

---

Silicon Integrated

---

## 蜂鸟 E200 系列 Core & SoC 简介

---

## Content

|   |                                      |    |
|---|--------------------------------------|----|
| 0 | REVISION HISTORY .....               | 1  |
| 1 | 背景简介 .....                           | 2  |
| 2 | 蜂鸟 E200 简介 .....                     | 4  |
| 3 | 蜂鸟 E200 开源型号及配置选项 .....              | 9  |
| 4 | 蜂鸟 E200 系列 CORE & SOC 原型快速使用说明 ..... | 12 |

---

## 0 Revision History

| Date         | Version | Author | Change Summary  |
|--------------|---------|--------|---|
| Sep 29, 2017 | 0.1     | Bob Hu | Initial version   |
| Oct 12, 2017 | 0.2     | Bob Hu | ➤ Removed the redundant E204 core in the introduction, and updated the ISA combination for E203 from IMC/EMC to IMAC/EMAC |
|              |         |        | ➤ Added E203_CFG_SUPPORT_AMO option in Table-4  |
| Oct 13, 2017 | 0.3     | Bob Hu | ➤ Updated the Chapter 2 for some introduction details   |
|              |         |        |   |
|              |         |        |   |
|              |         |        |   |
|              |         |        |   |

# 1 背景简介

由于 IoT 领域的发展迅速，对于超低功耗处理器核的需求巨大，传统的 8 位 8051 内核应用很多在向着 32 位架构迁移。除了选择商用的处理器 IP 之外，使用开放 RISC-V 架构的处理器核也引起了广泛的关注。

目前存在着诸多的开源与商用的 RISC-V 处理器核，对于众多开源实现加以分析，我们可以发现这些现象：

- 目前开源或商用的 RISC-V 实现主要以国外为主，没有一款中国本土的 RISC-V 处理器，难以取得与本土开发人员的交流和支持。
- 大多数的开源实现或者均来自于个人爱好者、或者来自于高校。其开发语言或者使用 VHDL，或者使用高级的 System Verilog。来自产业界工程团队且使用最稳健的 Verilog RTL 实现的开源 RISC-V 处理器尚不多见。
- 最知名的 Rocket Core 使用了高级的 Chisel 语言转换生成 Verilog RTL 代码，造成代码可读性很差，给不熟悉 Chisel 语言的业界工程师使用造成了困难。
- 面向 IoT 领域的高性能且超低功耗开源 RISC-V 处理器，可以选择的并不多，还难以对目前 IoT 领域的主流商用 ARM Cortex-M 系列处理器（2 级或者 3 级流水线实现）形成有效的替代。
- 绝大多数的开源处理器仅提供处理器核的实现，没有提供配套 SoC 和软件示例，用户若要将其使用起来且移植完整软件需要付出额外的努力。
- 绝大多数开源处理器仅提供处理器核的实现，但是并没有提供调试方案的实现，很少有开源处理器能够支持完整的 GDB 交互调试功能。
- 绝大多数开源处理器均文档匮乏。

蜂鸟 E200 系列处理器均可有效解决以上这些问题，与其他的 RISC-V 开源处理器实现相比，它具有如下显著特点：

- 蜂鸟 E200 系列由中国大陆本土研发团队开发，用户能够轻松与本土的开发人员取得交流和支持。
- 蜂鸟 E200 的研发团队来自业界一流的处理器设计公司，使用所有 EDA 工具均稳健支持的 Verilog 2001 语法编写的可综合 RTL 代码。
- 蜂鸟 E200 的代码为人工编写，添加丰富的注释且可读性强，非常易于理解。

- 蜂鸟 E200 专为 IoT 领域量身定做，其具有 2 级流水线深度，功耗和性能指标均优于目前主流商用的 ARM Cortex-M 系列处理器，能够在 IoT 领域完美替代 ARM Cortex-M 处理器。
- 蜂鸟 E200 不仅提供处理器核的实现，还提供完整的配套 SoC、详细的 FPGA 原型平台搭建，详细的软件运行实例。用户可以按照其步骤重现出整套系统，轻松将 E200 处理器核应用其到具体产品中。
- 蜂鸟 E200 不仅提供处理器核的实现、SoC 实现、FPGA 平台和软件示例。还实现了完整的调试方案，具备完整的 GDB 交互调试功能。是从硬件到软件，从模块到 SoC，从运行到调试，的一套完整解决方案。
- 蜂鸟 E200 系列提供详细文档和实例。

