Logos Family FPGAs Dedicated RAM Module (DRM) User Guide

(UG020002, V1.4)

(13.08.2022)

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.4	13.08.2022	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
GTP	Generic Technology Primitive
DRM	Dedicated RAM Module
SP	Single Port
SDP	Simple Dual Port
DP	True Dual Port
FIFO	First In First Out
NW	Normal-Write
RBW	Read-before-Write
TW	Transparent-Write
OR	Output Register

Related Documentation

The following documentation is related to this manual:

1. ***UG020007_Logos Family Products GTP User Guide***
2. ***Logos Family DRM RAM/FIFO IP User Guide***

Table of Contents

Tables

Figures

Chapter 1 Overview

Logos DRM functions a storage unit of up to 18K bits and its capacity can be independently configured as either two 9K bits or one 18K bits unit. Each DRM supports DP (True Dual Port) RAM mode, and can also be configured as SP (Single Port) RAM mode, SDP (Simple Dual Port) RAM mode, ROM mode, as well as synchronous/asynchronous FIFO mode with optional Rewrite/Resend. DRM resources also support Input Register (IR), Output Register (OR), and Core Latch, allowing superior performance when DRMs are cascaded. The total number of DRMs depends on the type of Logos family device.

The port data width of DP RAM goes up to 18 bits, with its two ports being completely independent apart from sharing RAM content, and supporting different clock domains; the port data width of SDP RAM goes up to 36 bits, with its two ports also supporting different clock domains, but one port is limited to write operations and the other is limited to read operations. With a small amount of external CLM resources, Logos SDP RAM can also be extended into a FIFO. In ROM mode, the port data width of the ROM can be up to 36 bits. The content of the DRM is typically initialized during the process of downloading configuration data. Of course, the content of the DRM can also be initialized using programmed configurations in other modes. In synchronous or asynchronous FIFO mode, one port is dedicated to FIFO data writing, and the other port is dedicated to FIFO data reading; read and write ports can support different clocks, but do not support mixed bit widths or byte enable write operations. FIFO mode supports an optional Rewrite/Resend mode in both synchronous and asynchronous modes.

The embedded DP RAM, SP RAM, SDP RAM, ROM, and synchronous/asynchronous FIFO modules with optional Rewrite/Resend can be conveniently generated through the IP Compiler tool embedded in the Pango Design Suite software by Shenzhen Pango Microsystems Co., Ltd.

1.1 Features

The DRM features of the Logos family FPGAs are shown in the table below.

Table 1-1 List of Features

Functions	Description
Memory Capacity	18K (or two 9K)
SP RAM Mode	Up to 36-bit data width supported
DP RAM Mode	Up to 18-bit data width supported
SDP RAM Mode	Up to 36-bit data width supported
ROM Mode	Up to 36-bit data width supported
FIFO Mode	Up to 36-bit data width supported
Normal Write Mode	Supported
Transparent Write Mode	Supported
Read-before Write Mode	Supported
Byte Write Enable	Supported
Input Register	Supported
Output Register	Supported (bypassable)
Address Input Hold	Supported
Synchronous/Asynchronous FIFO Control	Supported
Rewrite Function of FIFO Write Port	Supported
Resend Function of FIFO Read Port	Supported
Programmable ALMOST_FULL/ALMOST_EMPTY indicators	Supported

Note: Under SP mode, 32/36 bits write mode is prohibited from being set to TW or RBW modes, and should be set to the default NW mode, see 4.4 for details.

1.2 Resources

Table 1-2 Logos Family DRM Resource Scale

Device	Resource Quantity/18K
PGL12G	30
PGL22G	48
PGL22GS	48
PGL25G	60
PGL50G/PGL50H	134
PGL100H	286

1.3 Supported Configurations

Both DP RAM and SDP RAM support 9K and 18K modes. In DP RAM mode, DRMhas a maximum data width of 18 bits, and both A and B ports in the DRM module can independently perform read and write operations, each supporting an independent clock. In SDP RAM mode, the DRM data width increases to 36 bits, with one of the A and B ports dedicated to data writing and the other for data reading, both read and write ports also support independent clocks.

The port of GTP_DRM and GTP_FIFO supports two types of data widths: one is 2^N (including 1/2/4/8/16/32bit); the other is 9×2^N (including 9/18/36bit).

GTP_DRM's DP RAM and SDP RAM modes support mixed bit widths, meaning the two ports can be configured with different widths. For example, an SDP RAM can be configured to 16Kx1 at the write port and 512x32 at the read port, thereby saving the serial-to-parallel conversion logic from 1 bit to 32 bits.

GTP_DRM's SP RAM, ROM, and GTP_FIFO modes do not support mixed bit widths.

In DP RAM mode, the bit widths of the two ports can be independently configured according to DRM resources. DP RAM configuration and SDP RAM configuration in 9K DRM mode, as well as DP RAM configuration and SDP RAM configuration in 18K DRM mode are described in Table 3-3, Table 4-3, Table 3-2 and Table 4-2.

In the 9K mode for both SP RAM and ROM modes, DRM contains two ports, SP RAM mode allows independent read and write operations on these two ports, while in ROM mode the port is read-only; when sharing two pairs of ports, DRM contains only one port, SP RAM mode allows read and write operations on this port, while in ROM mode the port is read-only. The SP RAM and ROM modes in 9K mode, and the SP RAM and ROM modes in 18K mode have been described in Table 5-3, Table 6-3, Table 5-2 and Table 6-2.

In FIFO mode, one port is dedicated to data writing and the other port to data reading, read and write ports can use different clocks.

Multiple DRMs can be combined into larger DP RAM, SDP RAM, SP RAM, ROM, or FIFO through cascading. For this, DRM provides an additional 3-bit address extension (CS[2:0]), commonly used for deeply-extended applications.

See appendix for details of 8.1 Address and Data Port Mapping.

Chapter 2 DRM Primitives

As the basis for various modes, Logos family FPGAs DRM primitives have three types: GTP_DRM9K, GTP_DRM18K, and GTP_FIFO18K, with their supported modes shown in the table below:

Table 2-1 Logos Family FPGAs DRM Primitives

Primitives	**Supported Mode**
GTP_DRM9K	bit widths of x1, x2, x4, x8, x16, x32 (and x9, x18, x36)
GTP_DRM18K	Bit widths of x1, x2, x4, x8, x16, x32 (and x9, x18, x36)
GTP_FIFO18K	Bit widths of x1, x2, x4, x8, x16, x32 (and x9, x18, x36)

The Logos-supported GTP primitives and their ports, parameters, and other details are as specified in the "***UG020007_Logos Family Products GTP User Guide***".

2.1 GTP_DRM9K

GTP_DRM9K has a 9K bits storage unit, supporting multiple working modes, including DP (True Dual Port) RAM, SDP (Simple Dual Port) RAM, SP (Single Port) RAM or ROM mode. DRM supports configurable data widths and supports dual-port mixed bit widths in DP RAM and SDP RAM modes.

The DP RAM module has two independent ports, A and B, both of which can perform read and write operations independently, allowing simultaneous read or write, or one port to read while the other to write. Both ports support different clocks, with the bit width set independently. Simultaneous read and write operations on the same address via both ports are not possible, as this would cause read/write conflicts.

In SDP RAM, port A is fixed as the write port and port B as the read port; both ports support different clocks, with the bit width set independently. Simultaneous read and write operations on the same address via both ports are not possible, as this would cause read/write conflicts.

In SP RAM mode, the two ports of the DRM can be operated independently.

The DRM can be configured as ROM, with ROM contents initialized through the configuration interface.

The DRM's port write operations support three modes: Normal Write mode (NW), Transparent Write mode (TW), and Read before Write mode (RBW).

All ports are active-high by default.

The block diagram of GTP_DRM9K structure is shown in the figure below.

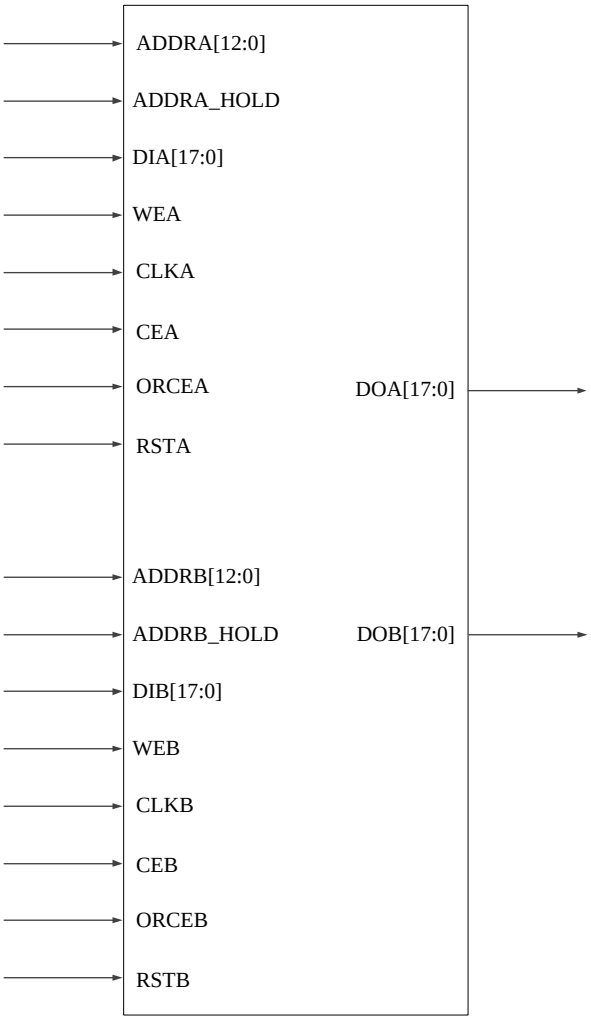

Figure 2-1 Block Diagram of GTP_DRM9K Structure

The table below lists the port names and descriptions for GTP_DRM9K.

Table 2-2 GTP_DRM9K Port List

Port	**Direction**	**Function Description**
ADDRA	Input	Port A address input bus
ADDRA_HOLD	Input	Port A address input hold
DIA	Input	Port A data input bus
WEA	Input	Port A write enable
CLKA	Input	Port A clock
CEA	Input	Port A clock enable
ORCEA	Input	Port A output register enable
RSTA	Input	Port A data register reset
DOA	Output	Port A data output bus
ADDRB	Input	Port B address input bus
ADDRB_HOLD	Input	Port B address input hold

Port	Direction	Function Description
DIB	Input	Port B data input bus
WEB	Input	Port B write enable
CLKB	Input	Port B clock
CEB	Input	Port B clock enable
ORCEB	Input	Port B output register enable
RSTB	Input	Port B data register reset
DOB	Output	Port B data output bus

The table below lists the parameters and descriptions for GTP_DRM9K.

Table 2-3 GTP_DRM9K Parameter List

Parameter	Description	Setting Value
DATA_WIDTH_A	Data width of Port A	1, 2, 4, 8, 16, 32, 9, 18, 36
DATA_WIDTH_B	Data width of Port B	1, 2, 4, 8, 16, 32, 9, 18, 36
WRITE_MODE_A	Port A write mode	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
WRITE_MODE_B	Port B write mode	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
DOA_REG	Port A output register	0 = Do not enable output register 1 = Enable output register
DOB_REG	Port B output register	0 = Do not enable output register 1 = Enable output register
RST_TYPE	Reset mode selection	"SYNC": Synchronous reset "ASYNC": Asynchronous reset "ASYNC_SYNC_RELEASE": Asynchronous reset with synchronous release
RAM_MODE	RAM mode selection	"TRUE_DUAL_PORT": DP RAM "SIMPLE_DUAL_PORT": SDP RAM "SINGLE_PORT": SP RAM "ROM": ROM
GRS_EN	Global reset enable signal (Internal Chip)	"FALSE": Global Reset not enabled; "TRUE": Global Reset enabled.
DOA_REG_CLKINV	Port A output register clock inversion	0 = Clock not inverted 1 = Clock inverted
DOB_REG_CLKINV	Port B output register clock inversion	0 = Clock not inverted 1 = Clock inverted
INIT_00 INIT_01 INIT_02 ... INIT_1F	RAM initialization configuration parameters; see Appendix Initialization Configuration Parameter Mapping for details	0 ~2^288-1
INIT_FILE	Initialization file path	"NONE": No initialization file is specified, and the initialization data will be the value set by the parameter INIT_XX; "XXX": XXX represents the specific initialization file path, which overrides the parameter INIT_XX as the initial value for DRM
BLOCK_X	Data cascade coordinates when	Depends on the cascaded DRM coordinates

Parameter	Description	Setting Value
	DRM9K is cascaded, used to map the same initialization file to different cascaded DRMs	
BLOCK_Y	Address cascade coordinates when DRM9K is cascaded, used to map the same initialization file to different cascaded DRMs	Depends on the cascaded DRM coordinates
RAM_DATA_WIDTH	Maximum data width after DRM cascading, used to map the same initialization file to different cascaded DRMs	Depends on the number of cascaded DRMs
RAM_ADDR_WIDTH	Maximum address width after DRM cascading, used to map the same initialization file to different cascaded DRMs	Depends on the number of cascaded DRMs
INIT_FORMAT	Initialization file format	"BIN": Binary "HEX": Hexadecimal

Note: SDP mode does not support write mode settings. When configuring DRM to SDP mode using GTP, cannot manually modify the parameter WRITE_MODE_A/B of write mode.

2.2 GTP_DRM18K

GTP_DRM18K has a 18K bits storage unit, supporting multiple working modes, including DP (True Dual Port) RAM, SDP (Simple Dual Port) RAM, SP (Single Port) RAM or ROM mode. DRM supports configurable data widths and supports dual-port mixed bit widths in DP RAM and SDP RAM modes.

The DP RAM module has two independent ports, A and B, both of which can perform read and write operations independently, allowing simultaneous read or write, or one port to read while the other to write. Both ports support different clocks, with the bit width set independently. Simultaneous read and write operations on the same address via both ports are not possible, as this would cause read/write conflicts.

In SDP RAM, port A is fixed as the write port and port B as the read port; both ports support different clocks, with the bit width set independently. Simultaneous read and write operations on the same address via both ports are not possible, as this would cause read/write conflicts.

In SP RAM mode, the two ports of the DRM can be operated independently. Simultaneous read and write operations on the same address via both ports are not possible, as this would cause read/write conflicts.

The DRM can be configured as ROM, with ROM contents initialized through the configuration interface.

The DRM's port write operations support three modes: Normal Write mode (NW), Transparent Write mode (TW), and Read before Write mode (RBW).

All ports are active-high by default.

The block diagram of GTP_DRM18K structure is shown in the figure below.

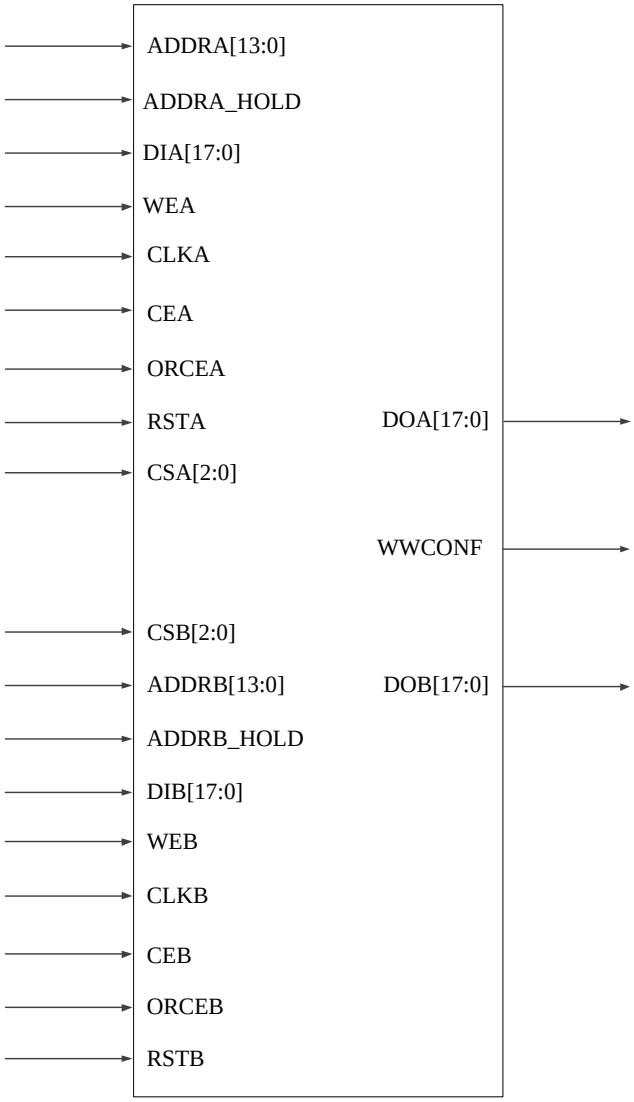

Figure 2-2 Block Diagram of GTP_DRM18K Structure

The table below lists the port names and descriptions for GTP_DRM18K.

Table 2-4 GTP_DRM18K Port List

Port	**Direction**	**Function Description**
ADDRA	Input	Port A address input bus
ADDRA_HOLD	Input	Port A address input hold
DIA	Input	Port A data input bus
CSA	Input	Port A address extension
WEA	Input	Port A write enable
CLKA	Input	Port A clock
CEA	Input	Port A clock enable
ORCEA	Input	Port A output register enable

Port	Direction	Function Description
RSTA	Input	Port A data register reset
DOA	Output	Port A data output bus
ADDRB	Input	Port B address input bus
ADDRB_HOLD	Input	Port B address input hold
DIB	Input	Port B data input bus
CSB	Input	Port B address extension
WEB	Input	Port B write enable
CLKB	Input	Port B clock
CEB	Input	Port B clock enable
ORCEB	Input	Port B output register enable
RSTB	Input	Port B data register reset
DOB	Output	Port B data output bus
WWCONF	Output	Write collision (this port is invalid in the Logos Family Products)

The table below lists the parameters and descriptions for GTP_DRM18K.

Table 2-5 GTP_DRM18K Parameter List

Parameter	Description	Setting Value
CSA_MASK[2:0]	Port A address extension control signal	0 ~ 7
CSB_MASK[2:0]	Port B address extension control signal	0 ~ 7
DATA_WIDTH_A	Maximum data width of Port A	1, 2, 4, 8, 16, 32, 9, 18, 36
DATA_WIDTH_B	Maximum data width of Port B	1, 2, 4, 8, 16, 32, 9, 18, 36
WRITE_MODE_A	Port A write mode	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
WRITE_MODE_B	Port B write mode	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
DOA_REG	Port A output register	0 = Do not enable output register 1 = Enable output register
DOB_REG	Port B output register	0 = Do not enable output register 1 = Enable output register
RST_TYPE	Reset mode selection	"SYNC": Synchronous reset "ASYNC": Asynchronous reset "ASYNC_SYNC_RELEASE": Asynchronous reset with synchronous release
RAM_MODE	RAM mode selection	"TRUE_DUAL_PORT": DP RAM "SIMPLE_DUAL_PORT": SDP RAM "SINGLE_PORT": SP RAM "ROM": ROM
WRITE_COLLISION_ARBITER	Write collision arbitration	"NULL": No arbitration
GRS_EN	Global reset enable signal (Internal Chip)	"FALSE": Global Reset not enabled; "TRUE": Global Reset enabled.
DOA_REG_CLKINV	Port A output register clock inversion	0 = Clock not inverted 1 = Clock inverted

Parameter	Description	Setting Value
DOB_REG_CLKINV	Port B output register clock inversion	0 = Clock not inverted 1 = Clock inverted
INIT_00 INIT_01 INIT_02 ... INIT_3F	RAM initialization configuration parameters; see Appendix Initialization Configuration Parameter Mapping for details	0 ~2^288-1
INIT_FILE	Initialization file path	"NONE": No initialization file is specified, and the initialization data will be the value set by the parameter INIT_XX; "XXX": XXX represents the specific initialization file path, which overrides the parameter INIT_XX as the initial value for DRM
BLOCK_X	Data cascade coordinates when DRM18K is cascaded, used to map the same initialization file to different cascaded DRMs	Depends on the cascaded DRM coordinates
BLOCK_Y	Address cascade coordinates when DRM18K is cascaded, used to map the same initialization file to different cascaded DRMs	Depends on the cascaded DRM coordinates
RAM_DATA_WIDTH	Maximum data width after DRM cascading, used to map the same initialization file to different cascaded DRMs	Depends on the number of cascaded DRMs
RAM_ADDR_WIDTH	Maximum address width after DRM cascading, used to map the same initialization file to different cascaded DRMs	Depends on the number of cascaded DRMs
INIT_FORMAT	Initialization file format	"BIN": Binary "HEX": Hexadecimal

Note:

1. SDP mode does not support write mode settings. When configuring DRM to SDP mode using GTP, cannot manually modify the write mode parameter WRITE_MODE_A/B.

