



HME pSRAM IP 用户手册

2026 年 9 月

京微齐力（北京）科技股份有限公司

注意

© 2025-2026 京微齐力（北京）科技股份有限公司
版权所有

未经京微齐力（北京）科技股份有限公司书面许可，不得以任何形式或方式，如电子、机械形式，包括影印、录音或其他数据储存和检索系统形式复制或转移此文档的任何部分，或将其翻译为其它任何语言或计算机语言。

所有商标均为京微齐力（北京）科技股份有限公司所有。

手册版本号

V1.1

联系我们

如果您在使用我们的产品过程中有任何疑问或问题，请与京微齐力（北京）科技股份有限公司或者您当地的代理商联系，或发送邮件至：

sales@hercules-micro.com

环境保护

本产品中包含的某些物质可能会对环境或人体健康有害，为避免将有害物质释放到环境中或危害人体健康，建议采用适当的方法回收本产品，以确保大部分材料可正确地重复使用或回收。有关处理或回收的信息，请与当地权威机构联系。

声明

本手册中包含的信息已经仔细检查并认为是完全可靠的。但是，不对手册中可能或潜在的错误负责。京微齐力（北京）科技股份有限公司保留停止发布或修改手册而不事先通知的权利。为确保获得最新的产品信息，建议用户及时更新手册版本。

本手册介绍的产品并没有被授权用作为生命保障设备或系统中的关键部件。在此使用到的术语有如下定义：

1) 生命保障设备或系统是满足以下条件的设备或系统，(a)被通过手术植入人体内或 (b)用来保障或维持生命，当按照标签上的使用说明正确使用时，有理由认为其工作的中断将会给使用者带来巨大的伤害；

2) 所谓关键部件是指生命保障设备或系统中满足以下条件的部件，即有理由认为该部件中断工作将会导致整个生命保障设备或系统中断工作，或者是影响到后者的安全性和有效性。

版本信息

下表列出了本产品手册的历史版本信息。

修订日期	版本号	修订内容
2025-09-09	V1.0	初版发布。
2026-09-16	V1.1	1. 更新 IP wizard 界面 2. 增加 H7 V213&S196 封装器件支持 3. 优化 internal training，支持 16bit pSRAM

目录

注意	1
版本信息	2
目录	3
图片目录	5
表格目录	6
开始前准备	7
关于本手册	7
缩写	7
寄存器类型说明	7
1 简介	8
1.1 特性	8
1.2 支持器件列表	8
1.3 内部框图	9
2 端口寄存器介绍	11
2.1 端口说明	11
2.2 pSRAM 参数描述	15
2.3 寄存器列表	16
2.4 寄存器定义	17
3 IP 核用法	19
3.1 时钟复位设计	19
3.2 pSRAM 初始化	21
3.3 pSRAM 校准	23
3.3.1 校准的作用和原理	23
3.3.2 写 training	24
3.3.3 读 training	24
3.3.4 PLL 相位调节方法	25
3.3.5 AHB 读写时序	26
3.4 pSRAM 控制时序	26
3.4.1 LOCAL BUS 写操作	26
3.4.2 LOCAL BUS 读操作	27
3.4.3 LOCAL BUS 非完整写操作	28
3.4.4 LOCAL BUS 非完整读操作	28
3.5 读写效率	29
3.5.1 Single latency 模式（不定 latency）	29
3.5.2 Double latency 模式	29
3.6 pSRAM IP 约束方法	30
3.6.1 时钟声明举例如下	30
3.6.2 时钟延迟约束	30
3.6.3 跨时钟域处理	31
3.7 pSRAM IP 的生成	31

3.7.1	IP wizard 参数设计	32
3.8	资源性能报告	34
3.8.1	系统设计框图.....	34
3.8.2	资源报告	34
3.8.3	性能报告	35
3.9	仿真.....	35
3.10	使用问题汇总	37
4	文件结构说明.....	38

图片目录

图 1-1 pSRAM 控制器框图	9
图 3-1 内部 PLL 模式下时钟复位设计	20
图 3-2 外部 PLL 模式下时钟复位设计	20
图 3-3 初始化流程图	21
图 3-4 pSRAM 读时钟校准	23
图 3-5 pSRAM 写时钟校准	24
图 3-6 PLL 动态相位调整时序图	26
图 3-7 AHB 读写时序	26
图 3-8 pSRAM 写操作 burstlen=64 时序图	27
图 3-9 pSRAM 读操作时序图	28
图 3-10 pSRAM 非完整写操作时序图	28
图 3-11 pSRAM 非完整读操作时序图	29
图 3-12 IP wizard 选择 IP 界面	32
图 3-13 pSRAM 例化第一页	32
图 3-14 pSRAM internal training 模式下例化第二页	33
图 3-15 pSRAM external training 模式下例化第二页	33
图 3-16 pSRAM ctrl IP 核测试系统框图	34
图 3-17 仿真配置界面	36
图 3-18 运行仿真界面	36
图 3-19 Modelsim 打印结果	37
图 3-20 仿真波形	37
图 3-21 latency 不一样	37

表格目录

表 2-1 pSRAM 控制器的端口说明	11
表 2-2 参数描述	15
表 2-3 寄存器列表	16
表 2-4 PLL 动态相位控制寄存器 (0x4)	17
表 2-5 training 控制寄存器 (0x8)	17
表 2-6 写数据寄存器 1 (0xc)	17
表 2-7 写地址寄存器 (0x10)	17
表 2-8 pSRAM 状态寄存器 (0x14)	17
表 2-9 写数据寄存器 2 (0x18)	18
表 3-1 CR0 的配置信息	21
表 3-2 CR1 的配置信息	22
表 3-3 动态调节相位专用的 PLL 端口	25

开始前准备

关于本手册

本手册是 pSRAM 控制器的用户手册，它提供技术参考，旨在帮助用户了解器件各功能的参数、特性，及如何使用器件设计应用。

如需了解该系列产品的详细信息，请访问 <http://www.hercules-micro.com>。

如果您在使用过程中有任何疑问或者建议，请发送邮件至 support@hercules-micro.com。

缩写

缩写	全称	中文术语
AHB	Advanced High-performance Bus	先进高性能总线
APB	Advanced Peripheral Bus	先进设备总线
ARM	Advanced RISC Machine	先进精简指令集计算机
EMB	Embedded Memory Block	嵌入式内存模块
FMAX	Max Frequency	最大时钟频率
FPGA	Field Programmable Gate Array	现场可编程门阵列
IP	Intellectual Property	知识产权
LUT	Look Up Table	查找表
MCU	Micro Control Unit	微控制器
OSC	Oscillator	晶振
PLL	Phase-Locked Loop	锁相环
pSRAM	Pseudo Static Random Access Memory	伪静态随机存储器
RAM	Random Access Memory	随机存储器
REG	Register	寄存器
RISC	Reduced Instruction Set Computer	精简指令集
UI	User Interface	用户界面

寄存器类型说明

- ☐ RO: 只读
- ☐ RW: 可读可写
- ☐ RWC: 可读可写，带清除
- ☐ W1C: 写 1 清除
- ☐ WO: 只写

1 简介

pSRAM (Pseudo Static Random Access Memory, 伪静态随机存储器) 是一种结合动态存储单元与静态接口特性的存储器, 主要应用于手机、智能语音设备、GPS接收器等便携式电子产品, 具有低功耗、接口简单及体积小等特点。pSRAM控制器IP用来操作FPGA片外或者片内pSRAM器件, 以达到扩展FPGA存储空间的作用。

1.1 特性

- ❑ 控制器支持 BURST 长度范围1~1024, 支持 wrap burst 和 linear burst 模式。
- ❑ 接口类型:
 - pSRAM 数据接口采用 LOCALBUS 自定义接口
 - pSRAM 初始化采用 AHB 接口
- ❑ 灵活配置 pSRAM 颗粒数量, 支持1~4颗。
- ❑ pSRAM 单颗粒位宽支持8位、16位。
- ❑ 支持每颗 pSRAM 容量最大256Mb。
- ❑ 支持独立工作模式和并接工作模式, 用户可根据实际需求选用不同模式
 - 并接工作模式是指 n 块 pSRAM 做位宽拼接, 同时访问 (目前只支持8位宽的 pSRAM)
 - 独立工作模式是指 n 块 pSRAM 各自独立控制
- ❑ 工作频率最高可到达250MHz。
- ❑ 支持选择 PLL IP 内置或者外置。
- ❑ 支持对 pSRAM 进行初始化 training, 可选用 FPGA 逻辑资源实现还是用硬核 MCU 去实现。
- ❑ 资源使用参考 (单片 pSRAM, 数据8位控制器)
 - 144 LUTS
 - 156 REG

1.2 支持器件列表

FPGA 型号	pSRAM 容量	pSRAM 配置
H7P20N0L176-M2H1	32Mb x 2	Onchip, custom
H7P20N0L176-M1H1	32Mb x 1	Onchip, custom
H7P20N0L128-M3H1	128Mb x 1	Onchip, custom
H7P20N0V213-M1H1	32Mb x 1	Onchip, custom

FPGA 型号	pSRAM 容量	pSRAM 配置
H7P20N0V213-M2H1	32Mb x 2	Onchip, custom
H7P20N0V213-M3H1	128Mb x 1	Onchip, custom
H7P20N0V213-M4H1	128Mb x 2	Onchip, custom
P2P50N0V324-M4H1	256Mb x 4	Onchip, custom
P2L50N0V324-M4H1	256Mb x 4	Onchip, custom
P2P50N0W168-M1H1	256Mb x 1	Onchip, custom
P2L50N0W168-M1H1	256Mb x 1	Onchip, custom
H7P20N0S196-M4H1	128Mb x 2	Onchip, custom
H7P20N0S196-M3H1	128Mb x 1	Onchip, custom
H7/P2 其它系列型号	外接（目前支持华邦 8 位和 16 位 pSRAM 颗粒）	custom