## 2 蜂鸟 E200 简介

蜂鸟 E200 系列处理器由武汉聚芯微电子有限公司和北京九天卫星科技发展有限公司联合开发,处理器研发团队拥有多年在国际一流公司开发处理器的经验。蜂鸟是世界上最小的鸟类,其体积虽小却有着极高的速度与敏锐度,可以说是世界上“能效比”最高的鸟类。E200 系列以蜂鸟命名便寓意于此,旨在将其打造成一款世界上最小面积与最高能效比的 RISC 处理器。

蜂鸟 E200 系列处理器核的特性简介如下:

- E200 系列处理器核采用 2 级流水线结构,通过一流的处理器架构设计,该 CPU 核的功耗与面积均优于同级 ARM Cortex-M0+核,实现业界最高的能效比与最低的成本。
- E200 系列处理器核能够运行 RISC-V 指令集,支持 RV32I/E/A/M/C/F/D 等指令子集的配置组合,支持机器模式 (Machine Mode Only)。
- E200 系列处理器核提供标准的 JTAG 调试接口,以及成熟的软件调试工具。
- E200 系列处理器核提供成熟的 GCC 编译工具链,以及 Linux 与 Windows 图形化软件开发工具。
- E200 系列处理器核配套 SoC 提供紧耦合系统 IP 模块,包括中断控制器,计时器, UART, QSPI, PWM 等,以及 Ready-to-Use 的 SoC 平台与 FPGA 原型系统。

蜂鸟 E200 主要面向极低功耗与极小面积的场景而设计,非常适合于替代传统的 8051 内核或者 Cortex-M 系列内核应用于 IoT 或其他低功耗场景。

同时蜂鸟 E200 作为结构精简的处理器核,可谓“蜂鸟虽小、五脏俱全”,源代码全部开源公开,文档翔实,非常适合作为大中专院校师生学习 RISC-V 处理器设计 (使用 Verilog 语言) 的教学实验或自学案例。

蜂鸟 E200 系列处理器的系统示意图如图 1 所示,其提供丰富的存储和接口,包括:

- 私有的 ITCM (指令紧耦合存储) 与 DTCM (数据紧耦合存储),实现指令与数据的分离存储同时提高性能。ITCM 与 DTCM 均基于单周期访问的单口 SRAM 实现, DTCM 的宽度为 32 比特,ITCM 的宽度为 64 比特以节省功耗。ITCM 与 DTCM 均提供外部访问接口可供外部模块访问。ITCM 不仅用于存储指令,也可以被作为数据存储访问。
- 中断接口用于与 SoC 级别的中断控制器连接。
- 调试接口用于与 SoC 级别的 JTAG 调试器连接。

- 系统总线接口，用于访存指令或者数据。可以将系统主总线接到此接口上，E200 可以通过该总线访问总线上挂载的片上或者片外存储模块。
- 紧耦合的私有外设接口，用于访存数据。可以将系统中的私有外设直接接到此接口上，使得 E200 无需经过与数据和指令共享的总线便可访问这些外设。
- 紧耦合的快速 IO 接口，用于访存数据。可以将系统中的快速 IO 模块直接接到此接口上，使得 E200 无需经过与数据和指令共享的总线便可访问这些模块。
- 所有的 ITCM、DTCM、系统总线接口、私有外设接口、快速 IO 接口均可以配置地址区间。

