器内核的实现中，在自定义的 CSR 寄存器 mcountinhibit 中额外增加了一个位控制域，软件可以配置该控制域将计时器关停，从而在不需要使用它们之时停止计数以达到省电的作用。有关 CSR 寄存器 mcountinhibit 的详情请参见《Nuclei_N200 系列指令架构手册》。

2.15. N200 系列内核的低功耗机制

N200 内核的低功耗机制体现在如下几个方面：

- N200 系列内核内部的各个主要单元的时钟在空闲时都会自动地会被门控关闭以节省静态功耗。
- N200 系列内核能够通过常见的 WFI（Wait for Interrupt）和 WFE（Wait for Event）机制支持休眠（Sleep）模式以实现较低的动态和静态功耗，有关 “Wait for Interrupt”

和“Wait for Event”的详情请参见《Nuclei_N200 系列指令架构手册》。

2.15.1. 进入休眠状态的时钟控制

N200 系列内核可以通过执行 WFI 指令进入休眠状态，有关“如何进入休眠状态”的具体详情请参见《Nuclei_N200 系列指令架构手册》。

N200 系列内核的输出信号 core_sleep_value 可以用于指示不同的休眠模式（0 或者 1），通常可以用休眠模式 0 作为浅度休眠，休眠模式 1 作为深度休眠。注意：当进入深度休眠模式之后，处理器内核将不能够再被 JTAG 调试接口进行调试。

处理器内核进入休眠状态时的时钟控制（参考方案）要点如下：

- 如图 2-1 中，当成功的执行了 WFI 后，N200 系列内核的输出信号 core_wfi_mode 会拉高，指示此处理器核处于执行 WFI 指令之后的休眠状态；SoC 系统层面可以使用 core_wfi_mode 控制外部的总门控时钟将处理器内核的主工作时钟 core_clk 关闭。
- 如果 N200 系列内核进入的是深度休眠模式（core_sleep_value 为 1），SoC 系统可以根据其实际情况决定是否将处内核的常开时钟 core_clk_aon 也关闭。

2.15.2. 退出休眠状态的时钟控制

处理器内核可以被中断（Interrupt）、事件（Event）或者 NMI 唤醒，有关“如何退出休眠状态”的具体详情请参见《Nuclei_N200 系列指令架构手册》。

处理器内核退出休眠状态时的时钟控制要点如下：

- 如果是等待中断（Interrupt）的唤醒，由于 N200 系列内核的中断需要经过 ECLIC 单元的处理和分发，中断只有通过了使能和优先级阈值等条件的判断之后，才能够唤醒内核。除此之外，还需特别注意处理器内核的常开时钟（core_clk_aon）是否关闭：
  - 如第 2.1 节中所述，由于 TIMER 受 core_clk_aon 驱动，因此：
    - 假设 SoC 系统层面已经将处理器内核的常开时钟（core_clk_aon）关闭，则 TIMER 单元由于无时钟，因此其无法产生计时器中断和软件中断。
  - 如第 2.1 节中所述，由于 ECLIC 受 core_clk_aon 驱动，因此：
    - 假设 SoC 系统层面已经将处理器内核的常开时钟（core_clk_aon）关闭，外部

中断信号线拉高之后必须一直保持，直到 SoC 系统层面将处理器内核的的常开时钟（core_clk_aon）再次打开。否则处理器内核的 ECLIC 单元由于没有时钟，无法采样到外部中断信号，从而导致处理器内核无法被唤醒。

- 如果是等待事件（Event）或者 NMI 的唤醒，则内核一旦（通过 core_clk_aon 时钟）采样到输入信号 rx_evt（Event 信号，高电平有效）或者 nmi（NMI 信号，上升沿有效），便从休眠状态中被唤醒。除此之外，还需特别注意处理器内核的常开时钟（core_clk_aon）是否关闭：
  - 假设 SoC 系统层面已经将处理器内核的常开时钟（core_clk_aon）关闭，输入信号 rx_evt 或者 nmi 拉高之后必须一直保持，直到 SoC 系统层面将处理器内核的的常开时钟（core_clk_aon）再次打开。否则处理器内核的 Event 和 NMI 采样逻辑由于没有时钟，无法采样到 Event 和 NMI，从而无法被唤醒。
- 处理器被唤醒后则会马上将输出信号 core_wfi_mode 拉低。假设 SoC 系统层面使用了 core_wfi_mode 控制内核的 core_clk 门控时钟，则随着 core_wfi_mode 信号的拉低，处理器内核的工作时钟 core_clk 将会重新被打开。

2.16. N200 系列内核的扩展指令机制

N200 系列内核提供一个可配置的扩展指令集接口（Nuclei Instruction Co-unit Extension，NICE），以支持用户进行自定义指令的扩展。有关 N200 系列如何通过 NICE 接口扩展协处理器以及 NICE 接口协议的详细介绍，请参见《Nuclei_N200 系列 NICE 使用说明》。