2. Logos does not support write collision arbitration, the parameter WRITE_COLLISION_ARBITER can only be set to "NULL", meaning no write collision arbitration is performed.

2.3 GTP_FIFO18K

DRM_FIFO supports 18K bits storage cells, configured in data width of 16Kx1, 8Kx2, 4Kx4, 2Kx9(8), 1Kx18(16), 512x36(32). When DRM is configured as FIFO mode, it is set to Simple Dual Port mode, and does not support mixed bit widths or bytes enable write operations.

DRM_FIFO can configure DRM into synchronous FIFO or asynchronous FIFO modes. In DRM SDP RAM mode, one port is dedicated to FIFO data writing and another port is dedicated to FIFO data reading, read and write ports can use different clocks. DRM_FIFO supports an optional Rewrite/Resend mode in both synchronous and asynchronous modes. The block diagram of GTP_DRM18K structure is shown in the figure below.

DRM_FIFO supports register mode, which is used in the same way as the GTP_DRM register mode.

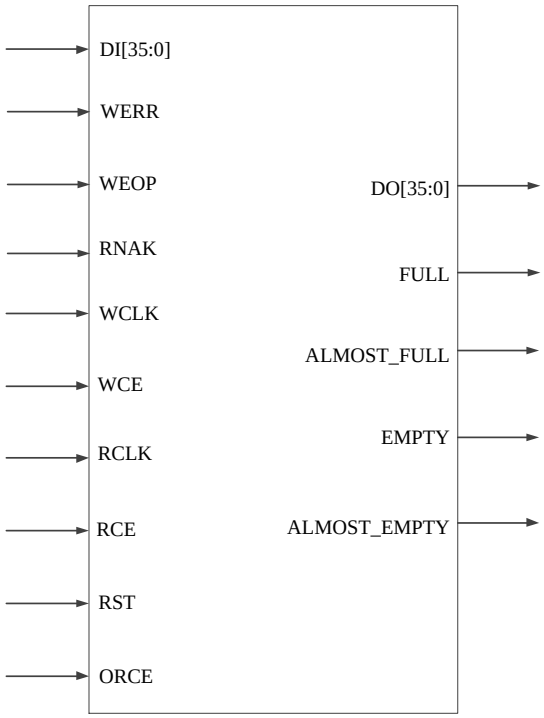

Figure 2-3 Block Diagram GTP_FIFO18K Structure

The table below lists the port names and descriptions for GTP_FIFO18K.

Table 2-6 GTP_FIFO18K Port List

Port	**Direction**	**Function Description**
DI	Input	Data input
WERR	Input	Write port error packet indicator
WEOP	Input	End-of-packet signal indicator
RNAK	Input	Read port resend indicator
WCLK	Input	Write clock signal
RCLK	Input	Read clock signal
WCE	Input	Write enable signal
RCE	Input	Read enable signal
RST	Input	Reset signal
ORCE	Input	Output register enable signal
DO	Output	Data output
EMPTY	Output	Read port empty flag
FULL	Output	Write port full flag
ALMOST_EMPTY	Output	Read port almost empty flag
ALMOST_FULL	Output	Write port almost full flag

The table below lists the parameters and descriptions for GTP_FIFO18K.

Table 2-7 GTP_FIFO18K Parameter List

Parameter	Function Description	Setting Value
GRS_EN	Global reset enable signal (Internal Chip)	"TRUE": Global Reset enabled "FALSE": Global Reset not enabled
ALMOST_FULL_OFFSET	When the FIFO is almost full and the difference between the write and read pointers is greater than or equal to ALMOST_FULL_OFFSET, the almost_full flag is set to 1.	16382: 16K address 8190: 8K address 4094: 4K address 2046: 2K address 1022: 1K address 510: 512 address
ALMOST_EMPTY_OFFSET	When the FIFO is almost empty and the difference between the read and write pointers is less than or equal to ALMOST_EMPTY_OFFSET, the almost_empty flag is set to 1.	16382: 16K address 8190: 8K address 4094: 4K address 2046: 2K address 1022: 1K address 510: 512 address
SYNC_FIFO	Asynchronous/Synchronous FIFO Selection	"TRUE": Use synchronous FIFO "FALSE": Use asynchronous FIFO
DATA_WIDTH	FIFO data width	1,2,4,8,9,16,18,32,36
REWRITE_EN	Rewrite enable	"TRUE": Rewrite enable "FALSE": Rewrite not enabled
RESEND_EN	Resend enable	"TRUE": Resend enable "FALSE": Resend not enabled
USE_EMPTY	Enable read empty flag	1: Read empty flag enabled 0: Read empty flag not enabled
USE_FULL	Enable write full flag	1: Write full flag enabled 0: Write full flag not enabled
DO_REG	Output register enable	1: Enabled 0: Not enabled

Chapter 3 DP Mode

3.1 Description of Mode

The port mode of RAM is determined by the parameter RAM_MODE; when the value of RAM_MODE is "TRUE_DUAL_PORT", RAM is configured in DP mode. This document will detail the 18K DP RAM, which has essentially the same structure and functions as the 9K.

DP RAM supports:

- DP read operation
- DP write operation
- One port to read and the other port to write
- Independent bit width settings for two ports

3.2 Port Descriptions

In DP RAM mode, the 18K and 9K DP RAM have two independent ports, A and B, but sharing DRM content. Their structures are completely symmetrical. The following figure shows the structural diagram of the 18K DP RAM.

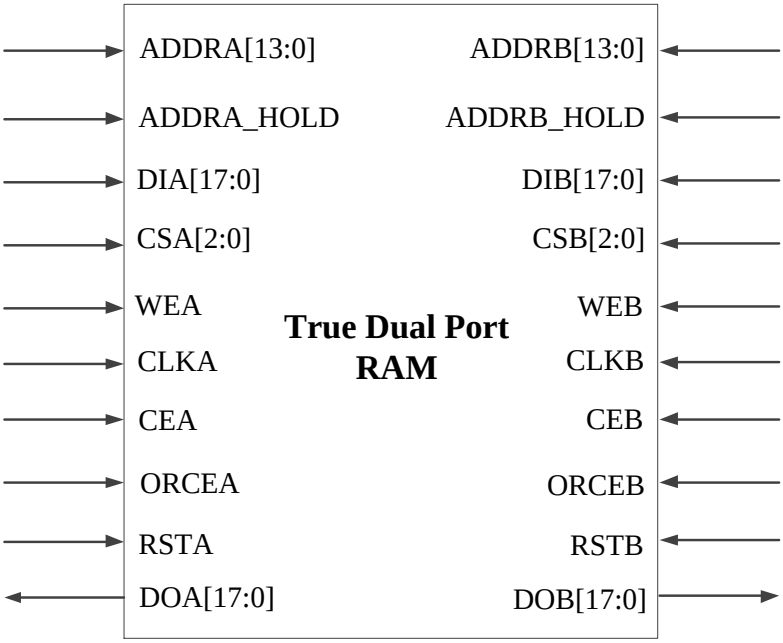

Figure 3-1 DP RAM Data Ports

The table below lists the port names and descriptions for DP RAM.

Table 3-1 DP RAM Port Naming and Description

Port	Direction	Description	Port	Direction	Description
DIA	Input	Port A data input bus	DIB	Input	Port B data input bus
ADDRA	Input	Port A address input	ADDRB	Input	Port B address input
ADDRA_HOLD	Input	Port A address hold	ADDRB_HOLD	Input	Port B address hold
WEA	Input	Port A write enable	WEB	Input	Port B write enable
CEA	Input	Port A clock enable	CEB	Input	Port B clock enable
CLKA	Input	Port A clock	CLKB	Input	Port B clock
RSTA	Input	Port A output register reset	RSTB	Input	Port B output register reset
CSA	Input	Port A address extension	CSB	Input	Port B address extension
ORCEA	Input	Port A output register enable	ORCEB	Input	Port B output register enable
DOA	Output	Port A data output bus	DOB	Output	Port B data output bus

Note:

1. For specific data widths, see the appendix Additional Information Bit for Bytes;

2. The 9K DRM has no CSA/CSB port and does not support address extension. For DRM cascading to extend address depth, it is recommended to use 18K DRM.

3.3 Bit Width Combinations

The port bit width of RAM is determined by the parameter DATA_WIDTH_A/DATA_WIDTH_B in the GTP. For example, when the value of DATA_WIDTH_A is 4, the bit width of port A is set to 4 bits. DP mode supports setting different bit widths for ports A and B.

The table below shows the allowed bit width combinations for True Dual Port RAM mode in 18K DRM mode.

Table 3-2 Allowed Bit Width Combinations for True Dual Port RAM Mode in 18K DRM Mode

			Port B						
			No Parity Bits					**With Parity Bits**	
			16Kx1	**8Kx2**	**4Kx4**	**2Kx8**	**1Kx16**	**2Kx9**	**1Kx18**
Port A	**No Parity Bits**	**16Kx1**	√	√	√	√	√		
		8Kx2	√	√	√	√	√		
		4Kx4	√	√	√	√	√		
		2Kx8	√	√	√	√	√		
		1Kx16	√	√	√	√	√		
	With Parity Bits	**2Kx9**						√	√
		1Kx18						√	√

Note: √ indicates the supported bit width combinations.

The following table shows the allowed bit width combinations for the True Dual Port RAM mode in the 9K DRM mode.

Table 3-3 Allowed Bit Width Combinations for True Dual Port RAM Mode in 9K DRM Mode

			Port B						
			No Parity Bits					**With Parity Bits**	
			8Kx1	**4Kx2**	**2Kx4**	**1Kx8**	**512x16**	**1Kx9**	**512x18**
Port A	**No Parity Bits**	**8Kx1**	√	√	√	√	√		
		4Kx2	√	√	√	√	√		
		2Kx4	√	√	√	√	√		
		1Kx8	√	√	√	√	√		
		512x16	√	√	√	√	√		
	With Parity Bits	**1Kx9**						√	√
		512x18						√	√

Note: √ indicates the supported bit width combinations.

3.4 Read and Write Operations

Depending on the data output from the same port during data writing, the port write operations of DRM support three modes: Normal Write mode (NW), Transparent Write mode (TW), and Read before Write mode (RBW), with the default write operation mode set to NW. The write operation mode of the RAM is determined by the parameter WRITE_MODE_A/WRITE_MODE_B in the GTP. When the value of WRITE_MODE_A is "NORMAL_WRITE", the write operation mode for port A of the DRM is set to NW. Under different write modes, the read timing is similar.

In DP mode, there are two relatively independent ports, and performing a read and a write operation on the same address through both ports simultaneously will cause read/write conflicts. DRM prohibits both ports from writing data to the same address simultaneously and from performing a read and a write on the same address at the same time; This kind of conflict need to be avoided in practical applications by using user logic.

3.4.1 Normal Write mode

As shown in the following Normal Write mode read-write timing diagram, when a user writes data to a port of the DRM, the output data of that port is not updated at this time.

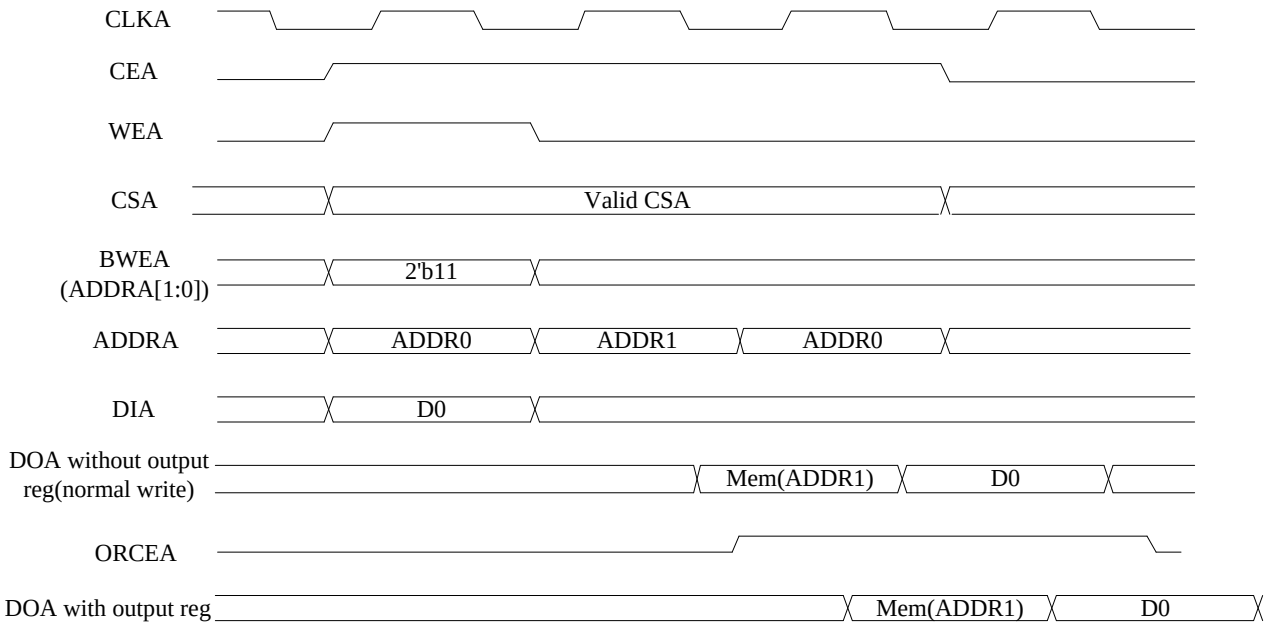

Figure 3-2 Normal Write Mode Read-Write Timing Diagram

3.4.2 Transparent Write mode

As shown in the Transparent Write mode read-write timing diagram below, when a user writes data to a port of the DRM, the written data is directly output to the output port at the same time as it is written to the RAM (i.e., the next clock cycle of the write operation).

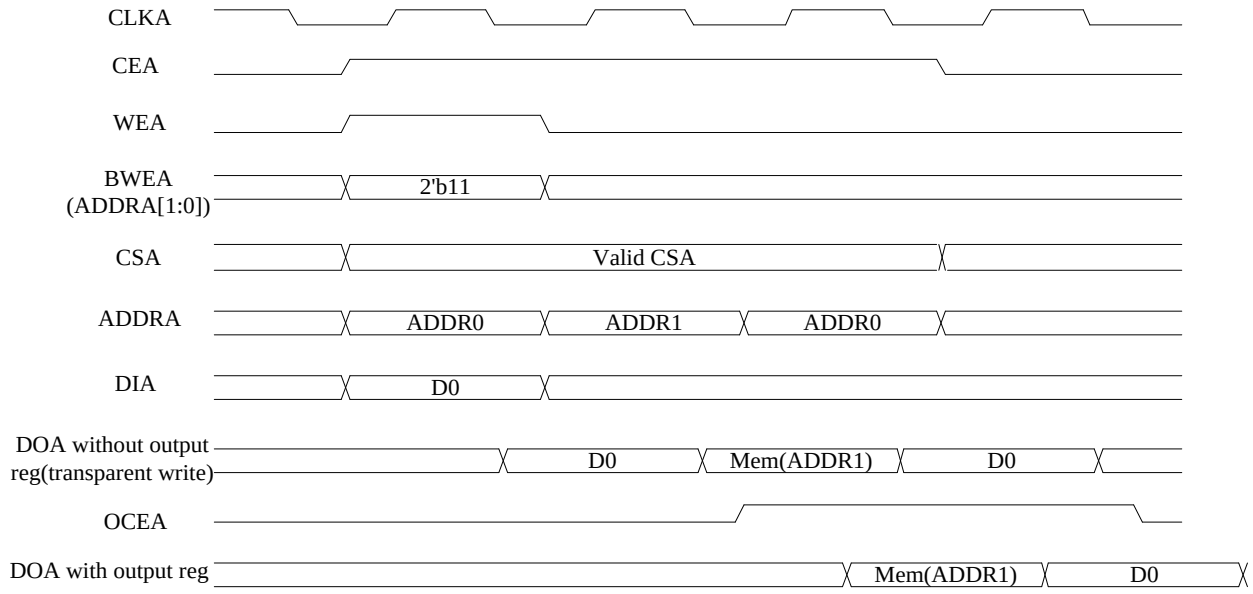

Figure 3-3 Transparent Write Mode Read-Write Timing Diagram

3.4.3 Read before Write mode

As shown in the Read before Write mode read-write timing diagram below, when a user writes data to a port of the DRM, the original data at that address is first read and then output to the output port in the next clock cycle of the write operation.

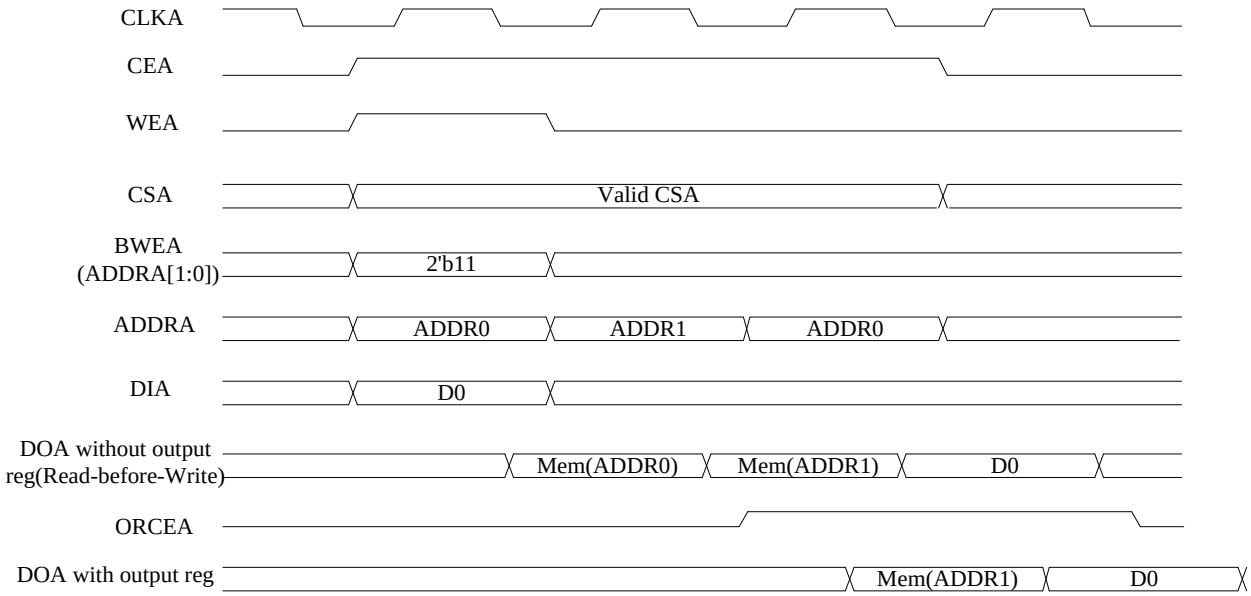

Figure 3-4 Read before Write Mode Read-Write Timing Diagram

3.5 Byte Enable

DRM supports the Byte-Write mode for write operations, which writes to selected data bytes through the BWEA/BWEB signals (active-high) while masking the writing to other bytes at the same address index. This mode is mainly used to operate on narrower data buses when the data bus width is limited. For example, operations can be performed only on 9-bit width of data on an 18-bit data bus.

In DP mode, when the port width is 2^N bits, byte enable for 16-bit write operations is supported, with each byte containing 8 bits. When the port width is 9×2^N bits, byte enable for 18-bit write operations is supported, with each byte containing 9 bits. For instance, when the data width of port A is 16 bits, BWEA contains two bits, with BWEA[1] controlling data[15:8] and BWEA[0] controlling data[7:0]; and so on. When BWEA[1:0]=2'b01, only data[7:0] is written to the current address during a valid write operation. During design, the following three aspects need to be considered:

- Ports A and B each has independent byte enable controls
  Port A: BWEA[1:0], Port B: BWEB[1:0]

- The byte enable signal is always valid in x18 bit data width mode; when not in use, this signal should be connected to a high level.
- The byte enable signal multiplexes with the lower address signal:
  BWEA[1:0] =ADDRA[1:0]
  BWEB[1:0] =ADDRB[1:0]

In DP mode, the byte enable write operation mode only supports the data port width for writing of x18(16). When the data port width for writing is x9(x8), the byte enable function is prohibited, as it duplicates the write enable function and becomes invalid.