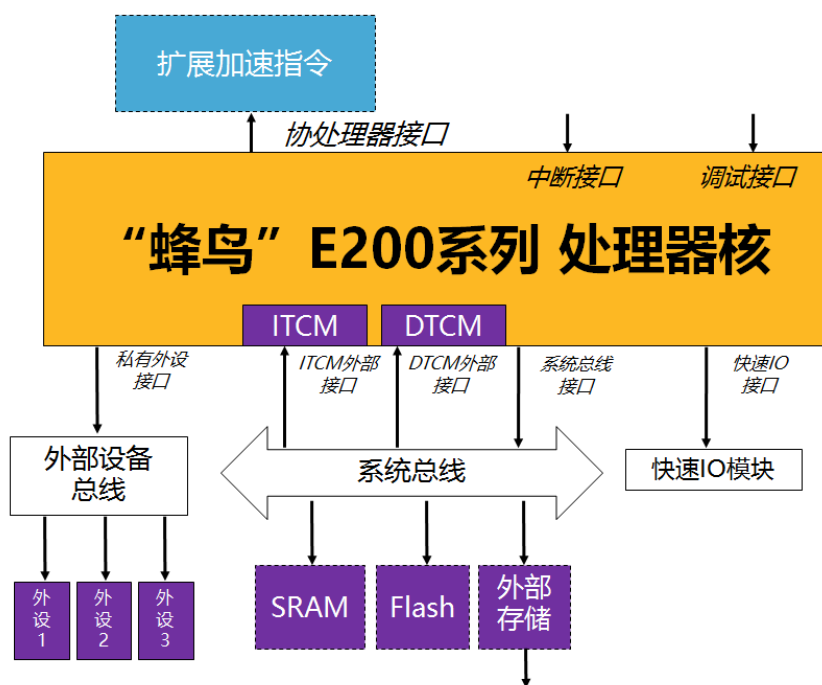


图 1 蜂鸟 E200 处理器系统示意图

蜂鸟 E200 是一个处理器系列，包含了多款不同的具体处理器型号，所有的 E200 系列 Core 均支持协处理器接口可用于自定义扩展指令，不同型号的具体差别如表 1 所示。

- E201 Core 是面积最小的 Core，可以配置为 RV32IC 或者 RV32EC 架构，不支持其他的扩展指令子集。

- E203 Core 可以配置为 RV32IMAC 或者 RV32EMAC 架构，支持面积优化的多周期硬件乘除法单元。
- E205 Core 为 RV32IMAC 架构，支持单周期的硬件乘法单元和多周期的硬件除法单元。
- E205f Core 为 RV32IMAFc 架构，在基本的 E205 基础上加入了单精度浮点运算单元。
- E205fd Core 为 RV32IMAFDC 架构，在基本的 E205 基础上加入了单精度和双精度浮点运算单元。

表 1 E200 处理器型号特性介绍

|                       | E201   | E203          | E205         | E205f | E205fd |
|-----------------------|--|---------------|--------------|-------|--------|
| 支持的 RISC-V 32 位架构指令子集 | IC<br>EC   | IMAC<br>EMAC  | IMAC         | IMAFc | IMAFDC |
| 硬件乘除法单元               | 无  | 有<br>(多周期乘法器) | 有<br>(单周期乘法) |       |        |
| 硬件单精度浮点器              | 无  | 无             | 无            | 有     | 有      |
| 硬件双精度浮点器              | 无  | 无             | 无            | 无     | 有      |
| ECC 保护 SRAM           | 可配置使用 ECC 对 ITCM 与 DTCM 进行保护，ECC 算法对于单比特错误能够纠正，双比特错误能够报错 |               |              |       |        |
| 可扩展性                  | 可以进行指令集扩展（支持协处理器接口）                                      |               |              |       |        |
| 基本外设                  | 提供 SPI, UART, PWM, GPIO, Timer, 中断控制器等基本外设               |               |              |       |        |

蜂鸟 E200 处理器核的功耗与面积以及性能参数非常的有竞争力，如表 2 中所示，蜂鸟 E203 核心的功耗面积和性能均优于 ARM 的 Cortex M0+ 处理器核（M0+ 是 ARM 最小面积的处理器核心）；蜂鸟 E205 核心的功耗面积和性能均优于 Cortex M3 处理器核。而 E205fd 更提供了目前仅在 Cortex M7 中才具备的双精度浮点特性（Cortex M7 是面积特别大的处理器核，难以适应极低功耗要求的场景）。

很多开源的处理器核仅有一个 Core 的实现，为了使其能够被完整使用起来，用户需要花费不少精力来构建完整的 SoC 平台、FPGA 平台、并且对于调试（Debugger）的支持更加的困难。蜂鸟 E200 不仅仅完全自主设计完成了 Core 的实现，还搭配完整的 SoC 平台，如图 2 和表 3 所示。该 SoC 使用可综合 Verilog RTL 代码基于开源的 Freedom E310 SoC 平台二次开发所得，请参见《Hummingbird\_E200\_Series\_Core\_SoC\_Quick\_Start\_Guide.pdf》了解更多 SoC 的介绍与信息。蜂鸟 E200 还提供软件开发环境，支持交互式硬件调试工具（GDB）的支持。用