3. N200 系列内核接口介绍

3.1. 时钟和复位接口

N200 系列内核的时钟和复位接口信号如表 3-1 中所示。

表 3-1 时钟和复位接口

信号名	方向	位宽	描述
core_clk_aon	Input	1	■ 此常开时钟用于驱动 N200 系列处理器内核内部的常开逻辑（Always-On Logics）。请参见第 2.1 节了解详情。
core_clk	Input	1	■ 此工作时钟用于驱动 N200 系列处理器内核内部的主要工作逻辑。请参见第 2.1 节了解详情。
por_reset_n	Input	1	■ 上电复位信号。该信号低电平有效，此复位信号将复位整个 N200 系列处理器内核，包括 JTAG 调试部分。 ■ 注意：此信号在内核内部会进行异步时钟同步处理，即将其处理成为“异步置位同步释放”的复位信号。
core_reset_n	Input	1	■ 系统复位信号。该信号低电平有效，此复位信号将复位除了 JTAG 调试部分之外的 N200 系列处理器内核主要功能部分。 ■ 注意：此信号在内核内部会进行异步时钟同步处理，即将其处理成为“异步置位同步释放”的复位信号。
reset_bypass	Input	1	■ 如上所述，异步复位信号需要在 N200 系列处理器内核内部进行“异步置位同步释放”的处理，需要使用到“几级寄存器同步（Synchronizer）”电路。 ■ 如果输入信号 reset_bypass 为高，则将此 Synchronizer 旁路（Bypass）掉，以便于测试目的（Design For Test）。 ■ 注意：如果输入信号 reset_bypass 为高，core_reset_n 复位信号会被旁路，仅有 por_reset_n 复位信号有效。
clkgate_bypass	Input	1	■ 如上所述，N200 系列处理器内核内部会使用到门控时钟。 ■ 如果输入信号 clkgate_bypass 为高，则将时钟门控（Bypass）掉，以便于测试（Design For Test）。

3.2. JTAG 调试接口

N200 系列内核的 JTAG 调试接口信号如表 3-2 中所示。

表 3-2 调试接口

信号名	方向	位宽	描述
jtag_TCK	Input	1	标准 JTAG TCK 信号

jtag_TMS	Input	1	标准 JTAG TMS 信号
jtag_TDI	Input	1	标准 JTAG TDI 信号
jtag_TDO	Output	1	标准 JTAG TDO 信号
jtag_DRV_TDO	Output	1	JTAG TDO 的输出使能信号，当 TDO 进行有效的输出之时，此使能信号为高电平，否则为低电平。

3.3. 外部中断接口

N200 系列内核的外部中断接口信号如表 3-3 中所示。

表 3-3 外部中断接口

信号名	方向	位宽	描述
nmi	Input	1	■ NMI（Non-Maskable Interrupt）中断输入信号。 ■ 注意： 1. nmi_irq 信号在处理器内核内部并没有进行异步时钟同步处理，因此，如果外部中断源与处理器内核处于异步时钟域，则系统集成者需要在外部对其进行异步时钟同步处理。 2. 有关 NMI 机制的详细介绍请参见《Nuclei_N200 系列指令架构手册》。
clic_irq	Input	可配置	■ 来自外部系统的外部中断信号，每个比特对应一个外部中断信号。 ■ 注意： 1. clic_irq 信号在处理器内核内部并没有进行异步时钟同步处理，因此，如果外部中断源与处理器内核处于异步时钟域，则系统集成者需要在外部对其进行异步时钟同步处理。 2. 有关 clic_irq 中断机制的详细介绍请参见《Nuclei_N200 系列指令架构手册》。

3.4. 总线接口

Nuclei N200 系列处理器内核的总线接口包含如下几类：

- （可配置）ILM 主接口（ILM Master Interface）：访问外部的 ILM 的接口。
- （可配置）DLM 主接口（DLM Master Interface）：访问外部的 DLM 的接口。
- （可配置）PPI 接口（Private Priperhal Interface）：访问外部私有外设总线的接口。
- （可配置）FIO 接口（Fast I/O Interface）：访问快速 I/O 总线的接口。

- MEM 接口（System Memory Interface）：访问系统总线的接口。

3.4.1. ILM 和 DLM 主接口

ILM 主接口用于 N200 系列处理器内核访问指令局部存储器（Instruction Local Memory，ILM）。

DLM 主接口用于 N200 系列处理器内核访问外部的数据局部存储器（Data Local Memory，DLM）。

DLM 主接口和 ILM 主接口均可配置为 AHB-Lite 接口或者 SRAM 接口。具体配置参数请参见本文档第 4 章。

假设 ILM 和 DLM 的 AHB-Lite 接口在外部需要进行仲裁，则有如下注意事项：

- DLM 主接口和 ILM 主接口如果在外部需要仲裁，需要确保 DLM 主接口相对 ILM 主接口拥有更高的优先级。
- N200 系列内核提供一个简单的配置选项（N20<x>_CFG_ILM_DLM_EXCLUSIVE），如果打开了此配置，则 ILM 和 DLM 的 AHB-Lite 接口不会同时发出总线 Transactions。此配置可以方便系统集成者使用非常简单的二选一选择器（Mux）来进行 ILM 和 DLM 的 AHB-Lite 总线仲裁，而无需考虑优先级的问题。