1.3 内部框图

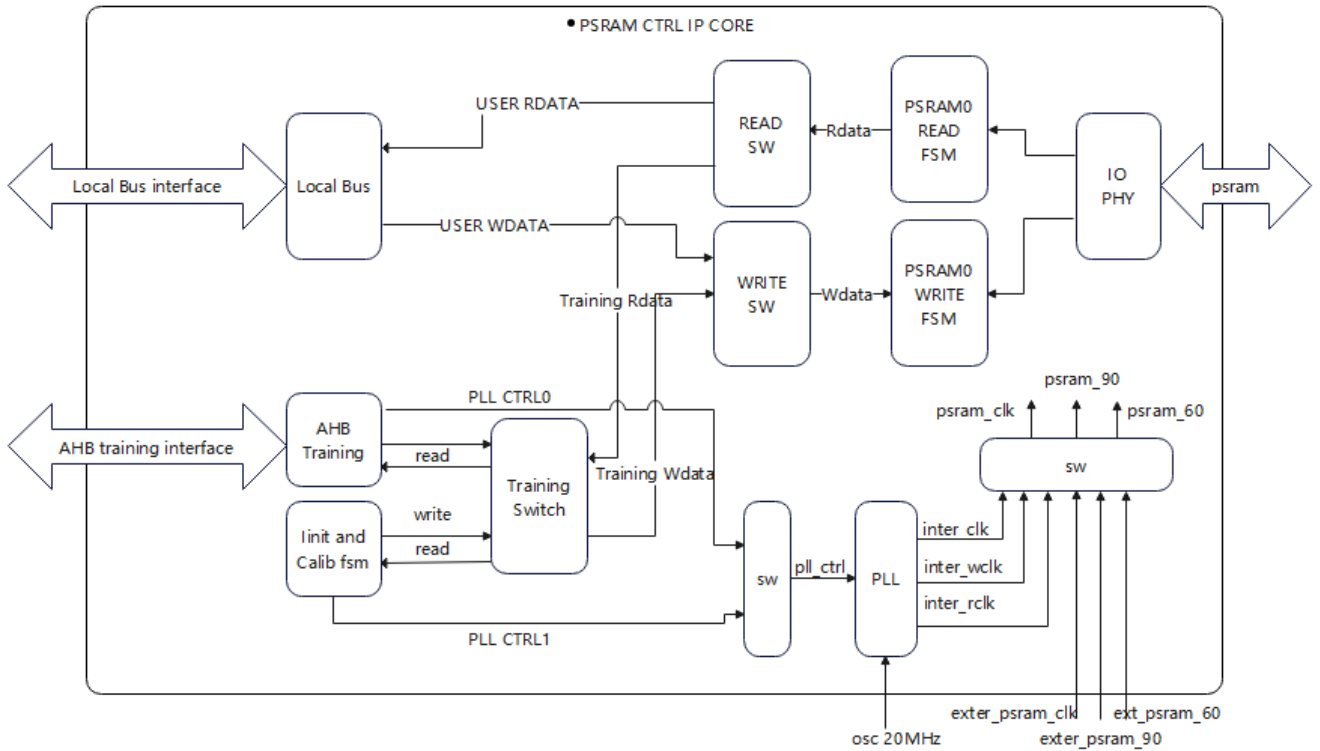


图 1-1 pSRAM 控制器框图

pSRAM 接口：采用 HyperBus 接口，这是一种双边沿传输，低总线数量，高吞吐量的接口。数据线数量根据 pSRAM 型号而定，目前单颗支持 8 位或者 16 位。

Local Bus 总线接口：用户数据接口总线是 Local Bus 总线接口，每颗 pSRAM 占 16 位或 32 位数据位宽，地址总线是 32 位，总线数量和位宽根据用户具体情况而定。

框图中 training 部分可以是 RTL 代码，也可以是预留 training（AHB 外部配置端口）接口，由用户通过 AHB 接到 MCU 上。该接口是标准的 AHB 从机接口，目前只支持 single 操作模式，用于读写控制 pSRAM 初始化和校准相关的逻辑。通过校准来动态调整 PLL 相位，选择最合适时钟相位。

2 端口寄存器介绍

2.1 端口说明

表 2-1 pSRAM 控制器的端口说明

端口名称	类型	位宽	描述
i_clk_sys	input	1	20M 输入，给内部 PLL 使用
i_rstn_in	input	1	系统复位，使用内部 PLL 时使用
o_clk_usrr	output	1	内部 PLL 输出时钟与内部接口总线同步
o_clk_psram	output	1	内部 PLL 输出时钟，与 pSRAM 控制器时钟同步
l_clk_local	input	1	外部 PLL 模式时，作为初始化配置时钟，内部 PLL 模式，不用该时钟
i_rst_xn_psram	input	1	外部 PLL 模式时，复位 pSRAM 控制器和初始化逻辑，低有效，内部 PLL 模式，不用该时钟
init_done	output	1	pSRAM 初始化完成
AHB 外部配置端口（从机）			
train_hrdata_s	output	32	外部初始化 AHB 接口
train_hsel_s	input	1	外部初始化 AHB 接口
train_hwrite_s	input	1	外部初始化 AHB 接口
train_hwdata_s	input	32	外部初始化 AHB 接口
train_haddr_s	input	32	外部初始化 AHB 接口
train_htrans_s	input	2	外部初始化 AHB 接口
PLL 相位控制接口			
ps_ck	output	1	外部 PLL 移相控制接口
ps_dir	output	1	外部 PLL 移相控制接口
ps_en	output	1	外部 PLL 移相控制接口
ps_sel	output	3	外部 PLL 移相控制接口
ps_done	input	1	外部 PLL 移相控制接口

端口名称	类型	位宽	描述
LOCAL BUS0 接口			
psram_rdata0	output	16/32	local 总线 pSRAM 读数据接口
wr_req_ack0	output	1	local 总线 pSRAM 写数据应答
rd_req_ack0	output	1	local 总线 pSRAM 读数据应答
psram_wr_reqin0	input	1	local 总线 pSRAM 写请求
wr_data0	input	16/32	local 总线 pSRAM 写数据接口
data_mask0	input	2	local 总线 pSRAM 写数据掩码
psram_addr_set0	input	32	local 总线 pSRAM 读写地址设置
psram_rd_reqin0	input	1	local 总线 pSRAM 读请求
wrburst_end0	input	1	local 总线 pSRAM 写结束
rdburst_end0	input	1	local 总线 pSRAM 读结束
wrburst_len0	input	10	local 总线 pSRAM 写突发长度
rdburst_len0	input	10	local 总线 pSRAM 读突发长度
linear_en0	input	1	local 总线 burst 模式设置, 0 代表 wrapburst, 1 代表 linearburst
psram_finish0	output	1	local 总线 pSRAM 操作完成信号
LOCAL BUS1 接口			
psram_rdata1	output	16/32	local 总线 pSRAM 读数据接口
wr_req_ack1	output	1	local 总线 pSRAM 写数据应答
rd_req_ack1	output	1	local 总线 pSRAM 读数据应答
psram_wr_reqin1	input	1	local 总线 pSRAM 写请求
wr_data1	input	16/32	local 总线 pSRAM 写数据接口
data_mask1	input	2	local 总线 pSRAM 写数据掩码
psram_addr_set1	input	32	local 总线 pSRAM 读写地址设置
psram_rd_reqin1	input	1	local 总线 pSRAM 读请求

端口名称	类型	位宽	描述
wrburst_end1	input	1	local 总线 pSRAM 写结束
rdburst_end1	input	1	local 总线 pSRAM 读结束
wrburst_len1	input	10	local 总线 pSRAM 写突发长度
rdburst_len1	input	10	local 总线 pSRAM 读突发长度
linear_en1	input	1	local 总线 burst 模式设置
psram_finish1	output	1	local 总线 pSRAM 操作完成信号
LOCAL BUS2 接口			
psram_rdata2	output	16/32	local 总线 pSRAM 读数据接口
wr_req_ack2	output	1	local 总线 pSRAM 写数据应答
rd_req_ack2	output	1	local 总线 pSRAM 读数据应答
psram_wr_reqin2	input	1	local 总线 pSRAM 写请求
wr_data2	input	16/32	local 总线 pSRAM 写数据接口
data_mask2	input	2	local 总线 pSRAM 写数据掩码
psram_addr_set2	input	32	local 总线 pSRAM 读写地址设置
psram_rd_reqin2	input	1	local 总线 pSRAM 读请求
wrburst_end2	input	1	local 总线 pSRAM 写结束
rdburst_end2	input	1	local 总线 pSRAM 读结束
wrburst_len2	input	10	local 总线 pSRAM 写突发长度
rdburst_len2	input	10	local 总线 pSRAM 读突发长度
linear_en2	input	1	local 总线 burst 模式设置
psram_finish2	output	1	local 总线 pSRAM 操作完成信号
LOCAL BUS3 接口			
psram_rdata3	output	16/32	local 总线 pSRAM 读数据接口
wr_req_ack3	output	1	local 总线 pSRAM 写数据应答

端口名称	类型	位宽	描述
rd_req_ack3	output	1	local 总线 pSRAM 读数据应答
psram_wr_reqin3	input	1	local 总线 pSRAM 写请求
wr_data3	input	16/32	local 总线 pSRAM 写数据接口
data_mask3	input	2	local 总线 pSRAM 写数据掩码
psram_addr_set3	input	32	local 总线 pSRAM 读写地址设置
psram_rd_reqin3	input	1	local 总线 pSRAM 读请求
wrburst_end3	input	1	local 总线 pSRAM 写结束
rdburst_end3	input	1	local 总线 pSRAM 读结束
wrburst_len3	input	10	local 总线 pSRAM 写突发长度
rdburst_len3	input	10	local 总线 pSRAM 读突发长度
linear_en3	input	1	local 总线 burst 模式设置
psram_finish3	output	1	local 总线 pSRAM 操作完成信号
pSRAM0 接口			
P_RSTN0	output	1	pSRAM IO
CK0	output	1	pSRAM IO
CK_N0	output	1	pSRAM IO
CS_N0	output	1	pSRAM IO
RWDS0	inout	1/2	pSRAM IO
DQ_RAM0	inout	8/16	pSRAM IO
pSRAM1 接口			
P_RSTN1	output	1	pSRAM IO
CK1	output	1	pSRAM IO
CK_N1	output	1	pSRAM IO
CS_N1	output	1	pSRAM IO