The byte enable write operation mode can coexist with NW, TW, or RBW write operation modes. The read/write timing diagram for the DRM in byte enable mode under TW write operation mode is shown in the following figure:

Figure 3-5 Byte Enable Read/Write Timing Diagram (TW Mode)

3.6 Internal Registers

The DRM internal registers include Input Register (IR), Output Register (OR), and Internal Latch.

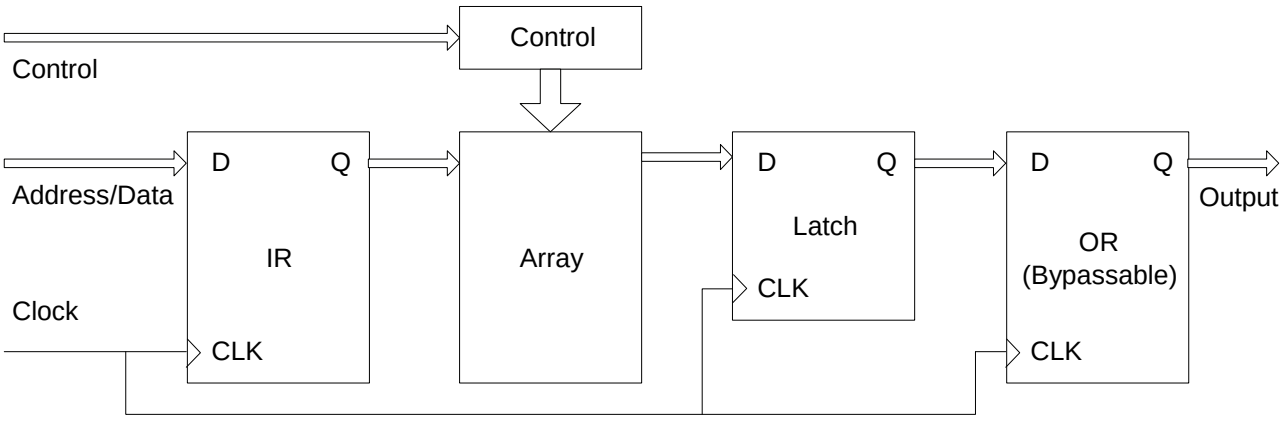

Figure 3-6 Logic Diagram of DRM Register

Port addresses, data, and some control signals can be sent to DRM via corresponding Input Registers (IR). The synchronized write operation can be implemented by IR. The IR cannot be bypassed or configured. An address hold selection pathway is built into the address input port, with its logic implementation as shown in the following figure:

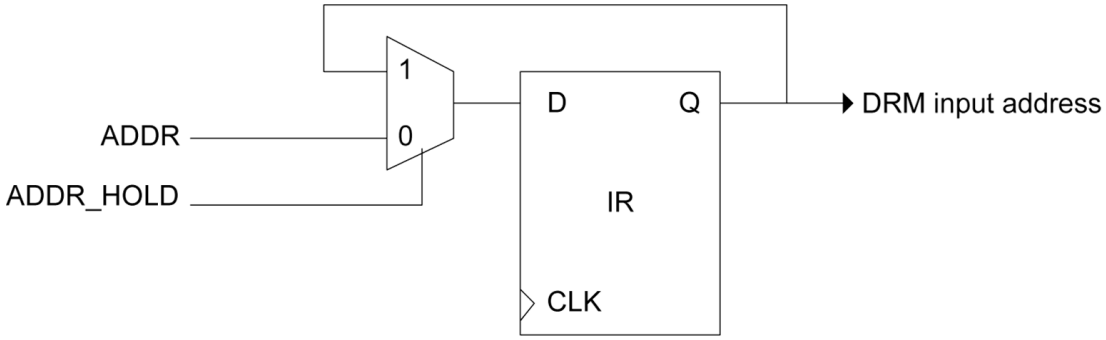

Figure 3-7 Logic Diagram of Address IR

The address hold selection pathway is controlled by the address hold signal ADDRA_HOLD. If ADDRA_HOLD remains at high level, the output address data from the input register remains unchanged, with the timing diagram as shown below:

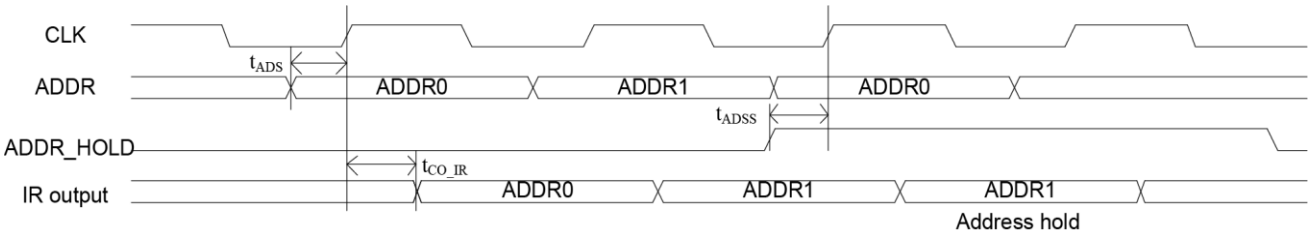

Figure 3-8 Timing Diagram of Address IR

For data output ports, DRM provides an optional Output Register (OR) for improved timing performance. Whether the output register is effective is determined by the GTP parameter DOA_REG/DOB_REG. For example, when DOA_REG is 1, port A output register is effective; when DOA_REG is 0, port A output register is bypassed. During read operations, when the output register is bypassed, DOA/DOB output is the latch output, occurring on the rising edge of the same read clock cycle. Port B is similar.

As shown in the following figure, when port A output register is effective, it can be controlled by an independent clock enable signal ORCEA. When ORCEA is at high level, the output data from the output register changes with the input on every rising edge of the clock cycle; when ORCEA is at low level, the output data from the output register remain the previous data; when the output register is bypassed, ORCEA is invalid. When ORCEA is a constant 1, the use of the output register increases the delay of port A read operation from one clock cycle to two clock cycles (Case 1); in pipeline design applications with flow control, user can also flexibly control ORCEA with logic (Case 2).

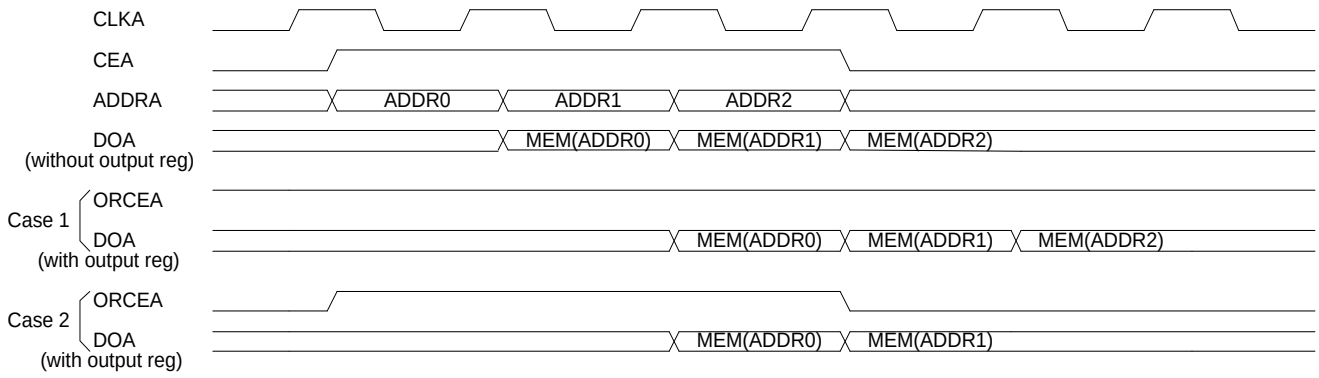

Figure 3-9 Read Timing with Output Register

3.7 Reset Signal

3.7.1 Synchronous and Asynchronous Reset

As shown in the following figure, DRM outputs (including Core Latch outputs and OR outputs) support synchronous or asynchronous reset:

1. Synchronous reset: RSTA/B is synchronized with CLKA/B, and it must satisfy the setup/hold time requirements for the clock edge;
2. Asynchronous reset: RSTA/B is independent of CLKA/B, and the output is reset at the edge of RSTA/B.

Ports A/B share the setting for synchronous/asynchronous reset.

In asynchronous reset mode, the de-assertion of reset is synchronized with the clock, with the RSTA/B signal functioning independently of the clock signal. When RSTA/B is active, it directly resets the internal latch and output register. With RSTA/B inactive, the internal latch and output register are synchronized to the clock edge to resume normal operation.

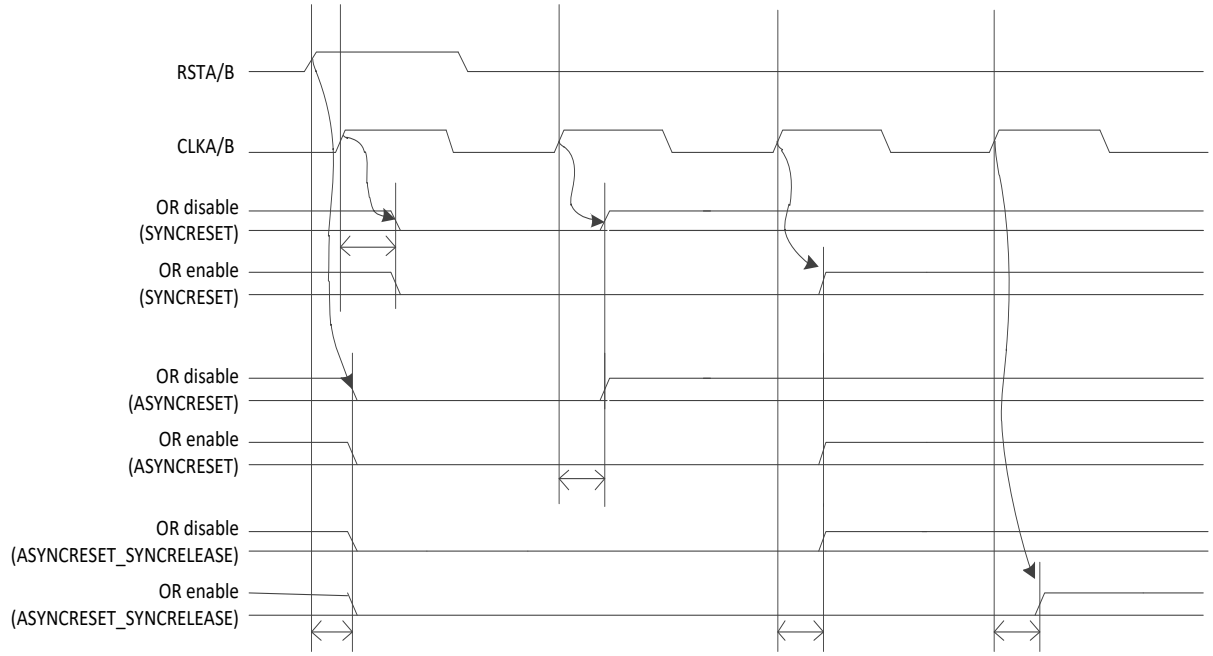

Figure 3-10 Output Register Reset Timing

3.7.2 Core Latch Reset

As shown in the following figure, in the DRM unit of the Logos family FPGAs, both ports A and B have output Latch registers. These Latches can be reset synchronously or asynchronously, controlled by the global reset GRSN or the local port reset RSTA/B.

Note: GSR_N has a reset mode of asynchronous reset. GSR_N reset can be disabled through configuring bits.

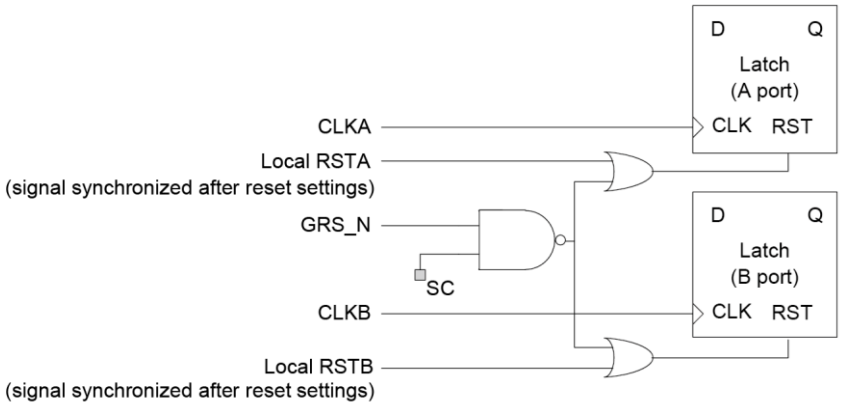

Figure 3-11 Core Reset

3.8 Global Signal Timing

As shown below, the bitstream download phase is between T0 and T1, including sc_bit and ROM initialization content. Before the moment T1, all DRM outputs are 1; after GLOGEN is released, all DRM output states depend on configuration and input.

The release times of GRS_N, GWEN, and DONE (not available in DRM) can be individually configured by the user through CCS, and can be set at T2, T3, or T4; after the DONE signal is set to 1, the device will enter user mode.

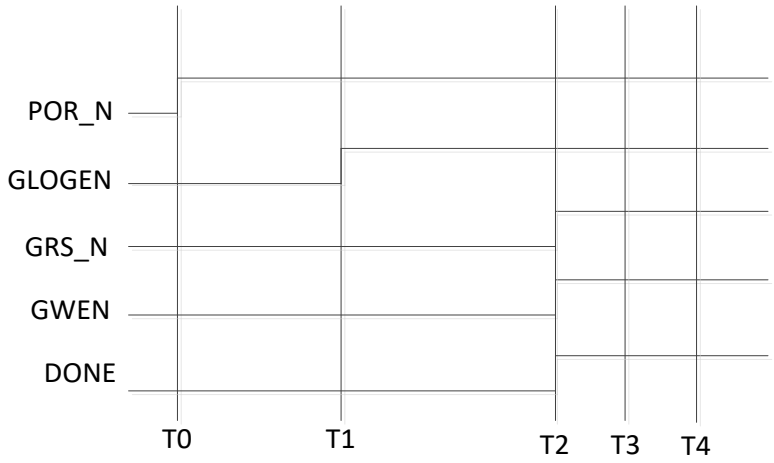

Figure 3-12 Global Signal Timing Diagram

3.9 Application Examples

3.9.1 Single DRM Configuration

This section illustrates the GTP configuration steps for a single 18K DRM. Users can also directly generate the DRM IP using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

The example is for mixed bit widths DP mode, with port A configured as 2Kx8, port B as 1Kx16 in byte enable write mode. Two ports have different clocks, enabling output registers, in TW write mode. The configuration steps for a 9K DRM are similar.

Configure a single 18K DRM as follows:

1. Configure the DRM parameters as described in the following table:

Table 3-4 DRM Parameter Configuration

Parameter Name	**Configuration Value**	**Description**
DATA_WIDTH_A	8	Configure port A to 2Kx8 mode
DATA_WIDTH_B	16	Configure port B to 1Kx16 mode
WRITE_MODE_A	"TRANSPARENT_WRITE"	Configure the write mode of port A to TW
WRITE_MODE_B		Configure the write mode of port B to TW

Parameter Name	Configuration Value	Description
DOA_REG	1	Enable port A output register
DOB_REG		Enable port B output register
RAM_MODE	"TRUE_DUAL_PORT"	Configure the DRM to DP mode

2. Connect the DRM ports as shown in the following table:

Table 3-5 DRM Port Connections

Port	Interfacing Signals	Description
ADDRA[13:0]	{addra[10:0],3'b111}	Connect the input address signal addra[10:0] to ADDRA[13:3], without using the byte enable function, and connect ADDRA[2:0] to high level. See 8.1 Address and Data Port Mapping for detailed address connection instructions
ADDRB[13:0]	{addrb[9:0], 2'b11, bweb[1:0]}	Connect the input address signal addrb[9:0] to ADDRB[13:4], using the byte enable function, with ADDRB[3:2] connected to high level, and ADDRB[1:0] connected to the byte enable signal
ADDRA_HOLD	1'b0	Do not use port A/B address hold function, connected to low level
ADDRB_HOLD		
DIA[17:0]	dia[7:0]	Connect the input data signal dia[7:0] to DIA[7:0], with the unused high-bit DIA ports floating. See 8.1 Address and Data Port Mapping for detailed data port connection instructions
DIB[17:0]	{1'b0, dib[15:8], 1'b0, dib[7:0]}	Connect the input data signal dib[15:0] to the GTP port {DIB[16:9], DIB[7:0]}, with DIB[8], DIB[17] as additional byte information bits. Leave floating or connect to low level when the data width is 2^N. See 8.4 Additional Information Bit for Bytes for details.
CSA[2:0]	3'b0	Address extension function is not used, connected to low level
CSB[2:0]		
DOA[17:0]	doa[7:0]	Connect the output data signal doa[7:0] to the GTP port DOA[7:0]
DOB[17:0]	{1'bz, dob[15:8], 1'bz, dob[7:0]}	Connect the output data signal dob[15:0] to the GTP port {DOB[16:9], DOB[7:0]}, with DOB[8], DOB[17] left floating

3. Other signals for A/B ports: Connect the port A/B's clock, clock enable, write enable, output register clock enable, and data register reset to the corresponding GTP ports;
4. Other unused parameters: Do not set the other unused parameters, use default values;
5. Other unused ports: Leave the other unused ports floating (ports that must be connected even if not used have been described above).

The configured GTP is as follows:

```
GTP_DRM18K #(
.DATA_WIDTH_A   (8),
.DATA_WIDTH_B   (16),
```

```
.WRITE_MODE_A   ("TRANSPARENT_WRITE"),
.WRITE_MODE_B   ("TRANSPARENT_WRITE"),
.DOA_REG          (1),
.DOB_REG          (1),
.RAM_MODE         ("TRUE_DUAL_PORT")
) GTP_DRM18K_inst (
.DOA              (doa[7:0]    ), // OUTPUT[17:0]
.DOB              (dob_gtp[17:0]), // OUTPUT[17:0]
.ADDRA            ({addra[10:0],3'b111}), // INPUT[13:0]
.ADDRB            ({addrb[9:0],2'b0,bweb}), // INPUT[13:0]
.CSA              (3'b0          ), // INPUT[2:0]
.CSB              (3'b0          ), // INPUT[2:0]
.DIA              (dia[7:0]    ), // INPUT[17:0]
.DIB              ({1'b0,dib[15:8],1'b0,dib[7:0]}), // INPUT[17:0]
.ADDRA_HOLD       (1'b0          ), // INPUT
.ADDRB_HOLD       (1'b0          ), // INPUT
.CEA              (cea           ), // INPUT
.CEB              (ceb           ), // INPUT
.CLKA             (clka          ), // INPUT
.CLKB             (clkb          ), // INPUT
.ORCEA            (orcea         ), // INPUT
.ORCEB            (orceb         ), // INPUT
.RSTA             (rsta          ), // INPUT
.RSTB             (rstb          ), // INPUT
.WEA              (wea           ), // INPUT
.WEB              (web           )    // INPUT
);
assign dob[15:0] = {dob_gtp[16:9],dob_gtp[7:0]};
```

3.9.2 Configuration for Multiple DRMs Cascaded

The cascading mentioned in this document specifically refers to address depth cascading. For data cascading, the configuration for individual DRM remains the same, only connecting the data ports, which will not be separately described in this document.

Address depths are cascaded through 18K DRMs without external logic to process the address. The DRM address up to 3 bits can be cascaded (CSA/CSB width is 3), with a total address width of 17 bits; the maximum data width for data cascading depends on the device's DRM resources.

This section illustrates the GTP cascading configuration steps for multiple 18K DRMs. Users can also directly generate the IP of cascaded DRM using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

Four DP 18K DRMs are cascaded in the example, each with ports A/B configured as 16Kx1, with a total data width of 2 bits and a total address width of 15 bits. Since the 9K DRM lacks address expansion related signals and parameters, it is not recommended to use 9K DRM for cascading.