3.4.2. ILM 主接口

N200 系列内核的 ILM 主接口可配置为 AHB-Lite 或者 SRAM。

- 当 ILM 接口配置为 AHB-Lite 时，ILM 主接口信号如表 3-4 中所示。
- 当 ILM 接口配置为 SRAM 时，ILM 主接口信号如表 3-5 中所示。

表 3-4 ILM AHB-Lite 接口信号表

信号名	方向	位宽	描述
ilm_htrans	Output	2	■ AHB-Lite 协议的 HTRANS 信号。 ■ 注：在 ILM 接口，只会发出 IDLE 和 NONSEQUENTIAL 两种类型的 Transaction。
ilm_haddr	Output	取决于 ILM 的大小配置	■ AHB-Lite 协议的 HADDR 信号

ilm_hsize	Output	3	■ AHB-Lite 协议的 HSIZE 信号。 ■ 注：在 ILM 接口，只会发出 32 比特（HSIZE 为 b010）的 Transaction。
ilm_hburst	Output	3	■ AHB-Lite 协议的 HBURST 信号。 ■ 注：在 ILM 接口，只会发出 SINGLE 类型（HBURST 为 b000）的 Transaction。
ilm_hprot	Output	4	■ AHB-Lite 协议的 HPROT 信号。 ■ 注意，在 ILM 接口： 1. HPROT[3]的值一定是 1，表示 Cacheable。 2. HPROT[2]的值一定是 0，表示 Non-bufferable。 3. HPROT[1]的值可以是 1（表示这是 Machine Mode 下的访问），或者是 0（表示这是 User Mode 下的访问）。 4. HPROT[0]的值可以是 1（表示这是取数据访问）或者是 0（表示这是取指令访问）。
ilm_hrdata	Input	32	■ AHB-Lite 协议的 HRDATA 信号。
ilm_hresp	Input	2	■ AHB-Lite 协议的 HRESP 信号。 ■ 注：在 ILM 接口仅支持 OKAY 和 ERROR 类型的反馈。
ilm_hready	Input	1	■ AHB-Lite 协议的 HREADY 信号。
ilm_hwrite	output	1	■ AHB-Lite 协议的 HWRITE 信号。
ilm_hmastlock	output	1	■ AHB-Lite 协议的 HLOCK 信号。
ilm_hwdata	output	32	■ AHB-Lite 协议的 HWDATA 信号。

表 3-5 ILM SRAM 接口信号表

信号名	方向	位宽	描述
ilm_cs	output	1	■ ILM 的 SRAM 接口 cs 信号。
ilm_addr	output	取决于 ILM 的大小配置	■ ILM 的 SRAM 接口 addr 信号。
ilm_byte_we	output	4	■ ILM 的 SRAM 接口 wem 信号。
ilm_wdata	output	32	■ ILM 的 SRAM 接口 ram_in 信号。
ilm_rdata	input	32	■ ILM 的 SRAM 接口 ram_out 信号。
clk_ilm_ram	output	1	■ ILM 的 SRAM 时钟信号。

3.4.3. DLM 主接口

N200 系列内核的 ILM 主接口可配置为 AHB-Lite 或者 SRAM。

- 当 ILM 接口配置为 AHB-Lite 时，ILM 主接口信号如表 3-6 中所示。
- 当 ILM 接口配置为 SRAM 时，ILM 主接口信号如表 3-7 中所示。

