

FPGA Cortex-M1 SoC DEMO

应用指南

(AN02007, V1.1)

(2020.12.24)

深圳市紫光同创电子有限公司

版权所有 侵权必究

文档版本修订记录

版本号	发布日期	修订记录
V0.1	2019-10-10	初始版本
V0.2	2019-10-29	增加 DDR 介绍
V0.3	2019-12-13	增加 ICACHE 介绍
V0.4	2020-03-02	增加 DCACHE 介绍
V0.4-1	2020-03-09	DCACHE 寻址空间至 16MB
V0.5	2020-03-25	优化 SoC 资源，剔除 DDR 外设及相关逻辑，DCACHE 寻址空间至 256MB
V0.6	2020-03-30	添加仿真说明与完善 DDR 颗粒与 CACHE 映射关系
V0.7	2020-07-30	增加 Ethernet DMAC 介绍
V0.8	2020-08-21	增加以太网网速测试和短包补零功能，以及更改 ICACHE 大小
V0.9	2020-09-25	增加 APB 接口描述
V1.0	2020-10-28	增加 UDP 加速功能描述和外设扩展描述
V1.1	2020-12-24	取消具体芯片的资源介绍

名词术语解释

Abbreviations 缩略语	Full Spelling 英文全拼	Chinese Explanation 中文解释
AHB	Advanced High performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	高级外设总线
ITCM	Instruction Tightly-Coupled Memory	指令紧耦合空间
DTCM	Data Tightly-Coupled Memory	数据紧耦合空间
NMI	Non-Maskable Interrupt	异常处理
MEM	Memory	存储空间
DDR	Double Data Rate	双倍速率
ICACHE	Instruction Cache	指令缓存
DCACHE	Data Cache	数据缓存
DMAC	Direct Memory Access controller	直接内存存取控制器
UDP	User Datagram Protocol	用户数据报协议

目录

1.	概述	1
1.1.	介绍	1
1.2.	主要功能	1
1.3.	设计信息	1
1.4.	Soc资源使用情况	2
2.	功能描述	3
2.1.	Cortex M1 SoC设计架构	3
2.1.1.	Cortex M1软核模块	3
2.1.2.	GPIO模块	3
2.1.3.	UART模块	4
2.1.4.	TIMER模块	4
2.1.5.	WATCHDOG模块	4
2.1.6.	SPI模块	4
2.1.7.	I2C模块	4
2.1.8.	MEM模块	4
2.1.9.	ICACHE模块	4
2.1.10.	DCACHE模块	5
2.1.11.	Ethernet DMAC模块	5
2.1.12.	UDP SPEEDUP模块	6
2.1.13.	其他模块	6
2.2.	接口列表	7
2.3.	宏参数定义	9
2.4.	寻址空间地址映射	11
2.5.	接口时序	12
3.	参考设计	12
3.1.	参考功能设计	12
3.2.	参考设计文件目录	13
3.3.	参考设计仿真	13
3.4.	参考设计上板验证	14
4.	注意事项	15

图目录

图 1 CORTEX M1 SoC架构框图	3
图 2 MEM时序图	4
图 3 ETHERNET DMAC系统结构图	5
图 4 UDP SPEEDUP系统结构图	6
图 5 源数据输入时序	6
图 6 AHB READ TRANSFER.....	12
图 7 AHB READ TRANSFER WITH WAITE STATES	12
图 8 AHB WRITE TRANSFER	13
图 9 AHB WRITE TRANSFER WITH WAITE STATES.....	13
图 10 APB无等待写	13
图 11 APB无等待读	13
图 12 数据拼接操作	12
图 13 文件目录	13

表目录

表 1 CORTEX M1软核信息	1
表 2 SoC各模块资源.....	2
表 3 CORTEX M1 SoC接口列表	7
表 4 功能模块使能宏说明表	9
表 5 FPGA CORTEX-M1标准外设地址映射	11

1. 概述

1.1. 介绍

本文档为深圳市紫光同创电子有限公司基于Cortex M1软核上的SoC DEMO产品及应用文档。本文当主要介绍了Cortex M1 SoC DEMO的功能列表、设计架构、接口定义、接口时序、支持器件以及参考设计等。

1.2. 主要功能

Cortex M1所组成的单核片上系统支持的主要功能如下：

- 支持32-bit Thumb-2 BL, MRS, MSR, ISB, DSB, and DMB指令集
- 三级流水线与哈弗结构
- SoC主频125MHz
- 支持AHB系统总线，AHB转APB总线桥
- 支持FreeRTOS操作系统
- 支持1、8、16、32个外部中断与NMI异常处理
- 支持ModelSim的通用加速仿真平台
- 支持16个GPIO（含16个GPIO输入中断）、2个UART（含接收中断）、2个TIMER（含中断）、1个WATCHDOG（含中断）、1个Master SPI(8个片选)、1个Master I2C（含中断）、1个MEM（软核可挂16个最大16MB的RAM）、1个Systick系统定时器
- 支持可配置的ITCM与DTCM，默认4KB，最大1MB（当不使用CACHE时，请根据实际工程需求配置合适的ITCM与DTCM空间）
- 支持ICACHE，固定值8KB，DDR颗粒映射范围0x00000000-0x00FFFFFF，解决ICACHE一致性问题
- 支持DCACHE，固定值8KB，DDR颗粒映射范围0x01000000-0x0FFFFFFF，解决DCACHE一致性问题
- 支持千兆以太网的RGMII通信，实现裸机Lwip与FreeRTOS Lwip协议栈
- 支持千兆以太网MAC层的发送加速功能，发送最高速率为990Mbps，用户数据长度固定为1460字节（帧总长为42+1460字节）
- 支持Bootloader，系统上电后指令从SPI-FLASH加载至DDR颗粒中
- 支持用户挂载AHB/APB外设或用户直接与软核外部寄存器/RAM数据交互