The four cascaded DRMs are named DRM_1_1, DRM_1_2, DRM_2_1, and DRM_2_2 respectively. Taking DRM_1_2 as an example, the first digit 1 represents the address cascading coordinate, with 2 levels of address cascading, extending the address from 14 bits of a single DRM to 15 bits; the second digit 2 represents the data cascading coordinate, with 2 levels of data cascading, extending the data from 1 bit of a single DRM to 2 bits. When cascading multiple 18K DRMs, configure them according to the following steps:

1. Single DRM configuration: Configure the four 18K DRMs as 16Kx1 DP 18K DRMs, following steps similar to those in 3.9.1 Single DRM Configuration;
2. Configure the DRM address extension control parameters as described in the table below:

Table 3-6 DRM Parameter Configuration

Parameter Name	**DRM Name**	**Configuration Value**
CSA_MARK	DRM_1_1	3'b000
	DRM_1_2	3'b000
	DRM_2_1	3'b001
	DRM_2_2	3'b001
CSB_MARK	DRM_1_1	3'b000
	DRM_1_2	3'b000
	DRM_2_1	3'b001
	DRM_2_2	3'b001

3. Connect the cascade-related ports of the DRM as shown in the table below:

Table 3-7 DRM Port Connections

Port	**DRM Name**	**Interfacing Signals**
CSA	DRM_1_1	{2'b0,addra[14]}
	DRM_1_2	

Port	DRM Name	Interfacing Signals
	DRM_2_1	
	DRM_2_2	
CSB	DRM_1_1	{2'b0,addrb[14]}
	DRM_1_2	
	DRM_2_1	
	DRM_2_2	
ADDRA[15:0]	DRM_1_1	{addra[13:0]}
	DRM_1_2	
	DRM_2_1	
	DRM_2_2	
ADDRB[15:0]	DRM_1_1	{addrb[13:0]}
	DRM_1_2	
	DRM_2_1	
	DRM_2_2	
DIA[35:0]	DRM_1_1	dia[0]
	DRM_1_2	dia[1]
	DRM_2_1	dia[0]
	DRM_2_2	dia[1]
DIB[35:0]	DRM_1_1	dib[0]
	DRM_1_2	dib[1]
	DRM_2_1	dib[0]
	DRM_2_2	dib[1]
DOA[35:0]	DRM_1_1	doa_0[0]
	DRM_1_2	doa_0[1]
	DRM_2_1	doa_1[0]
	DRM_2_2	doa_1[1]
DOB[35:0]	DRM_1_1	dob_0[0]
	DRM_1_2	dob_0[1]
	DRM_2_1	dob_1[0]
	DRM_2_2	dob_1[1]

4. Output data selection: When the addra[14] address signal is input to the register and experiences a one-beat delay, if the value of addra[14] is 0, port A select doa_0[1:0] as output data; if the value is 1, port A select doa_1[1:0] as output data. Port B functions in a similar manner.

The configured GTP is as follows:

```
GTP_DRM18K #(
.CSA_MASK       (3'b000),
.CSB_MASK       (3'b000),
```

```
.DATA_WIDTH_A    (1),
.DATA_WIDTH_B    (1),
.RAM_MODE        ("TRUE_DUAL_PORT"),
.WRITE_MODE_A    ("TRANSPARENT_WRITE"),
.WRITE_MODE_B    ("TRANSPARENT_WRITE")
) DRM_1_1 (
.DOA             ({doa_0[0]}), // OUTPUT[17:0]
.DOB             ({dob_0[0]}), // OUTPUT[17:0]
.ADDRA           ({addra[13:0]}), // INPUT[13:0]
.ADDRB           ({addrb[13:0]}), // INPUT[13:0]
.CSA             ({2'b0,addra[14]}), // INPUT[2:0]
.CSB             ({2'b0,addrb[14]}), // INPUT[2:0]
.DIA             ({dia[0]}    ), // INPUT[17:0]
.DIB             ({dib[0]}    ), // INPUT[17:0]
.ADDRA_HOLD      (addra_hold), // INPUT
.ADDRB_HOLD      (addrb_hold), // INPUT
.CEA             (cea         ), // INPUT
.CEB             (ceb         ), // INPUT
.CLKA            (clka        ), // INPUT
.CLKB            (clkb        ), // INPUT
.ORCEA           (1'b0        ), // INPUT
.ORCEB           (1'b0        ), // INPUT
.RSTA            (rsta        ), // INPUT
.RSTB            (rstb        ), // INPUT
.WEA             (wea         ), // INPUT
.WEB             (web         )  // INPUT
);
GTP_DRM18K #(
.CSA_MASK        (3'b000),
.CSB_MASK        (3'b000),
.DATA_WIDTH_A    (1),
.DATA_WIDTH_B    (1),
.RAM_MODE        ("TRUE_DUAL_PORT"),
.WRITE_MODE_A    ("TRANSPARENT_WRITE"),
```

```
.WRITE_MODE_B    ("TRANSPARENT_WRITE")
) DRM_1_2 (
.DOA              ({doa_0[1]}), // OUTPUT[17:0]
.DOB              ({dob_0[1]}), // OUTPUT[17:0]
.ADDRA             ({addra[13:0]}), // INPUT[13:0]
.ADDRB             ({addrb[13:0]}), // INPUT[13:0]
.CSA              ({2'b0,addra[14]}), // INPUT[2:0]
.CSB              ({2'b0,addrb[14]}), // INPUT[2:0]
.DIA              ({dia[1]}    ), // INPUT[17:0]
.DIB              ({dib[1]}    ), // INPUT[17:0]
.ADDRA_HOLD       (addra_hold), // INPUT
.ADDRB_HOLD       (addrb_hold), // INPUT
.CEA              (cea          ), // INPUT
.CEB              (ceb          ), // INPUT
.CLKA             (clka         ), // INPUT
.CLKB             (clkb         ), // INPUT
.ORCEA            (1'b0         ), // INPUT
.ORCEB            (1'b0         ), // INPUT
.RSTA             (rsta         ), // INPUT
.RSTB             (rstb         ), // INPUT
.WEA              (wea          ), // INPUT
.WEB              (web          )    // INPUT
);
GTP_DRM18K #(
.CSA_MASK         (3'b001),
.CSB_MASK         (3'b001),
.DATA_WIDTH_A    (1),
.DATA_WIDTH_B    (1),
.RAM_MODE         ("TRUE_DUAL_PORT"),
.WRITE_MODE_A    ("TRANSPARENT_WRITE"),
.WRITE_MODE_B    ("TRANSPARENT_WRITE")
) DRM_2_1 (
.DOA              ({doa_1[0]}), // OUTPUT[17:0]
.DOB              ({dob_1[0]}), // OUTPUT[17:0]
```

```
.ADDRA          ({addra[13:0]}), // INPUT[13:0]
.ADDRB          ({addrb[13:0]}), // INPUT[13:0]
.CSA            ({2'b0,addra[14]}), // INPUT[2:0]
.CSB            ({2'b0,addrb[14]}), // INPUT[2:0]
.DIA            ({dia[0]}    ), // INPUT[17:0]
.DIB            ({dib[0]}    ), // INPUT[17:0]
.ADDRA_HOLD     (addra_hold), // INPUT
.ADDRB_HOLD     (addrb_hold), // INPUT
.CEA            (cea         ), // INPUT
.CEB            (ceb         ), // INPUT
.CLKA           (clka        ), // INPUT
.CLKB           (clkb        ), // INPUT
.ORCEA          (1'b0        ), // INPUT
.ORCEB          (1'b0        ), // INPUT
.RSTA           (rsta        ), // INPUT
.RSTB           (rstb        ), // INPUT
.WEA            (wea         ), // INPUT
.WEB            (web         )  // INPUT
);
GTP_DRM18K #(
.CSA_MASK       (3'b001),
.CSB_MASK       (3'b001),
.DATA_WIDTH_A   (1),
.DATA_WIDTH_B   (1),
.RAM_MODE       ("TRUE_DUAL_PORT"),
.WRITE_MODE_A   ("TRANSPARENT_WRITE"),
.WRITE_MODE_B   ("TRANSPARENT_WRITE")
) DRM_2_2 (
.DOA            ({doa_1[1]}), // OUTPUT[17:0]
.DOB            ({dob_1[1]}), // OUTPUT[17:0]
.ADDRA          ({addra[13:0]}), // INPUT[13:0]
.ADDRB          ({addrb[13:0]}), // INPUT[13:0]
.CSA            ({2'b0,addra[14]}), // INPUT[2:0]
.CSB            ({2'b0,addrb[14]}), // INPUT[2:0]
```

```
.DIA                ({dia[1]}    ), // INPUT[17:0]
.DIB                ({dib[1]}    ), // INPUT[17:0]
.ADDRA_HOLD         (addra_hold), // INPUT
.ADDRB_HOLD         (addrb_hold), // INPUT
.CEA                (cea         ), // INPUT
.CEB                (ceb         ), // INPUT
.CLKA               (clka        ), // INPUT
.CLKB               (clkb        ), // INPUT
.ORCEA              (1'b0        ), // INPUT
.ORCEB              (1'b0        ), // INPUT
.RSTA               (rsta        ), // INPUT
.RSTB               (rstb        ), // INPUT
.WEA                (wea         ), // INPUT
.WEB                (web         )  // INPUT
);
//Output data selection
always @(posedge clka or posedge rsta)
   begin
      if (rsta)
         sel_a <= 1'b0;
      else if (~addra_hold & cea)
         sel_a <= addra[14];
      end
always @(posedge clkb or posedge rstb)
   begin
      if (rstb)
         sel_b <= 1'b0;
      else if (~addrb_hold & ceb)
         sel_b <= addrb[14];
      end
assign doa[1:0] = (sel_a)?doa_1[1:0]:doa_0[1:0];
assign dob[1:0] = (sel_b)?dob_1[1:0]:dob_0[1:0];
```

Chapter 4 SDP Mode

4.1 Description of Mode

The port mode of the RAM is determined by the parameter RAM_MODE; when the value of RAM_MODE is "SIMPLE_DUAL_PORT", the RAM enters SDP mode. This document will detail the 18K SDP RAM, which has essentially the same structure and functions as the 9K.

SDP RAM mode supports:

- Dual-port read and write, with one port as the read port while the other as the write port.
- Neither port can perform read and write operations simultaneously.
- Independent bit width settings for two ports
- For x32/x36 bit width, port A is fixed as the write port (active when WEA is high), and port B is fixed as the read port (active when WEB is low).

4.2 Port Descriptions

Each 9K and 18K DRM can also be configured as SDP (Simple Dual Port) RAM. In this mode, the DRM port data width is increased to 36 bits. The SDP RAM mode includes two pairs of ports (each pair has two different ports, A and B), with one of ports A and B dedicated to data writing and the other to data reading, and both reading and writing ports support different clocks. Similar to DP RAM, performing a read and a write operation on the same address through both ports simultaneously will cause read/write conflicts. The diagram below shows the structure of the 18K SDP RAM.

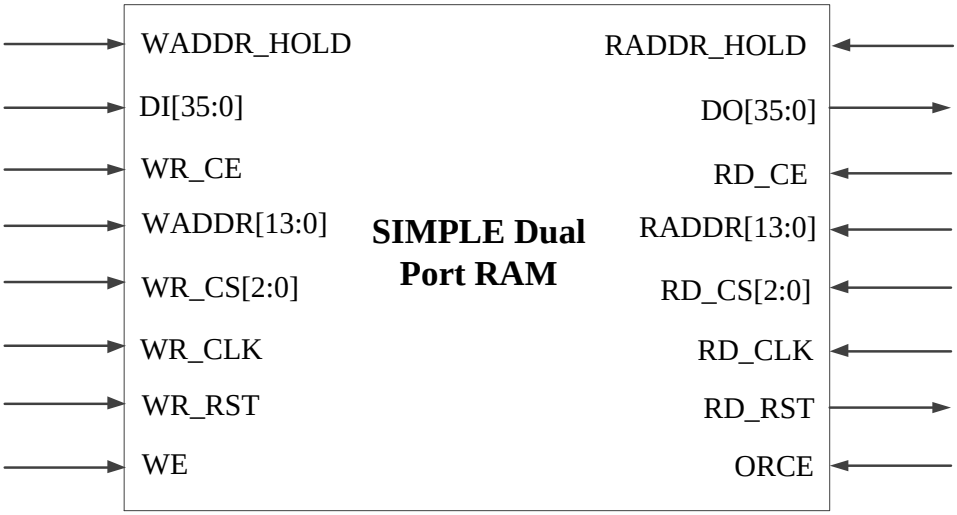

Figure 4-1 SDP RAM Data Ports

The table below lists the port names and descriptions.

Table 4-1 SDP RAM Port Names and Descriptions

Port	Direction	Description	Port	Direction	Description
WADDR_HOLD	Input	Write port address hold signal	RADDR_HOLD	Input	Read port address hold signal
DI	Input	Write port data input	DO	Output	Read port data output
WR_CE	Input	Write port input register clock enable	RD_CE	Input	Read port input register clock enable
WADDR	Input	Write port address input	RADDR	Input	Read port address input
WR_CS	Input	Write port address extension	RD_CS	Input	Read port address extension
WR_CLK	Input	Write port clock	RD_CLK	Input	Read port clock
WR_RST	Input	Write port output register reset	RD_RSTB	Input	Read port output register reset
WE	Input	Write enable	ORCE	Input	Output register clock enable

Note:

1. Under SDP mode with x32/x36 bit width, DI is formed by concatenating DIA/DIB from ports A/B, and DO likewise. See 8.1 Address and Data Port Mapping for details;

2. The 9K DRM has no CSA/CSB port and does not support address extension. For DRM cascading to extend address depth, it is recommended to use 18K DRM.

4.3 Bit Width Combinations

The port bit width of RAM is determined by the parameter DATA_WIDTH_A/DATA_WIDTH_B in the GTP. For example, when the value of DATA_WIDTH_A is 4, the bit width of port A is set to 4 bits. SDP mode supports setting different bit widths for ports A and B.

The following table shows the allowed bit width combinations for the Simple Dual Port RAM mode in the 9K DRM mode.

Table 4-2 Allowed Bit Width Combinations for Simple Dual Port RAM Mode in 9K DRM Mode

			Write Port 0/1								
			No Parity Bits						**With Parity Bits**		
			8Kx1	**4Kx2**	**2Kx4**	**1Kx8**	**512x16**	**256x32**	**1Kx9**	**512x18**	**256x36**
Read Port 0/1	**No Parity Bits**	**8Kx1**	√	√	√	√	√	√			
		4Kx2	√	√	√	√	√	√			
		2Kx4	√	√	√	√	√	√			
		1Kx8	√	√	√	√	√	√			
		512x16	√	√	√	√	√	√			
		256x32	√	√	√	√	√	√			
	With Parity Bits	**1Kx9**							√	√	√
		512x18							√	√	√
		256x36							√	√	√

Note: √ indicates the supported bit width combinations.

The following table shows the allowed bit width combinations for the Simple Dual Port RAM mode in the 18K DRM mode.

Table 4-3 Allowed Bit Width Combinations for Simple Dual Port RAM Mode in 18K DRM Mode

			Write Port 0/1								
			No Parity Bits						**With Parity Bits**		
			16Kx1	**8Kx2**	**4Kx4**	**2Kx8**	**1Kx16**	**512x32**	**2Kx9**	**1K x18**	**512x36**
Read Port 0/1	**No Parity Bits**	**8Kx1**	√	√	√	√	√	√			
		4Kx2	√	√	√	√	√	√			
		2Kx4	√	√	√	√	√	√			
		1Kx8	√	√	√	√	√	√			
		512x16	√	√	√	√	√	√			
		256x32	√	√	√	√	√	√			
	With Parity Bits	**2Kx9**							√	√	√
		1Kx18							√	√	√
		512x36							√	√	√

Note: √ indicates the supported bit width combinations.

4.4 Read and Write Operations

SDP mode does not support write mode settings. It is forbidden to set it to TW or RBW mode during use, and it must be set to the default NW mode. That is, the values of WRITE_MODE_A/WRITE_

MODE_B in GTP must be the default "NORMAL_WRITE", otherwise, the DRM will enter an abnormal write operation state.

In SDP mode, there are two relatively independent ports, and performing a read and a write operation on the same address through both ports simultaneously will cause read/write conflicts. DRM prohibits both ports from writing data to the same address simultaneously and from performing a read and a write on the same address at the same time; this must be avoided in practical applications through user logic.

The typical timing of the SDP RAM is shown in the diagram below. In the diagram, the SDP read/write ports are clocked simultaneously, with each signal name being the port name of SDP mode defined in Table 4-1, with the actual connected GTP port name in parentheses.

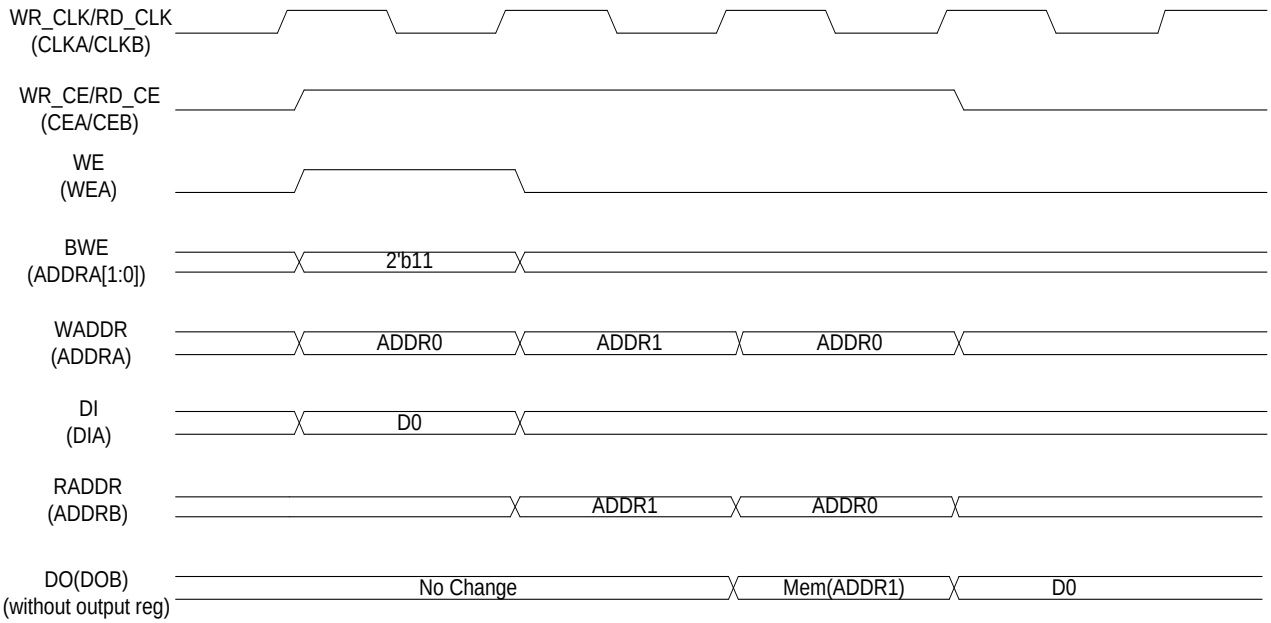

Figure 4-2 Typical Timing Diagram of SDP RAM

4.5 Byte Enable

The byte enable function of SDP mode is similar to that of DP mode. See 3.5 Byte Enable for details. The main differences of byte enable between SDP mode and DP mode are as follows:

- In SDP mode, one port is fixed as a write port, and the other port is fixed as a read port;
- The byte enable signal of SDP mode also supports x32(36)-bit data width. In this mode, read/write data is concatenated through ports A/B, BWE[3:0] = ADDRA[3:0].

The read/write timing diagram of DRM in the byte enable mode under SDP mode is shown in the following diagram, with each signal name being the port name of SDP mode defined in Table 4-1, with the actual connected GTP port name in parentheses.

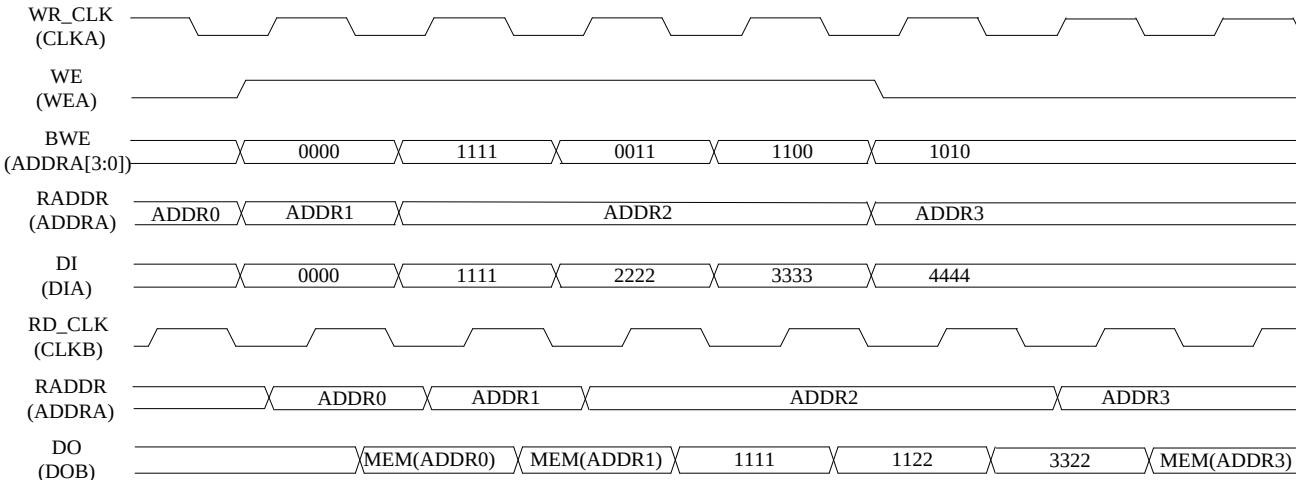

Figure 4-3 Byte Enable Read/Write Timing Diagram (SDP Mode)

4.6 Internal Registers

The internal registers of SDP mode are the same as those of DP mode. See 3.6 Internal Registers for details.