表 3-6 DLM AHB-Lite 接口信号表

信号名	方向	位宽	描述
dlm_htrans	Output	2	■ AHB-Lite 协议的 HTRANS 信号。 ■ 注：在 DLM 接口，只会发出 IDLE 和 NONSEQUENTIAL 两种类型的 Transaction。
dlm_hwrite	Output	1	■ AHB-Lite 协议的 HWRITE 信号
dlm_haddr	Output	取决于 DLM 的大小配置	■ AHB-Lite 协议的 HADDR 信号
dlm_hsize	Output	3	■ AHB-Lite 协议的 HSIZE 信号。 ■ 注：在 DLM 接口，可以发出 8、16 或者 32 比特的 Transaction。
dlm_hburst	Output	3	■ AHB-Lite 协议的 HBURST 信号。 ■ 注：在 DLM 接口，只会发出 SINGLE 类型（HBURST 为 b000）的 Transaction。
dlm_hprot	Output	4	■ AHB-Lite 协议的 HPROT 信号。 ■ 注，在 DLM 接口： 1. HPROT[3]的值一定是 1，表示 Cacheable。 2. HPROT[2]的值一定是 0，表示 Non-bufferable。 3. HPROT[1]的值可以是 1（表示这是 Machine Mode 下的取指令访问），或者是 0（表示这是 User Mode 下的数据访问）。 4. HPROT[0]的值一定是 1，表示这是数据访问。
dlm_hmastlock	output	1	■ AHB-Lite 协议的 HLOCK 信号。
dlm_master	Output	2	■ 该信号不是 AHB-Lite 的标准信号。 ■ 注：在 DLM 接口，此信号的值可以是 b01（表示这是 Debug-Mode 下的数据访问），或者是 b00（表示这是普通的数据访问）。
dlm_hwdata	Output	32	■ AHB-Lite 协议的 HWDATA 信号。
dlm_hrdata	Input	32	■ AHB-Lite 协议的 HRDATA 信号。
dlm_hresp	Input	2	■ AHB-Lite 协议的 HRESP 信号。 ■ 注：在 DLM 接口仅支持 OKAY 和 ERROR 类型的反馈。
dlm_hready	Input	1	■ AHB-Lite 协议的 HREADY 信号。

表 3-7 DLM SRAM 接口信号表

信号名	方向	位宽	描述

dlm_cs	output	1	■ DLM 的 SRAM 接口 cs 信号。
dlm_addr	output	取决于 DLM 的大小配置	■ DLM 的 SRAM 接口 addr 信号。
dlm_byte_we	output	4	■ DLM 的 SRAM 接口 wem 信号。
dlm_wdata	output	32	■ DLM 的 SRAM 接口 ram_in 信号。
dlm_rdata	input	32	■ DLM 的 SRAM 接口 ram_out 信号。
clk_dlm_ram	output	1	■ DLM 的 SRAM 时钟信号。

3.4.4. MEM 接口

MEM 主接口用于 N200 系列处理器内核访问外部的系统总线，MEM 接口为 AHB-Lite 协议。MEM 接口信号如表 3-8 中所示。

表 3-8 MEM 接口信号表

信号名	方向	位宽	描述
htrans	Output	2	■ AHB-Lite 协议的 HTRANS 信号。 ■ 注，在 MEM 接口： 1. 可以发出 IDLE 和 NONSEQUENTIAL 两种类型的 Transaction。 2. 如果配置了 Cache，则还可以发出 BUSY 和 SEQUENTIAL 类型的 Transaction。
hwrite	Output	1	■ AHB-Lite 协议的 HWRITE 信号
haddr	Output	32	■ AHB-Lite 协议的 HADDR 信号
hsize	Output	3	■ AHB-Lite 协议的 HSIZE 信号。 ■ 注：在 MEM 接口，可以发出 8、16 或者 32 比特的 Transaction。
hburst	Output	3	■ AHB-Lite 协议的 HBURST 信号。 ■ 注，在 MEM 接口： 1. 不是来自 I-Cache 的取指令操作会发出 SINGLE 类型（HBURST 为 b000）的 Transaction。 2. 来自 I-Cache 的取指令操作会发出 INCR8 类型（HBURST 为 b101）的 Transaction。 3. 不是来自 D-Cache 的数据访问操作会发出 INCR 类型（HBURST 为 b001）的 Transaction。 4. 来自 D-Cache 的数据访问操作会发出 WRAP8 类型（HBURST 为 b100）的 Transaction。
hprot	Output	4	■ AHB-Lite 协议的 HPROT 信号。 ■ 注，在 MEM 接口： 1. 如果是来自于 I-Cache 或者 D-Cache 的访问，HPROT[3]的值是 1（表示 Cacheable），否则是 0（表示 Non-Cacheable）。

			2. 如果是来自于 I-Cache 或者 D-Cache 的访问，HPROT[2]的值是 1（表示 Bufferable），否则是 0（表示 Non-Bufferable）。 3. HPROT[1]的值可以是 1（表示这是 Machine Mode 下的访问），或者是 0（表示这是 User Mode 下的访问）。 4. HPROT[0]的值可以是 1（表示这是数据访问）或者是 0（表示这是取指令访问）。
hmastlock	output	1	■ AHB-Lite 协议的 HLOCK 信号
master	Output	2	■ 该信号不是 AHB-Lite 的标准信号。 ■ 注：在 MEM 接口，此信号的值可以是 b01（表示这是 Debug-Mode 下的访问），或者是 b00（表示这是普通的数据访问），或者是 b10（表示这是普通的取指令访问）。
hwdata	Output	32	■ AHB-Lite 协议的 HWDATA 信号。
hrdata	Input	32	■ AHB-Lite 协议的 HRDATA 信号。
hresp	Input	2	■ AHB-Lite 协议的 HRESP 信号。 ■ 注：在 MEM 接口仅支持 OKAY 和 ERROR 类型的反馈。
hready	Input	1	■ AHB-Lite 协议的 HREADY 信号。