端口名称	类型	位宽	描述
RWDS1	inout	1/2	pSRAM IO
DQ_RAM1	inout	8/16	pSRAM IO
pSRAM2 接口			
P_RSTN2	output	1	pSRAM IO
CK2	output	1	pSRAM IO
CK_N2	output	1	pSRAM IO
CS_N2	output	1	pSRAM IO
RWDS2	inout	1/2	pSRAM IO
DQ_RAM2	inout	8/16	pSRAM IO
pSRAM3 接口			
P_RSTN3	output	1	pSRAM IO
CK3	output	1	pSRAM IO
CK_N3	output	1	pSRAM IO
CS_N3	output	1	pSRAM IO
RWDS3	inout	1/2	pSRAM IO
DQ_RAM3	inout	8/16	pSRAM IO

2.2 pSRAM 参数描述

表 2-2 参数描述

参数	类型	范围	默认值	描述
RD_CH_NUM	Binary	3'b000~3'b111	3'b011	读时钟对应 PLL 的通道
WR_CH_NUM	Binary	3'b000~3'b111	3'b010	写时钟对应 PLL 的通道
HYBRID_EN	Binary	0,1	0	使能 hybrid burst 功能 0: 使能; 1: 禁止
DRV_STR	Binary	000~111	3'b000	pSRAM 输出驱动强度配置

参数	类型	范围	默认值	描述
DQ_WIDTH	Integer	8,16,32	32	对应 pSRAM 的 DQ 数量
AD_WIDTH	Integer	1~27	24	对应 pSRAM 的地址位宽
TR_MODE	String	"internal" "external"	"internal"	0: 选择内部逻辑实现初始化 1: 选择外部 AHB_train 接口实现初始化
Fixed_Latency	Binary	0,1	1	1: 固定延迟模式 0: 可变延迟模式
Initial_Latency	Integer	6,7	7	pSRAM 的刷新延迟周期数
PASR	Binary		3'b000	pSRAM 刷新区域设置
EXT_PLL	Binary	0,1	0	1: 使用外部 PLL 生成的 pSRAM 时钟 0: 使用内部 PLL 生成的时钟
pSRAM_IDX	Integer	0,1,2,3	0	当例化单个 pSRAM 时, 选择 pSRAM 编号。其它情况填 0
CLK_FREQ	Integer		200	pSRAM 的时钟主频单位 MHz
C_MAX_BURST	Integer	1~64	16	最大突发长度
C_pSRAM_NUM	Integer	1,2,4	2	挂载 pSRAM 数量
BUS_TYPE	String	"LOCAL"	"LOCAL"	选择用户接口类型
C_DATA_WIDTH	Integer	16,32,64,128	32	总线接口宽度, 注意 AHB 接口只能设置为 32
C_ADDR_WIDTH	Integer	1~32	32	总线地址位宽
CLIB_EN	Binary	0,1	0	1: 使能校准功能; 0: 关闭校准功能

2.3 寄存器列表

表 2-3 寄存器列表

偏移地址	training 寄存器	读写	缺省值	描述
0x0	func_test	NC	0	测试寄存器
0x4	pll_set_reg	RW	0	PLL 动态相位控制寄存器
0x8	mcu_ctrl	RW	0	training 控制寄存器
0xc	mcu_wdata1	WO	0	写数据低 32 位
0x10	mcu_addr	WO	0	写地址

偏移地址	training 寄存器	读写	缺省值	描述
0x14	pSRAM_status	RO	0	pSRAM 状态寄存器
0x18	mcu_wdata2	WO	0	写数据高 32 位

2.4 寄存器定义

表 2-4 PLL 动态相位控制寄存器 (0x4)

位	功能	读写	缺省值	描述
[7]	相移使能	RW	0	拉高表示使能相移
[6]	相移方向	RW	0	0: 反方向移相 1: 正方向移相
[2:0]	PLL 输出通道选择	RW	0	对应 PLL 的输出时钟通道 0~7

表 2-5 training 控制寄存器 (0x8)

位	功能	读写	缺省值	描述
[12:11]	写数据掩码	RW	0	1: 表示此次操作对应的 byte 无效。
[8]	设置 init_done 标志	RW	0	1: pSRAM 初始化完成。
[7]	清除 latency 寄存器状态	RW	0	写 1 清除 latency 寄存器状态, 需要再写回 0。
[5]	pSRAM 复位控制	RW	0	复位需要先写 0, 再写 1。
[4]	pSRAM 寄存器模式设置	RW	0	1: 寄存器读写模式。 0: 数据读写模式。
[1]	training 读请求	RW	0	写 1 发起读请求, 需要再写回 0
[0]	training 写请求	RW	0	写 1 发起写请求, 需要再写回 0

表 2-6 写数据寄存器 1 (0xc)

位	功能	读写	缺省值	描述
[31:0]	写数据寄存器	WO	0	[15:0]写入 pSRAM0 的写数据。 [31:16]写入 pSRAM1 的写数据。

表 2-7 写地址寄存器 (0x10)

位	功能	读写	缺省值	描述
[31:0]	写地址寄存器	WO	0	[31:0]写入 pSRAM 的地址命令。

表 2-8 pSRAM 状态寄存器 (0x14)

位	功能	读写	缺省值	描述
[8]	pSRAM 控制状态	RO	0	表示 PLL 是否移相完成, 1 表示完成。
[4:1]	pSRAM latency 状态	RO	0	1: 表示读到的 pSRAM 正常操作时需要额外的延迟。 4:1 分别对应 pSRAM3~0
[0]	pSRAM 是否完成操作	RO	0	1: 表示 pSRAM 已完成操作。

位	功能	读写	缺省值	描述
				0: 标志操作进行中。

表 2-9 写数据寄存器 2 (0x18)

位	功能	读写	缺省值	描述
[31:0]	写数据寄存器 2	WO	0	[15:0]写入 pSRAM2 的数据。 [31:16]写入 pSRAM3 的数据。

3 IP 核用法

pSRAM 控制器 IP 核包含了以下几个功能模块：

- ☐ pSRAM init and calib
- ☐ Clk_rst(PLL)
- ☐ Local bus interface
- ☐ Ahb training interface
- ☐ pSRAM controller
- ☐ pSRAM phy

3.1 时钟复位设计

pSRAM 设置为外部 PLL 模式时：时钟一共有3个外部输入时钟 `i_clk_pSRAM`、`i_clk_pSRAM_90`和 `i_clk_pSRAM_60`，这3个时钟来自同一个时钟经过 PLL 产生，其相位初始关系分别为0度，90度，60度（经过 training 之后这个相移会有轻微变化）。

pSRAM 设置为内部 PLL 模式时：需要有 `i_clk_sys`（20MHz）作为内部 PLL 的输入源，内部生成 pSRAM 所需要全部时钟，并且 `o_clk_usr`（100MHz），`o_clk_pSRAM`（200MHz）会向外输出时钟，分别给 training 逻辑和 pSRAM 读写逻辑使用。其中 `o_clk_pSRAM` 与 pSRAM 控制器同步。

对设计中的 PLL 需要支持动态相位调节，training 过程中对 `i_clk_pSRAM_90`和 `i_clk_pSRAM_60`作调整。

- ☐ `i_rstn_in`：外部 PLL 模式时复位内部初始化逻辑（慢时钟域 `i_clk_local`），内部 PLL 时，作为 pSRAM 控制器系统全局复位。
- ☐ `i_clk_sys`：外部 PLL 模式作为内部初始化时钟（建议主频为 $F_{i_clk_pSRAM}/2$ ），内部 PLL 时，作为 PLL 的输入时钟源 20MHz。
- ☐ `i_rst_xn_pSRAM`：复位 pSRAM 控制器相关逻辑（非初始化部分），内部 PLL 模式时不接。