户可按照文档《Hummingbird\_E200\_Series\_Core\_SoC\_Quick\_Start\_Guide.pdf》中步骤便可快速搭建完整的 SoC 仿真平台或者 FPGA 原型平台，并运行软件示例。

表 2 “蜂鸟 E200 系列核”与“ARM Cortex M 系列核”对比表

|  | ARM<br>Cortex-M0                                | ARM<br>Cortex-M0+                               | ARM<br>Cortex-M3            | 蜂鸟<br>E201                                   | 蜂鸟<br>E203 | 蜂鸟<br>E205 |
|--|---|---|-----------------------------|--|------------|------------|
| Dhrystone<br>(DMIPS/MHz)   | 0.84<br>(Official)<br><br>1.21<br>(Max options) | 0.94<br>(Official)<br><br>1.31<br>(Max options) | 1.25                        | 1.171  | 1.262      | 1.355      |
| CoreMark<br>(CoreMark/MHz)   | 2.33  | 2.42  | 3.32                        | 1.352  | 2.226      | 3.327      |
| 最小配置逻辑门数<br>(K Gates)  | 12K   | 12K   | 36K                         | 10K  | 12K        | 20K        |
| 频率   | 不详  |   |                             | 180nm SIMC 工艺下 50 ~ 100MHz                   |            |            |
| 流水线深度  | 3   | 2   | 3                           | 2  | 2          | 2          |
| 乘法器  | 有   | 有   | 有                           | 无  | 有          | 有          |
| 除法器  | 无   | 无   | 有                           | 无  | 有          | 有          |
| ITCM（指令紧耦合存储）<br>DTCM（数据紧耦合存储）   | 不提供专用 ITCM 与 DTCM 接口                            |   | 提供 32 比特宽的专用 ITCM 与 DTCM 接口 | 提供 64 比特宽的专用 ITCM 接口<br>提供 32 比特宽的专用 DTCM 接口 |            |            |
| ECC 保护 SRAM  | 不提供 ECC 保护的 Memory                              |   |                             | 使用 ECC 保护 ITCM 和 DTCM                        |            |            |
| 可扩展性   | 不支持指令集扩展  |   |                             | 可以进行指令集扩展（支持协处理器接口）                          |            |            |
| 注：<br><br>（1） Cortex M0+的乘法器可以配置成单周期乘法器或多周期迭代乘法器，因此其 Dhrystone 性能数据有两组；CoreMark 性能数据采用单周期乘法或多周期乘法器信息不详。<br><br>（2） 本表格中有关 ARM Cortex M 系列处理器核的性能数据来自于其他公开信息，非官方数据。请以最新 ARM 官方数据为准。 |   |   |                             |  |            |            |

