



HME pSRAM IP User Guide

Sept. 2026

Hercules Microelectronics Co., Ltd.

Notes

Copyright © 2025-2026 Hercules Microelectronics Co., Ltd. All rights reserved.

No part of this document may be copied, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the written permission of Hercules Microelectronics Co., Ltd. All trademarks are the property of their respective companies.

Version Part Number

V1.1

Contact Us

If you have any problems or requirements during using our product, please contact

Hercules Microelectronics Co., Ltd., or your local distributors, or send e-mail to

sales@hercules-micro.com

Environmental Considerations

To avoid the harmful substances being released into the environment or harming human health, we encourage you to recycle this product in an appropriate way to make sure that most of the materials are reused or recycled appropriately. Please contact your local authorities for disposal or recycle information.

Warranty

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, Hercules Microelectronics Co., Ltd. reserves the right to discontinue or make changes, without prior notice, to any products herein to improve reliability, function, or design.

Hercules Microelectronics Co., Ltd. advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current.

The product introduced in this book is not authorized for use as critical components in life support devices or systems without the express written approval of Hercules Microelectronics Co., Ltd. As used herein: 1. Life support devices or systems are devices or systems that (a) are intended for surgical implant into the body or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user. 2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Revision History

The table below shows the revision history for this document.

Date	Version	Revision
2025-09-09	V1.0	Initial release.
2026-09-16	V1.1	1. Update IP wizard UI. 2. Add content about H7 V213&S196 package. 3. Optimize internal training mode to support 16bit pSRAM.

Table of Contents

Notes	1
Revision History	2
Table of Contents	3
Figure Contents	5
Table Contents	6
Before You Start	7
About This Datasheet	7
Legends	7
Register Type	7
1 Overview	8
1.1 Features	8
1.2 Device Family Support	8
1.3 Internal Block Diagram	10
2 Port and Register Description	11
2.1 Port Description	11
2.2 Parameter Description	16
2.3 Register List	18
2.4 Register Definition	18
3 IP Core Usage	20
3.1 Clock Reset Design	20
3.2 pSRAM Initialization	22
3.3 pSRAM Calibration	25
3.3.1 Purpose and Principle of Calibration	25
3.3.2 Write Training	26
3.3.3 Read Training	26
3.3.4 PLL Phase Adjustment	26
3.3.5 AHB Read/Write Timing	27
3.4 pSRAM Control Timing	28
3.4.1 LOCAL BUS Write Operation	28
3.4.2 LOCAL BUS Read Operation	29
3.4.3 LOCAL BUS Incomplete Write Operation	30
3.4.4 LOCAL BUS Incomplete Read Operation	30
3.5 Read and Write Efficiency	30
3.5.1 Single Latency Mode (Variable Latency)	30
3.5.2 Double Latency Mode	31
3.6 Constraint the pSRAM Controller IP	32
3.6.1 Examples of the Clock Constrains	32
3.6.2 Clock Delay Constraints	32
3.6.3 Cross Clock Domain Constraints	33
3.7 The Generation of the pSRAM Controller IP	33

3.7.1	IP wizard Parameter Definition	34
3.8	Resource Performance Report.....	36
3.8.1	System Design Block Diagram.....	36
3.8.2	Resource Report.....	36
3.8.3	Performance Report	37
3.9	Simulation	37
3.10	Issues Summary	39
4	Generate File Directory Structure.....	41

Figure Contents

Figure 1-1 pSRAM Block Diagram.....	10
Figure 3-1 Clock reset design in internal PLL mode.....	21
Figure 3-2 Clock reset design in external PLL mode.....	21
Figure 3-3 Initialization Flowchart.....	22
Figure 3-4 Read Clock Calibration for pSRAM.....	25
Figure 3-5 Write Clock Calibration for pSRAM.....	25
Figure 3-6 PLL Phase adjustment timing.....	27
Figure 3-7 AHB Read/Write timing.....	28
Figure 3-8 pSRAM Write operation burstlen=64 timing.....	29
Figure 3-9 pSRAM Read operation timing.....	29
Figure 3-10 pSRAM incomplete write operation timing.....	30
Figure 3-11 pSRAM incomplete read operation timing.....	30
Figure 3-12 IP wizard Select the pSRAM Controller IP.....	33
Figure 3-13 pSRAM Controller IP Wizard page1.....	34
Figure 3-14 pSRAM Controller internal training IP Wizard page2.....	35
Figure 3-15 pSRAM external training IP Wizard page2.....	35
Figure 3-16 Block diagram of the pSRAM controller IP test system.....	36
Figure 3-17 Simulation configuration.....	38
Figure 3-18 31Running the simulation interface.....	38
Figure 3-19 Results in Modelsim.....	39
Figure 3-20 Simulation waveform.....	39
Figure 3-21 pSRAM number is 4 and variable latency mode.....	39

Table Contents

Table 2-1 Port Description.....	11
Table 2-2 Description of The Parameters	16
Table 2-3 Register List	18
Table 2-4 PLL Dynamic Phase Control Register (Address: 0x4).....	18
Table 2-5 Training Control Register (Address: 0x8)	18
Table 2-6 Write Data Register 1 (Address: 0xc)	19
Table 2-7 Write Address Register (Address: 0x10).....	19
Table 2-8 Status Register (Address: 0x14).....	19
Table 2-9 Write Data Register 2 (Address: 0x18).....	19
Table 3-1 CR0 Configuration (W958D8NBYA)	22
Table 3-2 CR1 Configuration.....	24
Table 3-3 PLL Special Dynamic Phase Adjustment Ports	26

Before You Start

About This Datasheet

This document introduces the pSRAM controller. It serves as a technical reference, providing details on the parameters and characteristics of the device's features, and guides users on how to utilize the device in designing applications.

For more details of P2&H7 families, please go to <http://www.hercules-micro.com>.

If you have any questions or suggestions, please contact us at support@hercules-micro.com.

Legends

Abbreviation	Full Name	Abbreviation	Full Name
AHB	Advanced High-performance Bus	MCU	Micro Control Unit
APB	Advanced Peripheral Bus	OSC	Oscillator
ARM	Advanced RISC Machine	PLL	Phase-Locked Loop
EMB	Embedded Memory Block	pSRAM	Pseudo Static Random Access Memory
FMAX	Max Frequency	RAM	Random Access Memory
FPGA	Field Programmable Gate Array	REG	Register
IP	Intellectual Property	RISC	Reduced Instruction Set Computer
LUT	Look Up Table	UI	User Interface

Register Type

- ☐ RO: Read Only
- ☐ RW: Readable and Writable
- ☐ RWC: Readable and Writable, with clear function
- ☐ W1C: Write 1 to Clear
- ☐ WO: Write Only

1 Overview

pSRAM (Pseudo Static Random Access Memory) integrates dynamic memory cells with a static interface, making it ideal for portable electronics like smartphones, smart speakers, and GPS receivers. Its key advantages include low power consumption, a simple interface, and a small form factor. The pSRAM controller IP manages external or internal pSRAM devices to expand FPGA memory capacity. pSRAM is a mainstream low-cost, low-power RAM solution.

1.1 Features

- ❑ Support burst lengths 1–1024 (wrap/linear modes).
- ❑ Interfaces:
 - pSRAM data: Custom LOCAL BUS
 - Initialization: AHB interface
- ❑ Configurable pSRAM quantity (1–4 chips).
- ❑ Support 8-bit/16-bit per chip.
- ❑ Max capacity: 256Mb per chip.
- ❑ Modes:
 - Independent: Separate control of each pSRAM.
 - Concatenated: Multiple 8-bit pSRAMs combined for wider data access.
- ❑ Max frequency: 250MHz.
- ❑ Optional internal/external PLL.
- ❑ Support initialization training (FPGA logic or MCU-based).
- ❑ Resource usage (single 8-bit pSRAM controller):
 - 144 LUTs
 - 156 REGs

1.2 Device Family Support

FPGA Model	pSRAM Capacity	pSRAM Configuration
H7P20N0L176-M2H1	32Mb x 2	Onchip, custom

FPGA Model	pSRAM Capacity	pSRAM Configuration
H7P20N0L176-M1H1	32Mb x 1	Onchip, custom
H7P20N0L128-M3H1	128Mb x 1	Onchip, custom
H7P20N0V213-M1H1	32Mb x 1	Onchip, custom
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 others	External (only support winbond)	custom