3.4.5. PPI 接口

PPI 主接口用于 N200 系列处理器内核访问外部的私有外设总线。

注意：

- PPI 接口协议为 APB 协议，其详细描述如表 3-9 中所示。
- PPI 接口均为可配置接口，用户可以通过配置参数将其配置出来或者去除，具体配置参数请参见本文档第 4 章。

表 3-9 PPI 接口信号表

信号名	方向	位宽	描述
ppi_paddr	Output	32	APB 协议的 PADDR 信号
ppi_pwrite	Output	1	APB 协议的 PWRITE 信号
ppi_psel	Output	1	APB 协议的 PSEL 信号
ppi_dmode	Output	1	自定义信号，表示该 Transaction 是调试模式下的访问。

ppi_penable	Output	1	APB 协议的 PENABLE 信号
ppi_pprot	output	3	APB 协议的 PPROT 信号
ppi_pstrobe	output	4	APB 协议的 PSTRRB 信号
ppi_pwdata	Output	32	APB 协议的 PWDATA 信号
ppi_prdata	Input	32	APB 协议的 PRDATA 信号
ppi_pready	Input	1	APB 协议的 PREADY 信号
ppi_pslverr	Input	1	APB 协议的 PSLVERR 信号

3.4.6. FIO 接口

FIO 主接口用于 N200 系列处理器内核访问外部的快速 IO（Fast-IO）模块，譬如 GPIO 模块。

注意：

- FIO 接口的总线协议为自定义单周期读写访问接口协议，其详细描述如表 3-10 中所示。
- FIO 接口均为可配置接口，用户可以通过配置参数将其配置出来或者去除，具体配置参数请参见本文档第 4 章。

表 3-10 FIO 接口信号表

信号名	方向	位宽	描述
fio_cmd_valid	Output	1	■ 如果该信号为 1，则表示是一个有效的访问。
fio_cmd_addr	Output	32	■ 该访问的地址信号
fio_cmd_read	Output	1	■ 如果该信号为 1，则表示该访问是读操作，否则表示写操作。
fio_cmd_dmode	Output	1	■ 如果该信号为 1，则表示该访问是 Debug-Mode 下的操作。
fio_cmd_mmode	Output	1	■ 如果该信号为 1，则表示该访问是 Machine-Mode 下的操作。
fio_cmd_wdata	Output	32	■ 该访问的写数据
fio_cmd_wmask	Output	4	■ 该访问的写数据字节使能
fio_rsp_rdata	Input	32	■ 该访问的返回读数据。 ■ 注：返回读数据应该发生在 fio_icb_cmd_valid 为 1 时的同一个时钟周期。
fio_rsp_err	Input	1	■ 如果该信号为 1，则表示该访问的返回状态为错误。 ■ 注：返回状态应该发生在 fio_icb_cmd_valid 为 1 时的同一个时钟周期。

3.5. 扩展指令接口

有关扩展指令接口的详细信息，请参见《Nuclei_N200 系列扩展指令说明》。

3.6. Trace 接口

Trace 接口用于输出外部设备所需的 N200 系列处理器内核内部状态信息。Trace 接口信号如表 3-11 所示。

表 3-11 Trace 接口信号

信号名	方向	位宽	描述
trace_ivalid	output	1	■ 如果该信号为 1，则表示有指令正在提交或者处理器进入了 Trap。
trace_iexception	output	1	■ 如果该信号为 1，则表示处理器进入了异常 Trap 或者 NMI Trap。
trace_interrupt	output	1	■ 如果该信号为 1，则表示处理器进入了中断 Trap。
trace_cause	output	32	■ 该信号表示 Trap 的类型。
trace_tval	output	32	■ 该信号表示产生异常的数据。
trace_iaddr	output	32	■ 该信号表示当前提交的指令 PC。
trace_instr	output	32	■ 该信号表示当前提交的指令码。
trace_priv	output	2	■ 该信号表示处理器当前所处的特权模式。

3.7. 其他功能接口

表 3-12 其他功能接口信号表

信号名	方向	位宽	描述
tx_evt	Output	1	■ N200 系列处理器内核可以通过此输出信号 txevt 产生一个单周期脉冲信号，作为对外发送的 Event 信号。 ■ 请参见《Nuclei_N200 系列指令架构手册》了解 CSR 寄存器 txevt 的详细行为。
rx_evt	Input	1	■ 输入信号作为 Wait For Event 的唤醒信号，请参见第 2.15.2 节了解此输入信号的详情。 ■ 请参见《Nuclei_N200 系列指令架构手册》了解 Wait For Event 机制的详情。