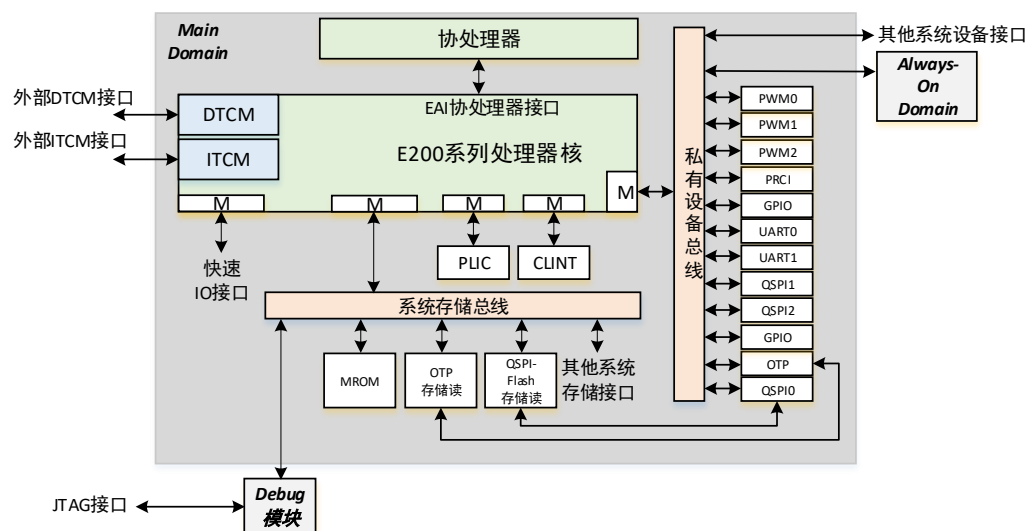


图 2 E200 系列处理器配套 SoC 结构图

表 3 E200 系列处理器配套 SoC 特性

|     | 特性                         | 描述         |
|-----|----------------------------|------------|
| CPU | 使用 Windows/Linux GCC 工具链开发 |            |
|     | 基于 E200 系列处理器核             |            |
|     | 使用标准 JTAG 调试接口             |            |
|     | 支持 GDB 交互式软件调试能力           |            |
|     | 支持中断控制器                    |            |
| 存储  | 片上 ITCM-SRAM（指令）           | 可配置大小      |
|     | 片上 DTCM-SRAM（数据）           | 可配置大小      |
|     | 可通过 QSPI 等接口外接其他片外存储       | 片外 Flash   |
| 外设  | 提供 PWM                     | 3 组        |
|     | 提供 SPI, QSPI               | 3 组        |
|     | 提供 GPIO                    | 32 个 pin 脚 |
|     | 提供 UART                    | 1 组        |
|     | 提供 WatchDog                | 1 组        |
|     | 提供 RTC（Real Time Counter）  | 1 组        |
|     | 提供计时器（Timer）               | 1 组        |