以下为复位设计示例：

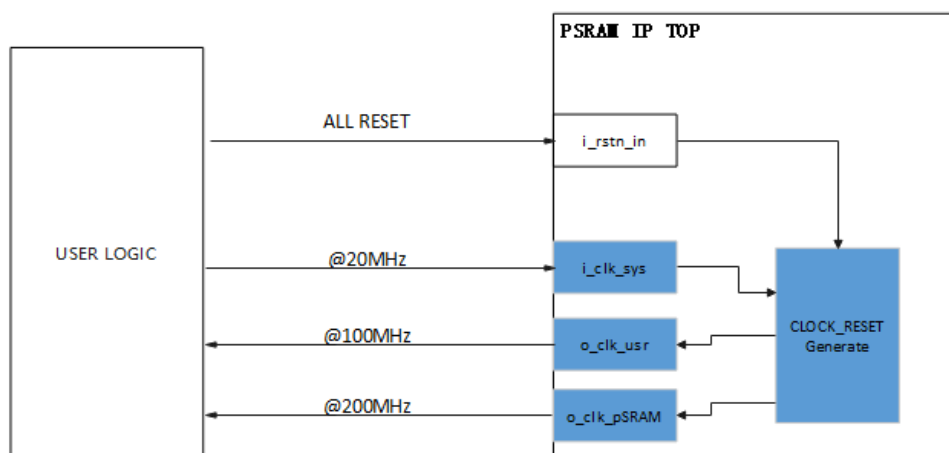


图 3-1 内部 PLL 模式下时钟复位设计

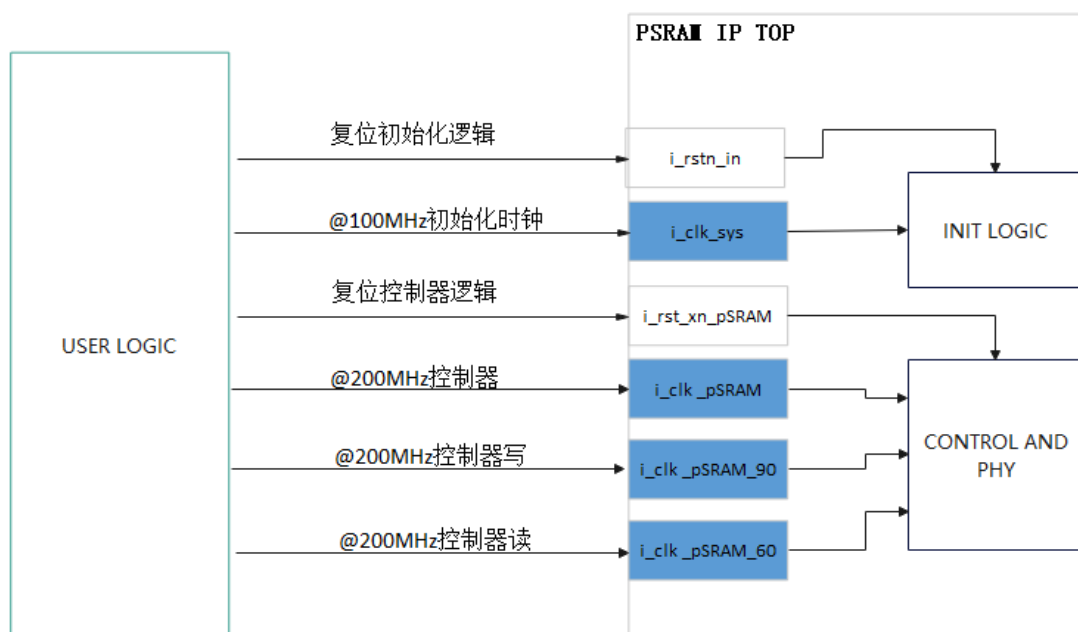


图 3-2 外部 PLL 模式下时钟复位设计

3.2 pSRAM 初始化

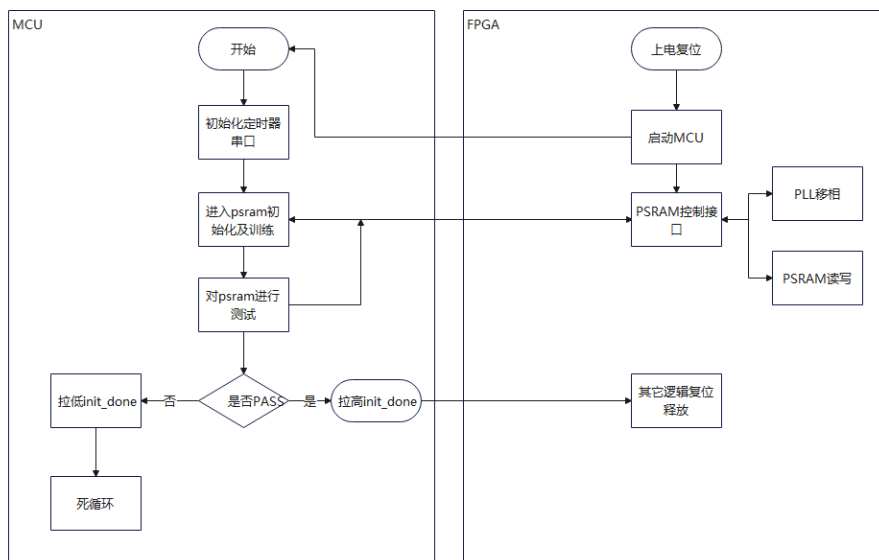


图 3-3 初始化流程图

pSRAM 的寄存器参数是由 ARM 使用{pSRAM_init()}函数或者 FPGA 逻辑进行设置的。由于 FPGA 逻辑是自动初始化和校准的，跟用户操作无关，以下将重点介绍 MCU 配置部分。

以 P2P50N0V324-M4H1 为例，第一个参数为 pSRAM Configuration Register 0（以下简称 CR0）的配置值，推荐值为 0x8f24 或 0x8f2c。第二个参数为 pSRAM Configuration Register 1（以下简称 CR1）的配置值，推荐值为 0xff80。用户可根据实际需求进行配置。

pSRAM 初始化完成后，可以运行函数 pSRAM_ctrl_mst_set(1)，拉高 pSRAM_ctrl bit8；此时，pSRAM 控制器会进入 User Mode，之后由 FPGA 接口进行读写操作。

下面列出了 W958D8NBYA 器件的配置寄存器信息作为参考。

表 3-1 CR0 的配置信息

CR0 位	功能	设置（二进制）
[15]	深度省电使能	0b - CR0[15]写入 0，器件进入深度省电（DPD） 1b - 正常运行（默认） 注意：退出深度省电后，pSRAM 会自动将 CR0[15]设置为 1
[14:12]	驱动强度	000b - 34 Ω（默认） 001b - 115 Ω 010b - 67 Ω 011b - 46 Ω 100b - 34 Ω

CR0 位	功能	设置（二进制）
		101b - 27 Ω 110b - 22 Ω 111b - 19 Ω
[11:8]	保留	1b - 保留（默认），保留供以后使用 写入 CR0 时，这几位应被设为 1 用于未来的兼容性
[7:4]	初始延迟	0000b - 5 个时钟延迟 @ 133MHz 最大频率 0001b - 6 个时钟延迟 @ 166MHz 最大频率 0010b - 7 个时钟延迟 @ 200MHz 最大频率（默认） 0010b - 7 个时钟延迟 @ 250MHz 最大频率 ... 1101b - 保留 1110b - 3 个时钟延迟 @ 85MHz 最大频率 1111b - 4 个时钟延迟 @ 104MHz 最大频率
[3]	固定延迟使能	0b - 可变初始延迟 - 1 倍或 2 倍初始延迟，在 CA 周期中根据 RWDS 而变化 1b - 固定 2 倍初始延迟（默认）
[2]	混合突发使能	0b: 使用混合方式包装突发序列 1b: 使用传统方式包装突发序列（默认）
[1:0]	突发长度	00b - 128 字节 01b - 64 字节 10b - 16 字节 11b - 32 字节（默认）

表 3-2 CR1 的配置信息

CR1 位	功能	设置（二进制）
[15:8]	保留	FFh - 保留（默认），保留供以后使用 写入 CR1 时，这几位应保持 FFh 用于未来的兼容性
[7]	保留	1b - 保留（默认） 写入 CR1 时，这一位应保持 1b
[6]	主时钟类型	0b - 差分 - CK#, CK 1b - 单端 - CK（默认）

CR1 位	功能	设置（二进制）
[5]	混合睡眠	0b - 正常运行（默认） 1b - 在CR1[5]中写1，启动器件的混合睡眠（HS）状态
[4:2]	部分阵列刷新	000b - 全阵列（默认） 001b - 底部1/2阵列 010b - 底部1/4阵列 011b - 底部1/8阵列 100b - 保留 101b - 顶部1/2阵列 110b - 顶部1/4阵列 111b - 顶部1/8阵列
[1:0] ^{*1}	分布式刷新闻隔	00b - 保留 01b - 4 μ s (t _{CSM}) 10b - 保留 11b - 保留

注意 1 CR1 的[1:0]位只读。

3.3 pSRAM 校准

3.3.1 校准的作用和原理

pSRAM 控制器运行共需要三个时钟，即 psram_clk、psram_clk_90 和 psram_clk_60。

在实际使用中，psram_clk_90 负责发送时钟，时钟保持跟数据有 90 度的相位差，以便接收端能够正确采样数据。然而真实情况下由于布局布线的或者颗粒本身的差异导致时钟跟数据并不是正好的 90 度的相位差，所以需要用 PLL 去对 psram_clk_90 去移相来找到合适的窗口中心点。

psram_clk_60 负责采样读进入 IO 的数据，与发送一样，这个数据的延迟也是各有差异，所以需要用 PLL 去对 psram_clk_60 去移相来找到合适的窗口中心点。

ARM 调用{psram_training_new();}函数进行 training，过程中调用函数{pll_adjust();}，从而产生控制信号 ps_ck、ps_dir、ps_en 和 pssel。这些信号共同对 PLL 的时钟输出进行相位调节，最终找到合适的时钟窗口。

FPGA 需要使用 u_apb2_fp_slave 模块连接 ARM 和 pSRAM 控制器。

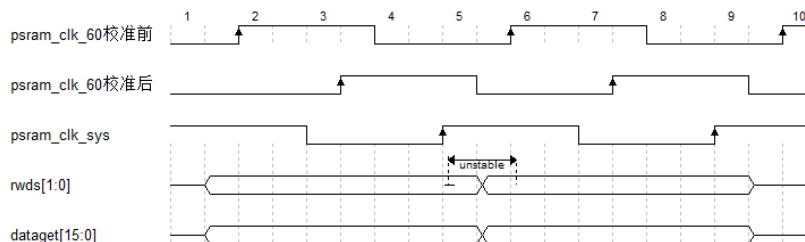


图 3-4 pSRAM 读时钟校准

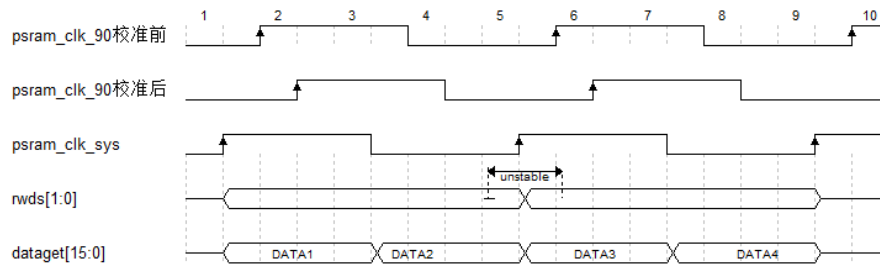


图 3-5 pSRAM 写时钟校准

3.3.2 写 training

ARM 调用{psram_wr_training_new();}函数，通过对 pSRAM 进行写操作来对 pSRAM 的 IO 时钟（即 psram_clk_90）调相。

- (1) 调用{psram_wr_training_basic_forward();}函数，确定 pSRAM 的 IO 时钟相位窗口的起始位置。在函数中对 pSRAM 进行两次数据写入操作，然后读状态寄存器进行判断。

如果两次状态寄存器的值都为 1，说明目前时钟的相位不对，需要调用函数{pll_adjust();}进行调相。返回进行下一次写入操作，继续读取状态寄存器，直到能够读出状态寄存器的值为 0，并且保持 88 次，函数返回调相次数 clk_phase_step。