4.7 Reset signal

The reset signals of SDP mode are the same as those of DP mode. See 3.7 Reset Signal for details.

4.8 Global Signal Timing

The global signal timing of SDP mode is the same as that of DP mode. See 3.8 Global Signal Timing for details.

4.9 Application Examples

4.9.1 Single DRM Configuration

This section illustrates the GTP configuration steps for a single 18K DRM. Users can also directly generate the DRM IP using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

The example is for a mixed bit widths SDP mode, with read port A configured as 2Kx9 and write port B configured as 512x36 (achieved by concatenating DOA/DOB of port A/B). Port A/B use different clocks, enabling output registers. The configuration steps for a 9K DRM are similar.

Configure a single 18K DRM as follows:

1. Configure the DRM parameters as described in the following table:

Table 4-4 DRM Parameter Configuration

Parameter Name	Configuration Value	Description
DATA_WIDTH_A	9	Configure port A to 2Kx9 mode
DATA_WIDTH_B	36	Configure port B to 512x36 mode
WRITE_MODE_A	"NORMAL_WRITE"	Configure the write mode of port A to NW
WRITE_MODE_B		Configure the write mode of port B to NW
DOA_REG	1	Enable port A output register
DOB_REG		Enable port B output register
RAM_MODE	"SIMPLE_DUAL_PORT"	Configure the DRM to SDP mode

2. Connect the DRM ports as shown in the following table:

Table 4-5 DRM Port Connections

Port	Interfacing Signals	Description
ADDRA[13:0]	{waddr[10:0],3'b111}	Port A serves as the write port, connecting the write address signal waddr[10:0] to ADDRA[13:3], with ADDRA[2:0] connected to high level. See 8.1 Address and Data Port Mappingfor detailed address connection instructions
ADDRB[13:0]	{raddr[8:0],5'b11111}	Port B serves as the read port, connecting the read address signal raddr[8:0] to ADDRB[13:5], with ADDRB[4:0] connected to high level
ADDRA_HOLD	1'b0	Do not use port A/B address hold function, connected to low level
ADDRB_HOLD		
DIA[17:0]	di[8:0]	Connect the write data signal di[8:0] to DIA[8:0], leaving the unused high-bit DIA ports floating. See 8.1 Address and Data Port Mapping for detailed data port connection instructions
DIB[17:0]	Floating or connecting to low level	Port B is a read port without data concatenation for writing, and the GTP port DIB is either floating or connected to a low level
CSA[2:0]	3'b0	Address extension function is not used, connected to low level
CSB[2:0]		
DOA[17:0]	do[17:0]	Ports A/B are concatenated to achieve a 36-bit data width, with DOA outputting the lower 18 bits of read data
DOB[17:0]	do[35:18]	Ports A/B are concatenated to achieve a 36-bit data width, with DOB outputting the upper 18 bits of read data
WEA	we	Connect the write enable signal we to the WEA of write port A
WEB	1'b0	Connect the write enable WEB of read port B to a low level

3. Other signals for read/write ports: Connect the clock, clock enable, output register clock enable, and data register reset of the read/write ports to the corresponding GTP ports A/B respectively;
4. Other unused parameters: Do not set the other unused parameters, use default values;
5. Other unused ports: Leave the other unused ports floating (ports that must be connected even if not used have been described above).

The configured GTP is as follows:

```
GTP_DRM18K #(
.DATA_WIDTH_A    (9),
.DATA_WIDTH_B    (36),
.WRITE_MODE_A    ("NORMAL_WRITE"),
.WRITE_MODE_B    ("NORMAL_WRITE"),
.DOA_REG         (1),
```

```
.DOB_REG            (1),
.RAM_MODE           ("SIMPLE_DUAL_PORT")
) GTP_DRM18K_inst (
.DOA                (do[17:0]    ), // OUTPUT[17:0]
.DOB                (do[35:18] ), // OUTPUT[17:0]
.ADDRA              ({waddr[10:0],3'b111}), // INPUT[13:0]
.ADDRB              ({raddr[8:0],5'b11111}), // INPUT[13:0]
.CSA                (3'b0          ), // INPUT[2:0]
.CSB                (3'b0          ), // INPUT[2:0]
.DIA                (di[8:0]      ), // INPUT[35:0]
.DIB                (                ), // INPUT[35:0]
.ADDRA_HOLD         (1'b0            ), // INPUT
.ADDRB_HOLD         (1'b0            ), // INPUT
.CEA                (wr_ce         ), // INPUT
.CEB                (rd_ce         ), // INPUT
.CLKA               (wr_clk        ), // INPUT
.CLKB               (rd_clk        ), // INPUT
.ORCEA              (orce            ), // INPUT
.ORCEB              (orce            ), // INPUT
.RSTA               (wr_rst        ), // INPUT
.RSTB               (rd_rst        ), // INPUT
.WEA                (we              ), // INPUT
.WEB                (1'b0            )    // INPUT
);
```

4.9.2 Configuration for Multiple DRMs Cascaded

Multiple 18K DRMs in SDP mode are cascaded in the same way as DP mode. See details in 3.9.2 Configuration for Multiple DRMs Cascaded .

Chapter 5 SP Mode

5.1 Description of Mode

The port mode of the RAM is determined by the parameter RAM_MODE; when the value of RAM_MODE is "SINGLE_PORT", the RAM enters SP mode. This document will detail the 18K SP RAM, which has essentially the same structure and functions as the 9K.

SP RAM mode supports:

- Single port read operation
- Single port write operation
- For x32/x36 bit width, port A is fixed as the write control port (active when WEA is high), and port B is fixed as the read control port (active when WEB is low).

5.2 Port Descriptions

Each 9K and 18K DRM can also be configured as SP (Single Port) RAM. In this mode, the data width of the DRM port is increased to 36 bits. In SP RAM mode, when two ports are shared, the DRM includes only one port, and SP RAM mode allows read and write operations on this port. The diagram below shows the structure of the 18K SP RAM.

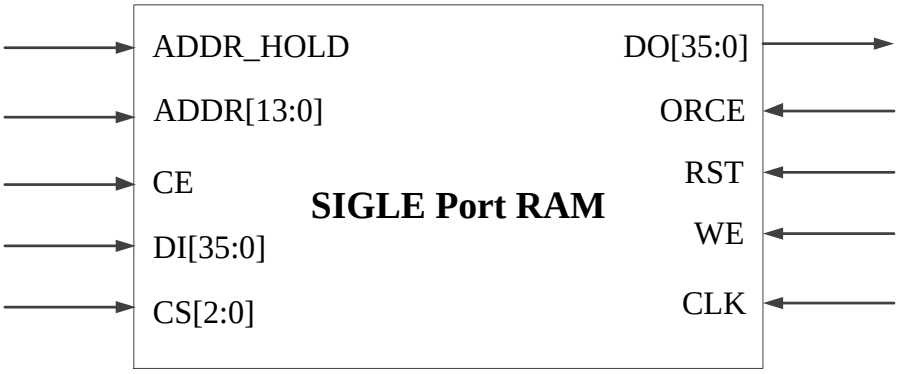

Figure 5-1 SP RAM Data Ports

The table below lists the port names and descriptions for SP RAM mode.

Table 5-1 SP RAM Port Names and Descriptions

Port	Direction	Description	Port	Direction	Description
ADDR_HOLD	Input	Address hold signal	DO	Output	Read data output
ADDR	Input	Address input	ORCE	Input	Output register clock enable
CE	Input	Input register clock	RST	Input	Output register reset

Port	Direction	Description	Port	Direction	Description
		enable			
DI	Input	Write data input	WE	Input	Write enable
CS	Input	Address extension	CLK	Input	Port clock

Note:

1. Under SP mode with x32/x36 bit width, DI is formed by concatenating DIA/DIB from ports A/B, and DO likewise. See 8.1 Address and Data Port Mapping for details;

2. The 9K DRM has no CSA/CSB port and does not support address extension. For DRM cascading to extend address depth, it is recommended to use 18K DRM.

5.3 Bit Width Combinations

The port bit width of RAM is determined by the parameter DATA_WIDTH_A / DATA_WIDTH_B in the GTP. For example, when the value of DATA_WIDTH_A is 4, the bit width of port A is set to 4 bits. Ports A and B in SP mode must be set to the same data width.

The table below shows the allowed bit width for Single Port RAM mode in 18K DRM mode.

Table 5-2 List of Single Port RAM Modes for 18K DRM Mode

Mode	16Kx1	8Kx2	4Kx4	2Kx8	1Kx16	512x32	2Kx9	1Kx18	512x36
SP RAM	√	√	√	√	√	√	√	√	√

Note:

1. √ indicates supported data width combinations;

2. In the Single Port RAM mode of 18K DRM mode, direct setting of TW, RBW write modes is prohibited for 512x32, 512x36 modes, additional configuration is required. See details in 5.4 Read and Write Operations.

The table below shows the allowed bit width for Simple Port RAM mode in 9Kb DRM mode.

Table 5-3 List of Single Port RAM Modes for 9K DRM Mode

Mode	8Kx1	4Kx2	2Kx4	1Kx8	512x16	256x32	1Kx9	512x18	256x36
SP RAM	√	√	√	√	√	√	√	√	√

Note:

1. √ indicates supported data width combinations;

2. In the Single Port RAM mode of 9K DRM mode, direct setting of TW, RBW write modes is prohibited for 256x32, 256x36 modes, additional configuration is required. See details in 5.4 Read and Write Operations.

5.4 Read and Write Operations

The read and write operations of each write mode in SP mode are similar to the single port in DP mode. See 3.4 Read and Write Operations for details.

In SP mode, DRM supports setting the write mode to NW (Normal Write) for all bit widths. For the data port widths of 18 bit and below in SP mode, the write mode can also be directly set to TW (Transparent Write) or RBW (Read before Write). However, for the data port widths of 32 bit and

36 bit in SP mode, it is prohibited to set the write mode directly to TW and RBW. It must be set to the default NW mode, with details as shown in the following table:

Table 5-4 Availability for write mode Settings in SP Mode

Data Port Width	**Supported or Not**	**Implementation of write modes**
1 (18K/9K)	Supported	In GTP, the mode parameter can be directly set to SP mode, and for 18K and 9K with the widths of 18 bit and below, the write modes can be directly set to NW, TW, and RBW
2 (18K/9K)	Supported	
4 (18K/9K)	Supported	
8 (18K/9K)	Supported	
9 (18K/9K)	Supported	
16 (18K/9K)	Supported	
18 (18K/9K)	Supported	
32 (18K/9K)	Not supported	In GTP, write modes cannot be set directly for 32/36 bit widths in SP mode, and the default NW must be used; TW, RBW implementations require additional configurations, see the following section for details
36(18K/9K)	Not supported	

TW and RBW modes cannot be implemented through direct configuration in DRM for 32/36 bits in SP mode; additional configurations are required, with the specific implementation as follows:

1. Configure both Port A and Port B to 16/18 bits in DP mode (for 32/36 bits SP respectively);
2. Set Port A and Port B to the same write mode; other configuration bits should also be the same;
3. The CE, CS, WE, ORCE, RST, CLK, and other control signals of Port A and Port B are connected to the same signals;
4. Splicing DIA and DIB as data inputs; splicing DOA and DOB as data outputs;
5. Both Port A and Port B address input ports should be connected to the address inputs; ADDRA[4] and ADDRB[4] should be connected to 0 or 1 respectively (ADDRA[4] is opposite to ADDRB[4]), as shown in the following diagram:

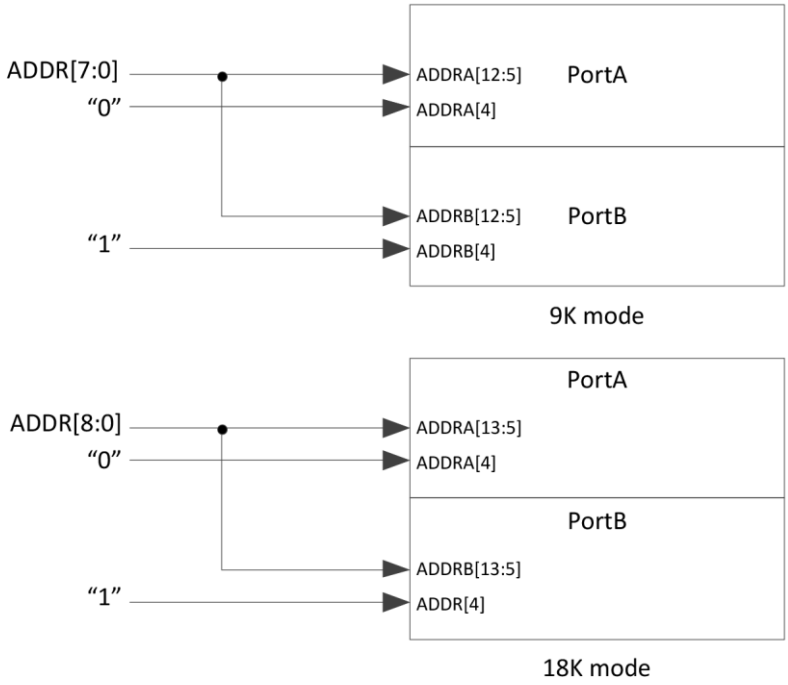

Figure 5-2 Address Line Configuration for TW and RBW Modes in 32/36 bits SP Mode

6. For the connection of BWEA and BWEB signals, refer to the connections in the 16/18 bits DP mode of two ports respectively.

5.5 Byte Enable

The byte enable function of SP mode is similar to that of SDP mode. See 3.5 Byte Enable.

5.6 Internal Registers

The internal registers of SP mode are the same as those of DP mode. See 3.6 Internal Registers.

5.7 Reset Signal

The reset signals of SP mode are the same as those of DP mode. See 3.7 Reset Signal.

5.8 Global Signal Timing

The global signal timing of SP mode is the same as of DP mode. See 3.8 Global Signal Timing.

5.9 Application Examples

5.9.1 Single DRM Configuration

This section illustrates the GTP configuration steps for a single 18K DRM. Users can also directly generate the DRM IP using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

The example is for SP mode, with read/write ports configured as 512x32, output register enabled, in NW write mode. The configuration steps for a 9K DRM are similar.

Configure a single 18K DRM as follows:

1. Configure the DRM parameters as described in the following table:

Table 5-5 DRM Parameter Configuration

Parameter Name	Configuration Value	Description
DATA_WIDTH_A	32	Configure port A to 512x32 mode
DATA_WIDTH_B		Configure port B to 512x32 mode
WRITE_MODE_A	"NORMAL_WRITE"	Configure the write mode of port A to NW
WRITE_MODE_B		Configure the write mode of port B to NW
DOA_REG	1	Enable port A output register
DOB_REG		Enable port B output register
RAM_MODE	"SINGLE_PORT"	Configure DRM to SP mode

2. Connect the DRM ports as shown in the following table:

Table 5-6 DRM Port Connections

Port	Interfacing Signals	Description
ADDRA[13:0]	{addr[8:0],5'b11111}	Port A/B address connects to the same address signal, connecting the input address signal addr[8:0] to ADDRA[13:5], ADDRA[4:0] to high level. The same with ADDRB. See detailed address connection instructions in 8.1 Address and Data Port Mapping
ADDRB[13:0]		
ADDRA_HOLD	1'b0	Do not use the A/B port address hold function, connecting to a low level. If the address hold function is used, the ports A/B must be connected to the same signal.
ADDRB_HOLD		
DIA[35:0]	{1'b0, di[15:8], 1'b0, di[7:0]}	The ports A/B are concatenated to implement a 32-bit write data port, with DIA inputting the lower 16-bit write data di[15:0]. For detailed data port connection instructions, see 8.1 Address and Data Port Mapping.
DIB[35:0]	{1'b0, di[31:24], 1'b0, di[23:16]}	The ports A/B are concatenated to implement a 32-bit write data port, with DIB inputting the upper 16-bit write data di[31:16]. DIB[8] and DIB[17] serve as byte additional information bits. For more details, see 8.4 Additional Information Bit for Bytes.
CSA[2:0]	3'b0	Do not use the address extension function, connecting to a low level. If the address extension function is used, the ports A/B must be connected to the same signal.
CSB[2:0]		

Port	Interfacing Signals	Description
DOA[35:0]	{1'bz, do[15:8], 1'bz, do[7:0]}	The ports A/B are concatenated to implement a 32-bit read data port, with DOA outputting the lower 16-bit read data do[15:0]. DOA[8] and DOA[17] should be left floating.
DOB[35:0]	{1'bz, do[31:24], 1'bz, do[23:16]}	The ports A/B are concatenated to implement a 32-bit read data port, with DOB outputting the upper 16-bit read data do[31:15]. DOB[8] and DOB[17] should be left floating.

3. Other signals for read/write ports: Connect the clock, clock enable, output register clock enable, data register reset and write enable of the read/write ports to the corresponding GTP ports A/B respectively; the ports A/B must be connected to the same signal.
4. Other unused parameters: Do not set the other unused parameters, use default values;
5. Other unused ports: Leave the other unused ports floating (ports that must be connected even if not used have been described above).

The configured GTP is as follows:

```
GTP_DRM18K #(
.DATA_WIDTH_A    (32),
.DATA_WIDTH_B    (32),
.WRITE_MODE_A    ("NORMAL_WRITE"),
.WRITE_MODE_B    ("NORMAL_WRITE"),
.DOA_REG         (1),
.DOB_REG         (1),
.RAM_MODE        ("SINGLE_PORT")
) GTP_DRM18K_inst (
.DOA             (doa[17:0] ), // OUTPUT[17:0]
.DOB             (dob[17:0] ), // OUTPUT[17:0]
.ADDRA           ({addr[8:0],5'b11111}), // INPUT[13:0]
.ADDRB           ({addr[8:0],5'b11111}), // INPUT[13:0]
.CSA             (3'b0        ), // INPUT[2:0]
.CSB             (3'b0        ), // INPUT[2:0]
.DIA             ({1'b0,di[15:8],1'b0,di[7:0]}), // INPUT[17:0]
.DIB             ({1'b0,di[31:24],1'b0,di[23:16]}), // INPUT[17:0]
.ADDRA_HOLD      (1'b0        ), // INPUT
.ADDRB_HOLD      (1'b0        ), // INPUT
.CEA             (ce          ), // INPUT
.CEB             (ce          ), // INPUT
```

```
.CLKA           (clk          ), // INPUT
.CLKB           (clk          ), // INPUT
.ORCEA          (orce         ), // INPUT
.ORCEB          (orce         ), // INPUT
.RSTA           (rst          ), // INPUT
.RSTB           (rst          ), // INPUT
.WEA            (we           ), // INPUT
.WEB            (we           )  // INPUT
);
assign do[31:0] = {dob[16:9],dob[7:0],doa[16:9],doa[7:0]};
```

5.9.2 Configuration for Multiple DRMs Cascaded

Multiple 18K DRMs in SP mode are cascaded in the same way as DP mode. See details in 3.9.2 Configuration for Multiple DRMs Cascaded .

Chapter 6 ROM Mode

6.1 Description of Mode

The port mode of the RAM is determined by the parameter RAM_MODE. When the value of RAM_MODE is "ROM", the RAM enters ROM mode. This document will detail the 18K ROM, which has essentially the same structure and functions as the 9K.

ROM mode only supports data read and does not support data write.

6.2 Port Descriptions

The DRM can be configured as ROM, with ROM contents initialized through the configuration interface. In ROM mode, the ports for both 9K and 18K modes are read-only. The following diagram illustrates the structure of ROM mode.

