

Logos 系列 FPGA 专用 RAM 模块(DRM)

用户指南

(UG020002, V1.4)

(2022.8.13)

深圳市紫光同创电子有限公司

版权所有 侵权必究

文档版本修订记录

版本号	发布日期	修订记录
V1.0	2018/4/6	初始版本
V1.1	2018/12/18	<ol style="list-style-type: none"> 封面新增最新的版本更新时间；页脚去掉版本信息 DRM 支持的原语 GTP_DRM18K_E1 名称修正为 GTP_DRM18K
V1.2	2021/4/20	<ol style="list-style-type: none"> 更新功能特性列表 第一章新增资源规模 修正表 4-3 18K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合； 新增 FIFO 模式 DRM原语新增GTP_DRM9K、GTP_DRM18K、GTP_FIFO18K 位宽组合新增备注 SDP 模式不可设置读写模式 字节使能功能新增备注 x9(x8)写数据端口模式下不支持字节使能功能，新增图 3-5 字节使能读写时序图（TW 模式） 读写操作新增 x32/x36 位宽下 GTP 设置为 SP 时不可设置读写模式的描述 删除原 11.1 输入信号极性控制 附录新增初始化配置参数映射 附录新增 DRM 端口信号说明 附录新增 GTP_DRM9K 例化模板、GTP_DRM18K 例化模板、GTP_FIFO18K 例化模板
V1.3	2021/7/13	<ol style="list-style-type: none"> 更新文档目录结构，删除部分多余的章节 各模式新增应用示例章节，举例说明了各模式 DRM 的配置、级联配置 更新文档内容，对原文档的部分端口名称进行统一，删除、修正部分不易理解的描述，修正部分图片，对部分内容新增了补充说明 更新功能特性列表的功能说明顺序
V1.4	2022/8/13	<ol style="list-style-type: none"> 删除 DRM 原语注释、补充默认值描述， 删除表 2-3 GTP_DRM9K 参数列表、表 2-5 GTP_DRM18K 参数列表、表 2-7 GTP_FIFO18K 参数列表中默认值 删除表 2-3 GTP_DRM9K 参数列表、表 2-5 GTP_DRM18K 参数列表、表 2-7 GTP_FIFO18K 参数列表、3.4 读写操作中”SIM_DEVICE”参数及相关描述

目 录

文档版本修订记录.....	1
目 录.....	2
表目录.....	6
图目录.....	8
名词术语解释.....	10
1 概述.....	11
1.1 功能特性列表.....	12
1.2 资源规模.....	12
1.3 支持的配置.....	12
2 DRM 原语.....	14
2.1 GTP_DRM9K.....	14
2.2 GTP_DRM18K.....	17
2.3 GTP_FIFO18K.....	20
3 DP 模式.....	23
3.1 模式介绍.....	23
3.2 数据端口.....	23
3.3 位宽组合.....	24
3.4 读写操作.....	25
3.4.1 Normal Write mode.....	26
3.4.2 Transparent Write mode.....	26
3.4.3 Read before Write mode.....	27
3.5 字节使能功能.....	27
3.6 内部寄存器.....	28
3.7 复位信号.....	30
3.7.1 同步、异步复位.....	30
3.7.2 Core Latch 复位.....	31
3.8 全局信号时序.....	31
3.9 应用示例.....	32

3.9.1 单个 DRM 配置.....	32
3.9.2 多个 DRM 级联配置.....	34
4 SDP 模式.....	40
4.1 模式介绍	40
4.2 数据端口	40
4.3 位宽组合	41
4.4 读写操作	42
4.5 字节使能功能	43
4.6 内部寄存器	44
4.7 复位信号	44
4.8 全局信号时序	44
4.9 应用示例	44
4.9.1 单个 DRM 配置.....	44
4.9.2 多个 DRM 级联配置.....	46
5 SP 模式	47
5.1 模式介绍	47
5.2 数据端口	47
5.3 位宽组合	48
5.4 读写操作	48
5.5 字节使能功能	50
5.6 内部寄存器	50
5.7 复位信号	50
5.8 全局信号时序	50
5.9 应用示例	51
5.9.1 单个 DRM 配置.....	51
5.9.2 多个 DRM 级联配置.....	53
6 ROM 模式	54
6.1 模式介绍	54
6.2 数据端口	54
6.3 位宽组合	54

6.4 读写操作	55
6.5 内部寄存器	55
6.6 复位信号	55
6.7 全局信号时序	55
6.8 应用示例	55
6.8.1 单个 DRM 配置.....	55
6.8.2 多个 DRM 级联配置.....	58
7 FIFO 模式.....	59
7.1 模式介绍	59
7.2 数据端口	60
7.2.1 异步 FIFO（不含丢包重发功能）	60
7.2.2 异步 FIFO（支持丢包功能）	61
7.2.3 异步 FIFO（包含丢包重发功能）	62
7.2.4 同步 FIFO（不含丢包重发功能）	63
7.2.5 同步 FIFO（支持丢包功能）	64
7.2.6 同步 FIFO（包含丢包重发功能）	65
7.3 位宽组合	66
7.4 读写操作	66
7.4.1 写入空 FIFO 时序	66
7.4.2 写入将满 FIFO 时序	66
7.4.3 满 FIFO 读出时序	67
7.4.4 将空 FIFO 读出时序	67
7.4.5 写丢包时序	68
7.4.6 读重发时序	68
7.4.7 支持丢包功能下写入空 FIFO	68
7.4.8 丢包重发功能下写入将满 FIFO	69
7.5 内部寄存器	69
7.6 复位信号	69
7.7 全局信号时序	69
7.8 应用示例	69

8 附录	72
8.1 地址和数据端口 MAPPING	72
8.2 初始化配置参数映射	75
8.3 DRM 端口信号说明	76
8.4 字节附加信息位	76
8.5 DRM_FIFO 位宽模式	77
8.6 GTP_DRM9K 例化模板.....	78
8.7 GTP_DRM18K 例化模板.....	79
8.8 GTP_FIFO18K 例化模板	82
免责声明	83

表目录

表 1-1 功能特性列表.....	12
表 1-2 Logos 系列 DRM 资源规模.....	12
表 2-1 Logos 系列 FPGA DRM 原语.....	14
表 2-2 GTP_DRM9K 端口列表.....	15
表 2-3 GTP_DRM9K 参数列表.....	16
表 2-4 GTP_DRM18K 端口列表.....	18
表 2-5 GTP_DRM18K 参数列表.....	19
表 2-6 GTP_FIFO18K 端口列表.....	21
表 2-7 GTP_FIFO18K 参数列表.....	22
表 3-1 DP RAM 端口命名及描述.....	24
表 3-2 18K DRM 模式下 True Dual Port RAM 模式允许的位宽组合.....	24
表 3-3 9K DRM 模式下 True Dual Port RAM 模式允许的位宽组合.....	25
表 3-4 DRM 参数配置.....	32
表 3-5 DRM 端口连接.....	32
表 3-6 DRM 参数配置.....	34
表 3-7 DRM 端口连接.....	35
表 4-1 SDP RAM 端口名称及描述.....	41
表 4-2 9K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合.....	41
表 4-3 18K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合.....	42
表 4-4 DRM 参数配置.....	44
表 4-5 DRM 端口连接.....	45
表 5-1 SP RAM 端口名称及描述.....	48
表 5-2 18K DRM 模式 Single Port RAM 模式列表.....	48
表 5-3 9K DRM 模式 Single Port RAM 模式列表.....	48
表 5-4 SP 模式读写模式设置支持情况.....	49
表 5-5 DRM 参数配置.....	51
表 5-6 DRM 端口连接.....	51
表 6-1 ROM 端口名称及描述.....	54
表 6-2 18K DRM 模式 ROM 模式列表.....	55

表 6-3 9K DRM 模式 ROM 模式列表.....	55
表 6-4 DRM 参数配置.....	56
表 6-5 DRM 端口连接.....	56
表 7-1 异步 FIFO 模式（不含丢包重发功能）端口说明.....	60
表 7-2 异步 FIFO 模式（支持丢包功能）端口说明.....	61
表 7-3 异步 FIFO 模式（包含丢包重发功能）端口说明.....	62
表 7-4 同步 FIFO 模式（不含丢包重发功能）端口说明.....	63
表 7-5 同步 FIFO 模式（支持丢包功能）端口说明.....	64
表 7-6 同步 FIFO（包含丢包重发功能）端口说明.....	65
表 7-7 18K GTP_FIFO 模式允许的位宽.....	66
表 7-8 FIFO 参数配置.....	70
表 7-9 FIFO 端口连接.....	70
表 8-1 18K DRM 模式下存储器对应的地址和数据端口 Mapping.....	72
表 8-2 9K DRM 模式下存储器对应的地址和数据端口 Mapping.....	73
表 8-3 不同位宽数据地址 Mapping（x1, x2, x4, x8, x16, x32 数据宽度）.....	74
表 8-4 不同位宽数据地址 Mapping（x9, x18, x36 数据宽度）.....	74
表 8-5 INIT_XX 参数映射.....	75
表 8-6 字节附加信息位列表.....	76
表 8-7 FIFO 丢包重发模式地址深度与数据宽度配置列表.....	77
表 8-8 含丢包重发模式的异步/同步 FIFO 模式数据位宽列表.....	77
表 8-9 不含丢包重发模式的异步/同步 FIFO 模式数据位宽列表.....	77

图目录

图 2-1 GTP_DRM9K 结构框图	15
图 2-2 GTP_DRM18K 结构框图	18
图 2-3 GTP_FIFO18K 结构框图	21
图 3-1 DP RAM 数据端口	23
图 3-2 Normal Write mode 读写时序图	26
图 3-3 Transparent Write mode 读写时序图	26
图 3-4 Read before Write mode 读写时序图	27
图 3-5 字节使能读写时序图 (TW 模式)	28
图 3-6 DRM 寄存器逻辑图	28
图 3-7 地址 IR 逻辑图	29
图 3-8 地址 IR 时序图	29
图 3-9 带输出寄存器的读时序	30
图 3-10 输出寄存器复位时序	30
图 3-11 Core 复位	31
图 3-12 全局信号时序示意图	31
图 4-1 SDP RAM 数据端口	40
图 4-2 SDP RAM 典型时序图	43
图 4-3 字节使能读写时序图 (SDP 模式)	43
图 5-1 SP RAM 数据端口	47
图 5-2 SP 模式的 32/36 bit 模式下 TW 和 RBW 模式地址线配置	50
图 6-1 ROM 模式数据端口	54
图 7-1 FIFO 顶层结构框图	59
图 7-2 异步 FIFO 模式 (不含丢包重发功能)	60
图 7-3 异步 FIFO 模式 (支持丢包功能)	61
图 7-4 异步 FIFO 模式 (包含丢包重发功能)	62
图 7-5 同步 FIFO 模式 (不含丢包重发功能)	63
图 7-6 同步 FIFO 模式 (支持丢包功能)	64
图 7-7 同步 FIFO (包含丢包重发功能)	65
图 7-8 写入空 FIFO 时序图	66

图 7-9 写入将满 FIFO 时序图	67
图 7-10 从满 FIFO 读出时序图	67
图 7-11 从将空 FIFO 读出时序.....	67
图 7-12 写丢包时序	68
图 7-13 读重发时序	68
图 7-14 写丢包模式下写入空 FIFO 时序图	68
图 7-15 丢包重发模式下写入将满 FIFO 时序图.....	69