1.3 Internal Block Diagram

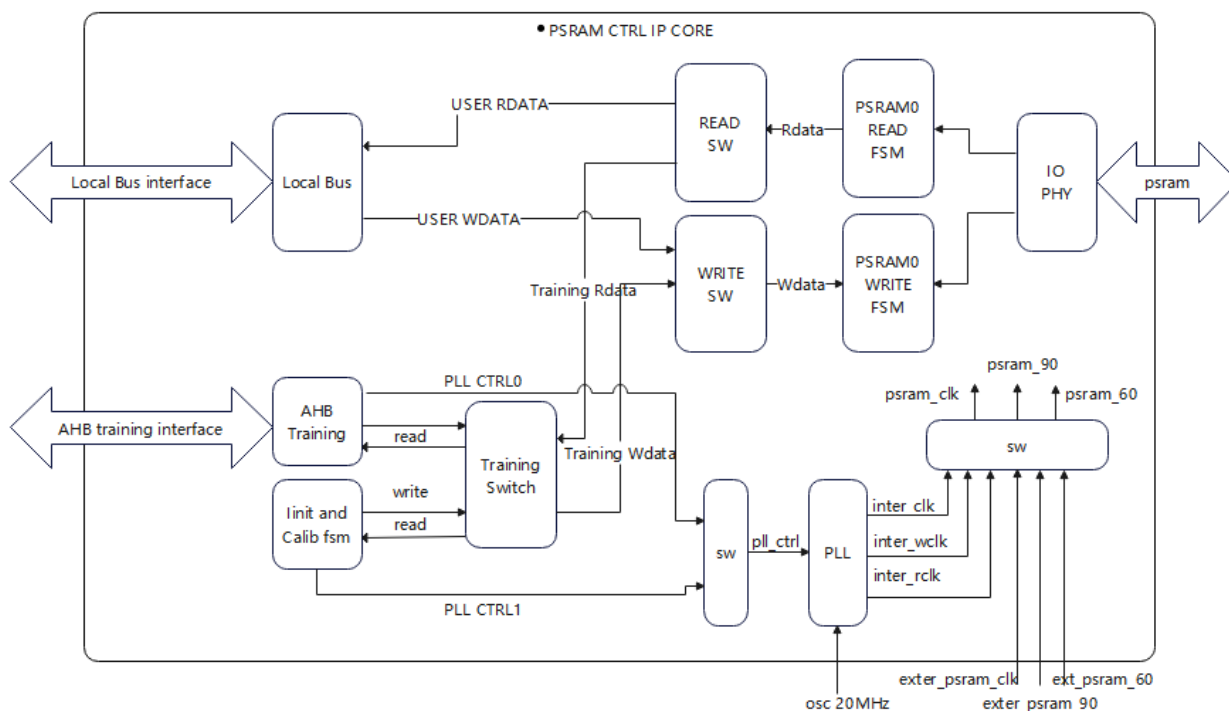


Figure 1-1 pSRAM Block Diagram

pSRAM Interface: It utilizes the HyperBus interface, which is a dual-edge data transfer interface featuring a low pin count and high throughput. The number of data lines depends on the pSRAM model, with current single devices supporting either 8-bit or 16-bit configurations.

Local Bus Interface: The user data interface bus is a Local Bus interface, with each pSRAM occupying 16-bit or 32-bit data width. The address bus is 32-bit, and the number and width of buses depend on the user's specific requirements.

Training can be implemented via RTL or an external AHB interface connected to the MCU. AHB Training Interface is a standard AHB slave interface. Currently, it only supports single transfer mode and is used for read/write control of the logic related to pSRAM initialization and calibration.

2 Port and Register Description

2.1 Port Description

Table 2-1 Port Description

Port Name	Type	Width	Description
i_clk_sys	input	1	20M input for the internal PLL
i_rstn_in	input	1	System reset when using the internal PLL
o_clk_usr	output	1	The internal PLL output clock is synchronized with the internal interface bus
o_clk_psram	output	1	The internal PLL output clock is synchronized with the pSRAM controller clock
l_clk_local	input	1	In external PLL mode, this clock serves as the initialization configuration clock; in internal PLL mode, it is unused.
i_rst_xn_psram	input	1	Assert low (active-low) in external PLL mode to reset the pSRAM controller and initialization logic; not used in internal PLL mode.
init_done	output	1	pSRAM initialization completed
AHB External Configuration Port			
train_hrdata_s	output	32	External AHB training interface
train_hsel_s	input	1	External AHB training interface
train_hwrite_s	input	1	External AHB training interface
train_hwdata_s	input	32	External AHB training interface
train_haddr_s	input	32	External AHB training interface
train_htrans_s	input	2	External AHB training interface

2 Port and Register Description

Port Name	Type	Width	Description
The PLL phase control interface			
ps_ck	output	1	External PLL phase-shift control interface
ps_dir	output	1	External PLL phase-shift control interface
ps_en	output	1	External PLL phase-shift control interface
ps_sel	output	3	External PLL phase-shift control interface
ps_done	input	1	External PLL phase-shift control interface
LOCAL BUS0 Interface			
psram_rdata0	output	16/32	Local Bus pSRAM read data interface
wr_req_ack0	output	1	Local Bus pSRAM write data response
rd_req_ack0	output	1	Local Bus pSRAM read data response
psram_wr_reqin0	input	1	local Bus pSRAM write request
wr_data0	input	16/32	Local Bus pSRAM write data interface
data_mask0	input	2	Local Bus pSRAM write data mask
psram_addr_set0	input	32	Local Bus pSRAM read and write address setting
psram_rd_reqin0	input	1	Local Bus pSRAM read request
wrburst_end0	input	1	Local Bus pSRAM write completion
rdburst_end0	input	1	Local Bus pSRAM read completion
wrburst_len0	input	10	Local Bus pSRAM write burst length
rdburst_len0	input	10	Local Bus pSRAM read burst length
linear_en0	input	1	Local Bus burst mode configuration, 0: Wrap Burst, 1: Linear Burst

2 Port and Register Description

Port Name	Type	Width	Description
psram_finish0	output	1	Local Bus pSRAM operation completion signal
LOCAL BUS1 Interface			
psram_rdata1	output	16/32	Local Bus pSRAM read data interface
wr_req_ack1	output	1	Local Bus pSRAM write data response
rd_req_ack1	output	1	Local Bus pSRAM read data response
psram_wr_reqin1	input	1	Local Bus pSRAM write request
wr_data1	input	16/32	Local Bus pSRAM write data interface
data_mask1	input	2	Local Bus pSRAM write data mask
psram_addr_set1	input	32	Local Bus pSRAM read and write address setting
psram_rd_reqin1	input	1	Local Bus pSRAM read request
wrburst_end1	input	1	Local Bus pSRAM write completion
rdburst_end1	input	1	Local Bus pSRAM read completion
wrburst_len1	input	10	Local Bus pSRAM write burst length
rdburst_len1	input	10	Local Bus pSRAM read burst length
linear_en1	input	1	Local Bus burst mode setting
psram_finish1	output	1	Local Bus pSRAM operation completion signal
LOCAL BUS2 Interface			
psram_rdata2	output	16/32	Local Bus pSRAM read data interface
wr_req_ack2	output	1	Local Bus pSRAM write data response
rd_req_ack2	output	1	Local Bus pSRAM read data response

2 Port and Register Description

Port Name	Type	Width	Description
psram_wr_reqin2	input	1	Local Bus pSRAM write request
wr_data2	input	16/32	Local Bus pSRAM write data interface
data_mask2	input	2	Local Bus pSRAM write data mask
psram_addr_set2	input	32	Local Bus pSRAM read and write address setting
psram_rd_reqin2	input	1	Local Bus pSRAM read request
wrburst_end2	input	1	Local Bus pSRAM write completion
rdburst_end2	input	1	Local Bus pSRAM read completion
wrburst_len2	input	10	Local Bus pSRAM write burst length
rdburst_len2	input	10	Local Bus pSRAM read burst length
linear_en2	input	1	Local Bus burst mode setting
psram_finish2	output	1	Local Bus pSRAM operation completion signal
LOCAL BUS3 Interface			
psram_rdata3	output	16/32	Local Bus pSRAM read data interface
wr_req_ack3	output	1	Local Bus pSRAM write data response
rd_req_ack3	output	1	Local Bus pSRAM read data response
psram_wr_reqin3	input	1	Local Bus pSRAM write request
wr_data3	input	16/32	Local Bus pSRAM write data interface
data_mask3	input	2	Local Bus pSRAM write data mask
psram_addr_set3	input	32	Local Bus pSRAM read and write address setting
psram_rd_reqin3	input	1	Local Bus pSRAM read request

2 Port and Register Description

Port Name	Type	Width	Description
wrburst_end3	input	1	Local Bus pSRAM write completion
rdburst_end3	input	1	Local Bus pSRAM read completion
wrburst_len3	input	10	Local Bus pSRAM write burst length
rdburst_len3	input	10	Local Bus pSRAM read burst length
linear_en3	input	1	Local Bus burst mode setting
psram_finish3	output	1	Local Bus pSRAM operation completion signal
pSRAM0 Interface			
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 Interface			
P_RSTN1	output	1	pSRAM IO
CK1	output	1	pSRAM IO
CK_N1	output	1	pSRAM IO
CS_N1	output	1	pSRAM IO
RWDS1	inout	1/2	pSRAM IO
DQ_RAM1	inout	8/16	pSRAM IO

2 Port and Register Description

Port Name	Type	Width	Description
pSRAM2 Interface			
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 Interface			
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 Parameter Description

Table 2-2 Description of The Parameters

Name	Type	Range	Default	Description
RD_CH_NUM	Binary	3'b000~3'b111	3'b011	Read clock PLL channel
WR_CH_NUM	Binary	3'b000~3'b111	3'b010	Write clock PLL channel
HYBRID_EN	Binary	0,1	0	Enable the hybrid burst function 0: enable; 1: disable