1.3. 设计信息

表 1 Cortex M1软核信息

Cortex M1软核	
支持器件	全系列FPGA产品，仅测PGL和PG2L
支持用户接口	高级高性能总线(AHB)
提供的设计文件	

Cortex M1设计文件	Verilog加密文件
Cortex M1参考设计	Verilog加密文件
Cortex M1仿真文件	Verilog加密文件
Cortex M1约束文件	fdc文件
开发工具支持	
RTL设计工具	PDS开发套件- Pango Design Suite 2020.4-rc3
仿真工具	ModelSim10.2c

1.4. Soc 资源使用情况

表 2 SoC各模块资源

模块名	LUT4	FF	DRM	说明
Cortex M1	2778	1061	4	不支持在线调试
GPIO	242	107	0	
UART0	187	116	0	
UART1	167	99	0	
TIMER0	169	77	0	
TIMER1	155	77	0	
WATCHDOG	215	114	0	
SPI	138	79	0	
I2C	302	159	0	
MEM	5	5	0	
ICACHE	557	532	5	大小8KB
DCACHE	924	443	5	大小8KB
Ethernet_DMACH	882	596	5	
UDP SPEEDUP	605	620	1	

2. 功能描述

2.1. Cortex M1 SoC 设计架构

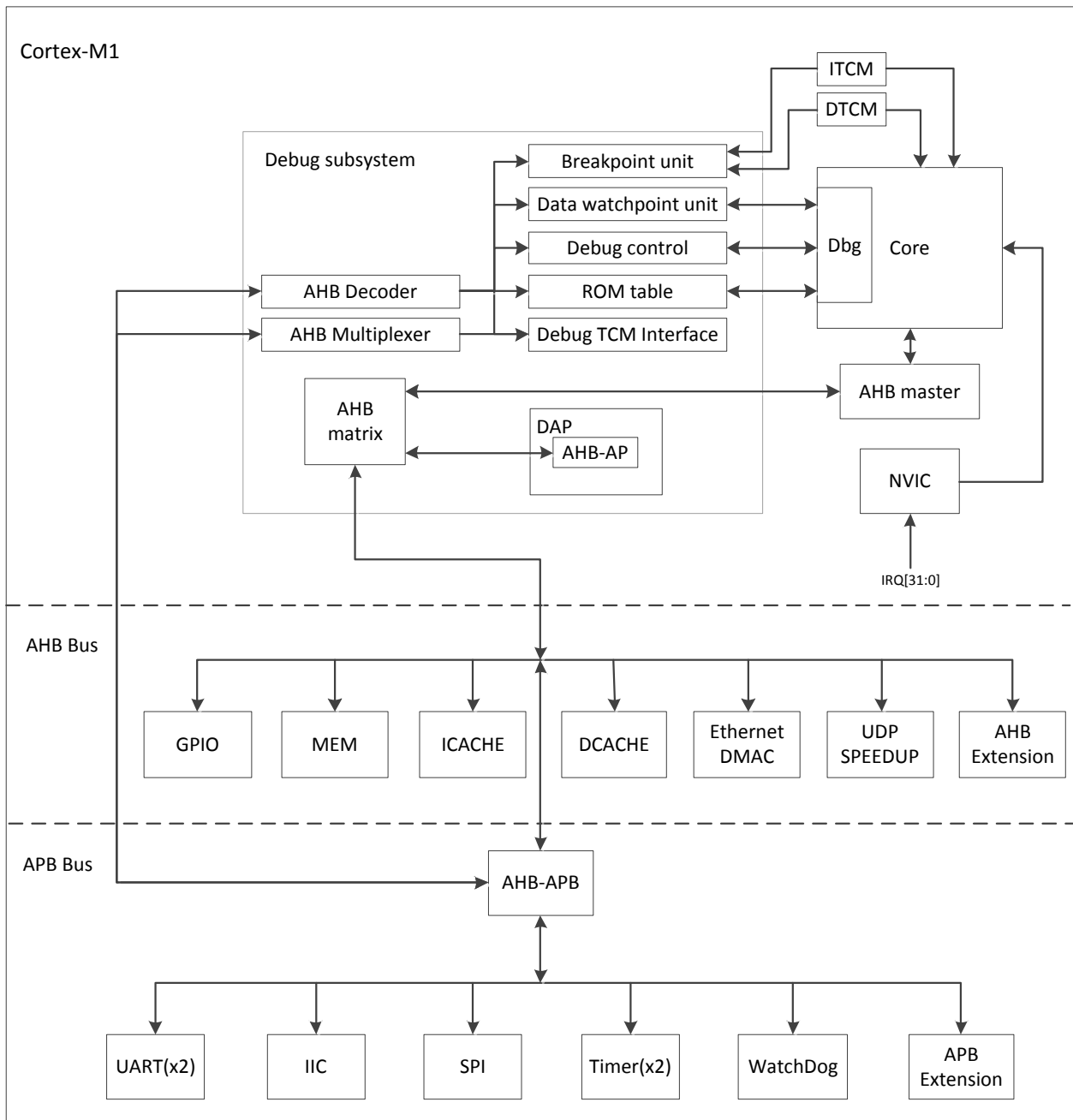


图 1 Cortex M1 SoC架构框图

2.1.1. Cortex M1 软核模块

该模块主要包括Cortex M1软核，以及Debug、Core、NVIC等模块组成，具体内容请参见官方已发布文档所示，即《Cortex-M1_TechnicalReferenceManualRevD_r1p0-00rel0.pdf》。

2.1.2. GPIO 模块

该模块规划了16个普通GPIO口，其中GPIO0至GPIO15均可作为外部输入去触发中断，内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

2.1.3.UART 模块

该模块由分频系数和数据收发控制组成，主要用于C代码调试中的字符打印，其逻辑部分完全遵循UART协议，支持输入中断，其内部详细的寄存器描述与配置请参见《FPGA Cortex-M1 软核编程应用指南》。

2.1.4.TIMER 模块

该模块是一个独立的可编程的32位计数器，当计数到0时产生独立的中断，TIMER内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