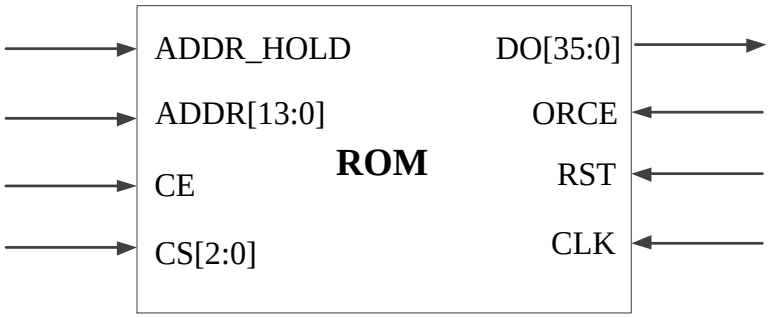

Figure 6-1 ROM Mode Data Ports

The table below lists the port names and descriptions for ROM mode.

Table 6-1 ROM Port Names and Descriptions

Port	Direction	Description	Port	Direction	Description
ADDR_HOLD	Input	Address hold signal	DO	Output	Read data output
ADDR	Input	Address input	ORCE	Input	Output register clock enable
CE	Input	Input register clock enable	RST	Input	Output register reset
CS	Input	Address extension	CLK	Input	Port clock

Note:

1. Under ROM mode with x32/x36 bit width, DI is formed by concatenating DIA/DIB from ports A/B, and DO likewise. For more details, see Address and Data Port Mapping.

2. The 9K DRM has no CSA/CSB port and does not support address extension. For DRM cascading to extend address depth, it is recommended to use 18K DRM.

6.3 Bit Width Combinations

The port bit width of RAM is determined by the parameter DATA_WIDTH_A / DATA_WIDTH_B in the GTP. For example, when the value of DATA_WIDTH_A is 4, the bit width of port A is set to 4 bits. Ports A and B in ROM mode must be set to the same data width.

The table below shows the allowed bit widths for ROM mode in 18K DRM mode.

Table 6-2 List of ROM Modes for 18K DRM Mode

Mode	**16Kx1**	**8Kx2**	**4Kx4**	**2Kx8**	**1Kx16**	**512x32**	**2Kx9**	**1Kx18**	**512x36**
ROM	√	√	√	√	√	√	√	√	√

Note: √ indicates the supported bit width combinations;

The table below shows the allowed bit widths for ROM mode in 9K DRM mode.

Table 6-3 List of ROM Modes for 9K DRM Mode

Mode	**8Kx1**	**4Kx2**	**2Kx4**	**1Kx8**	**512x16**	**256x32**	**1Kx9**	**512x18**	**256x36**
ROM	√	√	√	√	√	√	√	√	√

Note: √ indicates the supported bit width combinations;

6.4 Read and Write Operations

In ROM mode, DRM does not support write operations; the timing of read operations is the same as in DP mode. For details, see 3.4 Read and Write Operations.

6.5 Internal Registers

The internal registers in ROM mode are the same as in DP mode. For details, see 3.6 Internal Registers.

6.6 Reset signal

The reset signals in ROM mode are the same as in DP mode. For details, see 3.7 Reset Signal.

6.7 Global Signal Timing

The global signal timing in ROM mode is the same as in DP mode. For details, see 3.8 Global Signal Timing.

6.8 Application Examples

6.8.1 Single DRM Configuration

This section illustrates the GTP configuration steps for a single 18K DRM. Users can also directly generate the DRM IP using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

The example is for ROM mode, with the read/write port configured as 1Kx18, and output register enabled. The configuration steps for a 9K DRM are similar.

Configure a single 18K DRM as follows:

1. Configure the DRM parameters as described in the following table:

Table 6-4 DRM Parameter Configuration

Parameter Name	Configuration Value	Description
DATA_WIDTH_A	18	Configure port A to 1Kx18 mode
DATA_WIDTH_B		Configure port B to 1Kx18 mode
DOA_REG	1	Enable port A output register
DOB_REG		Enable port B output register
RAM_MODE	"ROM"	Configure DRM to ROM mode
INIT_00	{18'h0000f, ... 18'h00003, 18'h00002, 18'h00001, 18'h00000}	Set the initial value parameters of the ROM to the corresponding address values, that is, the 1-18 bits of INIT_00 corresponding to address 0 are set to 0; the 19-36 bits corresponding to address 1 are set to 1, and so on. See the specific initialization parameter mapping rules in 8.2 Initialization Configuration Parameter Mapping;
.....	.....	
INIT_3F	{18'h003ff, ... 18'h003f3, 18'h003f2, 18'h003f1, 18'h003f0}	

2. Connect the DRM ports as shown in the following table:

Table 6-5 DRM Port Connections

Port	Interfacing Signals	Description
ADDRA[13:0]	{addr[9:0],4'b1111}	Port A/B address connects to the same address signal, connecting the input address signal addr[9:0] to ADDRA[13:4], ADDRA[3:0] to high level. The same with ADDRB. See detailed address connection instructions in 8.1 Address and Data Port Mapping
ADDRB[13:0]		
ADDRA_HOLD	1'b0	Do not use the A/B port address hold function, connecting to a low level. If the address hold function is used, the ports A/B must be connected to the same signal.
ADDRB_HOLD		
DIA[17:0]	Float	ROM does not support write operations, with ports DIA/DIB floating
DIB[17:0]		
CSA[2:0]	3'b0	Do not use the address extension function, connecting to a low level. If the address control function is used, the ports A/B must be connected to the same signal.
CSB[2:0]		
DOA[17:0]	do[17:0]	Connect the output data signal do[17:0] to DOA[17:0]

Port	Interfacing Signals	Description
DOB[17:0]	Float	No data concatenation performed, with DOB floating
WEA	1'b0	ROM does not support write operations, connecting WEA/WEB to low level
WEB		

3. Other signals for read/write ports: Connect the clock, clock enable, output register clock enable, and data register reset of the read/write ports to the corresponding GTP ports A/B respectively; the ports A/B must be connected to the same signal.
4. Other unused parameters: Do not set the other unused parameters, use default values;
5. Other unused ports: Leave the other unused ports floating (ports that must be connected even if not used have been described above).

The configured GTP is as follows:

```
GTP_DRM18K #(
.DATA_WIDTH_A    (18),
.DATA_WIDTH_B    (18),
.DOA_REG         (1),
.DOB_REG         (1),
.RAM_MODE        ("ROM"),
.INIT_00         ({18'h0000f, 18'h0000e, 18'h0000d, 18'h0000c, 18'h0000b, 18'h0000a, 18'h00009, 18'h00008, 18'h00007, 18'h00006, 18'h00005, 18'h00004, 18'h00003, 18'h00002, 18'h00001, 18'h00000}),
...
...
.INIT_3F         ({18'h003ff, 18'h003fe, 18'h003fd, 18'h003fc, 18'h003fb, 18'h003fa, 18'h003f9, 18'h003f8, 18'h003f7, 18'h003f6, 18'h003f5, 18'h003f4, 18'h003f3, 18'h003f2, 18'h003f1, 18'h003f0})
) GTP_DRM18K_inst (
.DOA             (do[17:0]    ), // OUTPUT[17:0]
.DOB             (            ), // OUTPUT[17:0]
.ADDRA           ({addr[9:0],4'b1111}), // INPUT[13:0]
.ADDRB           ({addr[9:0],4'b1111}), // INPUT[13:0]
.CSA             (3'b0        ), // INPUT[2:0]
.CSB             (3'b0        ), // INPUT[2:0]
.DIA             (            ), // INPUT[35:0]
.DIB             (            ), // INPUT[35:0]
```

```
.ADDRA_HOLD      (1'b0          ), // INPUT
.ADDRB_HOLD      (1'b0          ), // INPUT
.CEA             (ce            ), // INPUT
.CEB             (ce            ), // INPUT
.CLKA            (clk           ), // INPUT
.CLKB            (clk           ), // INPUT
.ORCEA           (orce          ), // INPUT
.ORCEB           (orce          ), // INPUT
.RSTA            (rst           ), // INPUT
.RSTB            (rst           ), // INPUT
.WEA             (1'b0          ), // INPUT
.WEB             (1'b0          )  // INPUT
);
```

6.8.2 Configuration for Multiple DRMs Cascaded

Multiple 18K DRMs in ROM mode are cascaded in the same way as DP mode. See details in 3.9.2 Configuration for Multiple DRMs Cascaded .

Chapter 7 FIFO Mode

7.1 Description of Mode

DRM_FIFO consists of DRM and DRM_FIFO_CTRL modules, and the DTM_FIFO_CTRL module includes: read/write pointer generation and status flag generation logic. The top-level structural block diagram of FIFO is as follows:

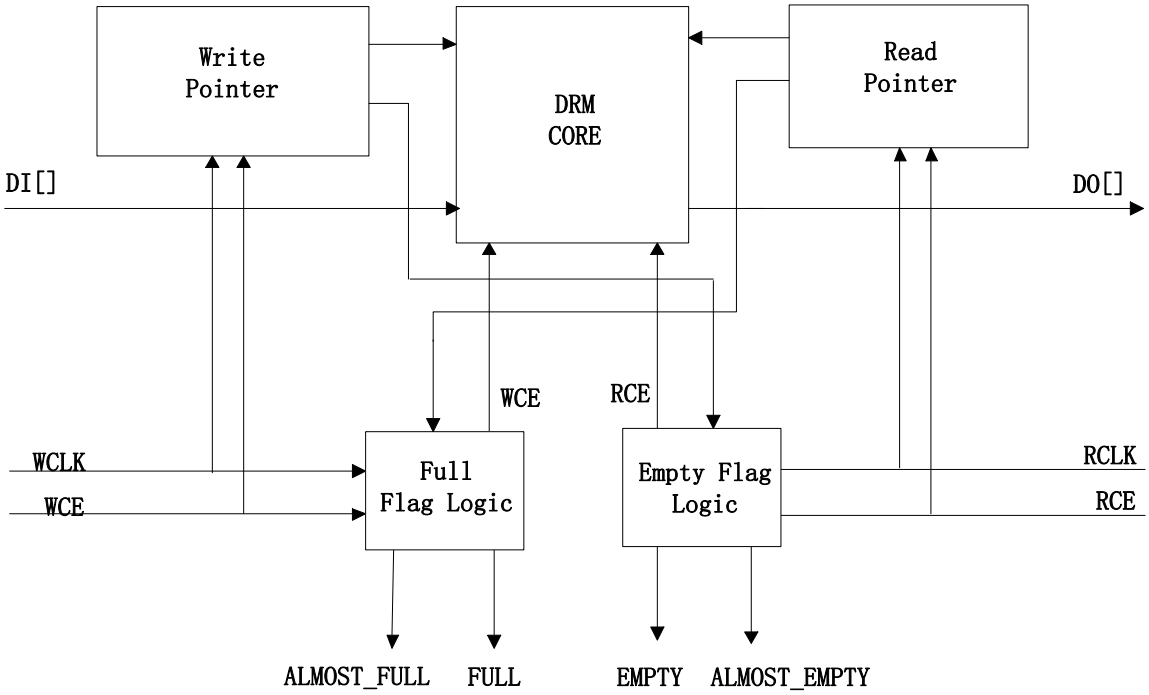

Figure 7-1 Top-Level Structural Block Diagram of FIFO

Users can instantiate GTP_DRM to configure it to SDP mode and then implement the DRM_FIFO_CTRL module through user logic, or directly instantiate GTP_FIFO18K to implement DRM_FIFO.

GTP_FIFO is divided into synchronous and asynchronous FIFO, determined by the parameter SYNC_FIFO. When the value of SYNC_FIFO is "TRUE", it enters synchronous FIFO mode; when the value of SYNC_FIFO is "FALSE", it enters asynchronous FIFO mode. In this mode, the read and write clock domains of the asynchronous FIFO are independent of each other. Write operations are synchronized to the write clock and read operations are synchronized to the read clock.

In asynchronous FIFO mode, when write enable is active and FIFO is not full, data is written to FIFO on the rising edge of the write clock; when read enable is active and FIFO is not empty, data is read from FIFO on the rising edge of the read clock. When the FIFO is full, a full signal is

generated and synchronized to the write clock domain. When the FIFO is empty, an empty signal is generated and synchronized to the read clock domain.

In synchronous FIFO mode, the read and write share the same clock. When write enable is active and FIFO is not full, data is written to FIFO on the rising edge of the clock; when read enable is active and FIFO is not empty, data is read from FIFO on the rising edge of the clock. A full signal is generated when the FIFO is written full, and an empty signal is generated when the FIFO is read empty. See appendix for details of 8.5 DRM_FIFO Bit Width Mode.

7.2 Port Descriptions

7.2.1 Asynchronous FIFO (Without Rewrite/Resend Function)

The data flow in this mode is shown in the figure below:

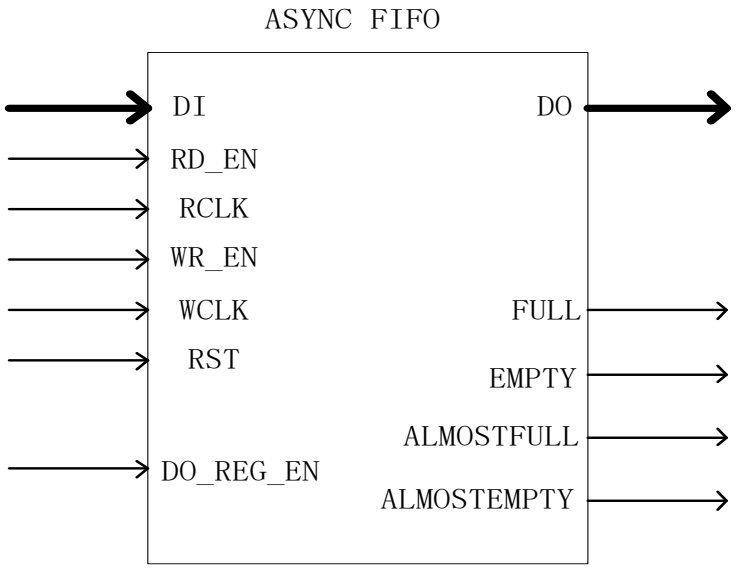

Figure 7-2 Asynchronous FIFO Mode (Without Rewrite/Resend Function)

Table 7-1 Port Description of Asynchronous FIFO Mode (Without Rewrite/Resend Function)

Port Name	Direction	Description
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high
DO_REG_EN	Input	Read register enable signal, active-high
DO	Output	Data output
FULL	Output	Full flag

Port Name	Direction	Description
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.2.2 Asynchronous FIFO (with Rewrite Function)

Rewrite function: When a wr_err packet error signal is detected at the asynchronous FIFO write port, the write address of the write port reverts back to the write address where the packet header is located. The rewrite function can be enabled separately, and the "wr_eop" requires occupying one bit of the DI input data port.

The data flow in this mode is shown in the figure below:

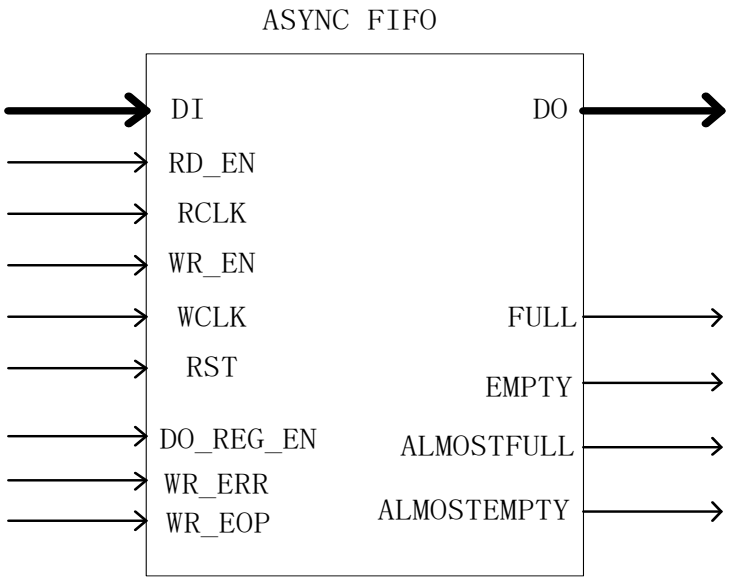

Figure 7-3 Asynchronous FIFO Mode (with Rewrite Function)

Table 7-2 Port Description of Asynchronous FIFO Mode (with Rewrite Function)

Port Name	Direction	Description
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high
DO_REG_EN	Input	Read register enable signal, active-high
WR_ERR	Input	Write port error packet indicator
WR_EOP	Input	End-of-packet signal indicator
DO[N:0]	Output	Data output

Port Name	Direction	Description
FULL	Output	Full flag
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.2.3 Asynchronous FIFO (with Rewrite/Resend Function)

Rewrite function: When a wr_err packet error signal is detected at the asynchronous FIFO write port, the write address of the write port reverts back to the write address where the packet header is located.

Resend function: When the asynchronous FIFO read clock detects an rd_nak signal, the read address reverts back to the address where the packet header is located.

Both the rewrite and resend functions need to be enabled simultaneously. The "wr_eop" requires occupying one bit of the DI input data port.

The data flow in this mode is shown in the figure below:

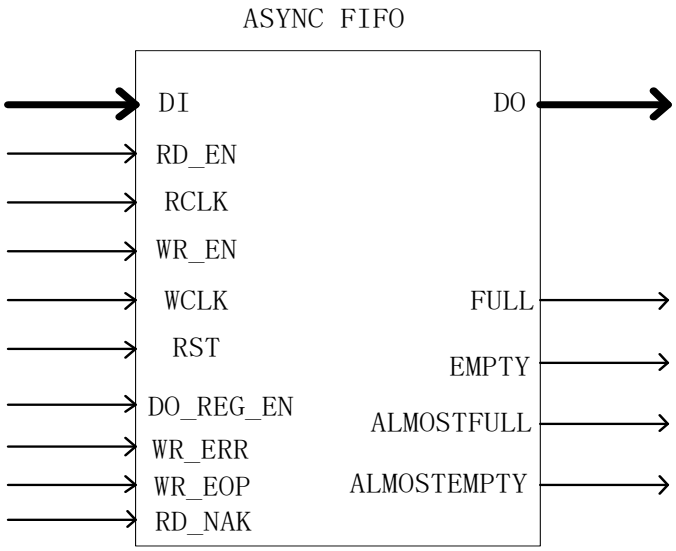

Figure 7-4 Asynchronous FIFO Mode (with Rewrite/Resend Function)

Table 7-3 Port Description of Asynchronous FIFO Mode (with Rewrite/Resend Function)

Port Name	Direction	Description
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high

Port Name	Direction	Description
DO_REG_EN	Input	Read register enable signal, active-high
WR_ERR	Input	Write port error packet indicator
WR_EOP	Input	End-of-packet signal indicator
RD_NAK	Input	Read port resend indicator
DO	Output	Data output
FULL	Output	Full flag
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.2.4 Synchronous FIFO (Without Rewrite/Resend Function)

In synchronous FIFO mode, the read and write share the same clock.

When write enable is active and FIFO is not full, data is written to FIFO on the rising edge of the clock.

When read enable is active and the FIFO is not empty, data is read from FIFO on the rising edge of the clock.

A full signal is generated when the FIFO is written full, and an empty signal is generated when the FIFO is read empty.

The data flow in this mode is shown in the figure below:

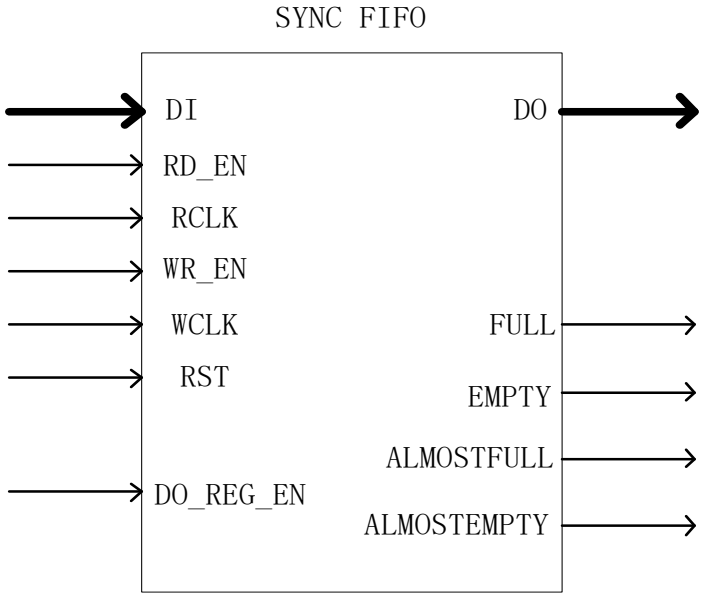

Figure 7-5 Synchronous FIFO Mode (Without Rewrite/Resend Function)

Table 7-4 Port Description of Synchronous FIFO Mode (Without Rewrite/Resend Function)

Port Name	Direction	Description
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high
DO_REG_EN	Input	Read register enable signal, active-high
DO	Output	Data output
FULL	Output	Full flag
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.2.5 Synchronous FIFO (with Rewrite Function)

Rewrite function: When a wr_err packet error signal is detected at the synchronous FIFO write port, the write address of the write port reverts back to the write address where the packet header is located. The rewrite function can be enabled separately, and the "wr_eop" requires occupying one bit of the DI input data port.