mtime_toggle_a	Input	1	■ 来自于 SoC 系统层面的 Real-Time-Clock 的脉冲信号，用于驱动 Core 内部 TIMER 单元的计时器。 ■ 注意： ● 该信号被当做异步输入信号。 ● 在 Core 内部会对该信号进行异步信号同步处理（使用几级寄存器进行同步）。 ● 在进行同步处理之后，会根据 Core 的主时钟对此信号进行上升沿和下降沿的检测，任何一个边沿检测到就会触发 TIMER 的计时器自增加一。 ● 建议此信号使用慢速时钟（譬如 32.768KHz）驱动的寄存器输出（即 2 分频）作为输入信号，处理器内核内部进行上下边沿检测后产生的自增频率即等于慢速时钟的频率，如图 3-1 中所示。慢速时钟的频率越低，则内部计时器的自增频率越低，可以降低动态功耗。
hart_id	Input	1	■ 该 Core 的 HART ID 指示信号，在 SoC 集成时，可以将此信号赋予某个常数值或者信号值。该 ID 号会体现在 Core 内部的 CSR 寄存器 mhartid 中。在单核情形下，可以将此信号直接接 0。
reset_vector	Input	32	■ 该输入信号用于指定处理器被 reset 后的 PC 初始值。在 SoC 层面可以通过控制此信号达到控制处理器核上电 PC 初始值的效果。
hart_halted	Output	1	■ 该输出信号如果为高电平，则指示此 Core 处于调试模式状态。
i_dbg_stop	input	1	■ 该信号如果位高电平，可以将 Core 的调试功能关闭，从而让外界的 JTAG Debugger 没法对 Core 进行调试。
ndmreset	output	1	■ 该信号是 JTAG 调试器发出的对整个 System 进行 reset 的请求。系统集成者可以用此信号 reset 整个 SoC，并且通过连接到 Core 的 core_reset_n（注意：不能够连接到 por_reset_n）达到复位 Core 的工作域的效果。
core_wfi_mode	Output	1	■ 该输出信号如果为高电平，则指示此 Core 处于执行 WFI 指令之后的 Sleep 状态。
core_sleep_value	Output	1	■ 该输出信号指示 Core 的 Sleep 模式。请参见《Nuclei_N200 系列指令架构手册》了解“进入休眠状态”的详情。

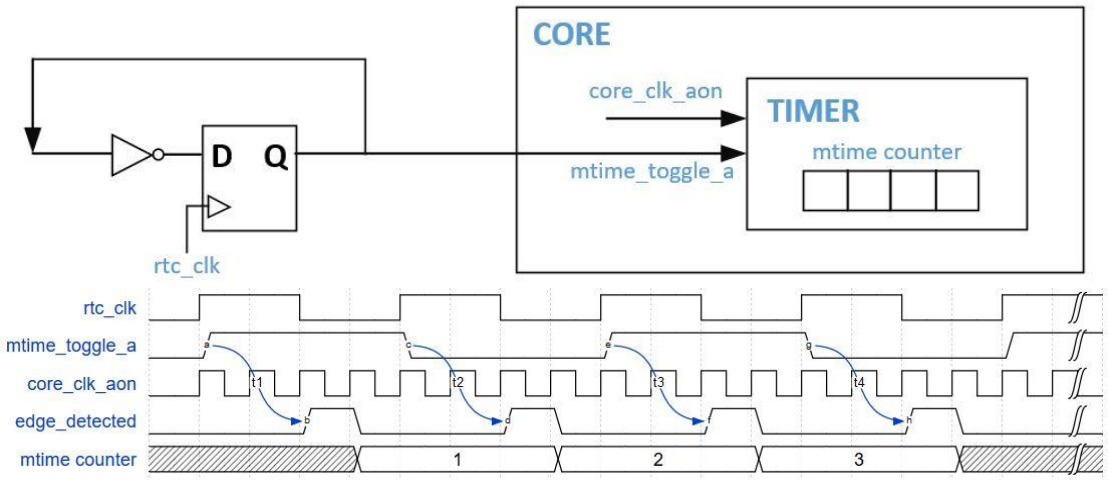

图 3-1 mtime_toggle_a 信号生成示意图

4. N200 系列内核配置选项介绍

N200 系列的每款处理器均具有一定的可配置性。通过修改每个内核 config.v 文件中的宏定义便可以实现不同的配置。

以 N205 内核为例，config.v 文件在 N200 的发布文件包中所在位置如下所示。

```
n200_rls_pkg
        |----rtl                          // 存放 RTL 的目录
            |----n205                     // N205 核和 SoC 的 RTL 目录
                |----core                 // 存放 core 相关模块的 RTL 代码
                    |----config.v         // 设定配置的源文件
```

config.v 中的具体的配置选项宏定义如表 4-1 所示。注意：宏“N20<x>_XXX”由具体的内核型号决定，譬如，如果内核代号为 N205，则该宏为 N205_XXX。