2.1.5.WATCHDOG 模块

该模块是“看门狗”模块，防止程序发生死循环，或者程序跑飞等错误情形，WATCHDOG中断触发为NMI异常类型，内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

2.1.6.SPI 模块

该模块主要实现Master SPI协议功能，支持4、8、16、32分频以及动态分频，可同时挂载8个从设备，SPI内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

2.1.7.I2C 模块

该模块主要实现Master I2C协议功能，支持标准100KHz与400KHz，同时当总线空闲时将会产生中断，I2C内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

2.1.8.MEM 模块

该模块主要实现外部memory与Cortex M1数据交互功能，在系统100Mhz AHB总线协议控制下可实现数据的写入和读出，并且可挂载16个最大16MB的RAM，MEM时序遵循图2所示时序，内部详细的寄存器描述与配置请参见《FPGA Cortex-M1软核编程应用指南》。

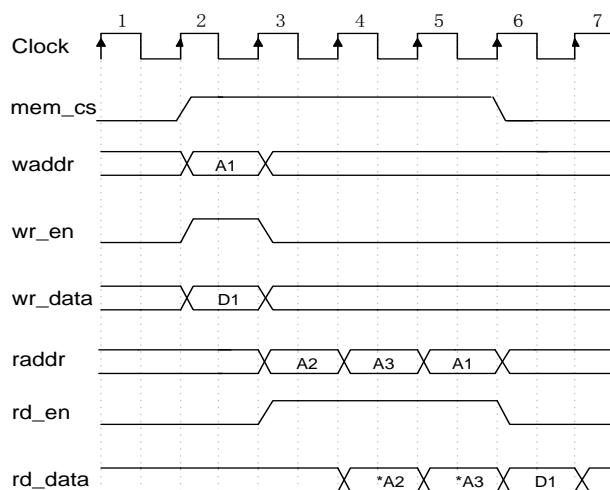


图 2 MEM 时序图

2.1.9.ICACHE 模块

该模块主要利用 AHB 总线协议的有等待读操作和 ICACHE 缓存指令功能，将指令从片外 DDR 颗粒存储空间内提取并输出至 Cortex M1 软核中，保证软核能正常运行。其中，ICACHE

大小固定配置为8KB,在DDR颗粒的映射范围为0x00000000-0x00FFFFFFF。同时,解决ICACHE指令一致性问题,即Cortex M1软核和软核外部逻辑共同访问DDR相同空间时,所引起指令不一致性问题。

当软核所需指令未在ICACHE空间(即未命中)时,ICACHE将会自动从DDR颗粒中提取所需指令并存储至对应位置处,之后软核将会把该指令送入软核中,完成指令的提取操作。

注意,ICACHE同时支持DDR颗粒的写操作,因此在系统进行Bootloder时,可利用ICACHE将SPI-FLASH中的指令写进DDR颗粒中。

2.1.10.DCACHE 模块

该模块主要利用AHB总线协议的有等待读/写操作和DCACHE缓存指令功能,将数据从片外DDR颗粒存储空间内提取并输出至Cortex M1软核中或将软核产生的数据存入DDR颗粒中。其中,DCACHE大小固定配置为8KB,在DDR颗粒的映射范围为0x01000000-0x0FFFFFFF。同时,解决DCACHE指令一致性问题,即Cortex M1软核和软核外部逻辑共同访问DDR相同空间时,所引起数据不一致性问题。

当软核所需指令未在ICACHE空间(即未命中)时,ICACHE将会自动从DDR颗粒中提取所需指令并存储至对应位置处,之后软核将会把该指令送入软核中,完成指令的提取操作。

注意,DCACHE支持AXI4读写操作DDR颗粒,可访问除ICACHE映射空间外的所有DDR颗粒片上空间,但只支持单字节的突发读写操作。

2.1.11.Ethernet DMAC 模块

该模块支持千兆以太网的RGMII通信,利用DMA使Ethernet MAC与DDR不通过Cortex M1而直接进行数据的传输,同时配合“CACHE一致性功能”实现以太网数据的更新和Cortex M1数据的回写,其Ethernet DMAC系统结构图如图3所示。

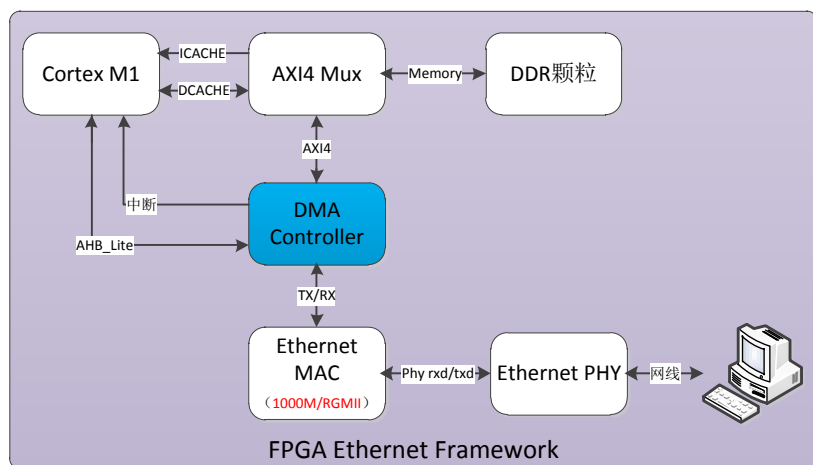


图 3 Ethernet DMAC系统结构图

注意,该模块利用我司的TSMAC IP实现了接收包过滤和发送包补零功能。其中,接收包过滤是将与目的MAC不匹配的包过滤掉,减少无用数据包的接收,以减轻嵌入式软核的数据

处理压力，如用户需要修改目的MAC，请在..\pgr_ARM_Cortex_M1\src\tsmac_phy\config_reg.v中修改第142和第165行即可，默认目的MAC为0x102030405060。而发送包补零是将不足64字节的发送包自动补零且重新计算CRC值，满足交换机的应用需求。