2 Port and Register Description

Name	Type	Range	Default	Description
DRV_STR	Binary	3'b000~3'b111	3'b000	pSRAM Output drive strength configuration
DQ_WIDTH	Integer	8,16,32	32	Number of DQ
AD_WIDTH	Integer	1~27	24	The address width of pSRAM
TR_MODE	String	"internal" "external"	"internal"	0: Select the internal logic to implement initialization 1: Select the external AHB_train interface to implement the initialization
Fixed_Latency	Binary	0,1	1	1: Fixed Latency mode 0: Variable Latency mode
Initial_Latency	Integer	6,7	7	The number of refresh Latency cycles of pSRAM
PASR	Binary		3'b000	pSRAM Refresh location settings
EXT_PLL	Binary	0,1	0	1: Generated pSRAM clock using an external PLL 0: Clock generated using the internal PLL
pSRAM_IDX	Integer	0,1,2,3	0	Device selection during single pSRAM instantiation. Set to 0 in otherwise
CLK_FREQ	Integer		200	pSRAM Clock main frequency, unit MHz
C_MAX_BURST	Integer	1~64	16	Maximum burst length
C_pSRAM_NUM	Integer	1,2,4	2	Number of pSRAM
BUS_TYPE	String	"LOCAL"	"LOCAL"	Select the user interface type
C_DATA_WIDTH	Integer	16,32,64,128	32	Bus Interface Width Setting (Note: AHB interface is fixed at 32-bit)
C_ADDR_WIDTH	Integer	1~32	32	Bus address bit width
CLIB_EN	Binary	0,1	0	1: Enable calibration function; 0: Disable calibration function

2.3 Register List

Table 2-3 Register List

Offset	Name	Type	Default	Description
0x0	func_test	NC	0	Test register
0x4	pll_set_reg	RW	0	The PLL dynamic phase control register
0x8	mcu_ctrl	RW	0	Training control register
0xc	mcu_wdata1	WO	0	Write the lower 32 bits data
0x10	mcu_addr	WO	0	Write address
0x14	pSRAM_status	RO	0	pSRAM status register
0x18	mcu_wdata2	WO	0	Write the higher 32 bits data

2.4 Register Definition

Table 2-4 PLL Dynamic Phase Control Register (Address: 0x4)

Bit	Name	Type	Default	Description
[7]	Phase shift enabling	RW	0	Raising the signal (to a high level) enables the phase shift.
[6]	Phase shift direction	RW	0	0: Reverse phase shift 1: Positive direction of phase shift
[2:0]	The PLL output channel selection	RW	0	The output clock channel corresponding to the PLL is 0~7

Table 2-5 Training Control Register (Address: 0x8)

Bit	Name	Type	Default	Description
[12:11]	Write data mask	RW	0	1: The byte corresponding to this operation is invalid
[8]	Set the init_done flag	RW	0	1: pSRAM initialization is completed
[7]	Clear the latency register status	RW	0	Write 1 to clear the latency register status and needs to write back to 0
[5]	pSRAM reset control	RW	0	The reset needs to write 0 first, and then write 1

2 Port and Register Description

Bit	Name	Type	Default	Description
[4]	pSRAM Register mode setting	RW	0	1: Register read and write mode 0: Data read and write mode
[1]	Training read request	RW	0	Write 1 to initiate read request; must write back 0 afterward to clear
[0]	Training write request	RW	0	Write 1 to initiate write request; must write back 0 afterward to clear

Table 2-6 Write Data Register 1 (Address: 0xc)

Bit	Name	Type	Default	Description
[31:0]	write data register 1	WO	0	[15:0] write data to the pSRAM 0 [31:16] write data to the pSRAM 1

Table 2-7 Write Address Register (Address: 0x10)

Bit	Name	Type	Default	Description
[31:0]	Write address register	WO	0	[31:0] write address command to the pSRAM

Table 2-8 Status Register (Address: 0x14)

Bit	Name	Type	Default	Description
[8]	pSRAM Control status	RO	0	Indicates whether the PLL completes the phase shift, and 1 indicates the completion
[4:1]	pSRAM latency state	RO	0	1: Indicates requires additional delay for normal operation 4:1 corresponds to pSRAM 3~0
[0]	Whether the operation is completed	RO	0	1: pSRAM operation completed 0: the operation is in progress

Table 2-9 Write Data Register 2 (Address: 0x18)

Bit	Name	Type	Default	Description
[31:0]	write data register 2	WO	0	[15:0] write data to the pSRAM 2 [31:16] write data to the pSRAM 3

3 IP Core Usage

The pSRAM controller IP includes:

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

3.1 Clock Reset Design

In the external PLL mode: there are three external input clocks `i_clk_pSRAM`, `i_clk_pSRAM_90` and `i_clk_pSRAM_60`. These three clocks are generated from the same clock through the PLL. The initial phase relationship is 0 degrees, 90 degrees and 60 degrees respectively (this phase shift changes slightly after training).

In the internal PLL mode: `i_clk_sys` (20 MHz) as the input source for internal PLL, all the clocks required for pSRAM internally, and `o_clk_usr` (100 MHz), `o_clk_pSRAM` (200 MHz) will output the clock to training logic and pSRAM read and write logic respectively. Where the `o_clk_pSRAM` is synchronized with the pSRAM controller.

The PLL in the design needs to support dynamic phase adjustment, adjusting for `i_clk_pSRAM_90` and `i_clk_pSRAM_60` during training.

- ☐ `i_rstn_in`: In external PLL mode reset internal initialization logic (slow clock domain `i_clk_local`), internal PLL, as global reset in pSRAM controller system.
- ☐ `i_clk_sys`: In external PLL mode initializes internal clock (recommended $F_{i_clk_pSRAM}/2$), in internal PLL, as 20MHz input clock source for PLL.
- ☐ `i_rst_xn_pSRAM`: reset the pSRAM controller related logic (non-initialized part), not connected in internal PLL mode.

The following is an example of the reset design:

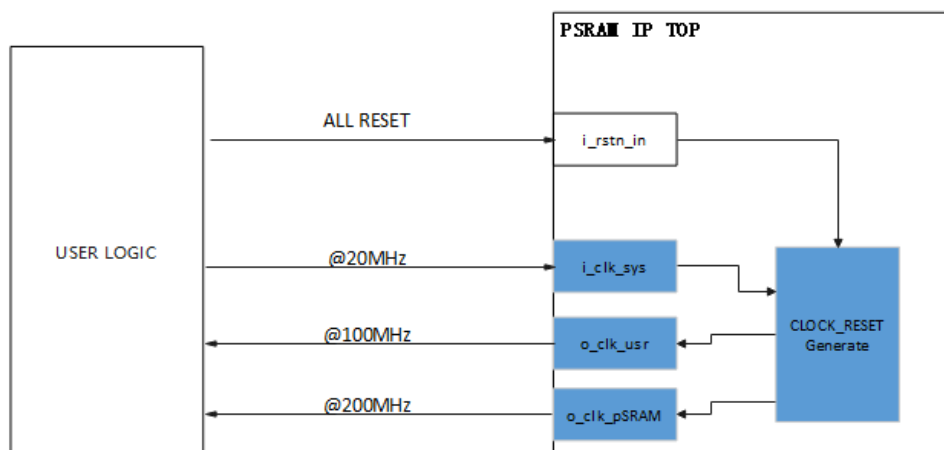


Figure 3-1 Clock reset design in internal PLL mode

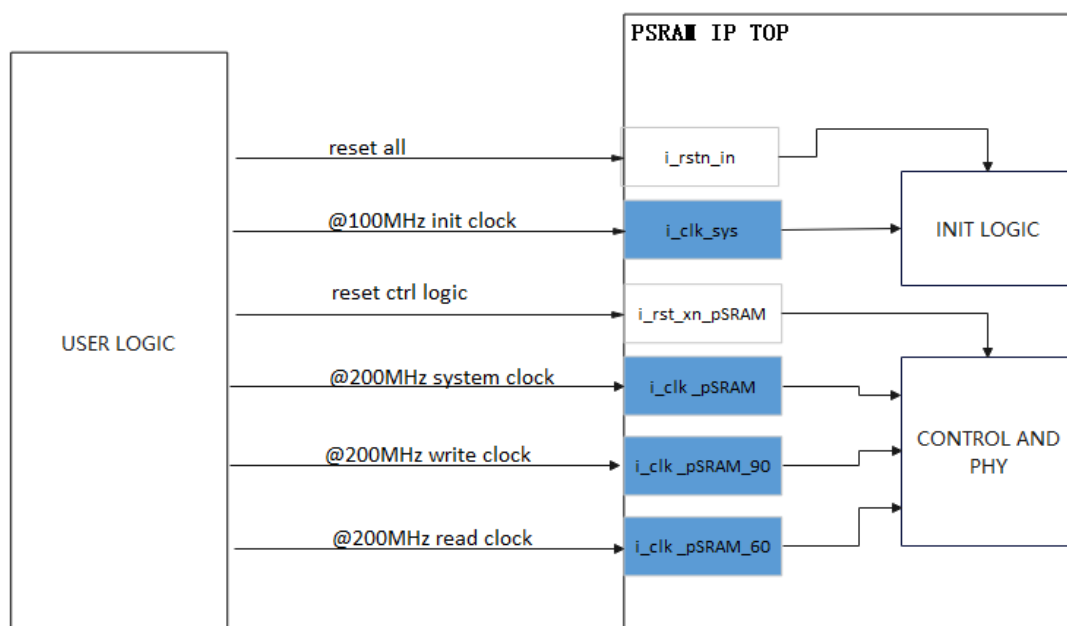


Figure 3-2 Clock reset design in external PLL mode

3.2 pSRAM Initialization

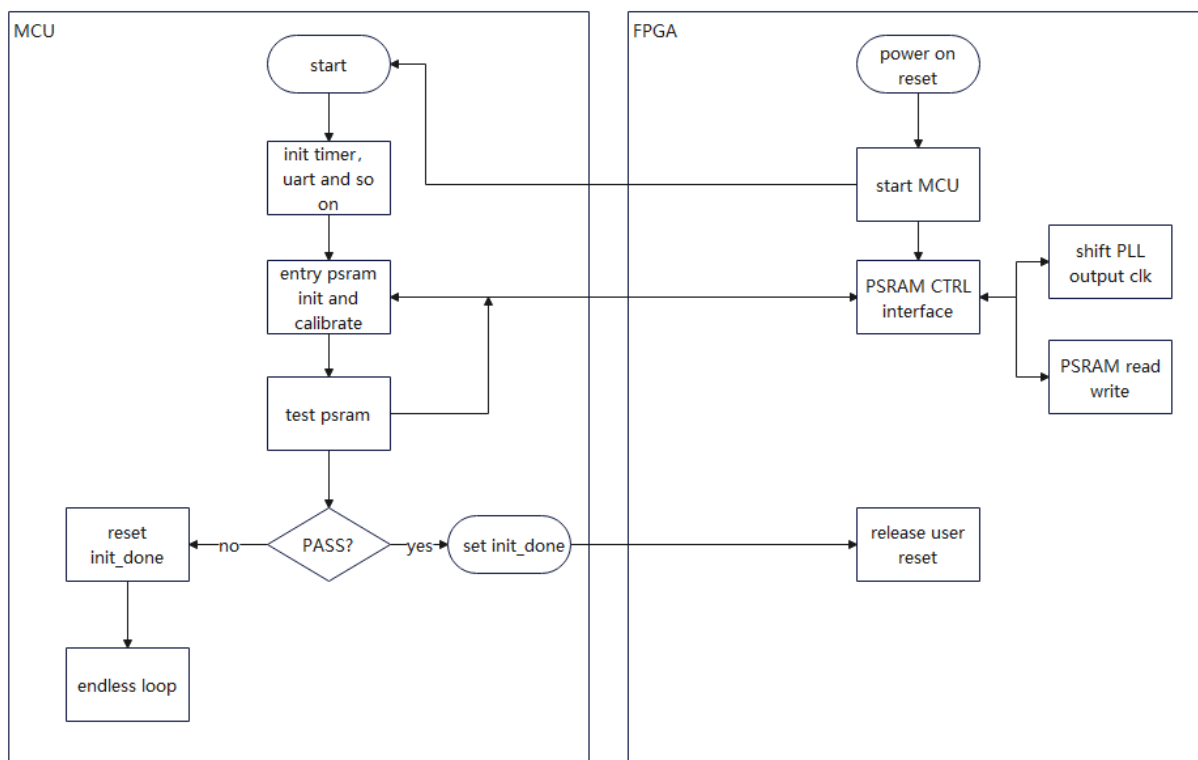


Figure 3-3 Initialization Flowchart

pSRAM registers are configured by ARM via {pSRAM_init()} or FPGA logic. This section focuses on MCU configuration.

For P2P50N0V324-M4H1:

- ❑ CR0 (Configuration Register 0): Recommended value 0x8f24 or 0x8f2c.
- ❑ CR1 (Configuration Register 1): Recommended value 0xff80.

After initialization, execute pSRAM_ctrl_mst_set(1) to enter User Mode.

Table 3-1 CR0 Configuration (W958D8NBYA)

CR0 bit	Name	Description
[15]	Deep Power Down Enable	1b - Normal operation (default) 0b - Writing 0 to CR0[15] causes the device to enter Deep Power Down (DPD) <i>Note: pSRAM will set the value of CR0[15] to "1" automatically after exit DPD.</i>
[14:12]	Drive	000b - 34 Ω (default)

CR0 bit	Name	Description
	Strength	001b - 115 Ω 010b - 67 Ω 011b - 46 Ω 100b - 34 Ω 101b - 27 Ω 110b - 22 Ω 111b - 19 Ω
[11:8]	Reserved	1b – Reserved When writing to CR0, these bits should be set to 1 for future compatibility
[7:4]	Initial Latency	0000b - 5 clock delay @ 133 MHz max frequency 0001b - 6 clock delay @ 166 MHz max frequency 0010b - 7 clock delay @ 200 MHz max frequency (default) 0010b - 7 clock delay @ 250 MHz max frequency ... 1101b – Reserved 1110b - 3 clock delay @ 85 MHz max frequency 1111b - 4 clock delay @ 104 MHz max frequency
[3]	Fixed Latency Enable	0b - Variable Latency – 1 or 2 times Initial Latency depending on RWDS during CA cycles. 1b - Fixed 2 times Initial Latency (default)
[2]	Hybrid Burst Enable	0b - Wrapped burst sequences in hybrid burst wrapping mode 1b - Wrapped burst sequences in legacy burst wrapping mode (default)
[1:0]	Burst-length	00b - 128 bytes 01b - 64 bytes 10b - 16 bytes 11b - 32 bytes (by default)

Table 3-2 CR1 Configuration

CR1 bit	Name	Description
[15:8]	Reserved	FFh - Reserved (default), reserved for Future Use. When writing CR1 to this register, these bits should keep FFh for future compatibility.
[7]	Reserved	1b - Reserved (default) When writing CR1 to this register, this bit should keep 1b.
[6]	Master Clock Type	0b - Differential - CK#, CK 1b - Single Ended - CK (default)
[5]	Hybrid Sleep	0b - Normal operation (default) 1b - Writing 1 to CR1[5] causes the device to enter Hybrid Sleep (HS) State
[4:2]	Partial Array Refresh	000b - Full Array (default) 001b - Bottom 1/2 Array 010b - Bottom 1/4 Array 011b - Bottom 1/8 Array 100b - Reserved 101b - Top 1/2 Array 110b - Top 1/4 Array 111b - Top 1/8 Array
[1:0] ^{*1}	Distributed Refresh Interval	00b - Reserved 01b - 4 μ s (t_{CSM}) 10b - Reserved 11b - Reserved

Note 1 CR1 [1:0] bit is read only.

3.3 pSRAM Calibration

3.3.1 Purpose and Principle of Calibration

The pSRAM controller operation requires three clocks, named `psram_clk`, `psram_clk_90`, and `psram_clk_60`.

In practice, the `psram_clk_90` clock is responsible for transmitting data, with its phase intentionally shifted by 90 degrees relative to the data signals to ensure correct sampling at the receiver. However, due to real-world factors such as PCB layout/routing variations or differences in memory device characteristics, the actual phase difference between the clock and data may not be exactly 90 degrees. Therefore, a PLL is used to adjust the phase shift of `psram_clk_90` to locate the optimal center of the sampling window.

Similarly, the `psram_clk_60` clock is used to sample read data entering the IO interface. As with transmission, the latency of this data can also vary. Thus, a PLL is employed to apply phase shifting to `psram_clk_60` in order to find the suitable center point of the sampling window.

ARM calls `{psram_training_new ();}` function for training, and calls the function `{pll_adjust ();}` in the process, thus generating the control signals `ps_ck`, `ps_dir`, `ps_en` and `ps_sel`. Together, these signals phase-shift the `clkout1` and `clkout2` of the PLL to find the full clock window.

FPGA needs to use `u_apb2_fp_slave` module to connect ARM and pSRAM controller.