名词术语解释

Abbreviations 缩略语	Full Spelling 英文全拼	Chinese Explanation 中文解释
GTP	Generic Technology Primitive	通用技术原语
DRM	Dedicated RAM Module	专用 RAM 模块
SP	Single Port	单口
SDP	Simple Dual Port	简单双口
DP	True Dual Port	双口
FIFO	First In First Out	先进先出存储器
NW	Normal-Write	普通写
RBW	Read-before-Write	先读后写
TW	Transparent-Write	透传写
OR	Output Register	输出寄存器

1 概述

Logos的DRM有高达18K bits的存储单元并且容量可被独立配置为2个9K bits或者1个18K bits。每个DRM都能支持DP（True Dual Port，双口）RAM模式，同时也可以被配置为SP（Single Port，单口）RAM模式，SDP（Simple Dual Port，简单双口）RAM模式，ROM模式，以及可选丢包重发的同步\异步FIFO模式。DRM资源还支持输入寄存器（IR）、输出寄存器（OR）以及Core Latch，这使得DRM级联使用时拥有更加出色的性能表现。DRM的总数取决于Logos系列器件类型。

DP RAM的端口数据位宽高至18bit，它的两个端口除了共享RAM内容之外完全独立，并支持不同的时钟域；SDP RAM的端口数据位宽高至36bit，它的两个端口也支持不同的时钟域，但它的一个端口仅限于写操作，另一个端口仅限于读操作。使用少量的外部CLM资源，Logos的SDP RAM还可以扩展成FIFO。ROM模式下，ROM的端口数据位宽高至36bits。DRM的内容通常在下载配置数据的过程中初始化。当然，其它模式也可以利用编程配置来初始化DRM的内容。在同步或者异步FIFO模式时，一个端口专用于FIFO数据写入，另一个端口专用于FIFO数据读取，读写端口可以支持不同的时钟，但不支持混合位宽及Byte enable的写操作。FIFO模式在同步和异步模式下都支持可选的丢包重发模式。

嵌入的DP RAM、SP RAM、SDP RAM、ROM以及可选丢包重发的同步\异步FIFO模块，都能便捷地通过深圳市紫光同创电子有限公司的软件Pango Design Suite内嵌的IP Compiler工具生成。

1.1 功能特性列表

如下表所示，为 Logos 系列 FPGA 的 DRM 功能特性列表。

表 1-1 功能特性列表

功能	说明
存储容量	18K(或 2 个 9K)
SP RAM 模式	支持最大 36 位数据宽度
DP RAM 模式	支持最大 18 位数据宽度
SDP RAM 模式	支持最大 36 位数据宽度
ROM 模式	支持最大 36 位数据宽度
FIFO 模式	支持最大 36 位数据宽度
Normal Write 写模式	支持
Transparent write 写模式	支持
Read-before Write 写模式	支持
Byte 写使能	支持
输入寄存器	支持
输出寄存器	支持（可旁路）
地址输入保持	支持
同步/异步 FIFO 控制	支持
FIFO 写入端的丢包功能	支持
FIFO 读出端的重发功能	支持
可编程的 ALMOST_FULL/ALMOST_EMPTY 指示	支持

注：SP 模式下 32/36 bit 数据位宽禁止设置读写模式为 TW、RBW，需要设置为默认的 NW，详见 4.4。

1.2 资源规模

表 1-2 Logos 系列 DRM 资源规模

器件	资源数量/18K
PGL12G	30
PGL22G	48
PGL22GS	48
PGL25G	60
Logos	134
PGL100H	286

1.3 支持的配置

DP RAM和SDP RAM均支持9K模式和18K模式。在DP RAM模式下，DRM有18-bit的数据最大位宽，DRM模块中A、B两个端口均可以独立进行读写操作，且均支持独立时钟。而

SDP RAM模式下，DRM数据位宽则增大至36 bits，A，B两个端口中一个端口专用于数据写入，另一个端口专用于数据读取，读写端口同样均支持独立时钟。

GTP_DRM、GTP_FIFO 的端口位宽支持两种类型：一种是数据位宽为 2^N （包括1/2/4/8/16/32bit）。另一种是数据位宽为 9×2^N （包括9/18/36bit）。

GTP_DRM 的 DP RAM 和 SDP RAM 模式支持混合数据位宽功能，即两个端口可以配置成不同位宽。例如，一个 SDP RAM，可以在写端口配置成 16Kx1，在读端口配置成 512x32，从而节约了从 1 比特到 32 比特的串并转换逻辑。

GTP_DRM的SP RAM、ROM模式和GTP_FIFO不支持混合位宽。

在DP RAM模式下，两个端口的位宽可以独立地根据DRM资源进行配置。9 K DRM模式下的DP RAM配置，SDP RAM配置，以及18 K DRM模式下的DP RAM配置，SDP RAM配置在表 3-3，表 4-3，表 3-2和表 4-2描述。

在SP RAM和ROM两种模式的9K模式下，DRM包含两个端口，SP RAM模式可以分别对这两个端口独立进行读写操作，而ROM模式下该端口只读；在两对端口共享情况下，DRM仅包含一个端口，SP RAM模式可以对该端口进行读写操作，而ROM模式下该端口只读。9 K模式下的SP RAM模式和ROM模式列表，以及18 K模式下的SP RAM模式和ROM模式列表已在表 5-3，表 6-3，表 5-2和表 6-2描述。

FIFO模式下，一个端口专用于数据写入另一个端口专用于数据读取，读写端口可以采用不同的时钟。

多个 DRM 可以通过级联扩展的方式组合成更大的 DP RAM, SDP RAM, SP RAM, ROM 或者 FIFO。对此，DRM 提供额外的 3 bit 地址扩展（CS[2:0]），常用于深度扩展的应用。

[地址和数据端口Mapping](#)详见附录。

2 DRM 原语

Logos 系列 FPGA DRM 的原语是各种模式的基础，其原语有三种：GTP_DRM9K、GTP_DRM18K 和 GTP_FIFO18K，其支持类型如下表所示：

表 2-1 Logos 系列 FPGA DRM 原语

原语	支持类型
GTP_DRM9K	支持 x1,x2,x4,x8,x16,x32（和 x9,x18,x36）的数据位宽
GTP_DRM18K	支持 x1,x2,x4,x8,x16,x32（和 x9,x18,x36）的数据位宽
GTP_FIFO18K	支持 x1,x2,x4,x8,x16,x32（和 x9,x18,x36）的数据位宽

Logos 支持的 GTP 原语及其端口、参数等说明以《UG020007_Logos 系列产品 GTP 用户指南》为准。

2.1 GTP_DRM9K

GTP_DRM9K 有 9K bits 存储单元，支持多种工作模式，包括双口（DP, True Dual Port）RAM，简单双口（SDP, Simple Dual Port）RAM，单口（SP, Single Port）RAM 或 ROM 模式。DRM 支持可配置的数据位宽，并在 DP RAM 和 SDP RAM 模式下支持双端口混合数据位宽。

DP RAM 模块有两个独立的端口 A、B，均可以独立进行读写操作，即可同时进行读操作，写操作，或一端口进行读操作一端口进行写操作。两端口均支持不同的时钟，且位宽可以独立设置。不可同时通过两个端口对同一地址进行一读一写操作，否则将引起冲突。

SDP RAM 中，A 端口固定为写入端口，B 端口固定为读出端口，两个端口支持不同时钟，且均有独立的位宽设置。不可同时通过两个端口对同一地址进行一读一写操作，否则将引起冲突。

SP RAM 模式下，可以对 DRM 的两个端口分别进行独立操作。不可同时通过两个端口对同一地址进行一读一写操作，否则将引起冲突。

DRM 可以配置为 ROM，通过配置接口可以初始化 ROM 内容。

DRM 的端口写操作支持 Normal Write mode (NW), Transparent Write mode (TW), Read before Write mode (RBW) 三种模式。

所有端口均默认为高电平有效。

GTP_DRM9K 的结构框图见下图。

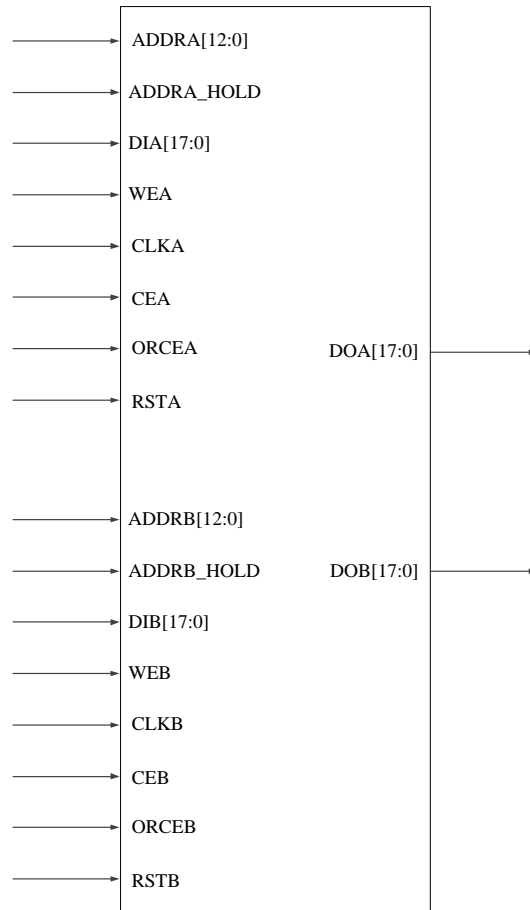


图 2-1 GTP_DRM9K 结构框图

下表罗列了 GTP_DRM9K 的端口名称及其描述。

表 2-2 GTP_DRM9K 端口列表

端口名称	方向	功能描述
ADDRA	输入	A端口输入地址
ADDRA_HOLD	输入	A端口地址输入选择
DIA	输入	A端口数据输入
WEA	输入	A端口写使能
CLKA	输入	A端口时钟
CEA	输入	A端口时钟使能
ORCEA	输入	A端口输出寄存器使能
RSTA	输入	A端口数据寄存器复位
DOA	输出	A端口数据输出
ADDRB	输入	B端口输入地址
ADDRB_HOLD	输入	B端口地址输入选择
DIB	输入	B端口数据输入
WEB	输入	B端口写使能
CLKB	输入	B端口时钟
CEB	输入	B端口时钟使能
ORCEB	输入	B端口输出寄存器使能

端口名称	方向	功能描述
RSTB	输入	B端口数据寄存器复位
DOB	输出	B端口数据输出

下表罗列了 GTP_DRM9K 的参数名称及其描述。

表 2-3 GTP_DRM9K 参数列表

参数名	描述	设置值
DATA_WIDTH_A	A端口数据最大位宽	1、2、4、8、16、32、9、18、36
DATA_WIDTH_B	B端口数据最大位宽	1、2、4、8、16、32、9、18、36
WRITE_MODE_A	A端口写模式	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
WRITE_MODE_B	B端口写模式	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
DOA_REG	A端口输出寄存	0: 不使能输出寄存 1: 使能输出寄存
DOB_REG	B端口输出寄存	0: 不使能输出寄存 1: 使能输出寄存
RST_TYPE	复位模式选择	"SYNC": 同步复位 "ASYNC": 异步复位 "ASYNC_SYNC_RELEASE": 异步复位同步释放
RAM_MODE	RAM模式选择	"TRUE_DUAL_PORT": 双口RAM "SIMPLE_DUAL_PORT": 简单双口RAM "SINGLE_PORT": 单口RAM "ROM": ROM
GRS_EN	全局复位使能信号 (芯片内部)	"FALSE":不使能全局复位; "TRUE":使能全局复位。
DOA_REG_CLKINV	A端口输出寄存器时钟翻转	0: 时钟不翻转 1: 时钟翻转
DOB_REG_CLKINV	B端口输出寄存器时钟翻转	0: 时钟不翻转 1: 时钟翻转
INIT_00 INIT_01 INIT_02 ... INIT_1F	RAM 初始化配置参数, 详细请参考附录 初始化配置参数映射	0 ~ 2 ²⁸⁸ -1
INIT_FILE	初始化文件路径, 该参数只在行为仿真中生效, 综合生成网表时RAM的初始化数据来源于初始化配置参数 INIT_XX	"NONE": 不指定初始化文件, 则初始化数据为参数 INIT_XX 设置的值; "XXX": XXX 表示具体初始化文件路径, 行为仿真中覆盖参数 INIT_XX 作为 DRM 的初始值