2.1.12.UDP SPEEDUP 模块

该模块支持千兆以太网的发送加速功能，利用 FPGA 硬件逻辑实现 FPGA 芯片与外部客户端的 UDP 快速通信，其 UDP 发送速率最高为 990Mbps(即源数据输入速率最高为 990Mbps)，用户数据长度固定为 1460 字节(帧总长 42+1460 字节)，UDP SPEEDUP 系统结构如图 4 所示。

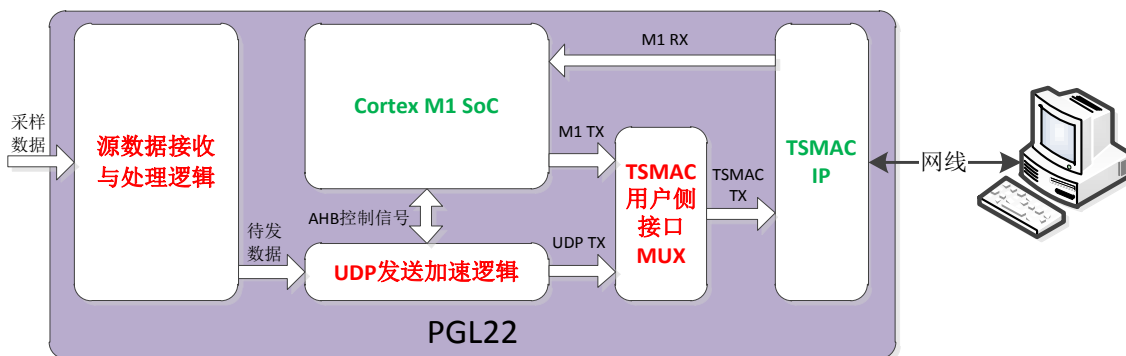


图 4 UDP SPEEDUP系统结构图

由于Ethernet DMAC模块的发送端处理速度无法满足实际的应用场景，因此在使用Ethernet DMAC与外部客户端建立网络连接后，需要使用UDP SPEEDUP模块大幅度提升发送端网速，两则配合且分时占用我司的TSMAC IP用户侧发送接口，以实现网络数据的外发，其详细的C操作流程请参见《FPGA Cortex-M1软核编程应用指南》。

注意，使用UDP SPEEDUP发送加速功能目的是将大量的源数据快速发送，因此源数据输入该模块时序入图5所示。

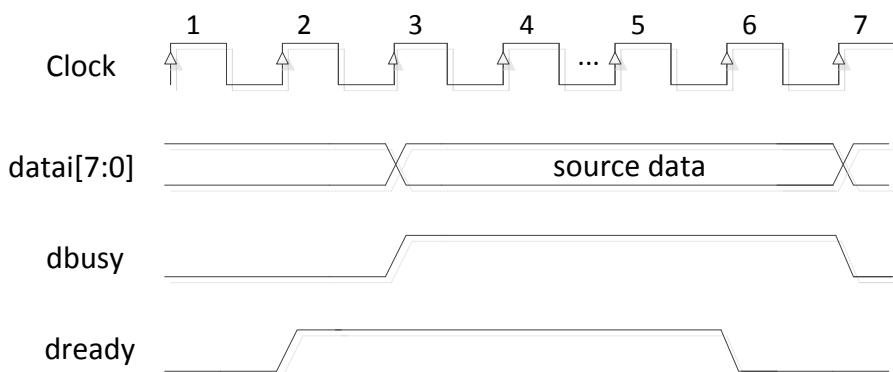


图 5 源数据输入时序

2.1.13.其他模块

PLL、双口RAM、软复位、按键复位、外部按键中断等均较为普通不做介绍。

在使用软核或客户扩展外设时，请注意以下几点内容：

- 1) 驱动AHB和APB外设的时钟均为Cortex M1软核的系统时钟；
- 2) Cortex M1所有的复位操作至少要持续3个时钟周期；
- 3) SysTick系统定时器精度可由软件设置，默认为1us。

2.2. 接口列表

表 3 Cortex M1 SoC接口列表

信号名	I/O	位宽	描述
顶层信号			
ex_clk_50m	I	1	晶振50MHz
rst_key	I	1	按键复位，低有效
gpio_in0	I	1	GPIO0按键中断
gpio_in1	I	1	GPIO1按键中断
LED	O	8	GPIO[7:0]
gpio_out	O	8	GPIO[15:8]
RX	I	1	UART数据接收
TX	O	1	UART数据发送
spi0_clk	O	1	SPI时钟
spi0_cs	O	1	SPI片选
spi0_mosi	O	1	SPI输出
spi0_miso	I	1	SPI输入
i2c0_sck	I/O	1	I2C时钟线
i2c0_sda	I/O	1	I2C数据线
pad_loop_in	I	1	低位温度补偿输入
pad_loop_in_h	I	1	高位温度补偿输入
pad_rstn_ch0	O	1	Memory复位
pad_ddr_clk_w	O	1	Memory差分时钟正端
pad_ddr_clkn_w	O	1	Memory差分时钟负端
pad_csn_ch0	O	1	Memory片选
pad_addr_ch0	O	16	Memory地址总线
pad_dq_ch0	I/O	16	数据总线
pad_dqs_ch0	I/O	2	数据时钟正端
pad_dqsn_ch0	I/O	2	数据时钟负端
pad_dm_rdqs_ch0	O	2	数据 Mask
pad_cke_ch0	O	1	Memory差分时钟使能
pad_odt_ch0	O	1	On Die Termination
pad_rasn_ch0	O	1	行地址strobe
pad_casn_ch0	O	1	列地址strobe.
pad_wen_ch0	O	1	写使能
pad_ba_ch0	O	3	Bank地址总线
pad_loop_out	O	1	低位温度补偿输出
pad_loop_out_h	O	1	高位温度补偿输出
phy_rst_n	O	1	PHY复位
rx_clki	I	1	RGMII接收时钟
phy_rx_dv	I	1	RGMII接收使能