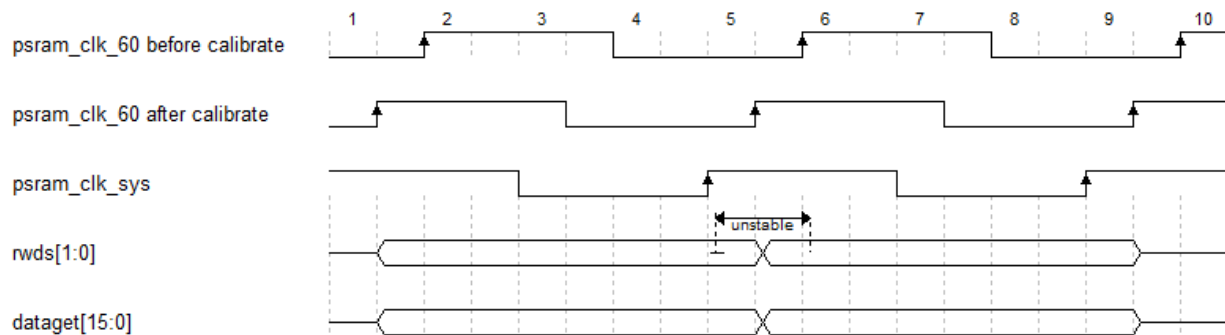


Figure 3-4 Read Clock Calibration for pSRAM

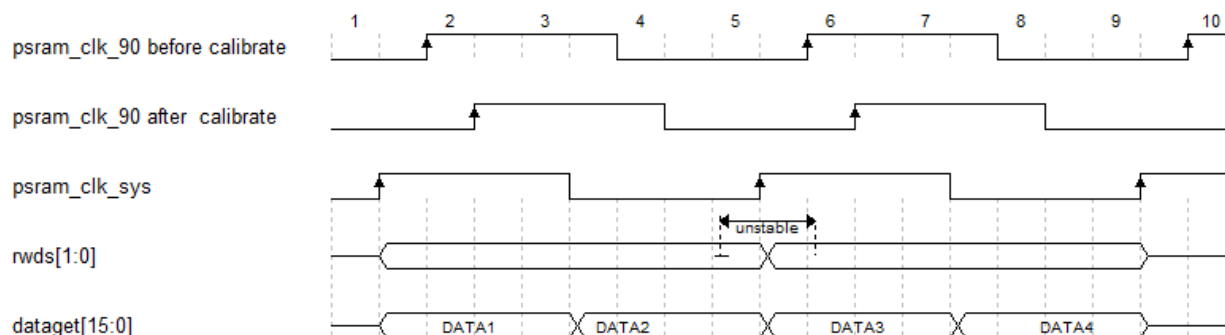


Figure 3-5 Write Clock Calibration for pSRAM

3.3.2 Write Training

ARM calls `psram_wr_training_new()`:

- (1) `psram_wr_training_basic_forward()` determines starting point of phase window.
- (2) `psram_wr_window_scan_forward()` determines size of phase window.
- (3) `pll_adjust()` centers the phase.

3.3.3 Read Training

ARM calls `psram_rd_cmd_training_new()`:

- (1) `psram_rd_cmd_training_basic_new()` determines starting point of phase window.
- (2) `psram_rd_window_scan_new()` determines size of phase window.
- (3) `pll_adjust()` centers the phase.

3.3.4 PLL Phase Adjustment

HME-P2/H7 PLLs support dynamic phase adjustment.

`{pll_adjust();}` function uses the PLL special dynamic phase adjustment port as shown in **Table 3-3**. For the pSRAM IO clock and the pSRAM controller, the phase increases or decreases $T_{vco}/64$. The timing diagram during the dynamic phase adjustment is shown in **Figure 3-6**.

Table 3-3 PLL Special Dynamic Phase Adjustment Ports

Name	Type	Width	Description
ps_ck	input	1	During dynamic phase adjustment, give the PLL a clock for receiving the relevant signals
ps_dir	input	1	Control the direction of the phase adjustment 0: Forward (left) adjustment 1: Backward (right) adjustment
ps_en	input	1	Dynamic Phase Adjustment Enable: Falling edge triggers single phase adjustment
ps_done	output	1	Phase adjustment completion signal given by the PLL
ps_sel	input	3	Select the clock channel requiring phase adjustment

Name	Type	Width	Description
			000: No selected
			001: Select channel 0
			010: Select channel 1
			011: Select channel 2
			100: Select channel 3
			101: Select channel 4
			110: Select channel 5
			111: Select channel 6

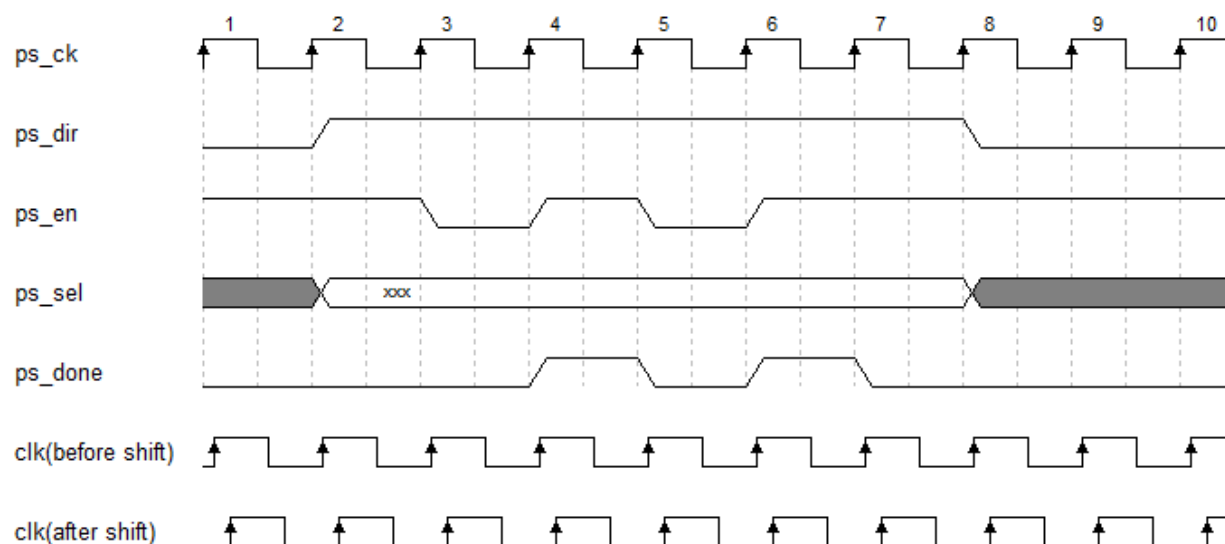


Figure 3-6 PLL Phase adjustment timing

3.3.5 AHB Read/Write Timing

AHB interface does not support burst operations, only supports simple AHB read and write operation.

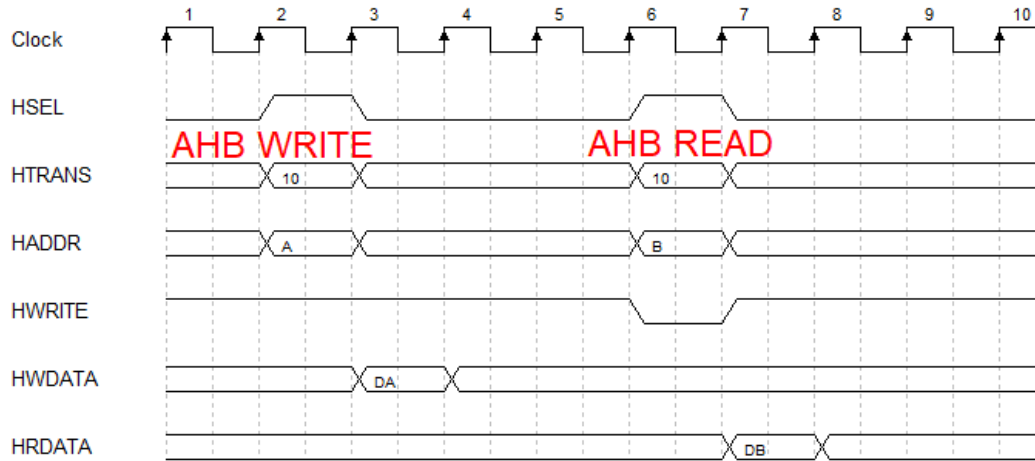


Figure 3-7 AHB Read/Write timing

3.4 pSRAM Control Timing

The pSRAM controllers of the HME-P2 and H7 series currently support complete read, write operations, and incomplete write operations with a maximum burst length of 128 bytes.

3.4.1 LOCAL BUS Write Operation

pSRAM_wr_reqin: Write request input to PSRAM controller (rising-edge triggered); pSRAM_addr_set: Write operation start address setup signal.

- (1) Execute pSRAM_addr_set configuration.
- (2) pSRAM_wr_reqin (rising-edge trigger) initiates write.
- (3) After sending command/address, wr_req_ack goes high.
- (4) 16 bits data are sent on each clock rising edge.

A total of 64 such data are sampled for a complete write operation, and the pSRAM controller will return to the initial state.

The timing diagram of the full write operation is shown in **Figure 3-8**.