参数名	描述	设置值
BLOCK_X	DRM9K 级联时数据级联坐标, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 坐标而定
BLOCK_Y	DRM9K 级联时地址级联坐标, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 坐标而定
RAM_DATA_WIDTH	DRM 级联后的数据最大位宽, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 个数而定
RAM_ADDR_WIDTH	DRM 级联后的地址最大位宽, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 个数而定
INIT_FORMAT	初始化文件格式	"BIN": 二进制 "HEX": 十六进制

注: SDP 模式不支持读写模式设置, 使用 GTP 配置 DRM 为 SDP 模式时, 不能手动修改设置读写模式参数 WRITE_MODE_A/B。

2.2 GTP_DRM18K

GTP_DRM18K 有 18K bits 存储单元, 支持多种工作模式, 包括双口 (DP, True Dual Port) RAM, 简单双口 (SDP, Simple Dual Port) RAM, 单口 (SP, Single Port) RAM 或 ROM 模式。DRM 支持可配置的数据位宽, 并在 DP RAM 和 SDP RAM 模式下支持双端口混合数据位宽。

DP RAM 模块有两个独立的端口 A、B, 均可以独立进行读写操作, 即可同时进行读操作, 写操作, 或一端口进行读操作一端口进行写操作。两端口均支持不同的时钟, 且位宽可以独立设置。不可同时通过两个端口对同一地址进行一读一写操作, 否则将引起冲突。

SDP RAM 中, A 端口固定为写入端口, B 端口固定为读出端口, 两个端口支持不同时钟, 且均有独立的位宽设置。不可同时通过两个端口对同一地址进行一读一写操作, 否则将引起冲突。

SP RAM 模式下, 可以对 DRM 的两个端口分别进行独立操作。不可同时通过两个端口对同一地址进行一读一写操作, 否则将引起冲突。

DRM 可以配置为 ROM, 通过配置接口可以初始化 ROM 内容。

DRM 的端口写操作支持 Normal Write mode (NW), Transparent Write mode (TW), Read before Write mode (RBW) 三种模式。

所有端口均默认为高电平有效。

结构框图见下图。

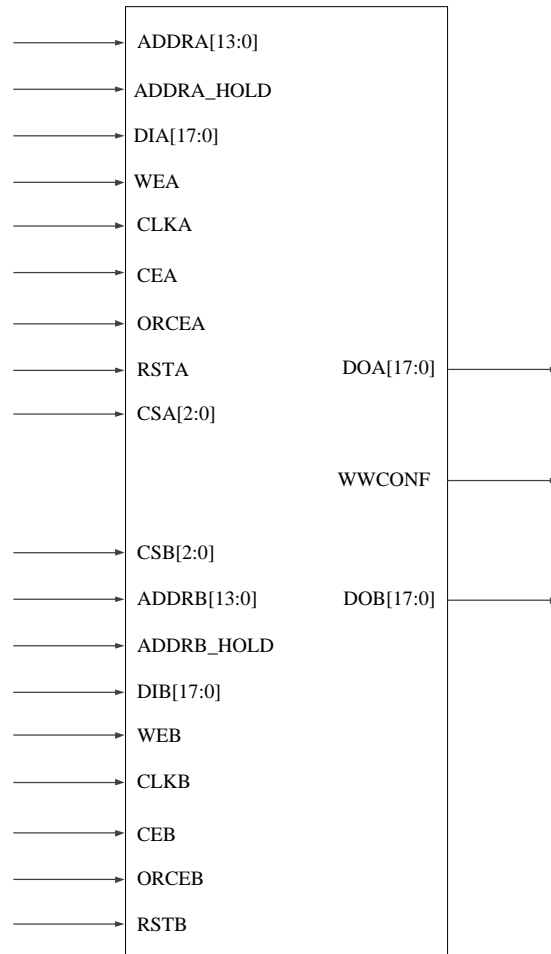


图 2-2 GTP_DRM18K 结构框图

下表罗列了 GTP_DRM18K 的端口名称及其描述。

表 2-4 GTP_DRM18K 端口列表

端口名称	方向	功能描述
ADDRA	输入	A端口输入地址
ADDRA_HOLD	输入	A端口地址输入选择
DIA	输入	A端口数据输入
CSA	输入	A端口地址扩展
WEA	输入	A端口写使能
CLKA	输入	A端口时钟
CEA	输入	A端口时钟使能
ORCEA	输入	A端口输出寄存器使能
RSTA	输入	A端口数据寄存器复位
DOA	输出	A端口数据输出
ADDRB	输入	B端口输入地址
ADDRB_HOLD	输入	B端口地址输入选择
DIB	输入	B端口数据输入

端口名称	方向	功能描述
CSB	输入	B端口地址扩展
WEB	输入	B端口写使能
CLKB	输入	B端口时钟
CEB	输入	B端口时钟使能
ORCEB	输入	B端口输出寄存器使能
RSTB	输入	B端口数据寄存器复位
DOB	输出	B端口数据输出
WWCONF	输出	写冲突（该端口在Logos无效）

下表罗列了 GTP_DRM18K 的参数名称及其描述。

表 2-5 GTP_DRM18K 参数列表

参数名	描述	设置值
CSA_MASK[2:0]	A端口地址扩展控制信号	0~7
CSB_MASK[2:0]	B端口地址扩展控制信号	0~7
DATA_WIDTH_A	A端口数据最大位宽	1、2、4、8、16、32、9、18、36
DATA_WIDTH_B	B端口数据最大位宽	1、2、4、8、16、32、9、18、36
WRITE_MODE_A	A端口写模式	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
WRITE_MODE_B	B端口写模式	"NORMAL_WRITE" "TRANSPARENT_WRITE" "READ_BEFORE_WRITE"
DOA_REG	A端口输出寄存	0: 不使能输出寄存 1: 使能输出寄存
DOB_REG	B端口输出寄存	0: 不使能输出寄存 1: 使能输出寄存
RST_TYPE	复位模式选择	"SYNC": 同步复位 "ASYN": 异步复位 "ASYN_SYNC_RELEASE": 异步复位同步释放
RAM_MODE	RAM模式选择	"TRUE_DUAL_PORT": 双口RAM "SIMPLE_DUAL_PORT": 简单双口RAM "SINGLE_PORT": 单口RAM "ROM": ROM
WRITE_COLLISION_ARBITER	写冲突仲裁	"NULL": 不进行仲裁
GRS_EN	全局复位使能信号（芯片内部）	"FALSE": 不使能全局复位; "TRUE": 使能全局复位。
DOA_REG_CLKINV	A端口输出寄存器时钟翻转	0: 时钟不翻转 1: 时钟翻转
DOB_REG_CLKINV	B端口输出寄存器时钟翻转	0: 时钟不翻转

参数名	描述	设置值
		1: 时钟翻转
INIT_00 INIT_01 INIT_02 ... INIT_3F	RAM 初始化配置参数, 详细请参考附录 初始化配置参数映射	0 ~ 2 ²⁸⁸ -1
INIT_FILE	初始化文件路径, 该参数只在行为仿真中生效, 综合生成网表时 RAM 的初始化数据来源于初始化配置参数 INIT_XX	"NONE": 不指定初始化文件, 则初始化数据为参数 INIT_XX 设置的值; "XXX": XXX 表示具体初始化文件路径, 行为仿真中覆盖参数 INIT_XX 作为 DRM 的初始值
BLOCK_X	DRM18K 级联时数据级联坐标, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 坐标而定
BLOCK_Y	DRM18K 级联时地址级联坐标, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 坐标而定
RAM_DATA_WIDTH	DRM 级联后的数据最大位宽, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 个数而定
RAM_ADDR_WIDTH	DRM 级联后的地址最大位宽, 用于将同一个初始化文件映射到不同的级联 DRM 中	视级联 DRM 个数而定
INIT_FORMAT	初始化文件格式	"BIN": 二进制 "HEX": 十六进制

注: 1.SDP 模式不支持读写模式设置, 使用 GTP 配置 DRM 为 SDP 模式时, 不能手动修改设置读写模式参数 WRITE_MODE_A/B。

2. Logos 不支持写冲突仲裁, 参数 WRITE_COLLISION_ARBITER 只能设置为"NULL", 即不进行写冲突仲裁。

2.3 GTP_FIFO18K

DRM_FIFO 支持 18K bits 存储单元, 可以配置成 16Kx1, 8Kx2, 4Kx4, 2Kx9(8), 1Kx18(16), 512x36(32)数据位宽模式。DRM 配置为 FIFO 模式时, DRM 配置成 Simple dual port 模式, 且不支持混合位宽, 不支持 Byte enable 的写操作。

DRM_FIFO 可将 DRM 配置成同步 FIFO 或异步 FIFO 模式。在 DRM SDP RAM 模式下, 一个端口专用于 FIFO 数据写入, 另一个端口专用于 FIFO 数据读取, 读写端口可以采用不同的时钟。DRM_FIFO 在同步和异步模式下都支持可选的丢包重发模式。结构框图见下图。

DRM_FIFO 支持寄存器模式, 具体使用方式和 GTP_DRM 的寄存器模式相同。

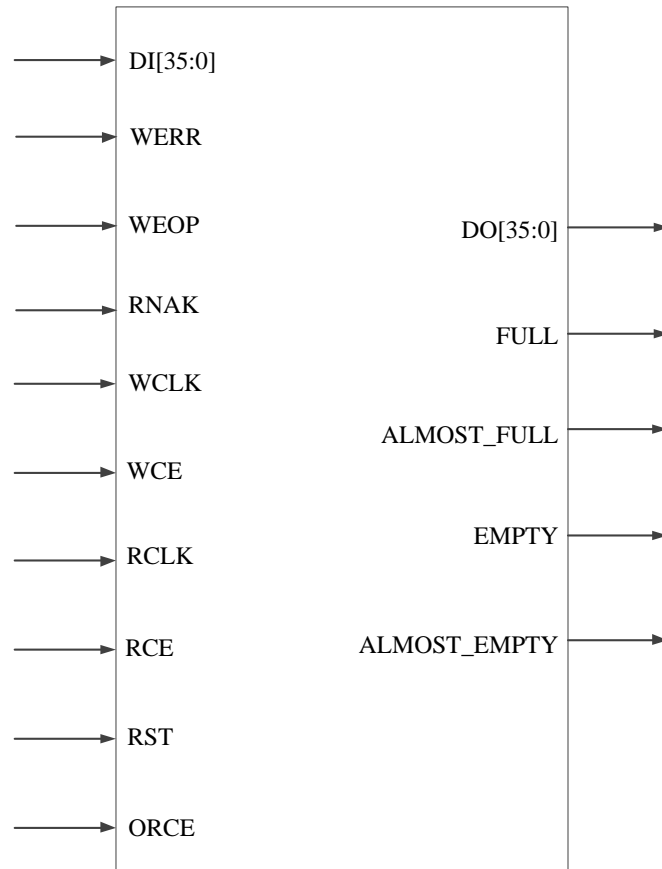


图 2-3 GTP_FIFO18K 结构框图

下表罗列了 GTP_FIFO18K 的端口名称及其描述。

表 2-6 GTP_FIFO18K 端口列表

端口名称	方向	功能描述
DI	输入	数据写入
WERR	输入	写端口错误数据包指示
WEOP	输入	包尾信号指示
RNAK	输入	读端口重发指示
WCLK	输入	写时钟信号
RCLK	输入	读时钟信号
WCE	输入	写使能信号
RCE	输入	读使能信号
RST	输入	复位信号
ORCE	输入	输出寄存器使能信号
DO	输出	数据读出
EMPTY	输出	读端口读空标志
FULL	输出	写端口写满标志
ALMOST_EMPTY	输出	读端口将空标志
ALMOST_FULL	输出	写端口将满标志