phy_rxd0	I	1	RGMII接收数据0
phy_rxd1	I	1	RGMII接收数据1
phy_rxd2	I	1	RGMII接收数据2
phy_rxd3	I	1	RGMII接收数据3
l0_sgmiiclk_shft	O	1	RGMII发送时钟
phy_tx_en	O	1	RGMII发送使能
phy_txd0	O	1	RGMII发送数据0
phy_txd1	O	1	RGMII发送数据1
phy_txd2	O	1	RGMII发送数据2
phy_txd3	O	1	RGMII发送数据3
Advanced High performance Bus (AHB) — 系统时钟 (即HCLK) 为HCLK时钟			
HTRANS	O	2	传输类型 2'b00: IDLE 2'b01: BUSY 2'b10: NONSE 2'b11: SEQ
HWRITE	O	1	读写控制 1'b1: write 1'b0: read
HADDR	O	32	总线读写地址
HSIZE	O	3	数据位宽 3'b000: 8bit 3'b001: 16bit 3'b010: 32bit
HBURST	O	3	保留
HMASTLOCK	O	1	保留
HWDATA	O	32	总线输出数据
HPROT	O	4	保留
HRDATA	I	32	总线输入数据
HRESP	I	2	系统设置为2'b00, 从机反馈okay
HREADY	I	1	系统设置为1'b1, 总线一直ready
Advanced Peripheral Bus (APB) — 系统时钟 (即HCLK) 为PCLK时钟			
PADDR	O	12	总线读写地址
PWRITE	O	1	读写使能 1'b1: 写 1'b0: 读
PWDATA	O	32	写数据
PENABLE	O	1	使能
PSEL	O	1	片选
PRDATA	I	32	读数据
PREADY	I	1	系统设置为1'b1, 总线一直ready
PSLVERR	I	1	系统设置为1'b0, 总线无错误
Source Data Bus (SDB) — 源数据总线			
DATAI	I	8	源数据

DBUSY	I	1	源数据有效信号
DREADY	O	1	源数据ready信号

2.3. 宏参数定义

Cortex M1软核包含较多宏文件，但CM1_OPTION_DEFS为关键宏文件，涉及系统参数与功能块的使能配置，因此重点介绍，其余宏文件保持默认配置即可。

表 4 功能模块使能宏说明表

CM1_OPTION_DEFS宏文件		
英文参数	中文说明	默认值
CM1_NUM_IRQ	中断数量配置 可配置为1、8、16、32个	16
CM1_OS	操作系统扩展使能配置 1) 0: 不支持操作系统扩展 2) 1: 支持操作系统扩展 注: 使用 SysTick, 配置为 1	1
CM1_SMALL_MUL	乘法器类型选择配置 1) 0: 快速乘法器, 3 周期完成乘法操作 2) 1: 最小面积乘法器, 周期数大于 3	0
CM1_BE8	大小端模式选择配置 1) 0: 小端模式 2) 1: 大端模式	0
CM1_ITCMSIZE	指令存储空间大小配置 1) 4'b0000 0 kB 2) 4'b0001 1 kB 3) 4'b0010 2 kB 4) 4'b0011 4 kB 5) 4'b0100 8 kB 6) 4'b0101 16 kB 7) 4'b0110 32 kB 8) 4'b0111 64 kB 9) 4'b1000 128 kB 10) 4'b1001 256 kB 11) 4'b1010 512 kB 12) 4'b1011 1 MB	4'b0011
CM1_DTCMSIZE	数据存储空间大小配置 1) 4'b0000 0 kB 2) 4'b0001 1 kB 3) 4'b0010 2 kB 4) 4'b0011 4 kB 5) 4'b0100 8 kB 6) 4'b0101 16 kB 7) 4'b0110 32 kB 8) 4'b0111 64 kB 9) 4'b1000 128 kB 10) 4'b1001 256 kB 11) 4'b1010 512 kB 12) 4'b1011 1 MB	4'b0011
CM1_ITCM_INIT	指令存储空间使能配置 1) 0: 关闭 2) 1: 打开	1

CM1_DTCM_INIT	数据存储空间使能配置 1) 0: 关闭 2) 1: 打开	1
CM1_ITCM_LA_EN	低地址段ITCM使能配置 1) 0: 关闭 2) 1: 打开	1
CM1_ITCM_UA_EN	高地址段ITCM使能配置 1) 0: 关闭 2) 1: 打开	0
CM1_SMALL_DEBUG	在线调试功能最小配置 1) 0: 精简版 2) 1: 全功能版	0
CM1_JTAG	JTAG在线调试接口使能配置 1) 0: 关闭 2) 1: 打开	1
CM1_SW	SW在线调试接口使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_DEBUG_ENABLE	在线调试模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_GPIO0	GPIO模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_MEM0	MEM模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_ICACHE	ICACHE模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_DCACHE	DCACHE模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_ETHERNET	ETHERNET 模块使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_TIMER0	TIMER0外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_TIMER1	TIMER1外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_UART0	UART0外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_UART1	UART1外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_WATCHDOG	看门狗外设使能配置 1) 0: 关闭 2) 1: 打开	1

CORTEXM1_APB_SPI0	SPI外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_I2C0	SPI外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_APB_PERIP	用户外设使能配置 1) 0: 关闭 2) 1: 打开	1
CORTEXM1_AHB_UDP	UDO加速模块使能配置 1) 0: 关闭 2) 1: 打开	1
UNCACHE	系统时钟切换配置 1) 注释 : 使用 DDR 时钟 2) 未注释: 使用PLL时钟	注释