表 4-1 N200 系列内核配置选项

类别	适用型号	功能	宏	描述
指令集相关	N205 N207	是否支持 A 扩展指令	N20<x>_CFG_HAS_AMO	■ 如果添加了此宏，则配置支持 A 扩展指令（原子操作指令）
		是否支持非对齐数据访问	N20<x>_CFG_MISALIGNED_ACCESS	■ 如果添加了此宏，则配置支持非对齐的 Load Store 访问
特权架构相关	N203 N205 N207	是否支持 User Mode	N20<x>_CFG_HAS_UMODE	■ 如果添加了此宏，则配置支持 User Mode。
PMP 相关	N203 N205 N207	是否支持 PMP	N20<x>_CFG_HAS_PMP	■ 如果添加了此宏，则配置支持物理存储器保护单元（PMP）。
		PMP 表项的个数	N20<x>_CFG_PMP_ENTRY_NUM	■ 定义 PMP 表项的个数： ● 8：表示 PMP 表项的个数为 8 个 ● 16 表示 PMP 表项的个数

				为 16 个
乘法器相关	N203 N205	支持多周期或者单周期乘法器	N20<x>_CFG_SHARE_MULDIV N20<x>_CFG_INDEP_MULDIV	■ 配置支持多周期乘法器，还是支持单周期乘法器。注意：除法器总是多周期。 ■ 如果两个宏都没有，则不支持乘法器和除法器
Regfile 复位	N203 N205 N207	是否为 Regfile 配置复位信号	N20<x>_CFG_REGFILE_RST	■ 如果添加了此宏，则为 Regfile 中的每个通用寄存器配置复位信号。 ■ 注意：如果增加了此配置，可以保证通用寄存器的复位初始值为 0，但是可能带来额外的逻辑开销（因为带复位的寄存器面积比不带复位的大）。
调试模块	N203 N205 N207	是否有调试模块	N20<x>_CFG_HAS_DEBUG	■ 如果添加了此宏，则配置使用调试模块。 ■ 注意：调试模块会增加大约 4K Gate 的逻辑开销。
		Debug 模块的基地址	N20<x>_CFG_DEBUG_BASE_ADDR	■ 配置 Debug 模块的地址区间基地址。注意：Debug 模块占据 4K 的地址空间。
		Hardware Trigger 的数目	N20<x>_CFG_DEBUG_TRIGM_NUM	■ 配置 Hardware Trigger 的数目（2/4/8） ■ 注意：每一组 Trigger 需要消耗约 64bits 寄存器的开销。
I-Cache 相关	N207	是否有 I-Cache	N20<x>_CFG_HAS_ICACHE	■ 如果添加了此宏，则配置使用 I-Cache。
		I-Cache 的大小	N20<x>_CFG_ICACH_ADDR_WIDTH	■ 配置 I-Cache 的大小，使用地址总线宽度作为其大小的衡量。譬如，假设 I-Cache 的大小

				为 1KByte，则此宏定义值为 10。
LM 相关	N205 N207	Local Memory 接口类型	N20<x>_CFG_LM_ITF_TYPE_AHBL N20<x>_CFG_LM_ITF_TYPE_SRAM	■ 定义 Local Memory 接口的类型，不同的宏指示不同的接口协议类型。
		ILM 的基地址	N20<x>_CFG_ILM_BASE_ADDR	■ 配置 ILM 的基地址。
		ILM 的大小	N20<x>_CFG_ILM_ADDR_WIDTH	■ 配置 ILM 的大小，使用地址总线宽度作为其大小的衡量。譬如，假设 ILM 的大小为 2KByte，则此宏定义值为 11。
		DLM 的基地址	N20<x>_CFG_DLM_BASE_ADDR	■ 配置 DLM 的基地址。
		DLM 的大小	N20<x>_CFG_DLM_ADDR_WIDTH	■ 配置 DLM 的大小，使用地址总线宽度作为其大小的衡量。譬如，假设 DLM 的大小为 2KByte，则此宏定义值为 11。
		ILM 和 DLM 的传输互斥	N20<x>_CFG_ILM_DLM_EXCLUSIVE	■ 如果添加了此宏，则保证 ILM 和 DLM 的 AHB-Lite 接口不会同时发出总线 Transactions。 ■ 此配置可以方便系统使用非常简单的二选一 Mux 来进行 ILM 和 DLM 的 AHB-Lite 总线仲裁。