- (2) 调用{psram_wr_window_scan_forward();}函数，确定 pSRAM 的 IO 时钟相位窗口的大小。在函数中对 pSRAM 进行两次数据写入操作，然后读状态寄存器进行判断。

如果两次状态寄存器的值中有一次为 0，那么窗口大小+1。返回进行下一次写入操作，继续读取状态寄存器，直到读出两次状态寄存器的值都为 1，跳出循环，返回窗口大小 ram_rdy。

- (3) 调用{pll_adjust();}函数，将 pSRAM 的 IO 时钟相位调节到窗口的中心位置，返回中心位置变量 clk_phase_step。

3.3.3 读 training

调用{psram_rd_cmd_training_new();}函数，通过对 pSRAM 初始化寄存器进行写入读取操作，从而对 pSRAM 控制器读操作时钟（即 psram_clk_60）调相。

- (1) 调用{psram_rd_cmd_training_basic_new();}函数，确定 pSRAM 控制器读操作时钟相位窗口的起始位置。在函数中对 CR0 进行写操作，然后读出，判断读出的值是否正确。

如果不正确，说明目前时钟的相位不对，需要调用函数{pll_adjust();}进行调相。返回进行下一次写操作，然后读出，判断读出的值是否正确。

如果正确，并且保持正确 33 次，函数返回调相次数 rd_phase_step。

- (2) 调用{psram_rd_window_scan_new();}函数，确定 pSRAM 控制器读操作时钟相位窗口的大小。在函数中对 CR0 进行写操作，然后读出，判断读出的值是否正确。

如果正确，那么窗口大小+1。返回进行下一次写操作，然后读出，直到读取值不正确，跳出循环，返回窗口大小 ram_rdy。

- (3) 调用函数{pll_adjust();}，将 pSRAM 控制器的读操作时钟相位调节到窗口的中心位置，返回中心位置变量 rd_phase_step。

3.3.4 PLL 相位调节方法

HME-P2 和 H7 系列芯片中的 PLL 都具有动态相位调节功能。

{pll_adjust();}函数使用如表 3-3 所示的 PLL 专用动态调节相位的端口，对 pSRAM IO 时钟和 pSRAM 控制器的读操作时钟进行动态相位调节，每调节一步，相位增加或减少 $T_{vco}/64$ 。动态相位调节时的时序图如图 3-6 所示。

表 3-3 动态调节相位专用的 PLL 端口

名称	IO	位宽	功能描述
ps_ck	input	1	动态调相时，给 PLL 用于接收相关信号的时钟
ps_dir	input	1	控制调相的方向 0: 往前（左）调 1: 往后（右）调
ps_en	input	1	动态调相使能信号，检测到下降沿进行一次动态调相
ps_done	output	1	PLL 给出的调相完成信号
ps_sel	input	3	选择需要调相的时钟通道 000: 不选择 001: 选择通道 0 010: 选择通道 1 011: 选择通道 2 100: 选择通道 3 101: 选择通道 4 110: 选择通道 5 111: 选择通道 6

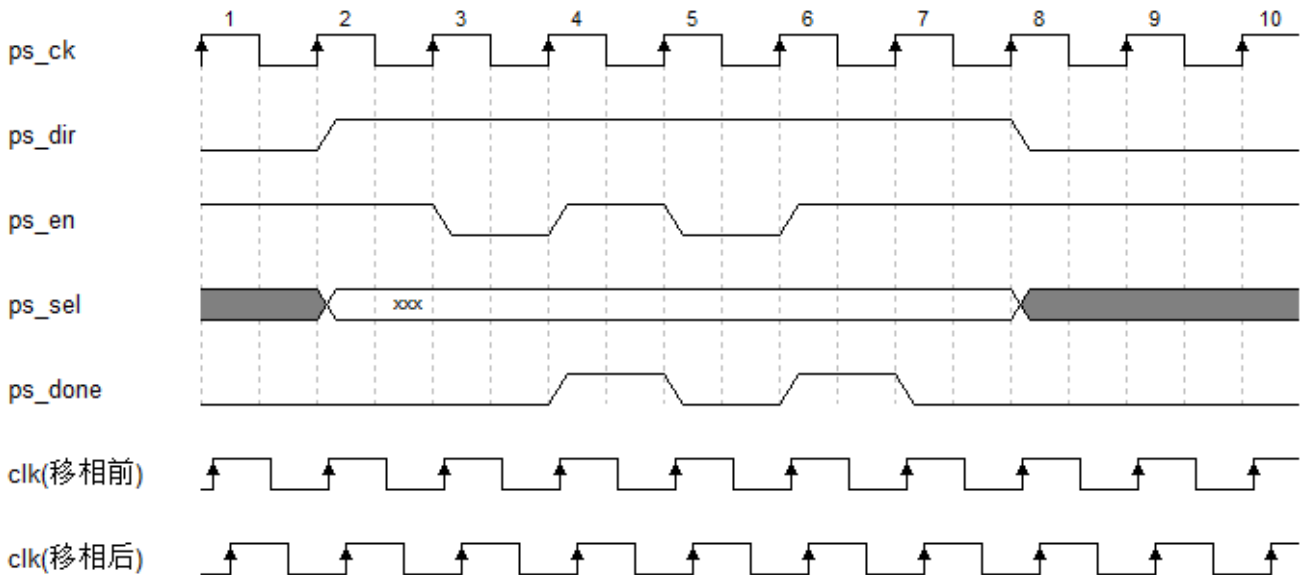


图 3-6 PLL 动态相位调整时序图

3.3.5 AHB 读写时序

AHB 接口不支持 burst 操作，只支持简单的 AHB 读写操作。

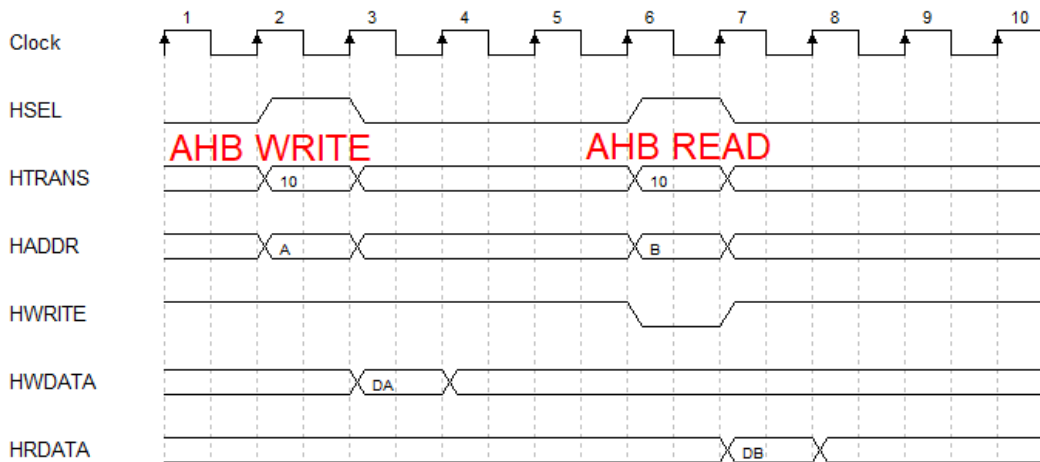


图 3-7 AHB 读写时序

3.4 pSRAM 控制时序

HME-P2 和 H7 系列的 pSRAM 控制器目前最高支持突发长度为 128 bytes 的完整读、写操作和非完整的写操作。

3.4.1 LOCAL BUS 写操作

pSRAM_wr_reqin 是输入 pSRAM 控制器的写请求信号，上升沿有效；pSRAM_addr_set 信号用于设置本次写操作的起始地址。

- (1) 完成 pSRAM_addr_set 的设置；

- (2) 控制器接收到 pSRAM_wr_reqin 的上升沿，开始一次写操作；
 - (3) pSRAM 控制器发送给 pSRAM 总线指令和地址，发送完成后 wr_req_ack 信号将被拉高；
 - (4) pSRAM 控制器将在每一个时钟上升沿发送一个 16 bits 数据 wrdata，并通过总线发送到 pSRAM。
- 一次完整的写操作共需采样 64 个这样的数据，完成后 pSRAM 控制器将返回初始状态。

完整写操作的时序图如图 3-8 所示。

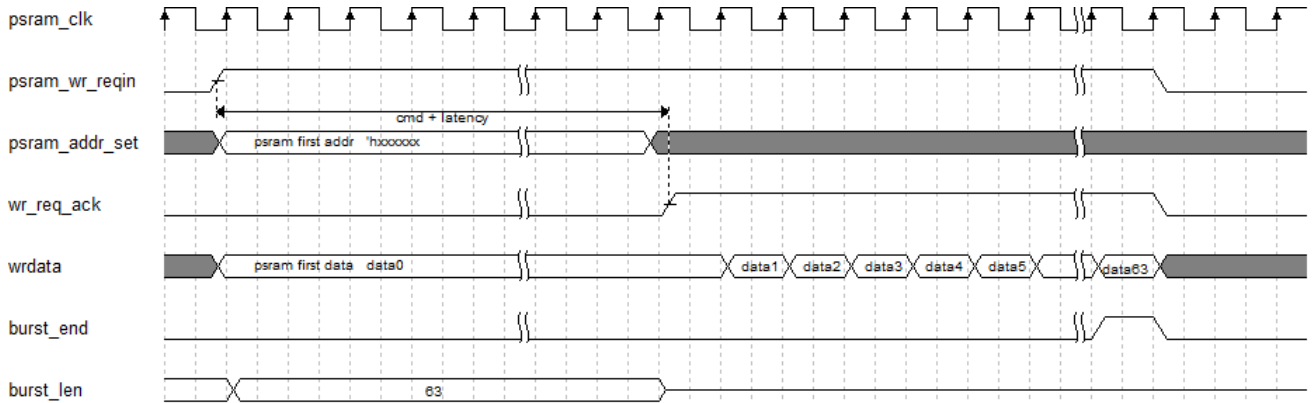


图 3-8 pSRAM 写操作 burstlen=64 时序图

3.4.2 LOCAL BUS 读操作

pSRAM_rd_reqin 是输入 pSRAM 控制器的读请求信号，pSRAM_addr_set 信号用于设置本次读操作的起始地址。

- (1) 完成 pSRAM_addr_set 的设置；
- (2) 控制器接收到 pSRAM_rd_reqin 的上升沿，开始一次读操作；
- (3) pSRAM 控制器发送给 pSRAM 总线指令和地址，发送完成后 rd_req_ack 信号将被拉高；
- (4) pSRAM 控制器会将 pSRAM 中读到的双沿 8 bits 的数据转换为 16 bits 的单沿数据，在每一个时钟上升沿输出一个到用户设置的出口。