2.4. 寻址空间地址映射

表 5 FPGA Cortex-M1标准外设地址映射

标准外设	基地址	描述
ITCM	0x00000000	片上指令存储空间
DTCM	0x20000000	片上数据存储空间
GPIO0	0x40000000	通用输入输出端口
TIMER0	0x50000000	定时器0
TIMER1	0x50001000	定时器1
UART0	0x50004000	串口0
UART1	0x50005000	串口1
Watch Dog	0x50008000	看门狗
I2C0	0x5000A000	SPI外设接口0
SPI0	0x5000B000	I2C外设接口0
MEM0	0x60000000~0x60FFFFFF	第0个Memory
	0x61000000~0x61FFFFFF	第1个Memory
	0x62000000~0x62FFFFFF	第2个Memory
	0x63000000~0x63FFFFFF	第3个Memory
	0x64000000~0x64FFFFFF	第4个Memory
	0x65000000~0x65FFFFFF	第5个Memory
	0x66000000~0x66FFFFFF	第6个Memory
	0x67000000~0x67FFFFFF	第7个Memory
	0x68000000~0x68FFFFFF	第8个Memory
	0x69000000~0x69FFFFFF	第9个Memory
	0x6A000000~0x6AFFFFFF	第10个Memory
	0x6B000000~0x6BFFFFFF	第11个Memory

	0x6C000000~0x6CFFFFFF	第12个Memory
	0x6D000000~0x6DFFFFFF	第13个Memory
	0x6E000000~0x6EFFFFFF	第14个Memory
	0x6F000000~0x6FFFFFFF	第15个Memory
ICACHE	0x10000000~0x1FFFFFFF 0x47000000~0x47FFFFFF	ICACHE指令寻址空间 DDR颗粒映射空间为 0x00000000~0x00FFFFFF
DCACHE	0x30000000~0x3FFFFFFF 0x48000000~0x48FFFFFF	DCACHE指令寻址空间 DDR颗粒映射空间为 0x01000000~0x0FFFFFFF
Ethernet DMAC	0x49000000~0x49FFFFFF	Ethernet DMAC寻址空间
UDP SPEEDUP	0x70000000~0x70000020	UDP SPEEDUP寻址空间
外部寄存器 或RAM	剩余地址为外部寄存器或 RAM所用	写出0x80000000地址处数据 (*(uint32_t*)(0x80000000)) = 0x00000001

2.5. 接口时序

此处仅介绍用户可使用的Cortex M1软核顶层AHB和APB接口时序,其余封装的UART、DDR等接口时序为标准接口,不做赘述。

本设计只涉及 **AHB 基本传输类型与有等待传输类型**,其基本读写时序入图6与图7所示,有等待传输类型时序如图8与图9所示。更为详尽的AHB总线协议介绍请参见《AMBA 3 AHB-Lite Protocol Specification》所示。

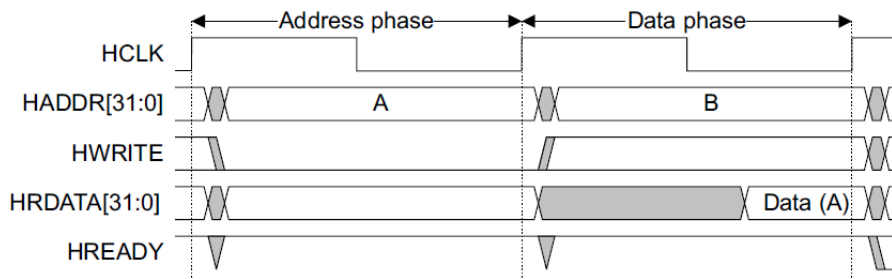


图 6 AHB Read Transfer

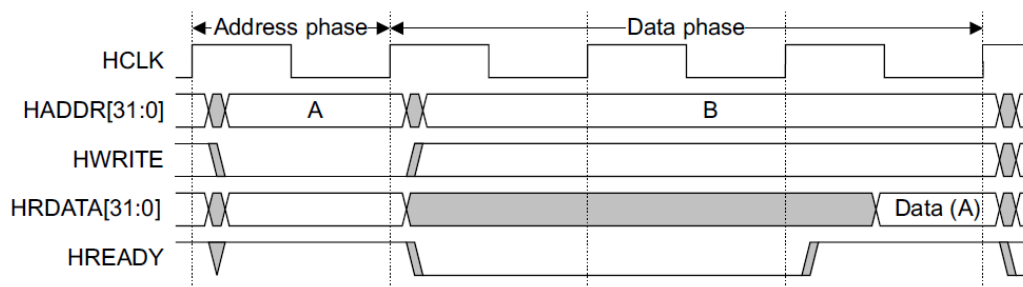


图 7 AHB Read Transfer with Waite States

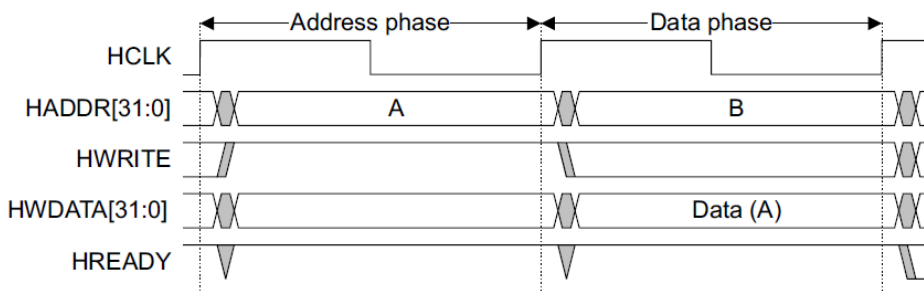


图 8 AHB Write Transfer

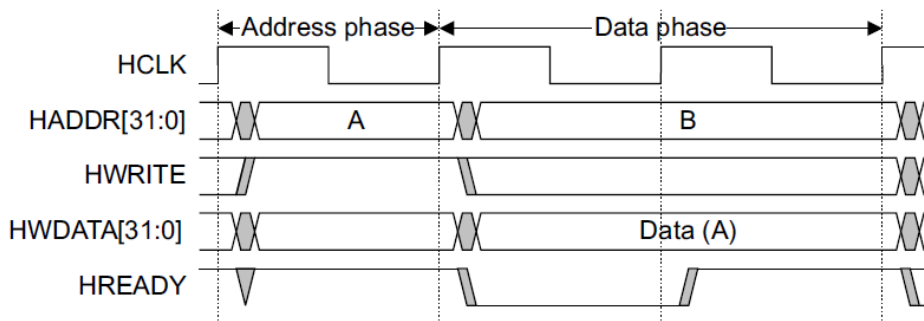


图 9 AHB Write Transfer with Waite States

本设计只涉及 APB 基无等待传输类型，其无等待读写时序入图 10 与图 11 所示。更为详尽的 APB 总线协议介绍请参见《AMBA 3 APB Protocol Specification》所示。