下表罗列了 GTP_FIFO18K 的参数名称及其描述。

表 2-7 GTP_FIFO18K 参数列表

参数命名	功能描述	设置值
GRS_EN	全局复位使能信号（芯片内部）	"TRUE":使能全局复位 "FALSE":不使能全局复位
ALMOST_FULL_OFFSET	当FIFO将满且写指针与读指针位置之差值等于ALMOST_FULL_OFFSET，产生almost_full标志。	16382: 16K地址 8190: 8K地址 4094: 4K地址 2046: 2K地址 1022: 1K地址 510: 512地址
ALMOST_EMPTY_OFFSET	当FIFO将空且读指针与写指针位置之差值等于ALMOST_EMPTY_OFFSET,产生almost_empty标志。	16382: 16K地址 8190: 8K地址 4094: 4K地址 2046: 2K地址 1022: 1K地址 510: 512地址
SYNC_FIFO	异步/同步FIFO选择	"TRUE":使用同步FIFO "FALSE":使用异步FIFO
DATA_WIDTH	FIFO数据位宽	1,2,4,8,9,16,18,32,36
REWRITE_EN	丢包使能	"TRUE": 无丢包功能; "FALSE": 使能丢包功能
RESEND_EN	重发使能	"TRUE": 无重发功能; "FALSE": 使能重发功能
USE_EMPTY	使能读空标志	1:使能读空标志; 0:不使能读空标志
USE_FULL	使能写满标志	1:使能写满标志; 0:不使能写满标志
DO_REG	使能输出寄存器使能	1:使能; 0:不使能

3 DP 模式

3.1 模式介绍

RAM 的端口模式由参数 RAM_MODE 决定，当参数 RAM_MODE 的值为 "TRUE_DUAL_PORT" 时，RAM 进入双口模式。本文将详细介绍 18K 的 DP RAM，9K 与 18K 结构与功能基本一致，仅位宽不同。

DP RAM 支持：

- DP 读操作
- DP 写操作
- 一端口读一端口写操作
- 两端口有独立的位宽设置

3.2 数据端口

DP RAM 模式下，18K、9K 的 DP RAM 拥有除了共享 DRM 内容之外完全独立的两个端口：A 端口和 B 端口。它们的结构完全对称，下图展示了 18K DP RAM 的结构图。

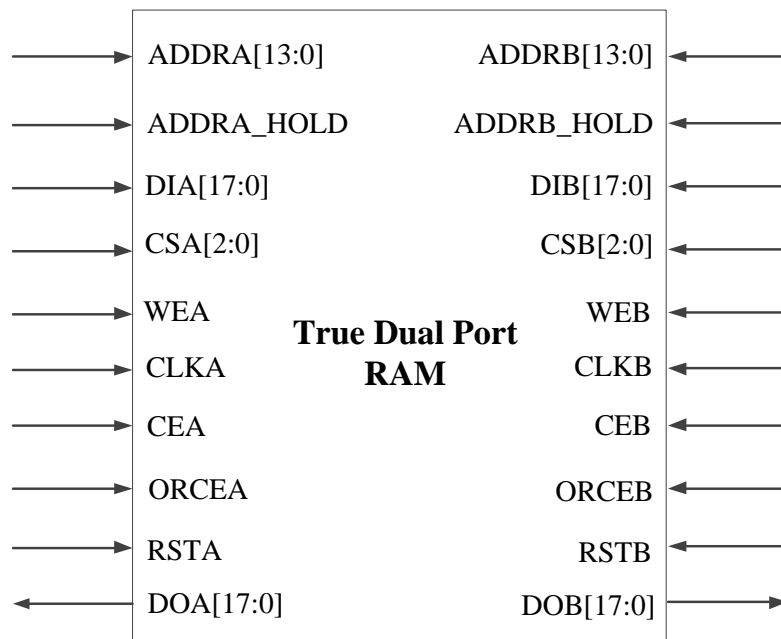


图 3-1 DP RAM 数据端口

下表罗列了 DPRAM 的端口名称和端口的描述。

表 3-1 DPRAM 端口命名及描述

端口命名	方向	描述	端口命名	方向	描述
DIA	输入	A 端口数据输入	DIB	输入	B 端口数据输入
ADDRA	输入	A 端口地址输入	ADDRB	输入	B 端口地址输入
ADDRA_HOLD	输入	A 端口地址锁存信号	ADDRB_HOLD	输入	B 端口地址锁存信号
WEA	输入	A 端口写使能	WEB	输入	B 端口写使能
CEA	输入	A 端口时钟使能	CEB	输入	B 端口时钟使能
CLKA	输入	A 端口时钟	CLKB	输入	B 端口时钟
RSTA	输入	A 端口输出寄存器复位	RSTB	输入	B 端口输出寄存器复位
CSA	输入	A 端口地址扩展	CSB	输入	B 端口地址扩展
ORCEA	输入	A 端口输出寄存器使能	ORCEB	输入	B 端口输出寄存器使能
DOA	输出	A 端口数据输出	DOB	输出	B 端口数据输出

注: 1.具体数据位宽详见附录字节附加信息位;

2.9K DRM 无 CSA/CSB 端口, 不支持地址扩展, 如需 DRM 级联进行地址深度扩展, 建议使用 18K DRM。

3.3 位宽组合

RAM 的端口位宽由 GTP 中的参数 DATA_WIDTH_A/DATA_WIDTH_B 决定, 例如, 当参数 DATA_WIDTH_A 的值为 4 时, A 端口数据位宽被设置为 4 bit。DP 模式支持 A、B 端口设置不同数据位宽。

如下表所示为 18K DRM 模式下 True Dual Port RAM 模式允许的位宽组合。

表 3-2 18K DRM 模式下 True Dual Port RAM 模式允许的位宽组合

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

注: √表示支持的位宽组合。

如下表所示为 9K DRM 模式下 True Dual Port RAM 模式允许的位宽组合。

表 3-3 9K DRM 模式下 True Dual Port RAM 模式允许的位宽组合

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

注：√表示支持的位宽组合。

3.4 读写操作

根据数据写入时同一端口输出的数据的不同，DRM 的端口写操作支持 Normal Write mode (NW)，Transparent Write mode (TW)，Read before Write mode (RBW) 三种模式，默认的写操作模式为 NW。RAM 的写操作模式由 GTP 中的参数 WRITE_MODE_A/WRITE_MODE_B 决定，当参数 WRITE_MODE_A 的值为 "NORMAL_WRITE" 时，DRM 的 A 端口的写操作模式被设置为 NW。不同写模式下，读时序是类似的。

DP 模式有两个相对独立的端口，若同时通过两个端口对同一地址进行一读一写操作会引起冲突。DRM 禁止两端口同时向同一地址写入数据、禁止两端口同时一读一写同一地址，需要在实际应用中通过用户逻辑加以规避。

3.4.1 Normal Write mode

如下图 Normal Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，此时该端口的输出数据不更新。

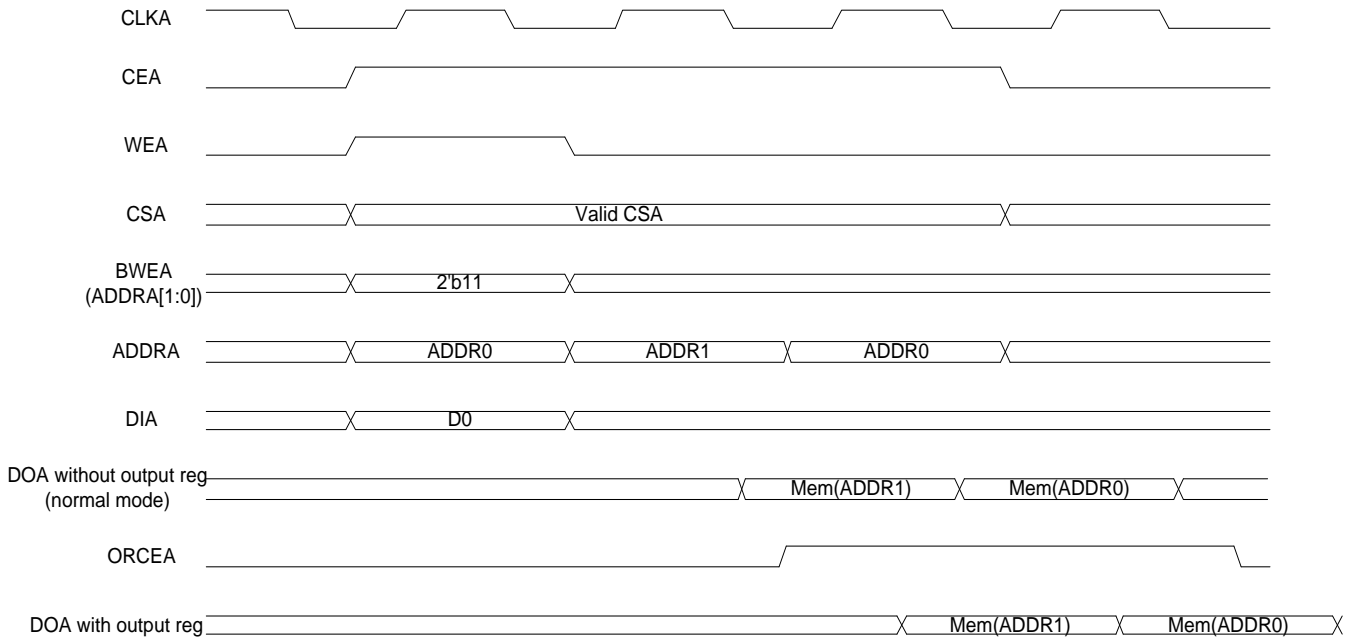


图 3-2 Normal Write mode 读写时序图

3.4.2 Transparent Write mode

如下图 Transparent Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，写入数据在写入 RAM 的同时（即写操作的下一个时钟周期）直接输出到输出端口。