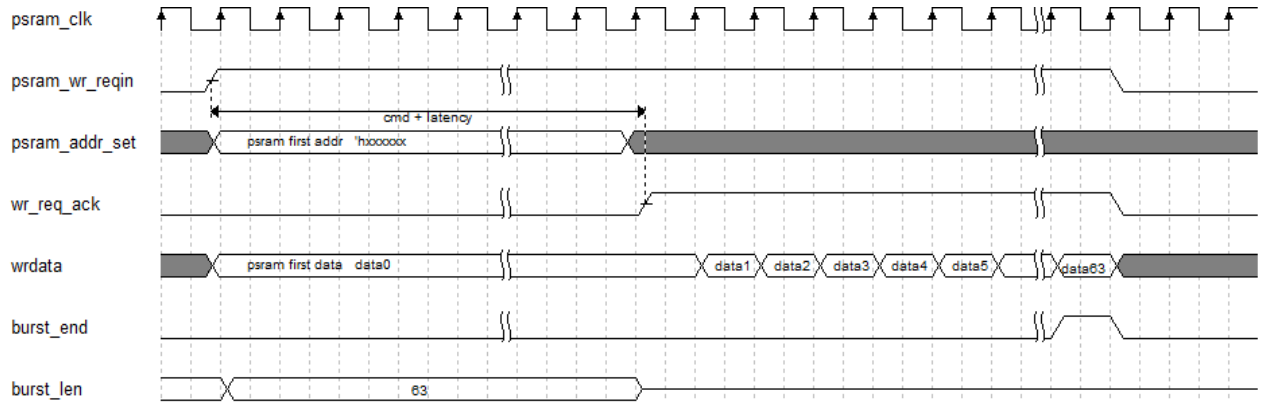


Figure 3-8 pSRAM Write operation burstlen=64 timing

3.4.2 LOCAL BUS Read Operation

psram_rd_reqin: Read request input to PSRAM controller; psram_addr_set: Read operation start address setup signal.

- (1) Execute psram_addr_set configuration.
- (2) psram_rd_reqin (rising-edge trigger) initiates read.
- (3) After sending command/address, rd_req_ack goes high.
- (4) Data is converted from 8-bit DDR to 16-bit SDR.

A total of 64 16-bits data will be sampled for a complete read operation, and the pSRAM controller will return to the initial state then.

The timing diagram of the complete read operation is shown in **Figure 3-9**.

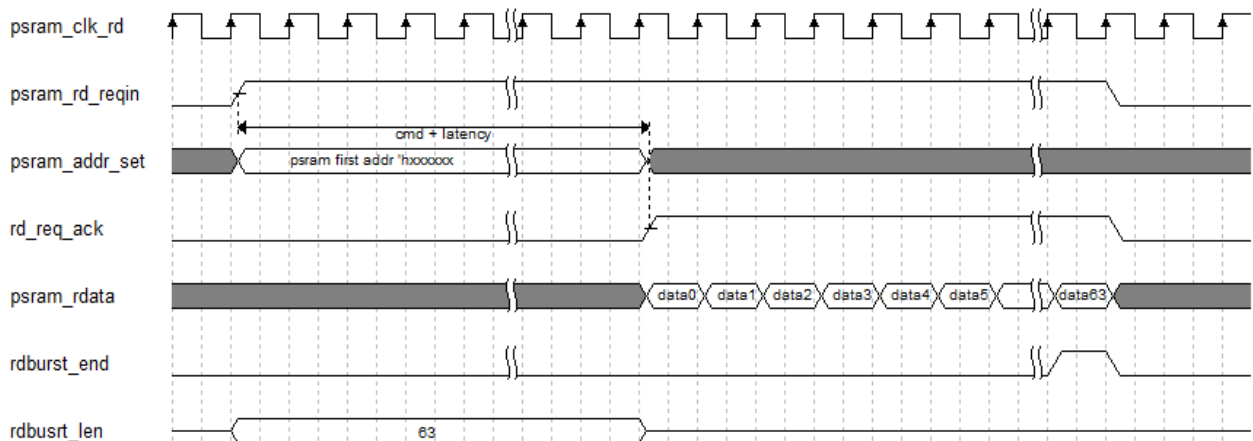


Figure 3-9 pSRAM Read operation timing

3.4.3 LOCAL BUS Incomplete Write Operation

The pSRAM controller can terminate it early during the write operation, as shown in **Figure 3-10**.

When writing the fourth 16-bits, the burst_end signal is asserted for a clock cycle, and the controller will directly terminate the write operation after receiving the signal.

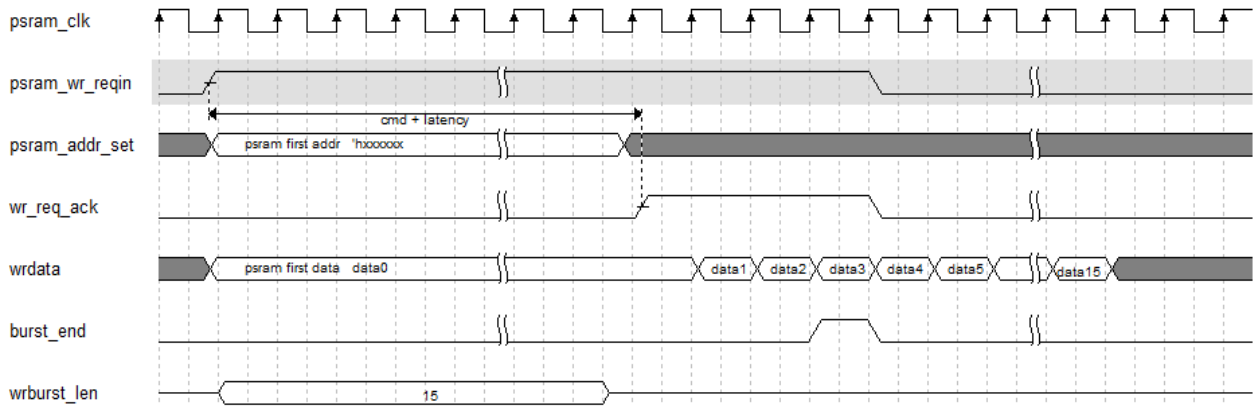


Figure 3-10 pSRAM incomplete write operation timing

3.4.4 LOCAL BUS Incomplete Read Operation

The pSRAM controller can terminate it early during the read operation, as shown in **Figure 3-11**.

When reading the fourth 16-bits data, assert the burst_end signal for one clock cycle, and the controller will directly terminate the write operation after receiving the signal.

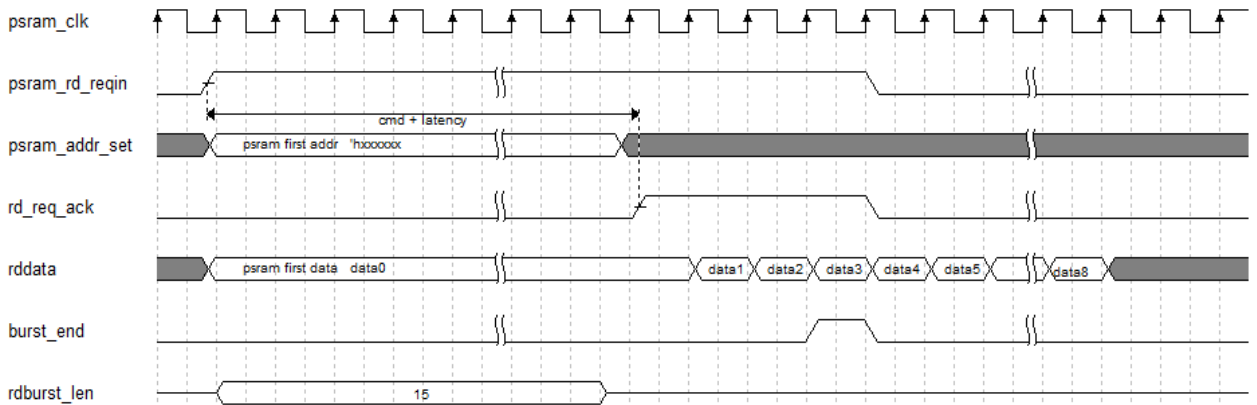


Figure 3-11 pSRAM incomplete read operation timing

3.5 Read and Write Efficiency

3.5.1 Single Latency Mode (Variable Latency)

The pSRAM CR0 configuration value is 0x8f24.

Write operation calculation: continuously assert the pSRAM_wr_reqin signal. The total cycle count is the number of clock cycles between two consecutive rising edges of ack, which totals 78 clock cycles, among which valid data occupies 64 clock cycles.

Read operation calculation: continuously assert the pSRAM_rd_reqin signal. The total cycle count is the number of clock cycles between two consecutive rising edges of ack, which totals 79 clock cycles, among which valid data occupies 64 clock cycles.

Theoretical write efficiency at 200 MHz main frequency and 128 burst length

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

Theoretical read efficiency at 200 MHz main frequency and 128 burst length

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

Theoretical write efficiency at 166 MHz main frequency and 128 burst length

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

Theoretical read efficiency at 166 MHz main frequency and 128 burst length

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

NOTE

In the actual operation, the actual efficiency is slightly lower than the value calculated above.

3.5.2 Double Latency Mode

The pSRAM CR0 configuration value is 0x8f2c.

Write operation calculation: continuously assert the pSRAM_wr_reqin signal. The total cycle count is the number of clock cycles between two consecutive rising edges of ack, which totals 85 clock cycles, among which valid data occupies 64 clock cycles.

Read operation calculation: continuously assert the pSRAM_rd_reqin signal. The total cycle count is the number of clock cycles between two consecutive rising edges of ack, which totals 86 clock cycles, among which valid data occupies 64 clock cycles.