The data flow in this mode is shown in the figure below:

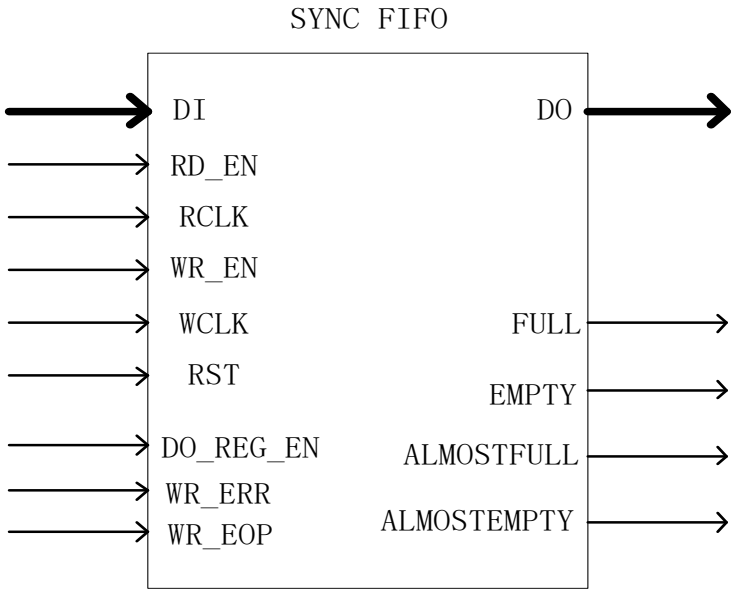

Figure 7-6 Synchronous FIFO Mode (with Rewrite Function)

Table 7-5 Port Description of Synchronous FIFO Mode (with Rewrite Function)

Port Name	Direction	Description
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high
DO_REG_EN	Input	Read register enable signal, active-high
WR_ERR	Input	Write port error packet indicator
WR_EOP	Input	End-of-packet signal indicator
DO	Output	Data output
FULL	Output	Full flag
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.2.6 Synchronous FIFO (with Rewrite/Resend Function)

Rewrite function: When a wr_err packet error signal is detected at the FIFO write port, the write address of the write port reverts back to the write address where the packet header is located.

Resend function: When the synchronous FIFO read clock detects an rd_nak signal, the read address reverts back to the address where the packet header is located.

Both the rewrite and resend functions need to be enabled simultaneously. The "wr_eop" requires occupying one bit of the DI input data port.

The data flow in this mode is shown in the figure below:

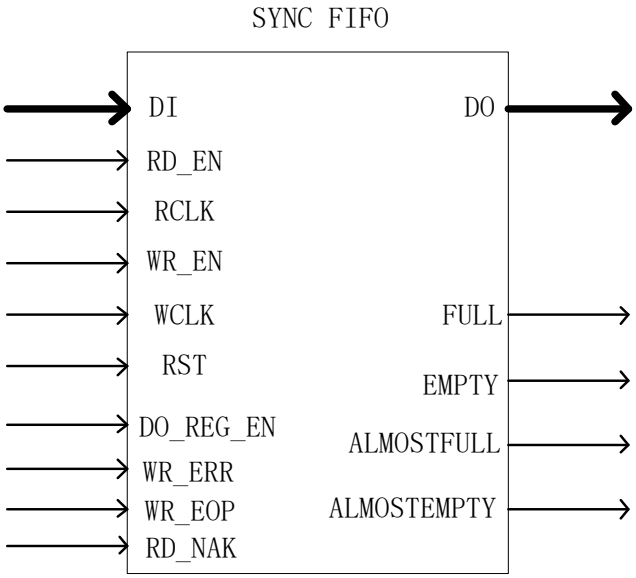

Figure 7-7 Synchronous FIFO (with Rewrite/Resend Function)

Table 7-6 Port Description of Synchronous FIFO (with Rewrite/Resend Function)

Port Name	**Direction**	**Description**
DI	Input	Data input
RD_EN	Input	Read enable signal, active-high
RCLK	Input	Read clock signals
WR_EN	Input	Write enable signal, active-high
WCLK	Input	Write clock signal
RST	Input	Reset signal, active-high
DO_REG_EN	Input	Read register enable signal, active-high
WR_ERR	Input	Write port error packet indicator
WR_EOP	Input	End-of-packet signal indicator
RD_NAK	Input	Read port resend indicator
DO[N:0]	Output	Data output
FULL	Output	Full flag
EMPTY	Output	Empty flag
ALMOSTFULL	Output	Almost full flag
ALMOSTEMPTY	Output	Almost empty flag

7.3 Bit Width Combinations

The port width of FIFO is determined by the parameter DATA_WIDTH in the GTP; for example, when the value of DATA_WIDTH is 4, the port data width is set to 4 bits. GTP_FIFO does not support mixed bit widths.

The table below shows the allowed bit widths for 18K GTP_FIFO mode.

Table 7-7 Allowed Bit Widths for 18K GTP_FIFO Mode

Mode	16Kx1	8Kx2	4Kx4	2Kx8	1Kx16	512x32	2Kx9	1Kx18	512x36
FIFO	√	√	√	√	√	√	√	√	√

Note: √ indicates the supported bit width combinations.

7.4 Read and Write Operations

7.4.1 Write Timing to Empty FIFO

The EMPTY signal indicates the FIFO is empty. When WR_EN is active and data is successfully written, in synchronous FIFO, the EMPTY signal is set to 0 after one RCLK clock cycle; in asynchronous FIFO, the EMPTY signal is set to 0 after two RCLK clock cycles. When WR_EN remains active, the ALMOST_EMPTY signal is set to 0 with a delay based on the configuration of ALMOST_EMPTY_OFFSET.

Write timing to empty asynchronous FIFO is shown below:

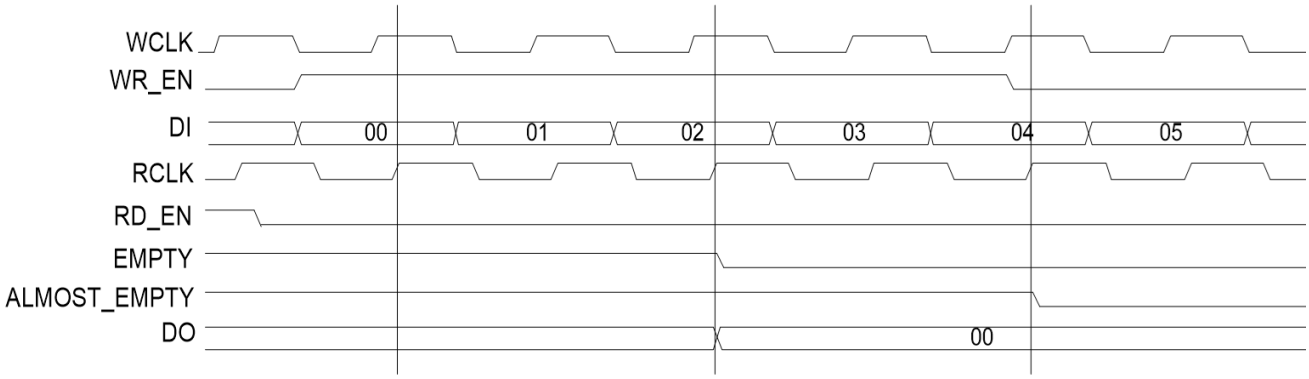

Figure 7-8 Write Timing to Empty Asynchronous FIFO

7.4.2 Write Timing to Almost Full Asynchronous FIFO

When the FIFO is almost full, ALMOST_FULL is set to 1 in advance, according to the configuration of ALMOST_FULL_OFFSET. When the FIFO is full, the write pointer will no longer be incremented.

Write timing to almost full asynchronous FIFO is shown below:

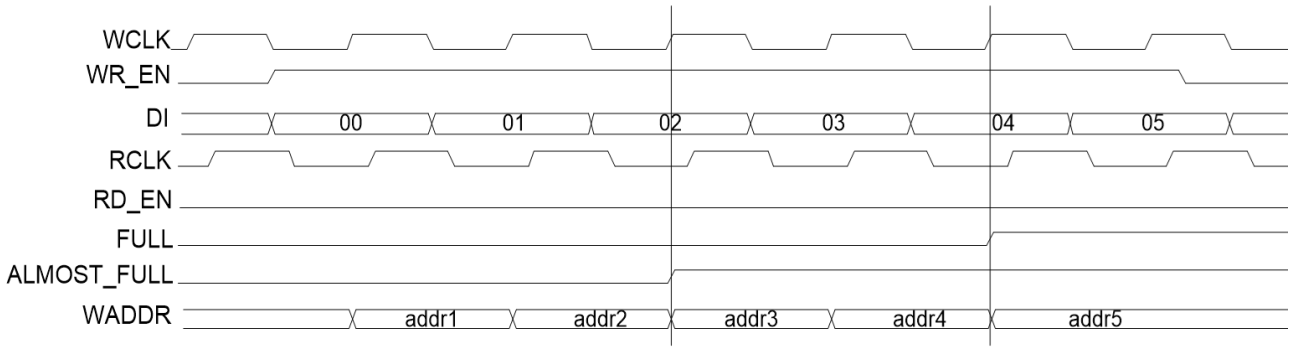

Figure 7-9 Write Timing to Almost Full Asynchronous FIFO

7.4.3 Read Timing from Full Asynchronous FIFO

The FULL signal indicates the FIFO is full. After data is read out when RD_EN is active, the FULL signal is set to 0 within one to two WCLK clock cycles. When RD_EN remains active, the ALMOST_FULL signal is set to 0 with a delay based on the configuration of ALMOST_FULL_OFFSET. Read timing from full FIFO is shown below:

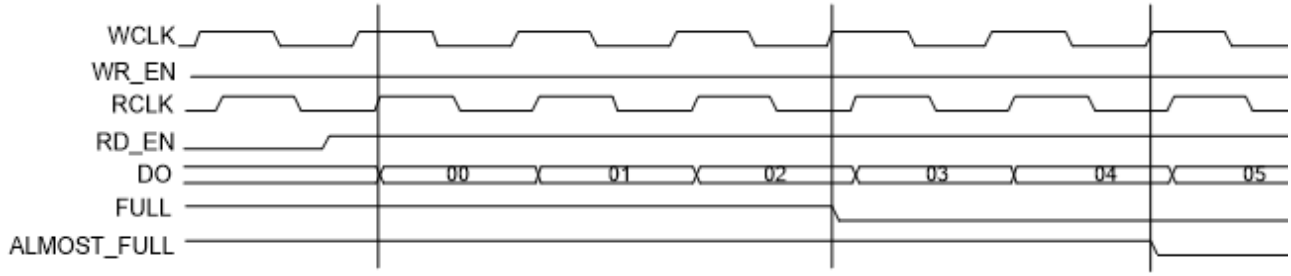

Figure 7-10 Read Timing from Full Asynchronous FIFO

7.4.4 Read Timing from Almost Empty FIFO

When the FIFO is almost empty, ALMOST_EMPTY is set to 1 in advance, based on the configuration of ALMOST_EMPTY_OFFSET. When the FIFO is empty, the read pointer will no longer be incremented. Read timing from almost empty asynchronous FIFO is shown below:

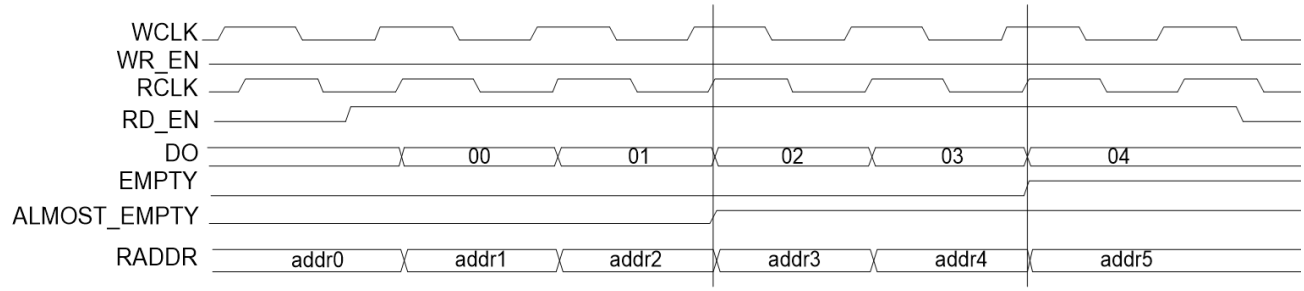

Figure 7-11 Read Timing from Almost Empty Asynchronous FIFO

7.4.5 Write Rewrite Timing

When the packet end indicator WR_EOP is set to 1 and the WR_ERR packet error indicator is active, the write pointer reverts to the packet header's position in the next WCLK cycle. Write rewrite timing is shown below:

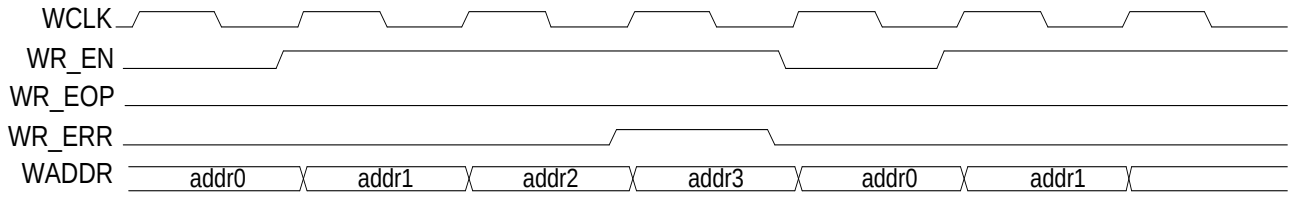

Figure 7-12 Write Rewrite Timing

7.4.6 Read Resend Timing

When the packet end indicator WR_EOP is set, the previous packet end address is marked, and the read pointer reverts to the current packet header's address after the RD_NAK signal is set.

Read resend timing is shown below:

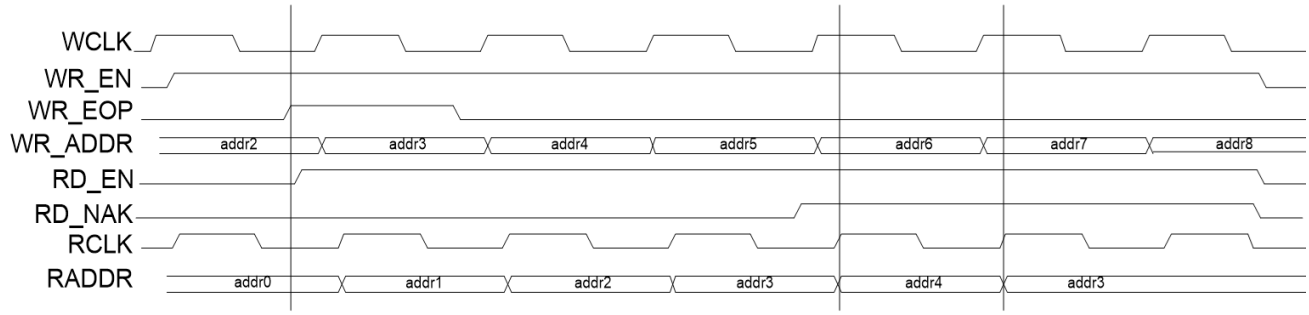

Figure 7-13 Read Resend Timing

7.4.7 Write to Empty FIFO with Rewrite Function

The EMPTY signal indicates the FIFO is empty. When WR_EN is active and the first WR_EOP packet end signal is active (WR_ERR equals 0), the EMPTY signal is cleared within the RCLK clock domain.

Write timing to empty FIFO with rewrite function is shown below:

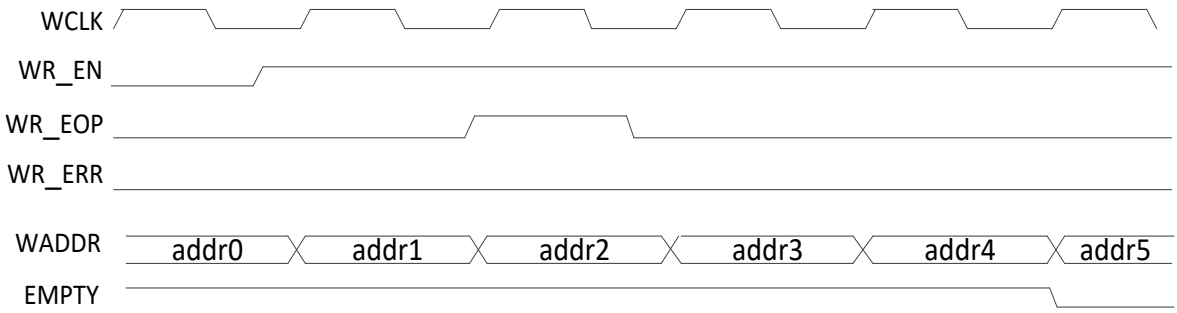

Figure 7-14 Write Timing to Empty FIFO with Rewrite Function

7.4.8 Write to Almost Full FIFO with Rewrite/Resend Function

The FULL signal indicates the FIFO is full, and when the read port reads the data at the read address containing WR_EOP and RD_NAK is "0", it indicates that the packet data read is valid. A FULL signal is generated when the data read is less than a full packet and the write pointer points to the address of the last valid packet read.

Write timing to almost full FIFO with Rewrite/Resend function is shown below:

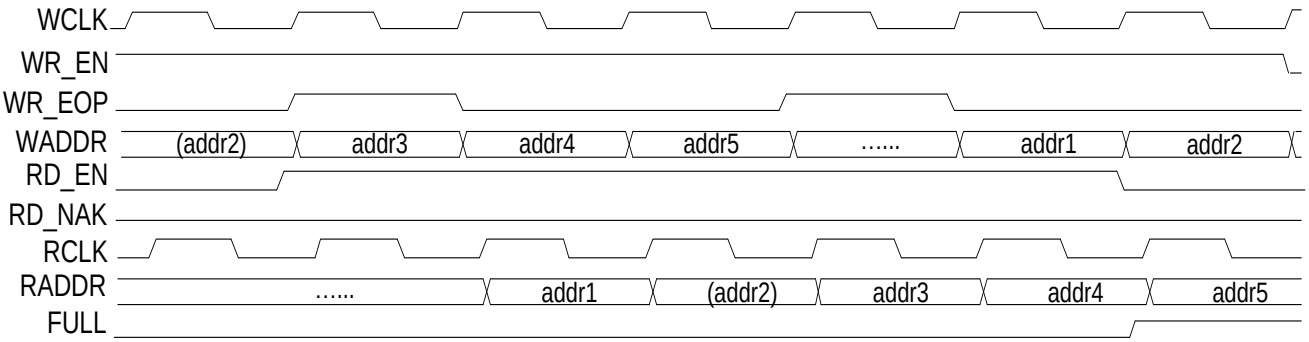

Figure 7-15 Write Timing to Almost Full FIFO with Rewrite/Resend Function

7.5 Internal Registers

The internal registers of FIFO mode are the same as those of DP mode. See details in 3.6 Internal Registers.

7.6 Reset signal

The reset signal of FIFO mode is the same as that of DP mode. See details in 3.7 Reset Signal.

7.7 Global Signal Timing

The global signal timing for FIFO mode is the same as that of DP mode. See details in 3.8 Global

Signal Timing.

7.8 Application Examples

This section illustrates the configuration steps for a single 18K GTP_FIFO. Users can also directly generate the IP of GTP_DRM FIFO using the IP Compiler tool embedded in the Pango Design Suite software, see the "Logos Family DRM RAM/FIFO IP User Guide" attached in the IP Compiler tool.

The example is for asynchronous FIFO mode, with the port data width configured as 1Kx18, and output registers enabled.