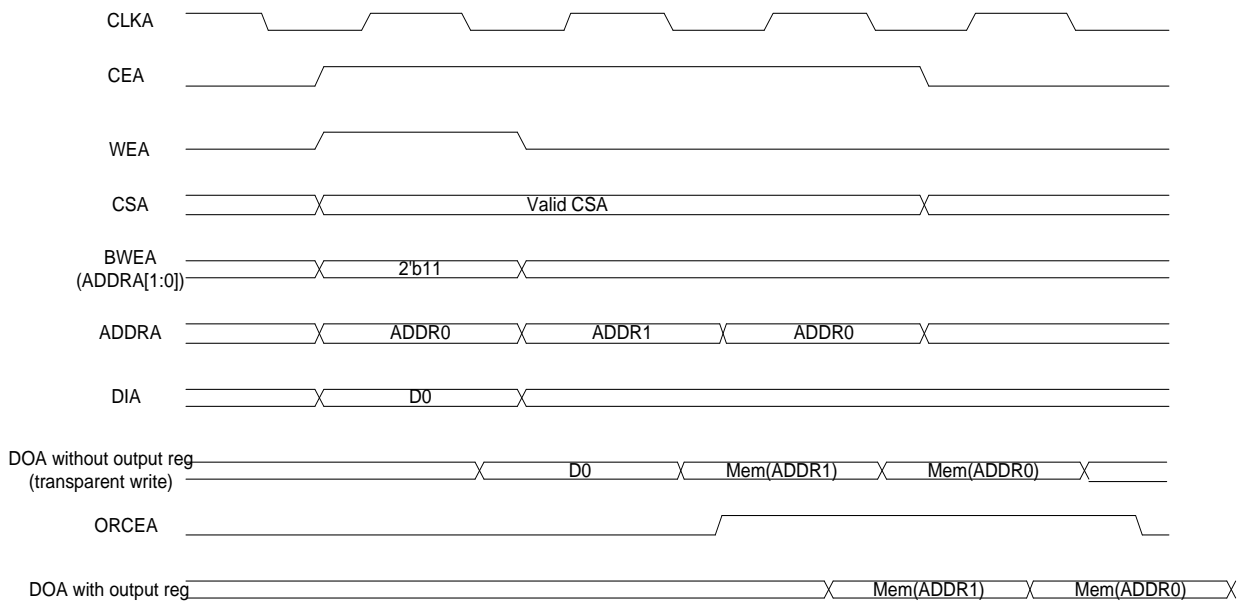


图 3-3 Transparent Write mode 读写时序图

3.4.3 Read before Write mode

如下图 Read before Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，将首先读出该地址所引的原数据并在写操作的下一个时钟周期输出到输出端口。

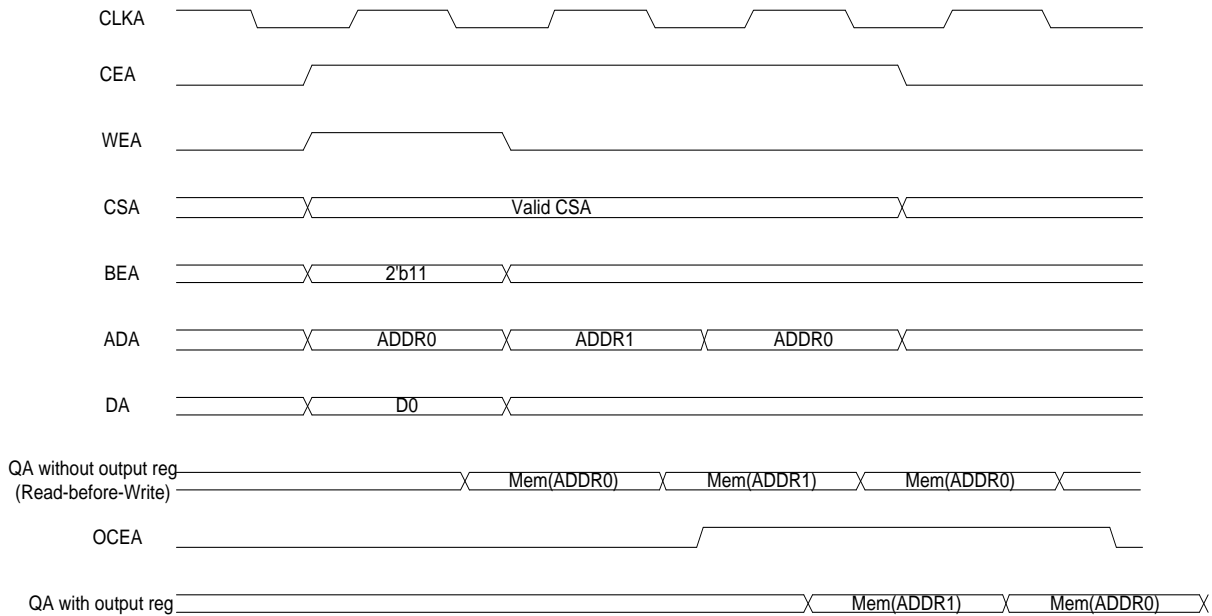


图 3-4 Read before Write mode 读写时序图

3.5 字节使能功能

DRM 支持写操作的 Byte-Write 模式，通过 BWEA/BWEB 信号（高有效）实现对选定数据字节写入，同时屏蔽对同一地址索引的其它字节的写入。该模式主要应用于在限定数据总线宽度时，只对比较窄的数据总线进行操作。比如可以在 18 位数据总线上只对 9 位宽度数据操作。

DP 模式下，当端口位宽为 2^N bit，支持 16 bit 的写操作的字节使能，此时每个字节包含 8 bit。当端口位宽为 9×2^N bit 时，支持 18 bit 的写操作的字节使能，此时每个字节包含 9 bit。例如，在 A 端口数据位宽为 16 bit 时，BWEA 包含两个比特，BWEA[1]控制 data[15:8]；BWEA[0]控制 data[7:0]；依次类推。当 BWEA[1:0]=2'b01 时，写操作有效时仅 data[7:0]被写入当前地址。设计时有以下三点需要考虑的地方：

- A 端口和 B 端口分别有独立的字节使能控制
A 端口：BWEA[1:0]，B 端口：BWEB[1:0]
- 字节使能信号在 x18 位数据宽度模式下总是有效，当不使用时，需要将该信号连接到高电平。
- 字节使能信号复用低位地址信号：

$BWEA[1:0] = ADDRA[1:0]$

$BWEB[1:0] = ADDR B[1:0]$

DP 模式下，字节使能写操作模式只支持 x18(16)写数据端口宽度，写数据端口宽度为 x9(x8)时禁止使用字节使能功能，此时字节使能功能与写使能功能重复，字节使能功能无效。

字节使能写操作模式可以与 NW、TW 或 RBW 写操作模式共存。TW 写操作模式下字节使能模式 DRM 的读写时序图如下图所示：



图 3-5 字节使能读写时序图 (TW 模式)

3.6 内部寄存器

DRM 内部寄存器包含输入寄存器 (IR)、输出寄存器 (OR) 和内部锁存器 (Latch)。

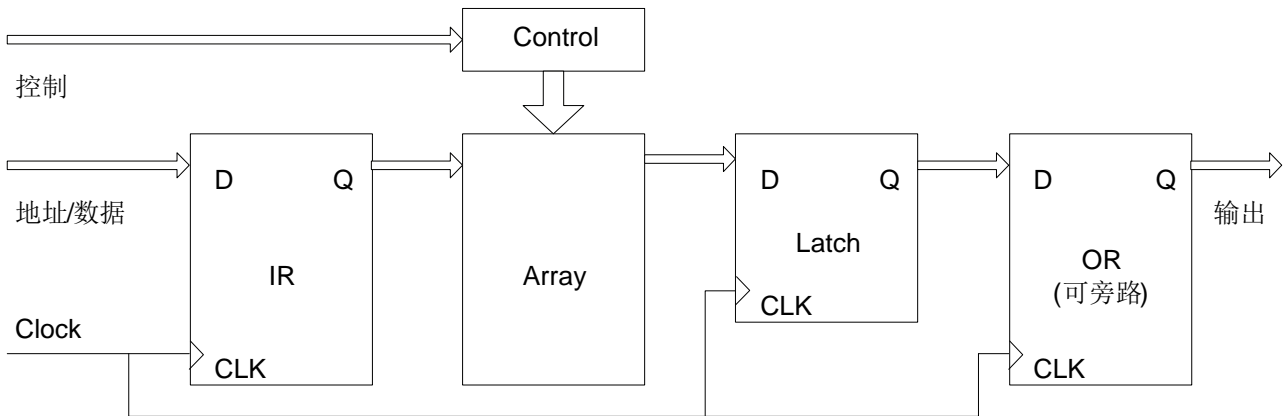


图 3-6 DRM 寄存器逻辑图

端口地址、数据、部分控制信号都有相应的输入寄存器 (IR)，通过 IR 可以实现同步写操作，IR 不可旁路，不可配置。在地址输入端口内置地址锁存选择通路，其逻辑实现如下图所示：

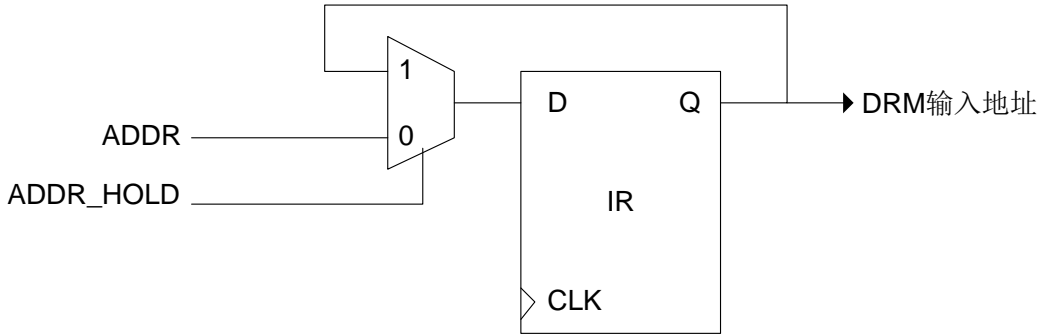


图 3-7 地址 IR 逻辑图

由地址锁存信号 ADDR_HOLD 控制地址锁存选择通路，ADDR_HOLD 保持高电平则输入寄存器的输出地址数据保持为当前输出不变，其时序图如下：

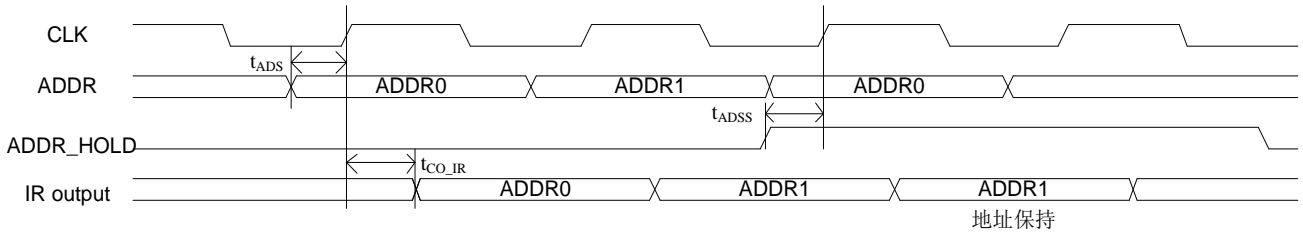


图 3-8 地址 IR 时序图

针对数据输出端口，DRM 特别提供了可选的输出寄存器（OR），以取得更好的时序性能，输出寄存器是否有效由 GTP 的参数 DOA_REG/DOB_REG 决定，例如 DOA_REG 为 1 时，A 端口输出寄存器有效，DOA_REG 为 0 时，A 端口输出寄存器被旁路。在读操作中，输出寄存器旁路时，DOA/DOB 输出为锁存器输出，在同一个读时钟周期时钟上升沿输出。B 端口与之类似。

如下图所示，A 端口输出寄存器有效时可由独立的时钟使能信号 ORCEA 控制，当 ORCEA 为高电平时，输出寄存器的输出数据每个时钟周期上升沿根据输入跳变；当 ORCEA 为低电平时，输出寄存器的输出数据维持之前的数据不变；输出寄存器旁路时 ORCEA 无效。当 ORCEA 为常量 1 时，输出寄存器的使用会把 A 端口读操作的延迟从一个时钟周期增加到两个时钟周期（Case 1）；在具有流控功能的流水线设计应用中，用户也可以用逻辑灵活控制 ORCEA（Case 2）。