Theoretical write efficiency at 200 MHz main frequency and 128 burst length

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

Theoretical read efficiency at 200 MHz main frequency and 128 burst length

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

Theoretical write efficiency at 166 MHz main frequency and 128 burst length

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

Theoretical read efficiency at 166 MHz main frequency and 128 burst length

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

NOTE In the actual operation, the actual efficiency is slightly lower than the value calculated above.

3.6 Constraint the pSRAM Controller IP

There is a constrains folder in the IP core pSRAM directory, and IP WIZARD automatically generates constraint files in the constrains folder in the project directory for user reference. This constraint file is only for reference and is not added in the project.

The pSRAM controller is working in higher frequency. There is the cross-clock domain logic in IP, so you need to follow the constraint rules, otherwise the compilation results may be work wrong, or the results may be unstable.

3.6.1 Examples of the Clock Constrains

```
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}]
```

Where T_clka is the pSRAM clock period (fast clock), and T_clkb is the initialized clock period (slow clock).

3.6.2 Clock Delay Constraints

```
#psram clock latency constrains

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

Clk_pSRAM_90 and Clk_pSRAM come out of the same PLL. When frequency is great than 133MHz, phase shift of 90 degrees relative to the Clk_pSRAM.

NOTE When frequency is below 133MHz, this constraint is not need.

3.6.3 Cross Clock Domain Constraints

```
#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 *}]
```

Clock domain crossing signals exist between Output #4 and Output #1 of the controller's internal PLL, requiring relaxation of timing constraints here.

3.7 The Generation of the pSRAM Controller IP

Starting with the new version since FX_2025.1.0, the IP wizard supports pSRAM instantiation.

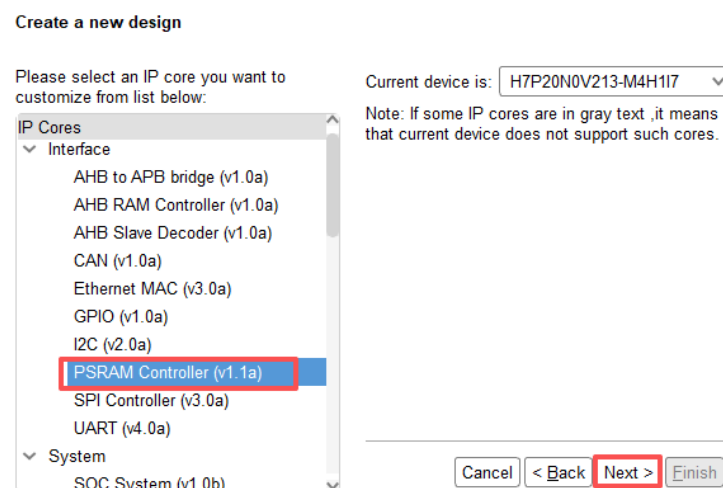


Figure 3-12 IP wizard Select the pSRAM Controller IP

3.7.1 IP wizard Parameter Definition

Create a new IP for pSRAM controller with specific parameter interface settings as follows:

Page 1 Parameters:

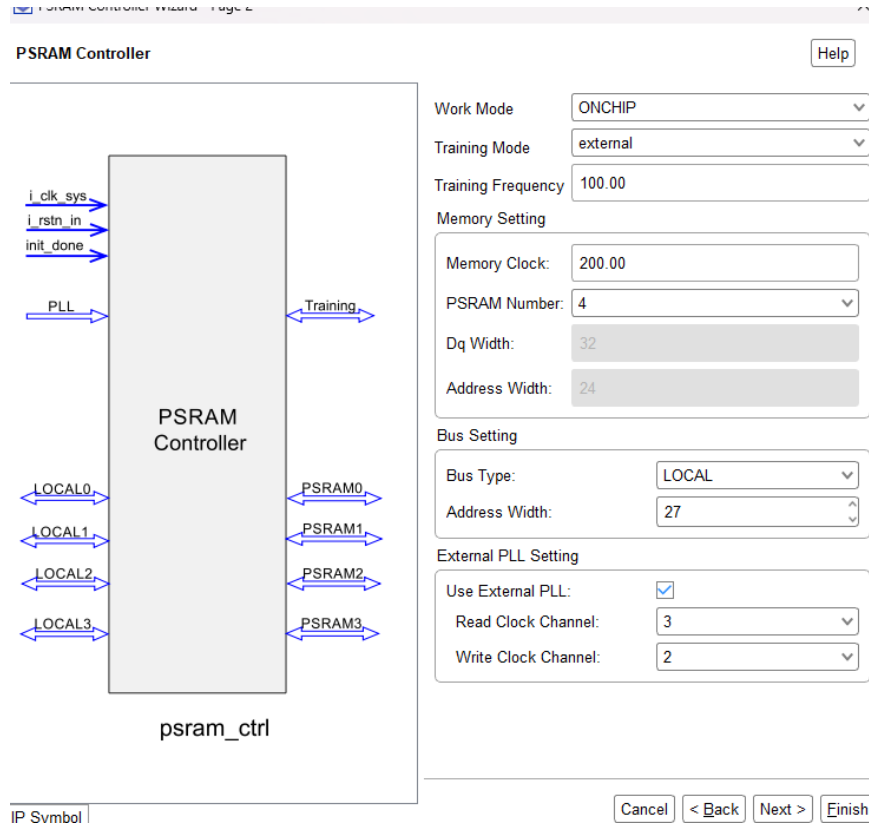


Figure 3-13 pSRAM Controller IP Wizard page1

- ❑ **Work Mode:** Choose ONCHIP or CUSTOM. In ONCHIP mode, the current chip part number is automatically matched with the internal sealing pSRAM model (the corresponding Memory Type). CUSTOM Mode is generally the user to choose the external pSRAM, this mode can be configured with more parameters.
- ❑ **Training MODE:** You can choose the internal logic training, or the external AHB (usually connected to the SoC) to do the training. Note: internal training mode only can init pSRAM register and do read calibration.
- ❑ **Training Frequency:** Corresponding to the main frequency of the training clock, the software uses this frequency to generate the correct timing constraint file (but the file will not be brought into the project, it's for reference only).
- ❑ **Memory Clock:** Select the frequency of the pSRAM operation, and users can enter the frequency by themselves. 200 MHz recommended.
- ❑ **pSRAM Number:** Set the number of pSRAM supported by the controller. H7 series: currently supports up to 2 pSRAMs; P2 series: currently supports up to 4 pSRAMs.

- ☐ Dq Width: pSRAM DQ line bit width (automatically filling in ONCHIP mode).
- ☐ Address Width: pSRAM address bit width (automatically filling in ONCHIP mode).
- ☐ Use External PLL: Use to configure whether to use the controller IP internal PLL. Internal PLL parameters have been set, H7 is 20 MHz input and 166 MHz output, P2 is 20 MHz input and 200 MHz output. If the frequency does not meet your needs, you can choose external PLL mode. The benefit of using the internal PLL is that the controller interface is simple and the clock reset logic is done internally.
 - In this mode there are the Read Clock Channel and Write Clock Channel options, corresponding to the external PLL output channel. If there are some errors, system will not work.
 - In external PLL mode, ensure training frequency is the actual value. UI will generate constraints file by filling this value.

Page 2 Parameters:

PSRAM Option

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

Figure 3-14 pSRAM Controller internal training IP Wizard page2

PSRAM Option

Burst Mode:	64	▼
Initial Latency:	7	▼

Figure 3-15 pSRAM external training IP Wizard page2

- ☐ Burst Mode: Set the maximum user interface Burst length (clock cycle), select according to user requirements. The supported length ranges of the pSRAM device are in the drop-down list.

- ❑ Fixed Latency: Set a fixed 2x Latency or variable latency. It can only be set in internal training mode and when the pSRAM num is 1. In external training mode, it is set by the external MCU. When the pSRAM num is greater than 1, variable latency mode is not supported.
- ❑ Initial Latency: Set the delay clock period.
- ❑ HyBrid Burst: When the pSRAM device supports this mode, the hybrid burst can be configured to increase the read and write efficiency.
- ❑ PASR: Internal refresh mode. default is 0 for all refresh.
- ❑ Drive Strength: Drive strength selection.

3.8 Resource Performance Report

3.8.1 System Design Block Diagram

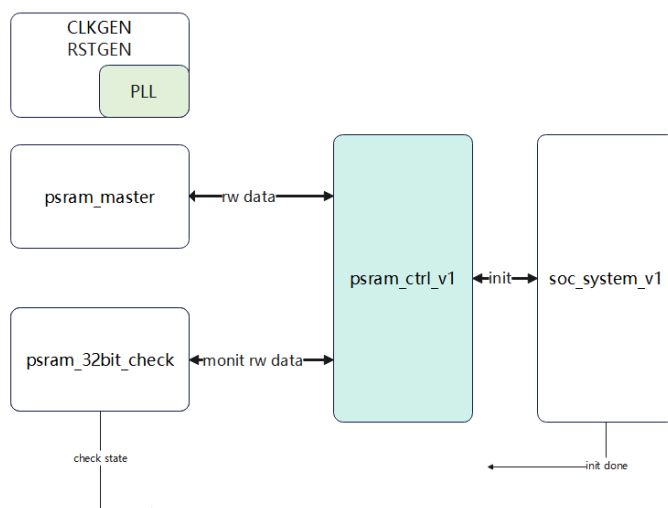


Figure 3-16 Block diagram of the pSRAM controller IP test system

3.8.2 Resource Report

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

Working Mode	LUT	REG	BRAM	PLL
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 psram_num=2 external training DQ_WIDTH =16	528	959	0	0

3.8.3 Performance Report

Devices	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 Simulation

(1) The IP generation has three files in the sim directory:

- Simulation Top Level: hme_ip_psram_ctrl_v1_tb.v
- Simulation model for the controller IP: hme_ip_psram_ctrl_v1.vp
- pSRAM Simulation model: W958D8NKY.v

(2) The configuration in Fuxi is as follows:

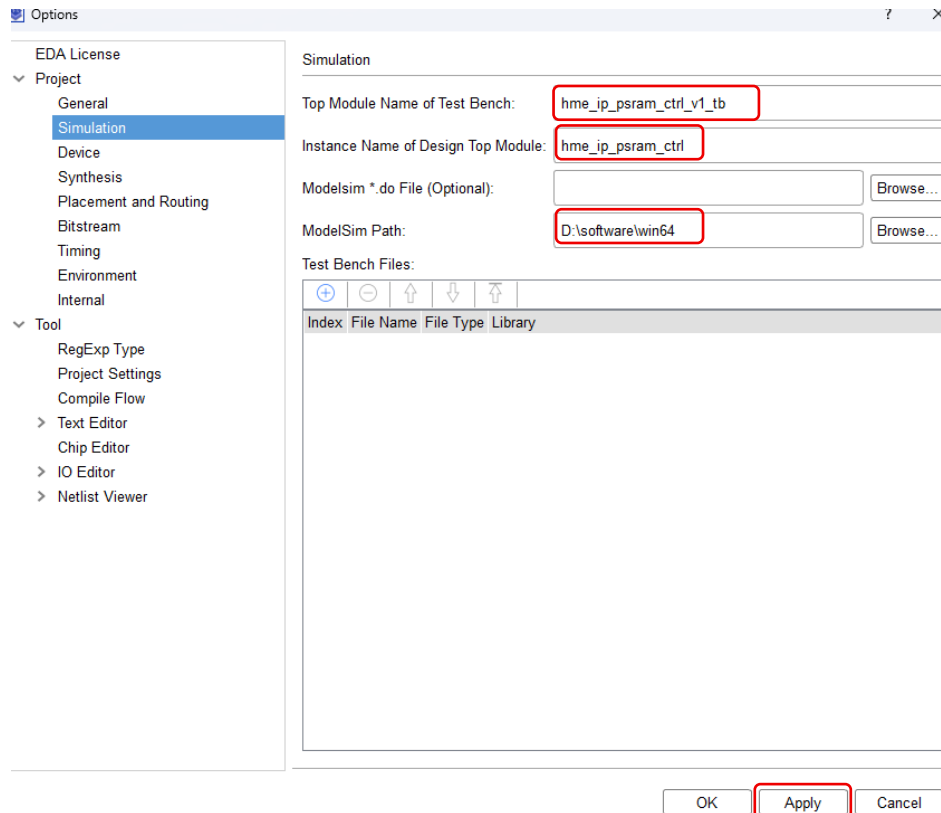


Figure 3-17 Simulation configuration

- (3) Click on **Apply** or **OK** after setting up.
- (4) Then run the simulation in the UI interface.

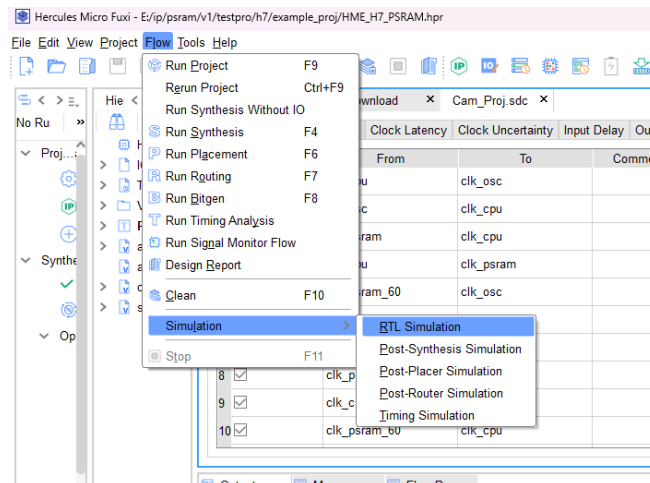


Figure 3-18 Running the simulation interface

- (5) Check the simulation results.