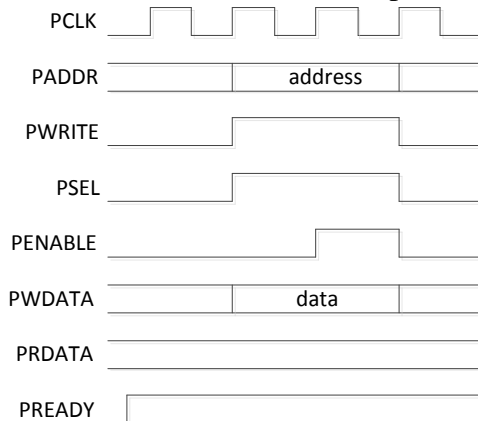


图 10 APB无等待写

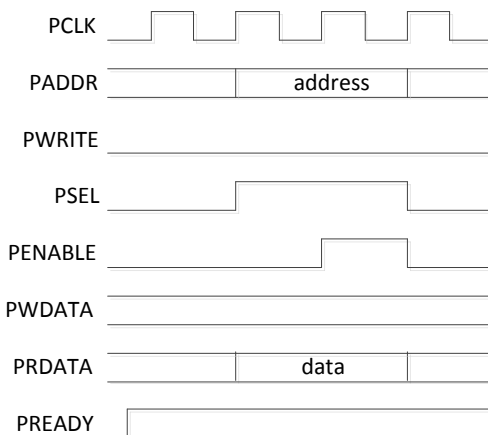


图 11 APB无等待读

3. 参考设计

3.1. 参考功能设计

Cortex M1 SoC设计如图1所示，系统时钟为125Mhz，内置软复位功能以及使用I/DCACHE加载指令/数据方式，将所有外设全覆盖使用，包括Bootloader在线调试功能。其功能模块介绍请参见第二章-Cortex M1 SoC设计架构所述，宏参数配置为默认值，寻址空间如表5所示。

本参考设计分为Bootloader部分和应用程序部分，其中Bootloader放置在片上ITCM中，即上电开始运行，而应用程序部分则通过Bootloader加载至DDR颗粒中，当初始化完成后，Cortex M1软核配合I/DCACHE实现应用程序的执行。

注意，Bootloader程序无需用户编写，用户只需关心应用程序设计即可。通过PDS Configuration插件工具，将应用工程所编译的BIN文件（PGL22平台起始地址为0x000C0000，PGL25平台起始地址为0x00100000，PG2L100H平台起始地址为0x00400000）与sbit数据流文件拼接，并利用同创cable直接下载到Flash芯片中。此后，板卡上电，当FPGA数据流文件加载完毕后，Bootloader将会自动将Flash空间的指令数据搬运至DDR颗粒中，完成指令初始化操作，其数据拼接操作如图12所示。