Configure a single 18K FIFO as follows:

1. Configure the FIFO parameters as described in the table below:

Table 7-8 FIFOz Parameter Configuration

Parameter Name	Configuration Value	Description
DATA_WIDTH	18	Configure the port to 1Kx18 mode
DO_REG	1	Enable port output register
SYNC_FIFO	"FALSE"	Asynchronous FIFO mode
USE_EMPTY	1	Empty flag bit enable
USE_FULL	1	Full flag bit enable
ALMOST_EMPTY_OFFSET	'd4	Almost empty signal is set when the number of data items stored in the FIFO is less than or equal to 4
ALMOST_FULL_OFFSET	'd1020	Almost full signal is set when the number of data items stored in the FIFO is larger than or equal to 1020

2. Connect the FIFO ports as shown in the table below:

Table 7-9 FIFO Port Connections

Port	Interfacing Signals	Description
DI[35:0]	di[17:0]	Connect the input data signal di[17:0] to DI[17:0], leaving high-bit DI ports floating or connected to a low level
DO[35:0]	do[17:0]	Connect the input data signal do[17:0] to DO[17:0], leaving the high-bit DO ports floating

3. Other signals for read/write ports: Connect the clock, read/write clock enable, output register clock enable, data register reset, and indicator of the read/write ports to the corresponding GTP read/write ports respectively;
4. Other unused parameters: Do not set the other unused parameters, use default values;
5. Other unused ports: Leave the other unused ports floating (ports that must be connected even if not used have been described above).

The configured GTP is as follows:

```
GTP_FIFO18K #(
.DATA_WIDTH                 (18),
.DO_REG                     (1),
.ALMOST_FULL_OFFSET     ('d1020),
.ALMOST_EMPTY_OFFSET    ('d4),
.USE_EMPTY                  (1),
.USE_FULL                   (1),
.SYNC_FIFO                  ("FALSE")
) GTP_FIFO18K_inst (
.DO                         (do[17:0]       ), // OUTPUT[35:0]
.DI                         (di[17:0]       ), // INPUT[35:0]
.ALMOST_EMPTY               (almost_empty), // OUTPUT
.ALMOST_FULL                (almost_full ), // OUTPUT
.EMPTY                      (empty          ), // OUTPUT
.FULL                       (full           ), // OUTPUT
.ORCE                       (orce           ), // INPUT
.RCE                        (rce            ), // INPUT
.RCLK                       (rclk           ), // INPUT
.RST                        (rst            ), // INPUT
.WCE                        (wce            ), // INPUT
.WCLK                       (wclk           )    // INPUT
);
```

Chapter 8 Appendix

8.1 Address and Data Port Mapping

The address and data port mapping corresponding to the 18K and 9K modes of module memory is shown in Table 8-1 and Table 8-2 respectively.

Table 8-1 Address and Data Port Mapping for Memory in 18K DRM Mode

Combinations Mode	Port A Address	Port B Address	Port A data input bus	Port B data input bus	Port A data output bus	Port B data output bus
18K DRM memory mode						
16Kx1	ADDRA [13:0]	ADDRB [13:0]	DIA[0]	DIB[0]	DOA[X], where X is any bit between 16~9, 7~0	DOB[X], where X is any bit between 16~9, 7~0
8Kx2	ADDRA [12:1]	ADDRB [12:1]	DIA[1:0]	DIB[1:0]	DOA[X+1:X], X=0,2,4,9,11,12,15	DOB[X+1:X],X=0,2,4,9,11,12,15
4Kx4	ADDRA [12:2]	ADDRB [12:2]	DIA[3:0]	DIB[3:0]	DOA[X+3:X], X=0,4,9,12	DOB[X+3:X],X=0,4,9,12
2Kx8	ADDRA [12:3]	ADDRB [12:3]	DIA[7:0]	DIB[7:0]	DOA[7:0] or DOA[16:9]	DOB[7:0] or DOB[16:9]
1Kx16	ADDRA [12:4]	ADDRB [12:4]	DIA[16:9,7:0]	DIB[16:9,7:0]	DOA[16:9,7:0]	DOB[16:9,7:0]
2Kx9	ADDRA [12:3]	ADDRB [12:3]	DIA[8:0]	DIB[8:0]	DOA[8:0] or DOA[17:9]	DOB[8:0] or DOB[17:9]
1Kx18	ADDRA [12:4]	ADDRB [12:4]	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]
512x32	ADDRA [12:5]	ADDRB [12:5]	{DIB[16:9,7:0], DIA[16:9,7:0]}	N/A	N/A	{DOB[16:9,7:0], DOA[16:9,7:0]}
512x36	ADDRA [12:5]	ADDRB [12:5]	{DIB[17:0], DIA[17:0]}	N/A	N/A	{DOB[17:0], DOA[17:0]}

Table 8-2 Address and Data Port Mapping for Memory in 9K DRM Mode

Combinations Mode	Port A Address	Port B Address	Port A data input bus	Port B data input bus	Port A data output bus	Port B data output bus
DRM port mode (9K module 0)						
8Kx1	ADDRA [12:0]	ADDRB [12:0]	DIA[0]	DIB[0]	DOA[X], where X is any bit between 16~9, 7~0	DOB[X], where X is any bit between 16~9, 7~0
4Kx2	ADDRA [12:1]	ADDRB [12:1]	DIA[1:0]	DIB[1:0]	DOA[X+1:X], X=0,2,4,9,11, 12,15	DOB[X+1:X], X=0,2,4,9,11, 12,15
2Kx4	ADDRA [12:2]	ADDRB [12:2]	DIA[3:0]	DIB[3:0]	DOA[X+3:X], X=0,4,9,12	DOB[X+3:X], X=0,4,9,12
1Kx8	ADDRA [12:3]	ADDRB [12:3]	DIA[7:0]	DIB[7:0]	DOA[7:0] or DOA[16:9]	DOB[7:0] or DOB[16:9]
1Kx9	ADDRA [12:3]	ADDRB [12:3]	DIA[8:0]	DIB[8:0]	DOA[8:0] or DOA[17:9]	DOB[8:0] or DOB[17:9]
512x16	ADDRA [12:4]	ADDRB [12:4]	DIA[16:9,7:0]	DIB[16:9,7:0]	DOA[16:9,7:0]	DOB[16:9,7:0]
512x18	ADDRA [12:4]	ADDRB [12:4]	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]
256x32	ADDRA [12:5]	ADDRB [12:5]	{DIB[16:9,7:0], DIA[16:9,7:0]}	N/A	N/A	{DOB[16:9,7:0], DOA[16:9,7:0]}
256x36	ADDRA [12:5]	ADDRB [12:5]	{DIB[17:0], DIA[17:0]}	N/A	N/A	{DOB[17:0], DOA[17:0]}

The data address mappings for data widths of x1, x2, x4, x8, x16, x32 and x9, x18, x36 are shown in Table 8-3 and Table 8-4 respectively.

Table 8-3 Data Address Mapping for Different Data Widths (x1, x2, x4, x8, x16, x32)

Data Width	Least-bit port address (compared with the least-bit port address in the maximum data width mode)																													
32	0																													
16	1																0													
8	3								2								1								0					
4	7				6				5				4				3				2				1				0	
2	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 8-4 Data Address Mapping for Different Data Widths (x9, x18, x36)

Data Width	Least-bit port address (compared with the least-bit port address in the maximum data width mode)																																	
36	0																																	
18	1																			0														
9	3									2										1												0		
Index	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

8.2 Initialization Configuration Parameter Mapping

INIT_XX is the initialization configuration parameter for DRM to initialize DRM and determine the initial value of the memory. By default, the initial values of DRM are all set to 0. GTP_DRM18K includes 64 initialization configuration parameters from INIT_00 to INIT_3F, and GTP_DRM9K includes 32 initialization configuration parameters from INIT_00 to INIT_1F. Each configuration parameter is 288 bits, configuring the 288 bits of memory at the corresponding DRM address.

The following formulas and the table show the position of the memory bit mapped by each INIT_XX (YY in the formula is the decimal number converted from the hexadecimal number XX):

Highest bit = [(YY+1)*288)]-1

Lowest bit = (YY)*288

Table 8-5 INIT_XX Parameter Mapping

Parameter	**Highest Bit**	**Lowest Bit**
INIT_00	287	0
INIT_01	575	288
INIT_02	863	576
.....	.....	.....
INIT_10	4895	4608
.....	.....	.....
INIT_1F	9215	8928
.....	.....	.....
INIT_3F	18431	18144

When the data width is 2^N, namely 1/2/4/8/16/32 bits, in the above table, every 9 bits of INIT_XX data only maps the lower 8 bits to memory; when the bit width is 9×2^N, namely 9/18/36 bits, all INIT_XX data is mapped to memory.For example, when the DRM is configured as 8x2K, INIT_00[7:0] is the initial value of the data is read out when the address addr corresponding to 0, and INIT_00[16:9] is the initial value of the data is read out when the address addr corresponding to 1; when the DRM is configured as 9x2K, INIT_00[8:0] is the initial value of the data is read out when the address addr corresponding to 0, and INIT_00[17:9] is the initial value of the data is read out when the address addr corresponding to 1.

8.3 DRM Port Signal Description

1. Address bus (ADDRA[13:0], ADDRB[13:0]): Ports A/B read/write address. The effective number of bits of the address bus depends on the read and write data width. For example, if DRM18K is configured to a read/write data width of 18 bits, the address bit width is 10 bits, with effective address inputs as ADDRA[13:4], ADDRB[13:4]. The remaining ADDRA[3:0] and ADDRB[3:0] need to be connected to a high level or byte enable signal (the byte enable signal reuses the lower-bit address, which needs to be connected to the byte enable signal when using the byte enable function). For detailed address mapping relationships, refer to the Appendix 8.1 Address and Data Port Mapping.
2. Address hold signals (ADDRA_HOLD, ADDRB_HOLD): When this port is set to a high level, the corresponding port address input remains the previous value, and does not vary with input address; when this port is set to a low level, the corresponding port address input does vary with input address. When the address hold is not in use, this port should be connected to a low level.
3. Data buses (DIA[17:0], DIB[17:0], DOA[17:0], DOB[17:0]): Read and write data ports for ports A and B. For detailed data port mapping, refer to Appendix 8.1 Address and Data Port Mapping.
4. Address extension signals (CSA[2:0], CSB[2:0]): Multiple DRMs can be cascaded and extended into larger DP RAMs, SDP RAMs, SP RAMs, ROMs, or FIFOs. For this, the ports A and B of the DRM provides an additional 3-bit address extension, commonly used for deeply-extended applications.
5. Write enable signals (WEA, WEB): Write operation control signals for the DRM. For its detailed write operation timing, refer to 3.4 Read and Write Operations.
6. Clock-related signals (CLKA, CLKB, CEA, CEB): Clock and enable signals for ports A and B of the DRM. Under SDP mode, CLKA and CEA are write clock and its enable signals, CLKB and CEB are read clock and its enable signals.
7. Output register related signals (ORCEA, ORCEB, RSTA, RSTB): Enable and reset signals for the output registers of ports A and B.

8.4 Additional Information Bit for Bytes

In data width modes of x9, x18, and x36, there is one additional information storage bit per byte, as shown in the following table:

Table 8-6 List of Additional Information Bit for Bytes

Port	Mode	Port Descriptions		Data input	
		Byte	Additional Information Bit	Byte	Additional Information Bit
A	18K, 9K	DIA[7:0]	DIA[8]	dia[7:0]	dia[8]
	18K, 9K	DIA[16:9]	DIA[17]	dia[16:9]	dia[17]
B	18K, 9K	DIB[7:0]	DIB[8]	dib[7:0]	dib[8]
	18K, 9K	DIB[16:9]	DIB[17]	dib[16:9]	dib[17]

8.5 DRM_FIFO Bit Width Mode

In FIFO Rewrite/Resend mode, the wr_eop signal occupies one data bit, and the FIFO data width needs to be configured as follows:

Table 8-7 Address Depth and Data Width Configuration for FIFO Rewrite/Resend Mode

FIFO Address and Bit Width Configuration	Available Bits for FIFO	wr_eop Occupied Bit
2Kx9	DI[7:0]	DI[8]
1Kx18	DI[17:9],DI[7:0]	DI[8]
512x36	DI[35:27],DI[25:0]	DI[26]

DRM_FIFO with Rewrite/Resend function supports the data widths as shown in the following table:

Table 8-8 Data Width List for Asynchronous/Synchronous FIFO Mode with Rewrite/Resend Function

		Write Port		
		2Kx9	1Kx18	512x36
Read Port	2Kx9	√		
	1Kx18		√	
	512x36			√

DRM_FIFO without Rewrite/Resend function supports the data widths as shown in the following table:

Table 8-9 Data Width List for Asynchronous/Synchronous FIFO Mode Without Rewrite/Resend Function

		Write Port					
		16Kx1	8Kx2	4Kx4	2Kx9 (2Kx8)	1Kx18 (1Kx16)	512x36 (512x32)
Read	16Kx1	√					

		Write Port					
		16Kx1	**8Kx2**	**4Kx4**	**2Kx9 (2Kx8)**	**1Kx18 (1Kx16)**	**512x36 (512x32)**
	8Kx2		√				
	4Kx4			√			
	2Kx9(2Kx8)				√		
	1Kx18(1Kx16)					√	
	512x36(512x32)						√

8.6 GTP_DRM9K Instantiation Template

```
GTP_DRM9K #(
    .GRS_EN("TRUE"),
    .DATA_WIDTH_A(18),
    .DATA_WIDTH_B(18),
    .DOA_REG(0),
    .DOB_REG(0),
    .DOA_REG_CLKINV(0),
    .DOB_REG_CLKINV(0),
    .RST_TYPE("SYNC"),
    .RAM_MODE("TRUE_DUAL_PORT"),
    .WRITE_MODE_A("NORMAL_WRITE"),
    .WRITE_MODE_B("NORMAL_WRITE"),
    .INIT_00(288'b0),
    .INIT_01(288'b0),
    .INIT_02(288'b0),
    .INIT_03(288'b0),
    .INIT_04(288'b0),
    .INIT_05(288'b0),
    .INIT_06(288'b0),
    .INIT_07(288'b0),
    .INIT_08(288'b0),
    .INIT_09(288'b0),
    .INIT_0A(288'b0),
    .INIT_0B(288'b0),
    .INIT_0C(288'b0),
```

```
    .INIT_0D(288'b0),
    .INIT_0E(288'b0),
    .INIT_0F(288'b0),
    .INIT_10(288'b0),
    .INIT_11(288'b0),
    .INIT_12(288'b0),
    .INIT_13(288'b0),
    .INIT_14(288'b0),
    .INIT_15(288'b0),
    .INIT_16(288'b0),
    .INIT_17(288'b0),
    .INIT_18(288'b0),
    .INIT_19(288'b0),
    .INIT_1A(288'b0),
    .INIT_1B(288'b0),
    .INIT_1C(288'b0),
    .INIT_1D(288'b0),
    .INIT_1E(288'b0),
    .INIT_1F(288'b0),
    .INIT_FILE("NONE"),
    .BLOCK_X(0),
    .BLOCK_Y(0),
    .RAM_DATA_WIDTH(9),
    .RAM_ADDR_WIDTH(10),
    .INIT_FORMAT("BIN")
) GTP_DRM9K_inst (
    .DOA(),          // OUTPUT[17:0]
    .DOB(),          // OUTPUT[17:0]
    .ADDRA(),        // INPUT[12:0]
    .ADDRB(),        // INPUT[12:0]
    .DIA(),          // INPUT[17:0]
    .DIB(),          // INPUT[17:0]
    .ADDRA_HOLD(),// INPUT
    .ADDRB_HOLD(),// INPUT
```

```
    .CEA(),          // INPUT
    .CEB(),          // INPUT
    .CLKA(),         // INPUT
    .CLKB(),         // INPUT
    .ORCEA(),        // INPUT
    .ORCEB(),        // INPUT
    .RSTA(),         // INPUT
    .RSTB(),         // INPUT
    .WEA(),          // INPUT
    .WEB()           // INPUT
);
```

8.7 GTP_DRM18K Instantiation Template

```
GTP_DRM18K #(
    .GRS_EN("TRUE"),
    .CSA_MASK('b000),
    .CSB_MASK('b000),
    .DATA_WIDTH_A(18),
    .DATA_WIDTH_B(18),
    .DOA_REG(0),
    .DOB_REG(0),
    .DOA_REG_CLKINV(0),
    .DOB_REG_CLKINV(0),
    .RST_TYPE("SYNC"),
    .RAM_MODE("TRUE_DUAL_PORT"),
    .WRITE_MODE_A("NORMAL_WRITE"),
    .WRITE_MODE_B("NORMAL_WRITE"),
    .WRITE_COLLISION_ARBITER("NULL"),
    .INIT_00(288'b0),
    .INIT_01(288'b0),
    .INIT_02(288'b0),
    .INIT_03(288'b0),
    .INIT_04(288'b0),
    .INIT_05(288'b0),
```

```
.INIT_06(288'b0),
.INIT_07(288'b0),
.INIT_08(288'b0),
.INIT_09(288'b0),
.INIT_0A(288'b0),
.INIT_0B(288'b0),
.INIT_0C(288'b0),
.INIT_0D(288'b0),
.INIT_0E(288'b0),
.INIT_0F(288'b0),
.INIT_10(288'b0),
.INIT_11(288'b0),
.INIT_12(288'b0),
.INIT_13(288'b0),
.INIT_14(288'b0),
.INIT_15(288'b0),
.INIT_16(288'b0),
.INIT_17(288'b0),
.INIT_18(288'b0),
.INIT_19(288'b0),
.INIT_1A(288'b0),
.INIT_1B(288'b0),
.INIT_1C(288'b0),
.INIT_1D(288'b0),
.INIT_1E(288'b0),
.INIT_1F(288'b0),
.INIT_20(288'b0),
.INIT_21(288'b0),
.INIT_22(288'b0),
.INIT_23(288'b0),
.INIT_24(288'b0),
.INIT_25(288'b0),
.INIT_26(288'b0),
.INIT_27(288'b0),
```

```
    .INIT_28(288'b0),
    .INIT_29(288'b0),
    .INIT_2A(288'b0),
    .INIT_2B(288'b0),
    .INIT_2C(288'b0),
    .INIT_2D(288'b0),
    .INIT_2E(288'b0),
    .INIT_2F(288'b0),
    .INIT_30(288'b0),
    .INIT_31(288'b0),
    .INIT_32(288'b0),
    .INIT_33(288'b0),
    .INIT_34(288'b0),
    .INIT_35(288'b0),
    .INIT_36(288'b0),
    .INIT_37(288'b0),
    .INIT_38(288'b0),
    .INIT_39(288'b0),
    .INIT_3A(288'b0),
    .INIT_3B(288'b0),
    .INIT_3C(288'b0),
    .INIT_3D(288'b0),
    .INIT_3E(288'b0),
    .INIT_3F(288'b0),
    .INIT_FILE("NONE"),
    .BLOCK_X(0),
    .BLOCK_Y(0),
    .RAM_DATA_WIDTH(9),
    .RAM_ADDR_WIDTH(11),
    .INIT_FORMAT("BIN")
) GTP_DRM18K_inst (
    .DOA(),          // OUTPUT[17:0]
    .DOB(),          // OUTPUT[17:0]
    .ADDRA(),        // INPUT[13:0]
```

```
    .ADDRB(),        // INPUT[13:0]
    .CSA(),          // INPUT[2:0]
    .CSB(),          // INPUT[2:0]
    .DIA(),          // INPUT[17:0]
    .DIB(),          // INPUT[17:0]
    .WWCONF(),       // OUTPUT
    .ADDRA_HOLD(),// INPUT
    .ADDRB_HOLD(),// INPUT
    .CEA(),          // INPUT
    .CEB(),          // INPUT
    .CLKA(),         // INPUT
    .CLKB(),         // INPUT
    .ORCEA(),        // INPUT
    .ORCEB(),        // INPUT
    .RSTA(),         // INPUT
    .RSTB(),         // INPUT
    .WEA(),          // INPUT
    .WEB()           // INPUT
);
```

8.8 GTP_FIFO18K Instantiation Template

```
GTP_FIFO18K #(
    .GRS_EN("TRUE"),
    .DATA_WIDTH(18),
    .DO_REG(0),
    .ALMOST_FULL_OFFSET('b00000000000000),
    .ALMOST_EMPTY_OFFSET('b00000000000000),
    .USE_EMPTY(0),
    .USE_FULL(0),
    .REWRITE_EN("FALSE"),
    .RESEND_EN("FALSE"),
    .SYNC_FIFO("FALSE")
) GTP_FIFO18K_inst (
    .DO(),           // OUTPUT[35:0]
```

```
    .DI(),              // INPUT[35:0]
    .ALMOST_EMPTY(),// OUTPUT
    .ALMOST_FULL(), // OUTPUT
    .EMPTY(),           // OUTPUT
    .FULL(),            // OUTPUT
    .ORCE(),            // INPUT
    .RCE(),             // INPUT
    .RCLK(),            // INPUT
    .RNAK(),            // INPUT
    .RST(),             // INPUT
    .WCE(),             // INPUT
    .WCLK(),            // INPUT
    .WEOP(),            // INPUT
    .WERR()             // INPUT
);
```

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage licence, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family products at any time without prior notice.