```
tb.u_psrpm: time 836476.000 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h0000000e, data out: 'h00
tb.u_psrpm: time 836479.358 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h0000003e, data out: 'hle
tb.u_psrpm: time 836481.858 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h0000003f, data out: 'h00
tb.u_psrpm: time 836484.358 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h0000003f, data out: 'hlf
tb.u_psrpm: time 836486.858 ns:DIE0-INFO: memory read task at clk pos, addr in: 'h00000040, data out: 'hxx
tb.u_psrpm: time 836489.358 ns:DIE0-INFO: memory read task at clk neg, addr in: 'h00000040, 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
```

Figure 3-19 Results in Modelsim

Normally after run simulation, it will print the word test pass.

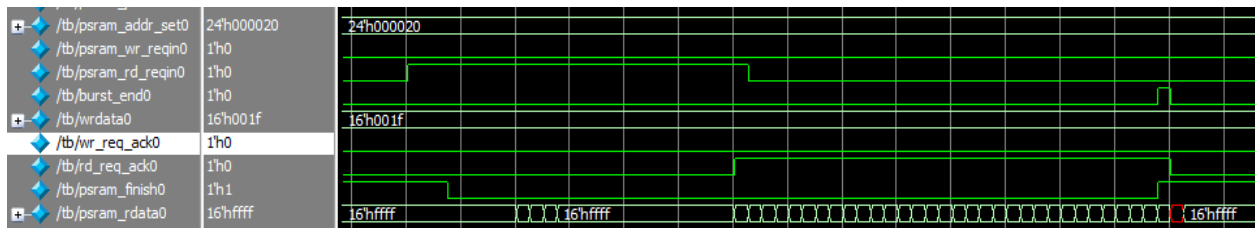


Figure 3-20 Simulation waveform

The figure is the read data waveform, and you can see the timing waveform of each signal.

3.10 Issues Summary

- (1) If there are multiple pieces of pSRAM, you must use fix latency mode. Otherwise, the internal automatic refresh time of each piece will lead to different latency at the same time, which will increase the complexity of the interface design. As shown in the figure.

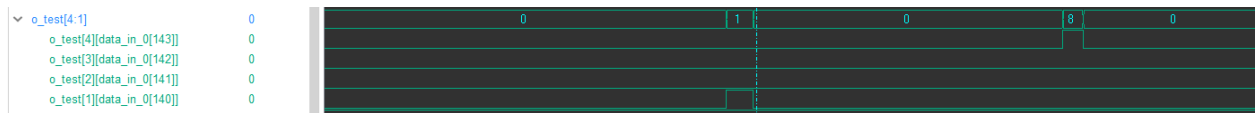


Figure 3-21 pSRAM number is 4 and variable latency mode

RTL code is: `o_test [4:1] = {lscy_mode3, lscy_mode2, lscy_mode1, lscy_mode0}`

As shown in the figure above, the `lscy_mode0` is pulled up first, indicating that only the pSRAM0 requires the double latency for this operation.

- (2) pSRAM Drive current problem

Depending on the hardware design, the driving current strength of the FPGA IO may affect the performance of the design. If the pSRAM test fails, you can try change the drive current.

- (3) pSRAM internal calibration failure

- Verify whether the pSRAM clock channel of the PLL matches the configuration during pSRAM IP generation. The rule is PLL OUT0~6 correspond to Channel 1~7.
- Internal training is only used for initialization without clock phase calibration. Open the auto-

generated file psram_ctrl_v1.v and set TR_MODE to "internal" and CLIB_EN = 0.

- Internal PLL mode can also be used for training calibration at this time, which avoids channel configuration errors.

4 Generate File Directory Structure

The pSRAM controller IP wizard for H7 and P2 generated files include: source files (src), simulation files (sim), example design files and related documents. The generated file directory structure is as below.

Path	Description
src	
src/p2_h7	IP Core file, used for synthesis
firmware	
firmware/p2_h7	.c and .h files of MCU firmware
constrains	
constrains/p2_h7/	sdc timing constraint files
sim	
sim/p2_h7/src	IP core simulation encrypted files
sim/p2_h7/hme_ip_psram_ctrl_v1_tb.v	Testbench top
sim/	Simulation models
doc	
doc/HME-pSRAM_IP_UserGuide.pdf	IP usage guide and reference design user guide
doc/HME_pSRAM_IP_Example_UserGuide.pdf	
example	
example/psram_example_h7.zip	Reference design, including the MCU and FPGA
example/psram_example_p2.zip	