		ILM 空间是否能够被 Load、Store 访问到	N20<x>_CFG _LSU_ACCESS_ILM	■ 如果不添加此宏，则 LSU 没有访问 ILM 的物理通路。 ■ 如果添加了此宏，则 LSU 可以访问 ILM，在此配置下如果 ILM 和 DLM 的地址区间配置得重叠，则 LSU 优先访问 ILM 空间。
PPI 接口相关	N205 N207	是否有 PPI 接口	N20<x>_CFG_HAS_PPI	■ 如果添加了此宏，则配置使用 PPI 接口。
		PPI 接口的基地址	N20<x>_CFG_PPI_BASE_ADDR	■ 配置 PPI 接口的基地址。
		PPI 地址区间宽度	N20<x>_CFG_PPI_ADDR_WIDTH	■ 配置 PPI 接口的地址区间，通过指定低位地址宽度来界定地址区间。譬如，如果该 WIDTH 定义为 20，基地址定义为 0x1000_0000，则表示 PPI 的地址区间为 0x1000_0000 ~ 0x100F_FFFF。
FIO 接口相关	N203 N205 N207	是否有 FIO 接口	N20<x>_CFG_HAS_FIO	■ 如果添加了此宏，则配置使用 FIO 接口。
		FIO 接口的基地址	N20<x>_CFG_FIO_BASE_ADDR	■ 配置 FIO 接口的基地址。
		FIO 接口的地址区间宽度	N20<x>_CFG_FIO_WIDTH	■ 配置 FIO 接口的地址区间，通过指定低位地址宽度来界定地址区间。譬如，如果该 WIDTH 定义为 20，基地址定义为 0x1000_0000，则表示 PPI 的地址区间为 0x1000_0000 ~ 0x100F_FFFF。
TIMER 和 ECLIC 相关	N203 N205 N207	TIMER 的基地址	N20<x>_CFG_TMR_BASE_ADDR	■ 配置 TIMER 单元的基地址。

	N203 N205 N207	ECLIC 的基地址	N20<x>_CFG_CLIC_BASE_ADDR	■ 配置 ECLIC 的基地址。
		ECLIC 中断数量	N20<x>_CFG_CLIC_IRQ_NUM	■ 配置 ECLIC 外部中断个数
		ECLIC 的中断级别编码最宽比特数	N20<x>_CFG_CLICINTCTLBITS	■ 用来指定 ECLIC 的中断级数，譬如假设此宏定义为 3 位，则可以编码 8 个中断级别；假设此宏定义为 8 位，则可以编码 256 个中断级别。
		ECLIC 的输出是否 Flop Clean	N20<x>_CFG_CLIC_FLOP_OUT	■ ECLIC 内部进行中断仲裁需要消耗大量组合逻辑。如果添加了此宏，就会将 ECLIC 的仲裁结果进行寄存后输出，从而优化 Timing。但是会增加一个 Cycle 的延迟。
Performance Boost（性能提升）相关选项	N203 N205 N207	是否为 Regfile 配置两个写端口	N20<x>_CFG_REGFILE_2WP	■ 如果添加了此宏，则为 Regfile 配置 2 个写端口，否则使用 1 个写端口。 ■ 注意：如果增加了此配置，可以提升性能 10%，但是可能会带来额外 4K Gates 的逻辑开销。
		是否有 Branch Prediction Unit	N20<x>_CFG_HAS_DYNAMIC_BPU	■ 如果添加了 DYNAMIC_BPU 宏，则配置使用动态分支预测器，否则默认使用静态预测器（BTFN）。
Timing Boost（时序提升）相关选项	N203 N205 N207	是否为置指令预取单元。	N20<x>_CFG_HAS_PREFETCH	■ 如果添加了此宏，则配置使用指令预取单元（为 ILM、I-Cache 和 BIU）。 ■ 注意：如果使用了指令预取单元，将会将（从 ILM、I-Cache 和 BIU）取回的指令放入预取单元，从而砍

				断对外界的 Timing Path，提升时序水平，但是会增加大概 2K Gates 的逻辑开销，且会降低性能 10%（由于增加了取指令的延迟）。
		是否延迟 Branch Flush	N20<x>_CFG_DELAY_BRANCH_FLUSH	■ 如果添加了此宏，则会将分支预测错误造成的 Flush 延迟一拍，从而优化 Timing Path（从 ALU->分支解析->Fetch Interface 关键路径将不复存在）。
		是否使得 LSU 的 AGU 具有私有的加法器	N20<x>_CFG_DEDICATED_AGU	■ 如果添加了此宏，则配置使得 LSU 的 AGU 具有私有的加法器。该配置会改善 Decode-Regfile-Operands-AGU-PMP-BIU/DLM-AHB-lite 路径的 Timing
		是否砍断 MEM 接口的 Timing	N20<x>_CFG_MEM_CUT_TIMING	■ 如果添加了此宏，则 MEM 接口的 CMD 和 RSP Channel 会各自添加一组 Ping-Pong Buffer，从而将外部的信号彻底砍断其 Timing Path，会让 MEM 接口的 Timing Path 与外界彻底断绝联系。
扩展指令接口	N203 N205 N207	是否配置扩展指令接口	N20<x>_CFG_HAS_NICE	■ 如果配置了此宏，则支持 NICE 接口。

注意：

■ 对于不适用的型号，即便配置了相关的宏也不会起作用。譬如对于 I-Cache 而言，其只适用于 N207 内核，假设在 N205 内核的 RTL 中配置了 N205_CFG_HAS_ICACHE，也不会生成 I-Cache 的 RTL 代码。