一次完整的读操作共有 64 个 16 bits 数据，完成后 pSRAM 控制器将返回初始状态。

完整读操作的时序图如图 3-9 所示。

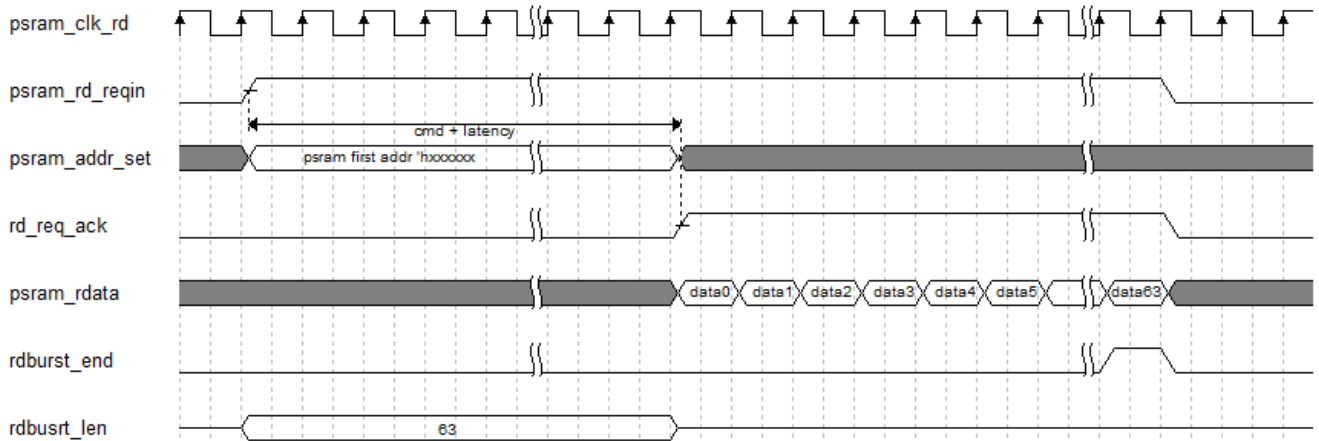


图 3-9 pSRAM 读操作时序图

3.4.3 LOCAL BUS 非完整写操作

pSRAM 控制器可以在写操作过程中将其提前终止，如图 3-10 所示。

在写入第 4 个 16 bits 数据时将 burst_end 信号拉高一个时钟周期，控制器接收到该信号后会直接终止这次写操作。

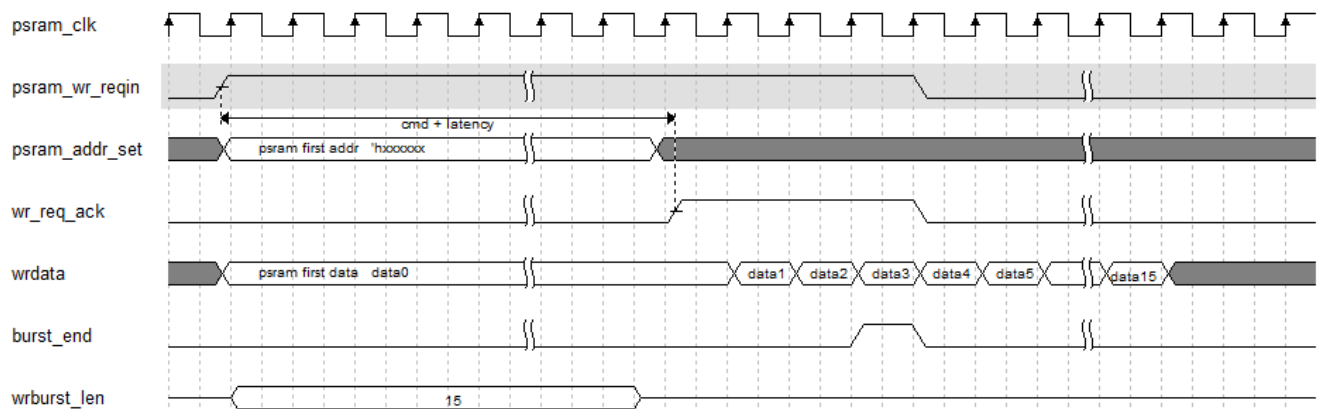


图 3-10 pSRAM 非完整写操作时序图

3.4.4 LOCAL BUS 非完整读操作

pSRAM 控制器可以在读操作过程中将其提前终止，如图 3-11 所示。

在读第 4 个 16bits 数据时将 burst_end 信号拉高一个时钟周期，控制器接收到该信号后会直接终止这次读操作。

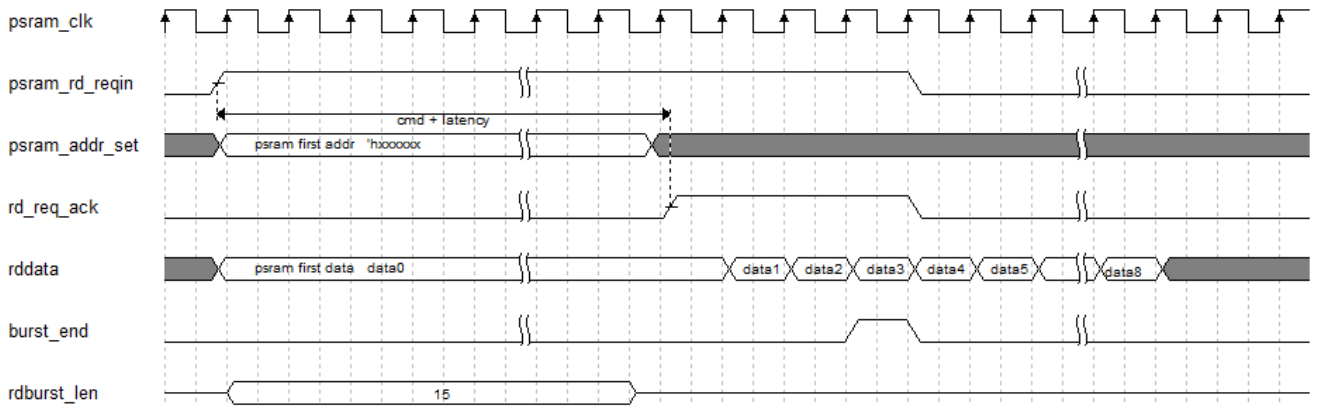


图 3-11 pSRAM 非完整读操作时序图

3.5 读写效率

3.5.1 Single latency 模式（不定 latency）

pSRAM CR0 配置值为 0x8f24。

写操作的计算：连续拉高 pSRAM_wr_reqin 信号，总周期为两次连续的 ack 上升沿之间的周期数，一共需要 78 个时钟周期，其中，有效数据占 64 个时钟周期。

读操作的计算：连续拉高 pSRAM_rd_reqin 信号，总周期为两次连续的 ack 上升沿之间的周期数，一共需要 79 个时钟周期，其中，有效数据占 64 个时钟周期。

在 200MHz 主频，128 burst 长度下理论写效率

$$64 / (7 + 7 + 64) = 82.05\%$$

在 200MHz 主频，128 burst 长度下理论读效率

$$64 / (7 + 7 + 64 + 1) = 81.01\%$$

在 166MHz 主频，128 burst 长度下理论写效率

$$64 / (6 + 6 + 64)ns = 84.21\%$$

在 166MHz 主频，128 burst 长度下理论读效率

$$64 / (6 + 6 + 64 + 1)ns = 83.12\%$$

注意

在实际运行时，由于每间隔一段时间会有一次 single latency 的延迟插入，实际效率要比以上计算出来的值略低。

3.5.2 Double latency 模式

pSRAM CR0 配置值为 0x8f2c。

写操作的计算：连续拉高 pSRAM_wr_reqin 信号，总周期为两次连续的 ack 上升沿之间的周期数，一共需要 85 个时钟周期，其中，有效数据占 64 个时钟周期。

读操作的计算：连续拉高 pSRAM_rd_regin 信号，总周期为两次连续的 ack 上升沿之间的周期数，一共需要 86 个时钟周期，其中，有效数据占 64 个时钟周期。

在 200MHz 主频，128 burst 长度下理论写效率

$$64/(7 + 14 + 64) = 75.29\%$$

在 200MHz 主频，128 burst 长度下理论读效率

$$64/(7 + 14 + 64 + 1) = 74.42\%$$

在 166MHz 主频，128 burst 长度下理论写效率

$$64/(6 + 12 + 64)ns = 78.05\%$$

在 166MHz 主频，128 burst 长度下理论读效率

$$64/(6 + 12 + 64 + 1)ns = 77.11\%$$

注意 在实际运行时，由于每间隔一段时间会有一次 double latency 的延迟插入，实际效率要比以上计算出来的值略低。

3.6 pSRAM IP 约束方法

在 IP core pSRAM 目录下有个 constrains 文件夹，IP WIZARD 利用此文件自动在工程目录下的 constrains 文件夹中生成了约束文件，可供用户参考使用。此约束文件只做参考，工程中并没有加入。

pSRAM 控制器内部主频比较高，涉及到跨时钟域逻辑，需要遵循约束规则，不然可能会出现编译结果功能错误，或者结果不稳定的情况。

3.6.1 时钟声明举例如下

```
create_clock -period $T_clkb -name {clk_cpu} [get_pins {*_internal_pll_U_clk_rst_u_pll_osc_pll_u0/CO3}]

create_clock -period $T_clka -name {clk_psram} [get_pins {*_internal_pll_U_clk_rst_u_pll_osc_pll_u0/CO0}]

create_clock -period $T_clka -name {clk_psram_90} [get_pins {*_internal_pll_U_clk_rst_u_pll_osc_pll_u0/CO1}]

create_clock -period $T_clka -name {clk_psram_60} -waveform { [expr $T_clka/6] [expr $T_clka/6+$T_clka/2]} [get_pins
{U_clk_rst_u_pll_osc_pll_u0/CO2}]
```

