

一 工程项目：

- 1 项目名称：cm3_ahbmtx_mcu
- 2 项目使用的资源与软件：
 - (1) VCS-2022
 - (2) Verdi-2022
 - (3) gcc-arm-none-eabi v10.3-2021.10
 - (4) Quartus Prime 2018
 - (5) Vivado 2021.1
 - (6) Spyglass-2021
 - (7) 野火串口助手
 - (8) J-Link commander v7.92f
- 3 项目使用的资源与硬件：
 - (1) ep4ce55 的 ad9361 开发板及 usb-blaster 下载器
 - (2) 正点原子 zynq 领航者开发板及 xilinx 下载器
 - (3) Jlink 盒子

二 工程目标：

- 1 实验现象：
 - (1) 搭建一个使用 ARM Cortex-M3 为 CPU、AHB 总线矩阵为基础的 MCU。
 - (2) 使用 Jlink 的 SWD 模式可以正确地读写指定地址。
 - (3) 使用 Jlink 的 SWD 模式可以正确地读写 SRAM。
 - (4) 当 CPU 开始运行时可运行在 ITCM 中存放的 bootloader。
 - (5) CPU 可以正确驱动 Uart 输入、输出。
 - (6) CPU 可以正确驱动 GPIO 输入、输出。
 - (7) CPU 可以正确驱动基本定时器进行计时操作。
 - (8) CPU 可以正确驱动网卡并于上位机建立连接。
 - (9) CPU 可以正确驱动网卡收发以太网帧。
- 2 实验内容：
 - (1) 在 MCU 内集成 CPU、总线、存储器、外设等。
 - (2) 使用 Jlink 的 SWD 模式读写指定地址。
 - (3) 使用 Jlink 的 SWD 模式读写 SRAM。
 - (4) ITCM 中存放一个 bootloader。
 - (5) 使用 CPU 驱动 Uart，完成输入、输出功能的实现。
 - (6) 使用 CPU 驱动 GPIO，完成输入、输出功能的实现。
 - (7) 使用 CPU 驱动基本定时器，完成定时的功能。
 - (8) 使用 CPU 驱动网卡与上位机建立连接。
 - (9) 使用 CPU 驱动网卡与上位机通信。
- 3 具体要求：
 - (1) CPU 使用 ARM Cortex-M3。
 - (2) 片内总线使用 AHB 总线矩阵。
 - (3) 使用异步 AHB2APB 桥。
 - (4) 使用 SWD 模型读写寄存器并进行仿真。
 - (5) 分别使用 RTL 和 FPGA IP 实现 SRAM 并进行读写操作仿真。

- (6) 在 ITCM 中存放一个 bootloader, 当 CPU 开始运行时即运行该程序。
- (7) 编写 Uart 的 RTL, 编写配套的 firmware 驱动并实现输入、输出的功能, 完成输入输出回环测试。
- (8) 编写 GPIO 的 RTL, 编写配套的 firmware 驱动并实现输入、输出的功能, 完成输入输出回环测试。
- (9) 编写基本定时器的 RTL, 编写配套的 firmware 驱动并实现计时功能, 包括中断方式定时。
- (10) 编写以太网的 SMA 接口的 RTL, 编写配套的 firmware 驱动并实现与上位机连接。
- (11) 编写以太网的 MAC 接口的 RTL, 编写配套的 firmware 驱动并实现与上位机通信。
- (12) 编写高级定时器的 RTL, 编写配套的 firmware 驱动并实现与上位机通信。

三 学习目标（下述内容非先后顺序）：

1 学习 MCU 的基本架构。

- (1) 了解 MCU 内部的基本架构。
- (2) 掌握 MCU 的集成方式。
- (3) 掌握 MCU 的仿真环境与 Testbench、激励的编写。

2 学习 ARM Cortex-M3 为 CPU:

- (1) 了解 ARM Cortex-M3 为 CPU 的基本功能和接口。
- (2) 掌握 ARM Cortex-M3 为 CPU 如何和总线进行链接。
- (3) 掌握 ARM Cortex-M3 为 CPU 与 SWD 的链接方式。
- (4) 掌握 ARM Cortex-M3 为 CPU 如何连接时钟信号和复位信号。
- (5) 掌握 ARM Cortex-M3 为 CPU 如何连接各种中断线, 包括同步中断和异步中断。

3 学习 AHB 总线矩阵:

- (1) 了解 AHB 总线矩阵的基本读写时序。
- (2) 了解 AHB 总线矩阵的生成。
- (3) 掌握 AHB 总线矩阵与 CPU、AHB2APB 桥、外设的连接方式。
- (4) 掌握 AHB 总线矩阵如何连接时钟信号和复位信号。

4 学习 AHB2APB 桥和 APB 总线:

- (1) 了解同步和异步 AHB2APB 桥的基本读写时序。
- (2) 了解 APB 总线的基本读写时序。
- (3) 掌握同步和异步 AHB2APB 桥和 AHB 总线矩阵、APB 总线的连接方式。
- (4) 掌握 APB 总线如何连接时钟信号和复位信号。
- (5) 掌握 APB 总线如何连接外设（寄存器）。

5 学习 SWD 时序与 JTAG 使用方法:

- (1) 了解 SWD 的基本工作原理与模型。
- (2) 掌握 SWD 读写指定地址的操作方法。
- (3) 掌握使用 Jlink 的 SWD 模式读写指定地址。

6 学习 SRAM:

- (1) 掌握 SRAM 读写接口与时序。
- (2) 掌握 SRAM 的 RTL 实现和 FPGA IP 核实现。

7 学习 bootloader:

- (1) 了解什么是 bootloader。
- (2) 掌握 bootloader 的 firmware 的用法。

8 学习 Uart:

- (1) 学习 Uart 及其通信协议。
- (2) 学习 Uart 的硬件 RTL 实现。
- (3) 对 Uart 进行功能测试和 spyglass 检查, 修改其中的错误和问题。
- (4) 学习 Uart 的软件驱动编写。
- (5) 掌握如何使用 Uart 并实现特定的功能。

9 学习 GPIO:

- (1) 学习 GPIO 的硬件 RTL 实现。
- (2) 对 GPIO 进行功能测试和 spyglass 检查, 修改其中的错误和问题。
- (3) 学习 GPIO 的软件驱动编写。
- (4) 掌握如何使用 GPIO 并实现特定的功能。

10 学习基本定时器:

- (1) 学习基本定时器的硬件 RTL 实现。
- (2) 对基本定时器进行功能测试和 spyglass 检查, 修改其中的错误和问题。
- (3) 学习基本定时器的软件驱动编写。
- (4) 掌握如何使用基本定时器并实现特定的功能。

11 学习以太网:

- (1) 学习以太网 SMA 和 MAC 的硬件 RTL 实现。
- (2) 对以太网 SMA 和 MAC 进行功能测试和 spyglass 检查, 修改其中的错误和问题。
- (3) 学习以太网 SMA 和 MAC 的软件驱动编写。
- (4) 掌握如何使用以太网 SMA 和 MAC 并实现特定的功能。

12 学习基本定时器:

- (1) 学习基本定时器的硬件 RTL 实现。
- (2) 对基本定时器进行功能测试和 spyglass 检查, 修改其中的错误和问题。
- (3) 学习基本定时器的软件驱动编写。
- (4) 掌握如何使用基本定时器并实现多路 PWM 输出、输入捕获、编码器输入的功能。

13 学习 VCS 与 Verdi 联合仿真环境:

- (1) 掌握如何使用 VCS 进行仿真并使用 Verdi 记录波形。

14 学习 gcc-arm-none-eabi 编译 firmware 环境:

- (1) 掌握如何使用 gcc-arm-none-eabi 编一个 cpu 可执行的程序。

15 学习 Quartus 和 Vivado 的 fpga 综合编译环境:

- (1) 掌握如何使用 Quartus 和 Vivado 生成或修改 IP 核, 包括 RAM 核和 PLL 核。
- (2) 掌握如何使用 Quartus 和 Vivado 进行 fpga 综合并下载 bit 文件。
- (3) 掌握如何使用 Quartus 和 Vivado 进行在线 debug。
- (4) 掌握如何使用 Quartus 和 Vivado 编写时序约束文件并进行时序分析。
- (5) 掌握如何使用 Quartus 和 Vivado 进行 ECO 操作。

16 学习 spyglass 环境:

- (1) 掌握如何编写 sgdc 约束文件。
- (2) 掌握如何使用 spyglass 进行语法和 cdc 分析。
- (3) 了解根据 spyglass 的分析结果对于 rtl 的意义。

四 可以重复利用的资源：

- 1 工程项目目录下 Library 下面的内容，包括 Altera 及 Xilinx 的部分仿真库、基本功能逻辑 RTL 的实现、Cortex-M3 的建议版本模型、CMSDK 模型（包括 AHB 总线矩阵、APB 总线）、SWD 模型、仿真结果监视器模型等。
- 2 工程项目目录下 User\Verilog，这里是 MCU 的基本集成结构，另外包含 Uart、GPIO、基本定时器、以太网 SMA 外设、高级定时器。
- 3 工程项目目录下 Library\drivers 和 Library\cmsis，这里是 MCU 的基本软件环境和 Uart、GPIO、基本定时器、以太网 SMA、高级定时器的 RTL 配套驱动程序。
- 4 便于使用者进行工作的“爱上实验室”开发工程环境，包含 setenv.sh 文件和 Script 目录下的 makefile 文件。

五 注意事项：

- 1 该项目会持续更新。
- 2 小编时间仓促，因此项目中有不完善的地方敬请大家提出建议。

六 “爱上实验室”开发工程环境的目录介绍：

项目目录下，各文件夹的含义是（含空文件夹，加粗的是本项目包含的文件）：

Doc 目录为项目文档，包括参考资料、设计文档、原理图等。

Library 目录为项目所使用到的库文件，包括仿真库、Monitor、Model 等。

Project 目录为项目所涉及到的工程，包括 arm-none-eabi 工程、fpga 工程等。

Script 目录包含“爱上实验室”开发工程环境专用的脚本、功能等。

Sim 目录为项目包含的仿真的目录，包含仿真工程、spyglass 工程、testbench、激励文件、脚本等、以及仿真专用代码、脚本等。

User 目录包含用户主要的项目文件，包含 rtl 代码、bootloader 等。

目前最适合的使用方法是在 Linux 环境下运行，很多脚本方便使用，可以体会到“爱上实验室”开发工程环境的便利。使用者只需根据自己的环境配置 setenv.sh 文件即可，默认使用的是 modulefiles 模块管理 eda 工具。

Source setenv.sh 后：

快捷键可以进行目录跳转，常见的例如：

cdc 跳转至开发环境专用脚本目录；

cdv 跳转至 verilog 目录；

cdfw 跳转至 firmware 目录；

cdq 跳转至 fpga 目录；

cds 跳转至仿真目录；

cdtv 跳转至仿真测试激励目录；

cdtb 跳转至仿真 testbench 目录等。

工程根目录下面的 script 文件夹里，即为“爱上实验室”开发工程环境专用的脚本、功能，目前包括以下种类：

RTL 功能仿真；

带 firmware 程序的 RTL 仿真；

AHB 总线矩阵生成;

连接 APB 总线的寄存器生成;

firmware 编译;

fpga 综合;

带 firmware 的综合仿真等。

用户可以根据自己的需要，建立对应的 `makefile.xxxx` 文件，同时在总的 `makefile` 文件中加入对应的命令即可。在“爱上实验室”开发工程环境的任意目录下，使用 `run xxx` 命令，即可开始进行 xxx 功能的操作，例如仿真等操作。