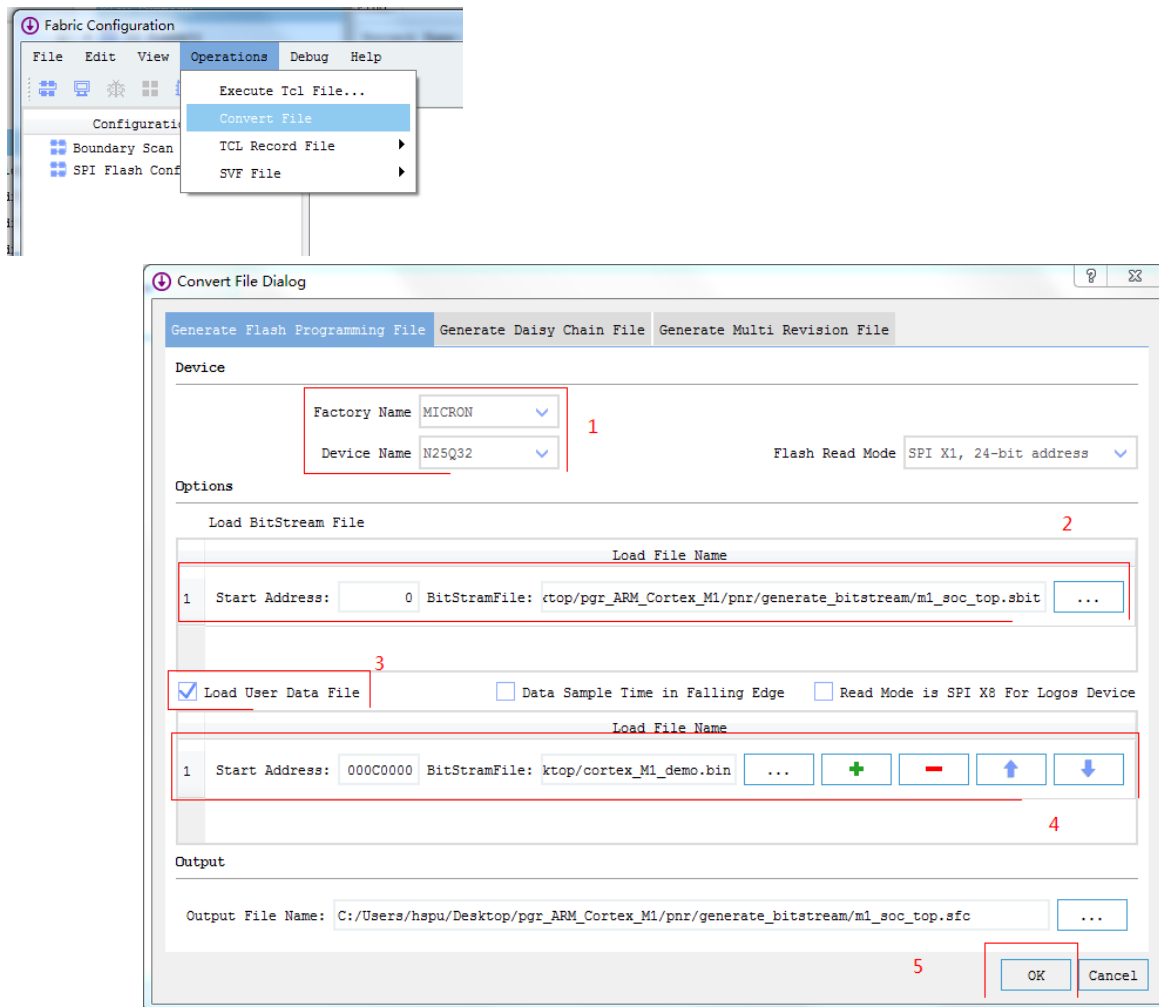


图 12 数据拼接操作

3.2. 参考设计文件目录

pgr_ARM_Cortex_M1设计实例目录结构图：

bench	//仿真test bench
docs	//设计文档-PDS工程使用手册
ip	//设计调用相关IP
pnr	//工程目录
generate_bitstream	// .sbit
ipcore	//IP中用到的IP
source	//fbc文件
ARM_M1_SoC_Top.pds	//工程文件
itcm0	//机器码文件0
itcm1	//机器码文件1
itcm2	//机器码文件2
itcm3	//机器码文件3
simulation	//仿真工程目录
src	//设计实例包含的RTL文件
ahb	//AHB代码
ahb_decoder	//AHB_LITE从设备选择
ahb_def_slave	//AHB_LITE默认从设备
ahb_mux	//AHB_LITE从设备相关接口配置
logic	//外设
cmsdk_ahb_dcache	//DCACHE
cmsdk_ahb_ethernet_dmac	//Ethernet_DMACH
cmsdk_ahb_gpio	//GPIO
cmsdk_ahb_icache	//ICACHE
cmsdk_ahb_to_apb	//AHB转APB桥
cmsdk_apb_i2c	//I2C
cmsdk_apb_mem	//MEM
cmsdk_apb_slave_mux	//APB从设备接口配置
cmsdk_apb_spi	//SPI
cmsdk_apb_subsystem	//APB外设顶层
cmsdk_apb_timer	//TIMER
cmsdk_apb_uart	//UART
cmsdk_apb_watchdog	//看门狗
cmsdk_iop_gpio	//GPIO子模块
m1_core	//Cortex M1相关文件
tsmac_phy	//PANGO TSMAC IP
udp_hw_speedup	//PANGO UDP HW SPEEDUP
M1_soc_top	//Demo顶层
rst_gen	//系统复位

图 13 文件目录

3.3. 参考设计仿真

当进行加密文件仿真时，将需要使用..\pgr_ARM_Cortex_M1_eval\simulation文件夹下自带的itcm0、itcm1、itcm2、itcm3、LRU0.dat、LRU0.dat，以及用户的应用C工程编译生成后的memory.dat、memory_used.dat、address.dat三个文件，并也放置在..\pgr_ARM_Cortex_M1_eval\simulation工程目录下，之后点击sim.bat即可运行ModelSim仿真，实现基于ModelSim的快速仿真。

此外，若PDS版本与手册中不一致，请跟新仿真库至..\pgr_ARM_Cortex_M1_eval\simulation中，操作方法如《FPGA Cortex-M1 SoC快速使用手册》-3.3节所述：

3.4. 参考设计上板验证

当需要上板实测参考设计功能是，需要将Bootloader工程编译生成的4个可执行机器码，即itcm0、itcm1、itcm2、itcm3，与LRU0.dat、LRU0.dat放置在..\pgr_ARM_Cortex_M1_eval\pnr工程目录下，同时将CM1_OPTION_DEFS宏文件中的CM1_ITCM_SIZE、CM1_DTCM_SIZE配置成工程需要的大小，以及打开CM1_ITCM_INIT与CM1_DTCM_INIT。之后，将数据拼接后的sfc文件烧入flash中，SPI-FLASH烧录方法请见《Fabric_Configuration_User_Guide.pdf》。

当上述操作均完成后，板卡上电即可运行C代码所设计应用功能。

C代码功能设计与IDE编译软件使用说明请分别参见《FPGA Cortex-M1软核编程应用指南》与《Cortex-M1软核IDE快速应用指南》。

4. 注意事项

当不需要使用DDR相关设备时，即不使用ICACHE、DCACHE和Ethernet DMAC，按以下步骤配置系统宏参数。

首先，将CM1_OPTION_DEFS宏文件中“`define UNCACHE”取消注释，其作用是把系统时钟从DDR的axi_clk0切换到外部PLL时钟clkout0上。

然后，将CM1_OPTION_DEFS宏文件中的CM1_ITCM_SIZE、CM1_DTCM_SIZE配置成工程需要的大小，打开CM1_ITCM_INIT与CM1_DTCM_INIT，关闭CORTEXM1_AHB_ICACHE、CORTEXM1_AHB_DCACHE和CORTEXM1_AHB_ETHERNET。

接着，将用户C设计编译生成的4个可执行机器码，即itcm0、itcm1、itcm2、itcm3放置在..

最后，利用PDS软件生成数据流.sbit文件并烧入FPGA或SPI-FLASH中，即可上板运行C代码所设计功能。

C代码功能设计与IDE编译软件使用说明请分别参见《FPGA Cortex-M1软核编程应用指南》与《Cortex-M1软核IDE快速应用指南》。