其中 T_clka 是 pSRAM 时钟周期（快时钟），T_clkb 是初始化时钟周期（慢时钟）。

3.6.2 时钟延迟约束

```
#psram clock latency constrains

set_clock_latency [expr $T_clka /4] [get_clocks {clk_psram_90}] -rise -fall
```

Clk_pSRAM_90与 Clk_pSRAM 从相同 PLL 出来，并相移90度，当主频大于等于133MHz 时这里需要设置时钟之间的 latency，以便约束工具正确的分析时序，以及驱动布局布线。

注意 主频如果小于 133MHz，这条约束需要去除。

3.6.3 跨时钟域处理

```
#multi bit slow to fast async constrains, extend src data period to 2 cycle des_clk

set_max_delay [expr $T_clka*2] -from [get_cells {*_data_slow_reg*}] -to [get_cells {*_data_fast_reg*}]

set_max_delay [expr $T_clka*2] -from [get_cells {*_dvalid_slow_reg*}] -to [get_cells {*_dvalid_slow2fast_reg0_reg*}]

set_max_delay [expr $T_clka*2] -from [get_cells {*_data_slow_d0_reg*}] -to [get_cells {*_data_fast_d0_reg*}]

#single bit slow to fast async constrains

set_false_path -from [get_cells {*_sig_slow_r_reg*}] -to [get_cells {*_sig_fast_r_reg*}]

#multi bit fast to slow async constrains

set_max_delay [expr $T_clkb*2] -from [get_cells {*_dvalid_fast_reg*}] -to [get_cells {*_dvalid_fast2slow_reg0_reg*}]

set_max_delay [expr $T_clkb*2] -from [get_cells {*_data_fast_reg*}] -to [get_cells {*_data_slow_reg*}]

#single bit fast to slow async constrains

set_false_path -from [get_cells {*_sig_fast_reg*}] -to [get_cells {*_sig_slow_reg*}]
```

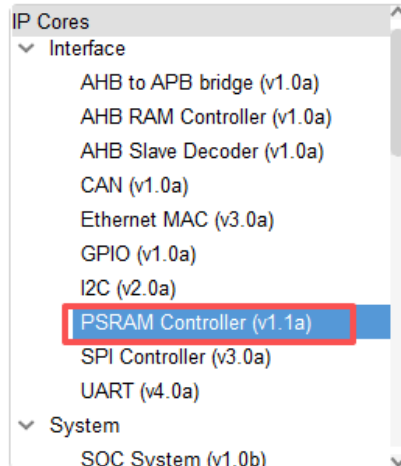
控制器内部 PLL 的第4路和第1路输出之间有跨时钟传输信号，这里需要放宽约束条件。

3.7 pSRAM IP 的生成

福晞（FX_2025.1.0以后更新版本）IP wizard 都支持 pSRAM 的例化。

Create a new design

Please select an IP core you want to customize from list below:



Current device is: H7P20N0V213-M4H117

Note: If some IP cores are in gray text, it means that current device does not support such cores.

Cancel < Back Next > Finish

图 3-12 IP wizard 选择 IP 界面

3.7.1 IP wizard 参数设计

新创建一个 pSRAM controller 的 IP，具体参数界面设置如下：

第一页参数：

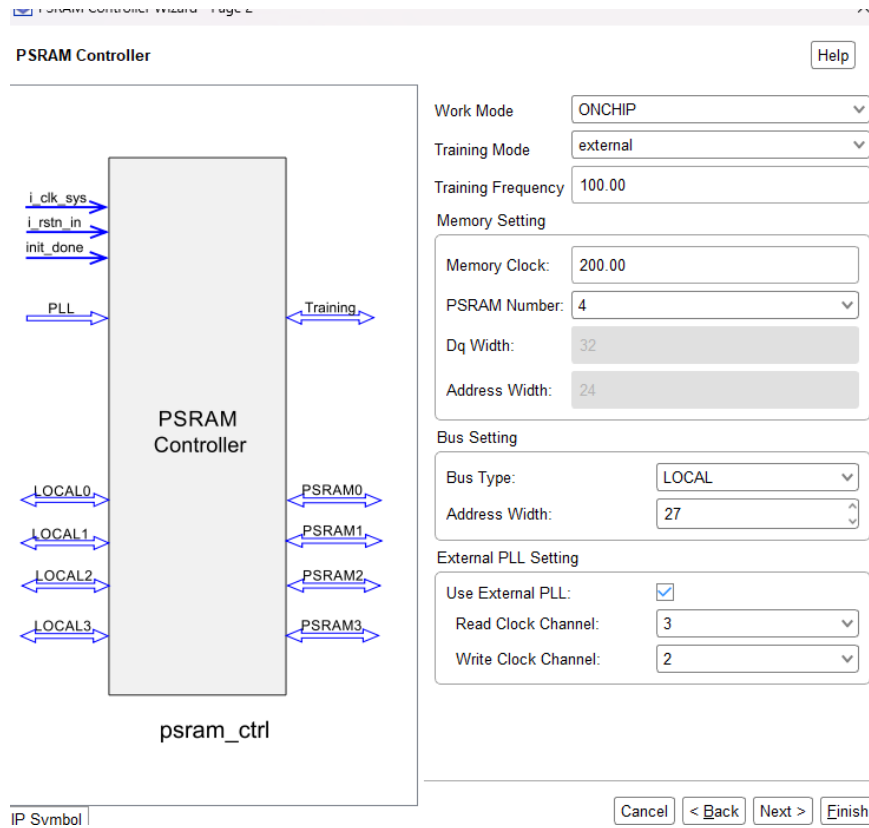


图 3-13 pSRAM 例化第一页

- ❑ **Work Mode:** 可选 ONCHIP 和 CUSTOM 两种方式。ONCHIP 模式下，自动匹配当前芯片型号内部合封 pSRAM 型号（即对应的 Memory Type）。CUSTOM 模式一般是用户选择外接 pSRAM，这种模式可配置的参数更多。
- ❑ **Training MODE:** 可以选择 internal training，或者是外部 AHB 从接口（通常接 SoC）做 training。需要注意内部校准模式只有寄存器初始化和读校准功能。
- ❑ **Training Frequency:** 对应 training 时钟的主频，软件利用该频率生成正确的时序约束文件（但是该文件不会被带入工程，只供参考）。
- ❑ **Memory Clock:** 选择 pSRAM 运行的频率，用户可以自己输入频率，推荐 200MHz。
- ❑ **pSRAM Number:** 设置控制器支持的 pSRAM 数量。H7 目前最多 2 个，P2 目前最多支持 4 个。
- ❑ **Dq Width:** pSRAM 数据线位宽（ONCHIP 模式下自动填写）。
- ❑ **Address Width:** pSRAM 地址位宽（ONCHIP 模式下自动填写）。
- ❑ **Use External PLL:** 用于配置是否使用控制器内部 PLL。内部 PLL 参数已经设定，H7 是 20MHz 输入 166MHz 输出，P2 是 20MHz 输入 200MHz 输出。如果频率不满足您的需求，可以选择外部例化 PLL。使用内部 PLL 的好处是控制器接口简单，时钟复位逻辑内部已经完成。

- 在此模式下有 Read Clock Channel 和 Write Clock Channel 选项，对应外部 PLL 生成的 pSRAM 读时钟和写时钟的输出通道。如果填写不对，将会导致无法初始化成功。
- 另外在 Use External PLL 模式下，需要填写 training 时钟的主频，此项参数仅用于生成约束文件，请正确填写。

第二页参数：

PSRAM Option

Burst Mode:	64	▼
Fixed Latency:	Fixed	▼
Initial Latency:	7	▼
HyBrid Burst:	Enable	▼
PASR:	0	▼
Drive Strength:	34 ohms	▼

图 3-14 pSRAM internal training 模式下例化第二页

PSRAM Option

Burst Mode:	64	▼
Initial Latency:	7	▼

图 3-15 pSRAM external training 模式下例化第二页

- ❑ **Burst Mode:** 设置最大用户接口 Burst 长度（Clock cycle），根据用户需求自行选择。pSRAM 器件支持的长度范围都在列表里。
- ❑ **Fixed Latency:** 设置固定 2 倍延迟或者可变延迟。只有 internal training 模式下，pSRAM num 为 1 的时候才可设置。external training 模式时由外部 MCU 设置。pSRAM num 大于 1 的时候不支持 variable latency 模式。
- ❑ **Initial Latency:** 设置延迟时钟周期。
- ❑ **HyBrid Burst:** 当 pSRAM 器件支持这个模式的时候，可以配置混合 burst 来增加读写效率。
- ❑ **PASR:** 内部刷新方式，默认 0，全部刷新。
- ❑ **Drive Strength :** 驱动强度选择。

3.8 资源性能报告

3.8.1 系统设计框图

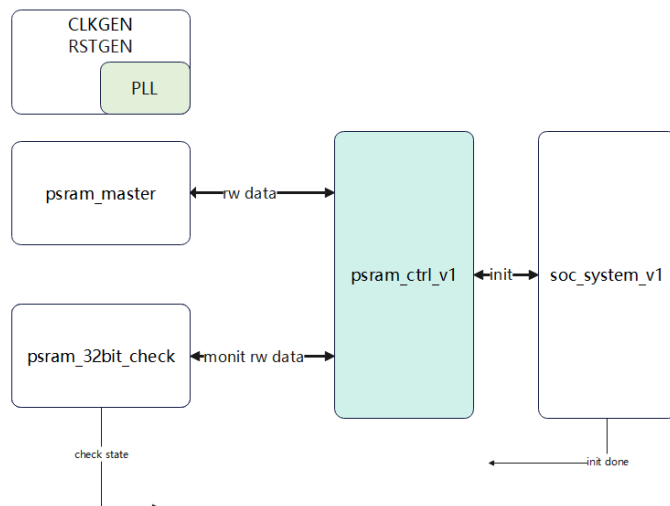


图 3-16 pSRAM ctrl IP 核测试系统框图

3.8.2 资源报告

Working Mode	LUT	REG	BRAM	PLL
Example project	1382	1594	8	1
Only psram controller psram_num=1 external training DQ_WIDTH =16	384	795	0	0
Only psram controller psram_num=1 internal training DQ_WIDTH =16	684	864	0	0
Only psram controller psram_num=1 external training DQ_WIDTH =8	354	651	0	0
Only psram controller psram_num=1 internal training DQ_WIDTH =8	630	799	0	0
Only psram controller psram_num=4 external training DQ_WIDTH =32	900	1536	0	0
Only psram controller	528	959	0	0