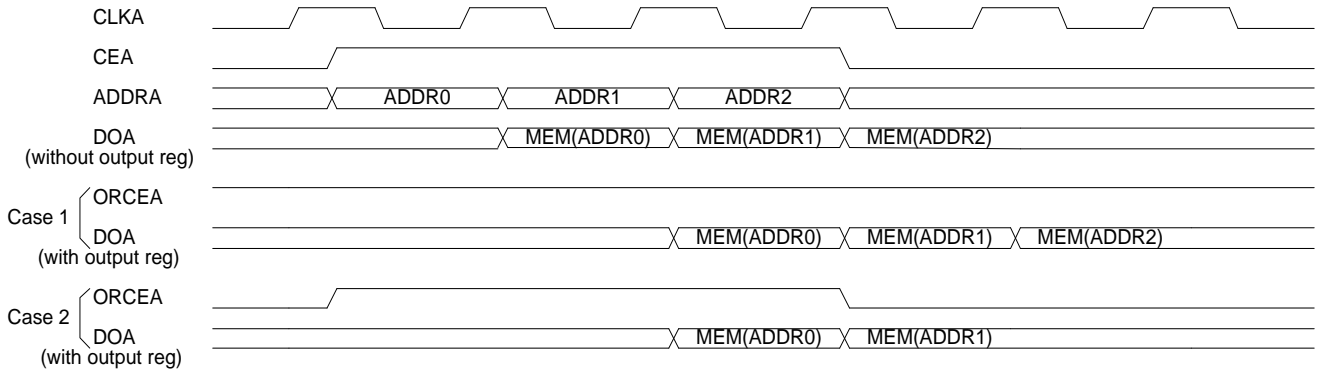


图 3-9 带输出寄存器的读时序

3.7 复位信号

3.7.1 同步、异步复位

如下图所示，DRM 输出(包括 Core 中 Latch 输出和 OR 输出)支持同步或者异步复位：

- 1.同步复位：RSTA/B 同步于 CLKA/B，需满足对时钟沿的建立/保持时间；
- 2.异步复位：RSTA/B 独立于 CLKA/B，输出复位于 RSTA/B 的边沿。

A/B 两个端口共用同步/异步复位选择设置。

异步复位同步复位失效，当 RSTA/B 有效时，独立于时钟复位内部 Latch 以及输出寄存器，当 RSTA/B 失效时，内部 Latch 以及输出寄存器同步于时钟沿恢复正常功能。

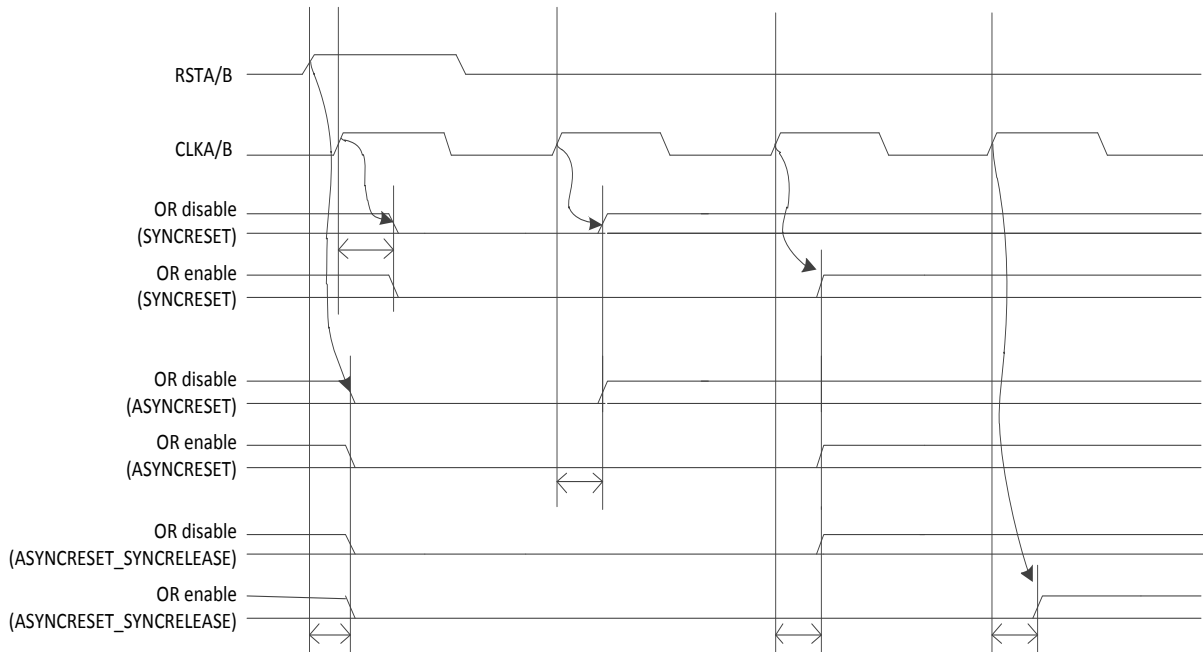


图 3-10 输出寄存器复位时序

3.7.2 Core Latch 复位

如下图所示在 Logos 系列 FPGA 的 DRM 单元中, A/B 两个端口都包括有数据输出 Latch 锁存器。这些 Latch 可以被同步或异步复位。其复位控制来自全局复位 GRSN 或者本地端口复位 RSTA/B。

注: GSR_N 为异步复位。GSR_N 复位可以通过配置位禁止。

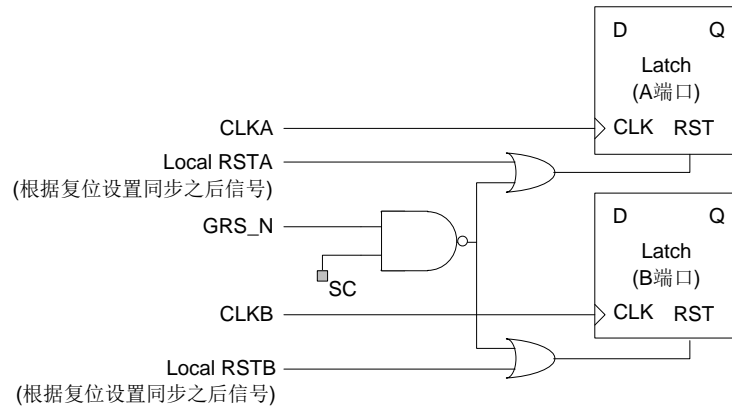


图 3-11 Core 复位

3.8 全局信号时序

如下图所示, T0 至 T1 之间为下载位流阶段, 包括 sc_bit 和 rom 初始化内容, T1 时刻之前, DRM 所有输出为 1; GLOGEN 释放后, DRM 所有输出状态, 与配置和输入有关。

GRS_N 和 GWEN 以及 DONE (DRM 中不可用) 的释放时刻, 用户通过 CCS 可分别独立配置, 可配置在 T2 或 T3 或 T4 时刻; 在 DONE 信号置 1 后, 进入用户模式。

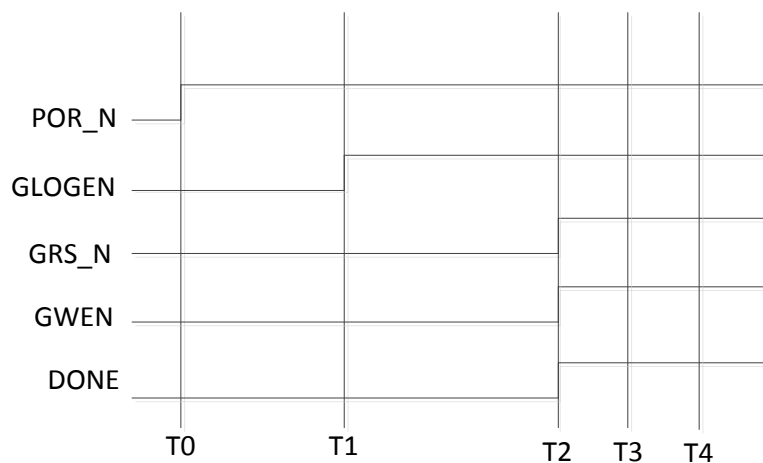


图 3-12 全局信号时序示意图

3.9 应用示例

3.9.1 单个 DRM 配置

本小节举例说明了单个 18K DRM 的 GTP 配置步骤,用户还可以通过 Pango Design Suite 软件内嵌的 IP Compiler 工具直接生成 DRM 的 IP, 详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例为混合位宽 DP 模式, A 端口配置为 2Kx8, B 端口配置为 1Kx16 且使用字节使能写模式, 两端口不同时钟, 使能输出寄存器, 读写模式 TW。9K DRM 的配置步骤与之类似。

对单个 18K DRM 按如下步骤进行配置:

a. 按下表所述对 DRM 的参数进行配置:

表 3-4 DRM 参数配置

参数名称	配置值	说明
DATA_WIDTH_A	8	将 A 端口配置为 2Kx8 模式
DATA_WIDTH_B	16	将 B 端口配置为 1Kx16 模式
WRITE_MODE_A	"TRANSPARENT_WRITE"	将 A 端口读写模式配置为 TW
WRITE_MODE_B		将 B 端口读写模式配置为 TW
DOA_REG	1	使能 A 端口输出寄存器
DOB_REG		使能 B 端口输出寄存器
RAM_MODE	"TRUE_DUAL_PORT"	将 DRM 配置为双口模式

b. 按下表所示对 DRM 的端口进行连接:

表 3-5 DRM 端口连接

端口名称	连接信号	说明
ADDRA[13:0]	{addra[10:0],3'b111}	将输入地址信号 addra[10:0]接到 ADDRA[13:3], 不使用字节使能功能, ADDRA[2:0]接高电平, 详细地址连接说明见 地址和数据端口 Mapping
ADDRB[13:0]	{addrb[9:0], 2'b11, bweb[1:0]}	将输入地址信号 addrb[9:0]连接到 ADDRb[13:4], 使用字节使能功能, ADDRb[3:2]接高电平, ADDRb[1:0]接字节使能信号
ADDRA_HOLD	1'b0	不使用 A/B 端口地址锁存功能, 接低电平
ADDRB_HOLD		
DIA[17:0]	dia[7:0]	将输入数据信号 dia[7:0]连接到 DIA[7:0], 未使用的 DIA 高位端口悬空, 详细数据端口连接说明见 地址和数据端口 Mapping
DIB[17:0]	{1'b0, dib[15:8], 1'b0, dib[7:0]}	将输入数据信号 dib[15:0]连接到 GTP 端口 {DIB[16:9], DIB[7:0]}, DIB[8]、DIB[17]为字节附加信息位, 详见 字节附加信息位 , 数据位宽为 2^N 时悬空或接低电平
CSA[2:0]	3'b0	未使用地址扩展功能, 接低电平
CSB[2:0]		