### 3 蜂鸟 E200 开源型号及配置选项

蜂鸟 E200 系列中，目前开源的具体型号为蜂鸟 E203 处理器核。

蜂鸟 E200 系列中的每款处理器均有一定的可配置性。以开源的蜂鸟 E203 核为例，通过修改其目录下的 config.v 文件中的宏定义便可以实现不同的配置。

config.v 文件在 e200\_opensource 目录的结构如下。

```
e200_opensource
|----rtl                // 存放 RTL 的目录
|----e203               // E203 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 所示。

表 4 E203 处理器核配置选项

| 宏   | 描述   | 推荐默认值           |
|---|--|-----------------|
| <b>E203_CFG_DEBUG_HAS_JTAG</b>  | 如果添加了此宏，则配置使用 JTAG 调试接口  | 使用              |
| <b>E203_CFG_ADDR_SIZE_IS_16</b><br><b>E203_CFG_ADDR_SIZE_IS_24</b><br><b>E203_CFG_ADDR_SIZE_IS_32</b> | 三选一，用于配置处理器的寻址地址宽度为 16 位，24 位或者 32 位                           | 32 位            |
| <b>E203_CFG_SUPPORT_MCYCLE_MINSTRET</b>   | 如果添加了此宏，则配置使用 MCYCLE 和 MINSTRET 这两个 64 位的 Performance Counters | 使用              |
| <b>E203_CFG_REGNUM_IS_32</b><br><b>E203_CFG_REGNUM_IS_16</b>  | 二选一，用于配置使用 32 个通用寄存器（RV32I）或者 16 个通用寄存器（RV32E）                 | 32 个            |
| <b>E203_CFG_HAS_ITCM</b>  | 如果添加了此宏，则配置使用 ITCM   | 使用              |
| <b>E203_CFG_ITCM_ADDR_BASE</b>  | 配置 ITCM 的基地址   | 0x8000_0000     |
| <b>E203_CFG_ITCM_ADDR_WIDTH</b>   | 配置 ITCM 的大小，使用地址总线宽度作为大小的衡量，譬如假设 ITCM 的大小为 1KByte，则此宏定义值为 10   | 16<br>(64KByte) |
| <b>E203_CFG_HAS_DTCM</b>  | 如果添加了此宏，则配置使用 DTCM   | 使用              |
| <b>E203_CFG_DTCM_ADDR_BASE</b>  | 配置 DTCM 的基地址   | 0x9000_0000     |

|                                     |   |                 |
|-------------------------------------|---|-----------------|
| <b>E203_CFG_DTCM_ADDR_WIDTH</b>     | 配置 DTCM 的大小，使用地址总线宽度作为大小的衡量，譬如假设 DTCM 的大小为 1KByte，则此宏定义为 10   | 14<br>(16KByte) |
| <b>E203_CFG_REGFILE_LATCH_BASED</b> | 如果添加了此宏，则配置使用 Latch 作为通用寄存器组 (Regfile) 的基本单元。如果没有添加此宏，则使用 D Flip-Flops 作为基本单元。  | 使用 Latch        |
| <b>E203_CFG_PPI_ADDR_BASE</b>       | 配置私有外设接口 (PPI: Private Peripheral Interface) 的基地址   | 0x1000_0000     |
| <b>E203_CFG_PPI_BASE_REGION</b>     | 配置 PPI 接口的地址区间，通过指定高位的区间来界定地址区间。譬如如果该 REGION 定义为 31:28，基地址定义为 0x1000_0000。则表示 PPI 的地址区间为 0x1000_0000 ~ 0x1FFF_FFFF。   | 31:28           |
| <b>E203_CFG_FIO_ADDR_BASE</b>       | 配置快速 IO 接口 (FIO: Fast IO Interface) 的基地址  | 0xf000_0000     |
| <b>E203_CFG_FIO_BASE_REGION</b>     | 配置 FIO 接口的地址区间，通过指定高位的区间来界定地址区间。譬如如果该 REGION 定义为 31:28，基地址定义为 0xf000_0000。则表示 FIO 的地址区间为 0xf000_0000 ~ 0xffff_FFFF。   | 31:28           |
| <b>E203_CFG_CLINT_ADDR_BASE</b>     | 配置 CLINT 接口的基地址。<br>有关 CLINT 介绍，请参见《Hummingbird_E200_Series_Core_SoC_Quick_Start_Guide.pdf》                           | 0x0200_0000     |
| <b>E203_CFG_CLINT_BASE_REGION</b>   | 配置 CLINT 接口的地址区间，通过指定高位的区间来界定地址区间。譬如如果该 REGION 定义为 31:16，基地址定义为 0x0200_0000。则表示 PLIC 的地址区间为 0x0200_0000 ~ 0x0200_FFFF | 31:16           |
| <b>E203_CFG_PLIC_ADDR_BASE</b>      | 配置 PLIC 接口的基地址<br>有关 PLIC 介绍，请参见《Hummingbird_E200_Series_Core_SoC_Quick_Start_Guide.pdf》                              | 0x0C00_0000     |

|                                      |  |  |
|--------------------------------------|--|--|
| <b>E203_CFG_PLIC_BASE_REGION</b>     | 配置 PLIC 接口的地址区间，通过指定高位的区间来界定地址区间。譬如如果该 REGION 定义为 31:24，基地址定义为 0x0C00_0000。则表示 PLIC 的地址区间为 0x0C00_0000 ~ 0x0CFF_FFFF | 31:24  |
| <b>E203_CFG_HAS_ECC</b>              | 如果添加了此宏，则配置使用 ECC 对 ITCM 和 DTCM 的 SRAM 进行保护  | 不使用 ECC。<br>注意：此选项的功能并未开源，因此相关代码并不具备，即便添加了配置宏也不起作用   |
| <b>E203_CFG_HAS_EAI</b>              | 如果添加了此宏，则配置使用协处理器接口  | 不使用协处理器接口。<br>注意：此选项的功能并未开源，因此相关代码并不具备，即便添加了配置宏也不起作用 |
| <b>E203_CFG_SUPPORT_SHARE_MULDIV</b> | 如果添加了此宏，则配置使用面积优化的多周期乘除法单元   | 使用多周期乘除法<br>注意：此选项对于开源的 E203 核必须配置，不可修改。             |
| <b>E203_CFG_SUPPORT_AMO</b>          | 如果添加了此宏，则配置支持 RISC-V 的“A”指令集扩展   | 支持 RISC-V 的“A”指令集扩展<br>注意：此选项对于开源的 E203 核必须配置，不可修改。  |

## 4 蜂鸟 E200 系列 Core & SoC 原型快速使用说明

请参见另外文档《Hummingbird\_E200\_Series\_Core\_SoC\_Quick\_Start\_Guide.pdf》。