Working Mode	LUT	REG	BRAM	PLL
psram_num=2 external training DQ_WIDTH =16				

3.8.3 性能报告

器件	FMAX
P2P50N0V324-XXX	250MHz
P2L50N0V324-XXX	250MHz
P2P50N0W168-XXX	250MHz
P2L50N0W168-XXX	250MHz
H7P20N0L176-XXX	200MHz
H7N0L176-XXX	200MHz
H7P20N0L128-XXX	250MHz
H7P20N0V213-XXX	200MHz
H7P20N0S196-XXX	250MHz

3.9 仿真

- IP 生成之后 sim 目录下有三个文件：
 - hme_ip_psram_ctrl_v1_tb.v 仿真顶层
 - hme_ip_psram_ctrl_v1.vp 控制器仿真模型
 - W958D8NKY.v pSRAM 仿真模型
- 在福晞中配置如下：

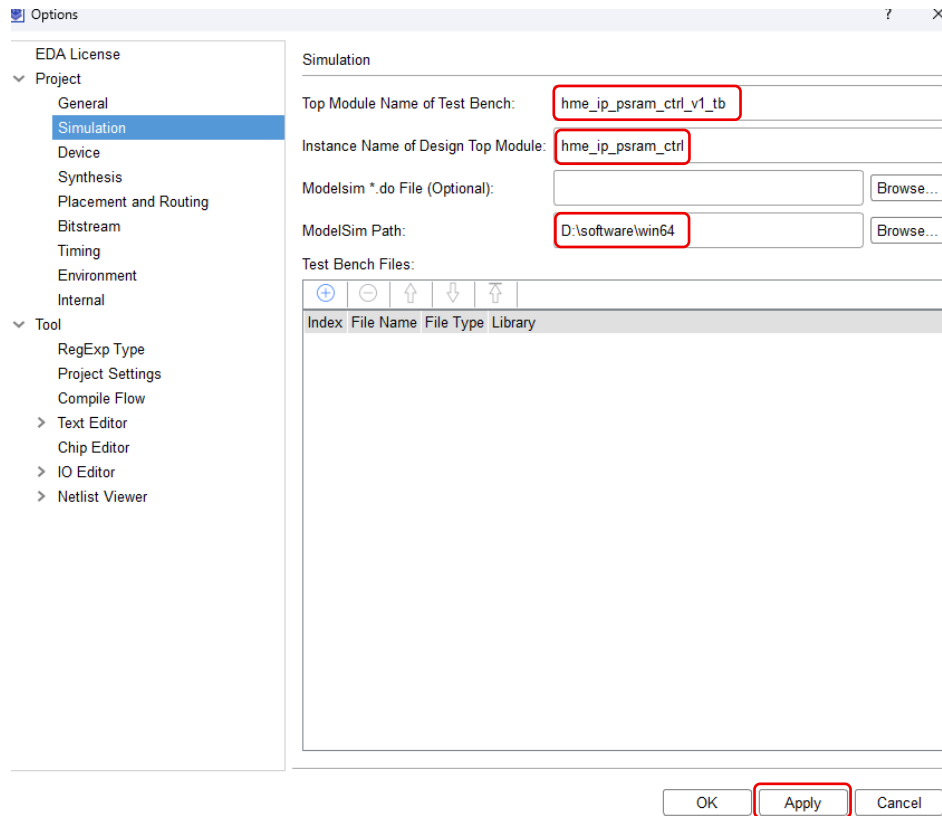


图 3-17 仿真配置界面

3. 设置好后点击 Apply 或 OK。
4. 然后在 UI 界面运行仿真。

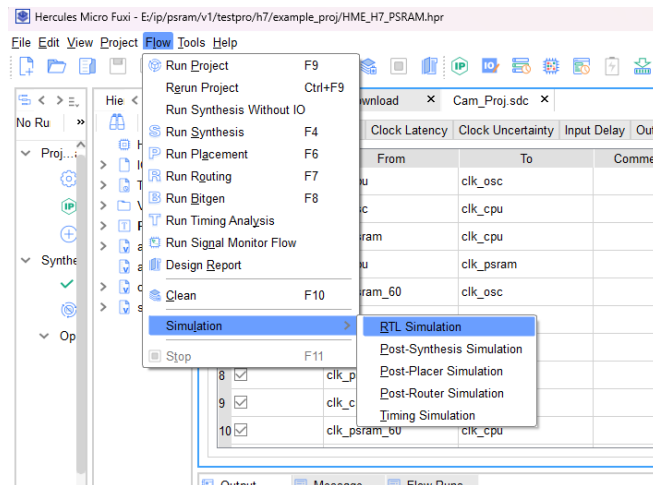


图 3-18 运行仿真界面

5. 检查仿真结果

```
tb.u_psrpm: time 836479.358 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h000000e, data out : 'h00
tb.u_psrpm: time 836481.858 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h000003e, data out : 'hle
tb.u_psrpm: time 836484.358 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h000003f, data out : 'h00
tb.u_psrpm: time 836486.858 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h000003f, data out : 'hlf
tb.u_psrpm: time 836489.358 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h0000040, data out : 'hxx
tb.u_psrpm: time 836490.608 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h0000040, data out : 'hxx
tb.u_psrpm: time 836490.608 ns:DIE0-INFO: Chip disable (0), addr_cmd_count 0, tCSM LOW period is 250.000 ns
tb.u_psrpm: time 836490.608 ns:DIE0-INFO: *****/
psram burst wrtie read test pass
** Note: $finish : src/tb.v(240)
Time: 837505608 ps Iteration: 0 Instance: /tb
1
Break in Module tb at src/tb.v line 240
```

图 3-19 Modelsim 打印结果

正常仿真读写对比通过后，会打印 test pass 字样。

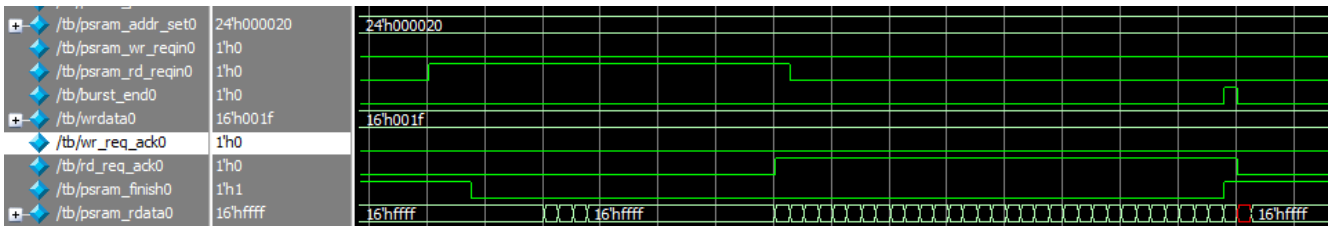


图 3-20 仿真波形

图中是读数据波形，可以看到各个信号的时序波形。

3.10 使用问题汇总

- (1) 如果内部有多片 pSRAM 并联位宽扩展，您需要使用 fix latency 模式，否则由于每片自动刷新时间不一致会导致同一时刻 latency 不一样，从而增加接口设计的复杂度。如下图所示。

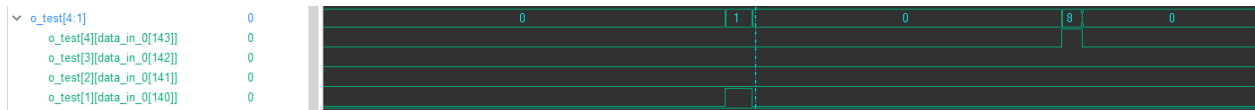


图 3-21 latency 不一样

RTL 代码 o_test[4:1]={ltyc_mode3,ltyc_mode2,ltyc_mode1,ltyc_mode0}

如上图，ltyc_mode0 先拉高，表明此次操作只有 pSRAM0 需要 double latency。

- (2) pSRAM 驱动电流问题

根据不同的硬件设计，FPGA 侧 pSRAM IO 的驱动电流强度可能会影响设计的性能。若遇到 pSRAM 测试不通过或者误码的情况，您可以尝试调整驱动电流。

- (3) pSRAM 使用内部校准失败

- 检查 PLL 的 pSRAM 时钟通道是否与 pSRAM IP 生成时候对应，规则是 PLL OUT0~6 对应通道 1~7。

- internal training 只做初始化，不做时钟相位校准：打开生成的 psram_ctrl_v1.v（自动生成的文件），将 TR_MODE 改为 "internal", CLIB_EN =0。

- 此时也可以使用内部 PLL 模式来做 training 校准，这样不会有通道设置错误的问题了。

4 文件结构说明

pSRAM IP 文件是由 wizard 自动生成的，主要包括：源文件（src）、仿真文件（sim）、参考设计和相关文档。详情如下表所示。

文件路径	描述
src 文件目录	
src/p2_h7	IP 核文件，用于综合
firmware 文件目录	
firmware/	MCU 固件.c 和.h 文件
constrains 约束目录	
constrains/p2_h7/	sdc 时序约束文件
sim 仿真目录	
sim/p2_h7/src	IP 核仿真加密文件
sim/p2_h7/hme_ip_psram_ctrl_v1_tb.v	仿真测试激励顶层
sim/	pSRAM 仿真模型
doc 文档目录	
doc/HME-pSRAM_IP_UserGuide.pdf doc/HME_pSRAM_IP_Example_UserGuide.pdf	IP 使用文档和参考设计文档
example 参考设计目录	
example/psram_example_h7.zip	参考设计，包含 MCU 和 FPGA 工程
example/psram_example_p2.zip	