端口名称	连接信号	说明
DOA[17:0]	doa[7:0]	将输出数据信号 doa[7:0]连接到 GTP 端口 DOA[7:0]
DOB[17:0]	{1'bz, dob[15:8], 1'bz, dob[7:0]}	将输出数据信号 dob[15:0]连接到 GTP 端口 {DOB[16:9], DOB[7:0]}, DOB[8]、DOB[17]则悬空

c. A/B 端口其余信号：将 A/B 端口的时钟、时钟使能、写使能、输出寄存器时钟使能、数据寄存器复位分别连接到相应的 GTP 端口上；

d. 其余不使用的参数：其余不使用的参数不进行设置，使用默认值；

e. 其余不使用的端口：其余不使用的端口悬空（不使用也必须进行连接的端口已在上文描述）。

配置后的 GTP 如下所示：

```
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 多个 DRM 级联配置

本文所述级联特指地址深度级联，数据级联时单个 DRM 配置相同，只需进行数据端口拼接，本文不再单独进行描述。

通过 18K DRM 进行地址深度级联，不增加外部逻辑对地址进行处理，DRM 地址最大可级联 3bit（CSA/CSB 位宽为 3），总地址位宽为 17bit；数据级联时最大数据位宽取决于器件 DRM 资源。

本小节举例说明了对多个 18K DRM 进行 GTP 级联配置的步骤，用户还可以通过 Pango Design Suite 软件内嵌的 IP Compiler 工具直接生成级联的 DRM 的 IP，详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例级联了 4 个 A/B 端口都配置为 16Kx1 的 DP 18K DRM，总数据位宽 2bit，总地址位宽 15bit。由于 9K DRM 无地址扩展相关信号和参数，因此不推荐使用 9K DRM 进行级联。

级联的 4 个 DRM 分别命名为 DRM_1_1, DRM_1_2, DRM_2_1, DRM_2_2。以 DRM_1_2 为例，第一个数 1 为地址级联坐标，共 2 级地址级联，地址由单个 DRM 的 14bit 扩展为 15bit；第二个数 2 为数据级联坐标，共 2 级数据级联，数据由单个 DRM 的 1bit 扩展为 2bit。对多个 18K DRM 进行级联时，按如下步骤进行配置：

- a. 单个 DRM 配置：按照与单个 DRM 配置类似步骤对 4 个 18K DRM 分别配置为 16Kx1 的 DP 18K DRM；
- b. 按下表所述对 DRM 的地址扩展控制参数进行配置：

表 3-6 DRM 参数配置

参数名称	DRM 名称	配置值
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

c. 按下表所示对 DRM 的级联相关端口进行连接:

表 3-7 DRM 端口连接

端口名称	DRM 名称	连接信号
CSA	DRM_1_1	{2'b0,addra[14]}
	DRM_1_2	
	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]

d. 输出数据选择: addra[14]通过寄存器延迟一拍后为 0 时 A 端口输出数据为 doa_0[1:0], 为 1 时 A 端口输出数据为 doa_1[1:0], B 端口也与之类似。

配置后的 GTP 如下所示:

```
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
);
//输出数据选择
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 clk b or posedge rstb)
begin
if (rstb)
sel_b <= 1'b0;
else if (~addrb_hold & ceb)
sel_b <= addrb[14];
end

```

```
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];
```

4 SDP 模式

4.1 模式介绍

RAM 的端口模式由参数 RAM_MODE 决定，当参数 RAM_MODE 的值为 "SIMPLE_DUAL_PORT" 时，RAM 进入简单双口模式。本文将详细介绍 18K 的 SDP RAM，9K 与 18K 结构与功能基本一致，仅位宽不同。

SDP RAM 模式支持：

- 双端口读和写，但是其中一个端口作为读端口，另一个端口作为写端口。
- 任意一个端口不能同时做读或者写操作。
- 两端口有独立的位宽设置
- x32/x36 位宽 A 端口固定为写入端口(WEA 为高时有效)，B 端口固定为读出端口(WEB 为低时有效)。

4.2 数据端口

每个 9K、18K 的 DRM 也能被配置为 SDP（简单双口）RAM。在这个模式下，DRM 端口数据位宽增大至 36 bits，SDP RAM 模式包含两对端口（每对有 A、B 两个不同的端口），A、B 两个端口中一个端口专用于数据写入，另一个端口专用于数据读取，读写端口同样均支持不同的时钟。与 DPRAM 相同的是，若同时通过两个端口对同一地址进行读写操作会引起冲突。下图展示了 18K SDP RAM 的结构图。

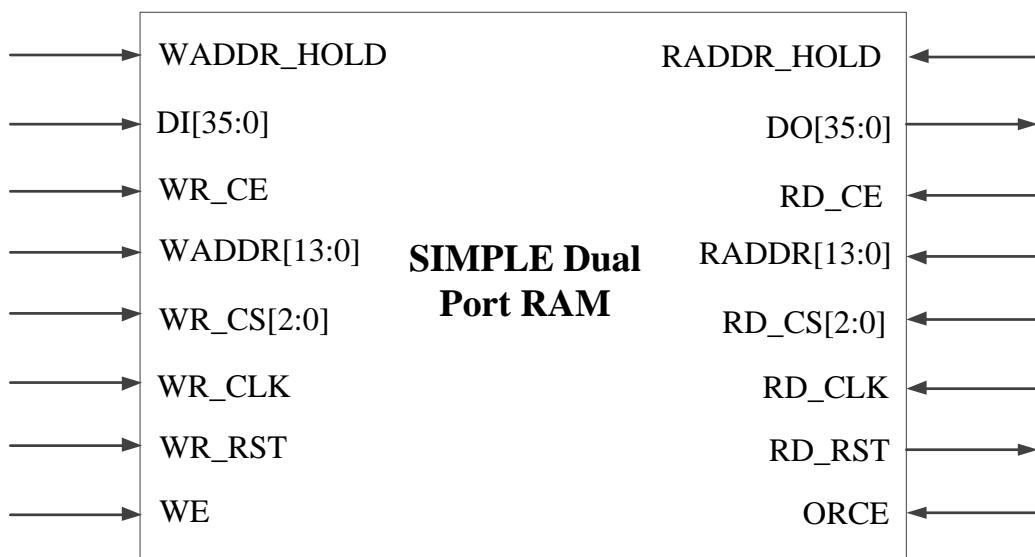


图 4-1 SDP RAM 数据端口

下表罗列了端口名称和端口的描述。

表 4-1 SDP RAM 端口名称及描述

端口名称	方向	描述	端口名称	方向	描述
WADDR_HOLD	输入	写端口地址锁存信号	RADDR_HOLD	输入	读端口地址锁存信号
DI	输入	写端口数据输入	DO	输出	读端口数据输出
WR_CE	输入	写端口输入寄存器时钟使能	RD_CE	输入	读端口输入寄存器时钟使能
WADDR	输入	写端口地址输入	RADDR	输入	读端口地址输入
WR_CS	输入	写端口地址扩展	RD_CS	输入	读端口地址扩展
WR_CLK	输入	写端口时钟	RD_CLK	输入	读端口时钟
WR_RST	输入	写端口输出寄存器复位	RD_RSTB	输入	读端口输出寄存器复位
WE	输入	写使能	ORCE	输入	输出寄存器时钟使能

注：1.SDP 模式的 x32/x36 位宽下，DI 由 A/B 端口的 DIA/DIB 拼接而成，DO 同理，详见[地址和数据端口 Mapping](#)；

2.9K DRM 无 CSA/CSB 端口，不支持地址扩展，如需 DRM 级联进行地址深度扩展，建议使用 18K DRM。

4.3 位宽组合

RAM 的端口位宽由 GTP 中的参数 DATA_WIDTH_A/DATA_WIDTH_B 决定，例如，当参数 DATA_WIDTH_A 的值为 4 时，A 端口数据位宽被设置为 4 bit。SDP 模式支持 A、B 端口设置不同数据位宽。

如下表所示为 9K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合。

表 4-2 9K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合

		写端口 0/1									
		No Parity Bits						With Parity Bits			
		8Kx1	4Kx2	2Kx4	1Kx8	512x16	256x32	1Kx9	512x18	256x36	
读端口 0/1	No Parity Bits	8Kx1	√	√	√	√	√	√			
		4Kx2	√	√	√	√	√	√			
		2Kx4	√	√	√	√	√	√			
		1Kx8	√	√	√	√	√	√			
		512x16	√	√	√	√	√	√			
		256x32	√	√	√	√	√	√			
	With Parity Bits	1Kx9							√	√	√
		512x18							√	√	√
		256x36							√	√	√

注：√表示支持的位宽组合。

如下表所示为 18K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合。

表 4-3 18K DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合

			写端口 0/1								
			No Parity Bits					With Parity Bits			
			16Kx1	8Kx2	4Kx4	2Kx8	1Kx16	512x32	2Kx9	1K x18	512x36
读端口 0/1	No Parity Bits	8Kx1	√	√	√	√	√	√			
		4Kx2	√	√	√	√	√	√			
		2Kx4	√	√	√	√	√	√			
		1Kx8	√	√	√	√	√	√			
		512x16	√	√	√	√	√	√			
		256x32	√	√	√	√	√	√			
	With Parity Bits	2Kx9							√	√	√
		1Kx18							√	√	√
		512x36							√	√	√

注：√表示支持的位宽组合。

4.4 读写操作

SDP 模式不支持读写模式设置，在使用时禁止设置成 TW 或 RBW 模式，只能设置为默认的 NW 模式，即 GTP 中参数 WRITE_MODE_A/WRITE_MODE_B 的值必须为默认的 "NORMAL_WRITE"，否则 DRM 将进入非正常的写操作状态。

SDP 模式有两个相对独立的端口，若同时通过两个端口对同一地址进行读写操作会引起冲突。DRM 禁止两端口同时向同一地址写入数据、禁止两端口同时读写同一地址，需要在实际应用中通过用户逻辑加以规避。

SDP RAM 的典型时序如下图所示，图中 SDP 读写端口同时钟，各信号名称为表 4-1 中定义的 SDP 模式端口名称，括号中为实际连接的 GTP 端口名称。

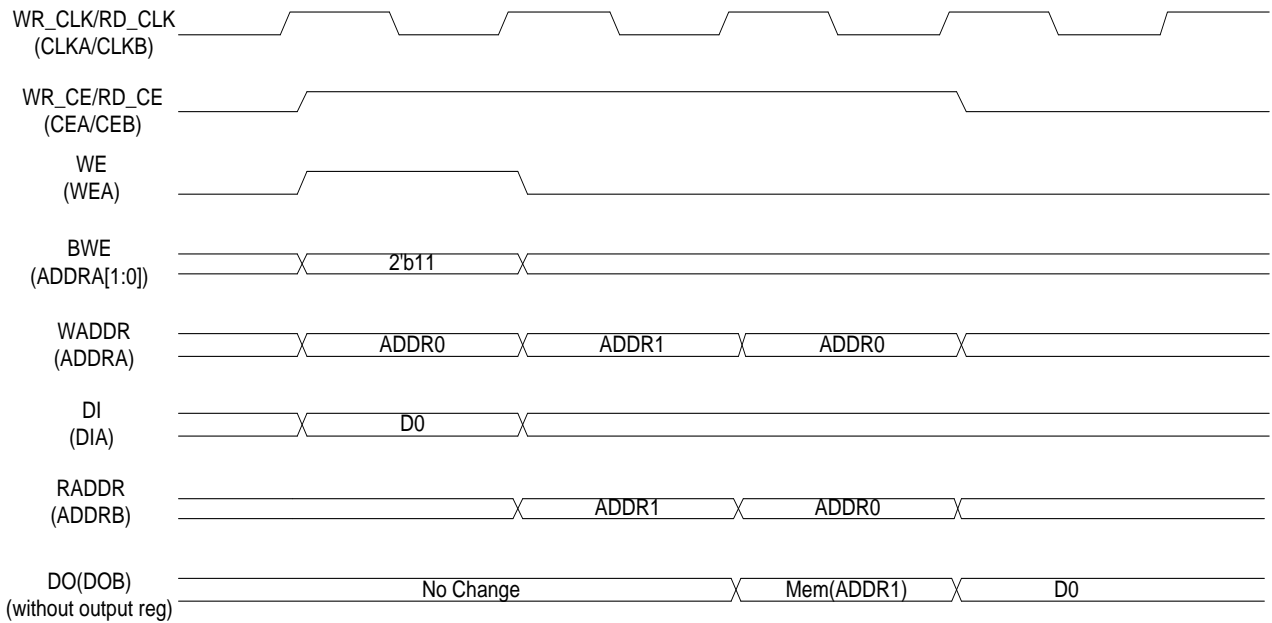


图 4-2 SDP RAM 典型时序图

4.5 字节使能功能

SDP 模式的字节使能功能与 DP 模式类似，详见[字节使能功能](#)。SDP 模式的字节使能与 DP 模式主要差异为以下两点：

- SDP 模式一个端口固定为写端口，一个端口固定为读端口；
- SDP 模式字节使能信号还支持 x32(36)位数据宽度模式，此时，读写数据通过 A/B 端口拼接而成， $BWE[3:0] = ADDR_A[3:0]$ 。

SDP 模式下字节使能模式 DRM 的读写时序图如下图所示，图中各信号名称为表 4-1 中定义的 SDP 模式端口名称，括号中为实际连接的 GTP 端口名称。

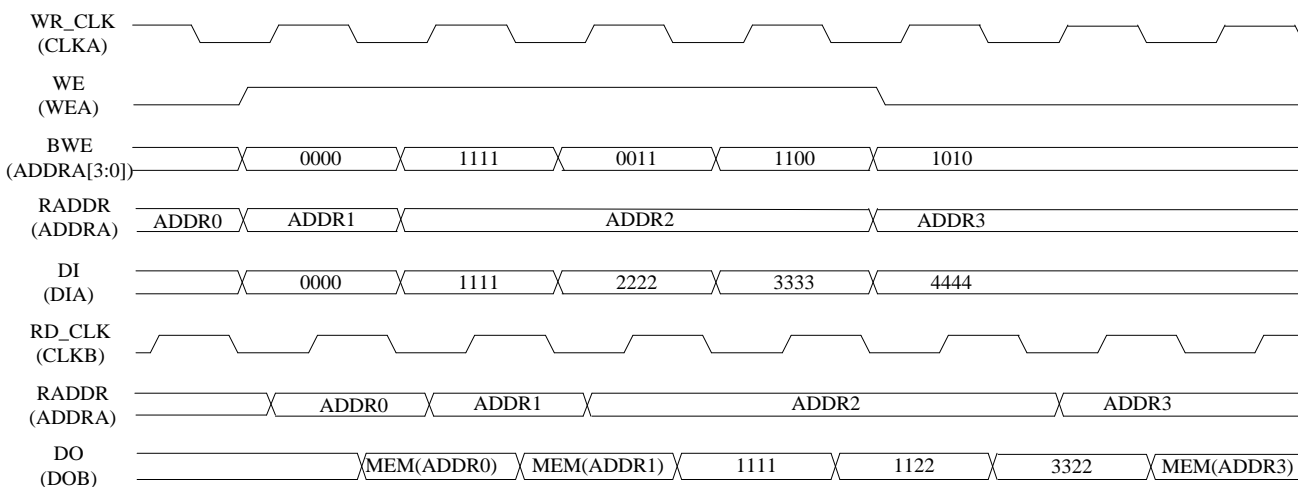


图 4-3 字节使能读写时序图（SDP 模式）

4.6 内部寄存器

SDP 模式的内部寄存器和 DP 模式相同，详见[内部寄存器](#)。

4.7 复位信号

SDP 模式的复位信号和 DP 模式相同，详见[复位信号](#)。

4.8 全局信号时序

SDP 模式的全局信号时序和 DP 模式相同，详见[全局信号时序](#)。

4.9 应用示例

4.9.1 单个 DRM 配置

本小节举例说明了单个 18K DRM 的 GTP 配置步骤，用户还可以通过 Pango Design Suite 软件内嵌的 IP Compiler 工具直接生成 DRM 的 IP，详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例为混合位宽 SDP 模式，读端口 A 配置为 2Kx9，写端口 B 端口配置为 512x36（由 A/B 端口 DOA/DOB 拼接实现），两端口不同时钟，使能输出寄存器。9K DRM 的配置步骤与之类似。

对单个 18K DRM 按如下步骤进行配置：

- a. 按下表所述对 DRM 的参数进行配置：

表 4-4 DRM 参数配置

参数名称	配置值	说明
DATA_WIDTH_A	9	将 A 端口配置为 2Kx9 模式
DATA_WIDTH_B	36	将 B 端口配置为 512x36 模式
WRITE_MODE_A	"NORMAL_WRITE"	将 A 端口读写模式配置为 NW
WRITE_MODE_B		将 B 端口读写模式配置为 NW
DOA_REG	1	使能 A 端口输出寄存器
DOB_REG		使能 B 端口输出寄存器
RAM_MODE	"SIMPLE_DUAL_PORT"	将 DRM 配置为简单双口模式

b. 按下表所示对 DRM 的端口进行连接:

表 4-5 DRM 端口连接

端口名称	连接信号	说明
ADDRA[13:0]	{waddr[10:0],3'b111}	A 端口做写端口,将写地址信号 waddr[10:0]连接到 ADDRA[13:3], ADDRA[2:0]接高电平,详细地址连接说明见 地址和数据端口 Mapping
ADDRB[13:0]	{raddr[8:0],5'b11111}	B 端口做读端口,将读地址信号 raddr[8:0]连接到 ADDRb[13:5], ADDRb[4:0]接高电平
ADDRA_HOLD	1'b0	不使用 A/B 端口地址锁存功能,接低电平
ADDRB_HOLD		
DIA[17:0]	di[8:0]	将写数据信号 di[8:0]连接到 DIA[8:0],未使用的 DIA 高位端口悬空,详细数据端口连接说明见 地址和数据端口 Mapping
DIB[17:0]	悬空或接低电平	B 端口为读端口且未进行写入数据拼接,GTP 端口 DIB 悬空或接低电平
CSA[2:0]	3'b0	未使用地址扩展功能,接低电平
CSB[2:0]		
DOA[17:0]	do[17:0]	A/B 端口拼接实现 36bit 数据位宽,DOA 输出低 18bit 读数据
DOB[17:0]	do[35:18]	A/B 端口拼接实现 36bit 数据位宽,DOB 输出高 18bit 读数据
WEA	we	将写使能信号 we 连接到写端口 A 的 WEA 上
WEB	1'b0	读端口 B 的写使能 WEB 接低电平

c. 读写端口其余信号:将读写端口的时钟、时钟使能、输出寄存器时钟使能、数据寄存器复位分别连接到相应的 GTP 相应 A/B 端口上;

d. 其余不使用的参数:其余不使用的参数不进行设置,使用默认值;

e. 其余不使用的端口:其余不使用的端口悬空(不使用也必须进行连接的端口已在上文描述)。

配置后的 GTP 如下所示:

```
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 多个 DRM 级联配置

SDP 模式多个 18K DRM 级联与 DP 模式相同，详见[多个 DRM 级联配置](#)。

5 SP 模式

5.1 模式介绍

RAM 的端口模式由参数 RAM_MODE 决定，当参数 RAM_MODE 的值为 "SINGLE_PORT" 时，RAM 进入单口模式。本文将详细介绍 18K 的 SPRAM，9K 与 18K 结构与功能基本一致，仅位宽不同。

SPRAM 模式支持：

- 单端口读操作
- 单端口写操作
- x32/x36位宽A端口固定为写入控制端口（WEA为高时有效），B端口固定为读出控制端口（WEB为低时有效）。

5.2 数据端口

每个 9K、18K 的 DRM 也能被配置为 SP（单口）RAM。在这个模式下，DRM 端口数据位宽增大至 36 bits，SPRAM 模式在两个端口共享情况下，DRM 仅包含一个端口，SPRAM 模式可以对这个端口进行读写操作。下图展示了 18K SPRAM 的结构图。

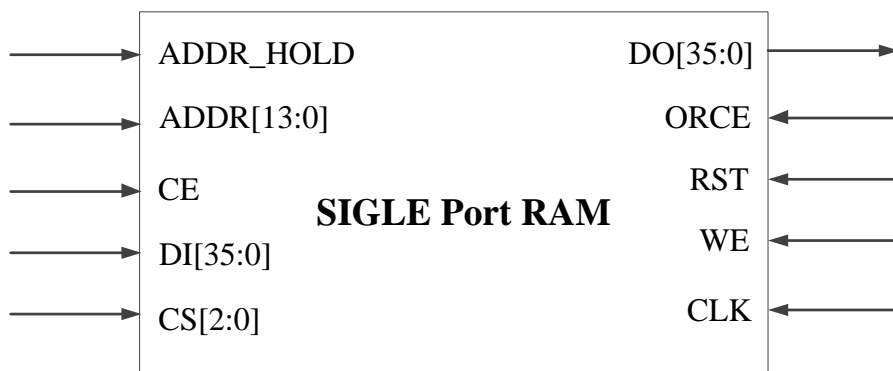


图 5-1 SP RAM 数据端口

下表罗列了 SPRAM 模式下的端口名称和端口的描述。

表 5-1 SP RAM 端口名称及描述

端口名称	方向	描述	端口名称	方向	描述
ADDR_HOLD	输入	地址锁存信号	DO	输出	读数据输出
ADDR	输入	地址输入	ORCE	输入	输出寄存器时钟使能
CE	输入	输入寄存器时钟使能	RST	输入	输出寄存器复位
DI	输入	写数据输入	WE	输入	写使能
CS	输入	地址扩展	CLK	输入	端口时钟

注：1.SP 模式的 x32/x36 位宽下，DI 由 A/B 端口的 DIA/DIB 拼接而成，DO 同理，详见[地址和数据端口 Mapping](#)；
2.9K DRM 无 CSA/CSB 端口，不支持地址扩展，如需 DRM 级联进行地址深度扩展，建议使用 18K DRM。

5.3 位宽组合

RAM 的端口位宽由 GTP 中的参数 DATA_WIDTH_A/DATA_WIDTH_B 决定，例如，当参数 DATA_WIDTH_A 的值为 4 时，A 端口数据位宽被设置为 4 bit。SP 模式的 A、B 端口需设置为相同数据位宽。

如下表所示为 18K DRM 模式下 Single Port RAM 模式允许的位宽。

表 5-2 18K DRM 模式 Single Port RAM 模式列表

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

注：1.√表示支持的位宽组合；

2.18K DRM 模式下 Single Port RAM 模式，512x32、512x36 模式禁止直接设置 TW、RBW 读写模式，需要额外配置，详见[读写操作](#)。

如下表所示为 9Kb DRM 模式下 Simple Port RAM 模式允许的位宽。

表 5-3 9K DRM 模式 Single Port RAM 模式列表

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

注：1.√表示支持的位宽组合；

2.9K DRM 模式下 Single Port RAM 模式，256x32、256x36 模式禁止直接设置 TW、RBW 读写模式，需要额外配置，详见[读写操作](#)。

5.4 读写操作

SP 模式的各读写模式的读写操作与 DP 模式的单端口类似，详见[读写操作](#)。

DRM 在 SP 模式下各位宽都支持设置读写模式为 NW（Normal Write）模式，SP 模式下 18 bit 及以下数据端口位宽读写模式还可直接设置为 TW（Transparent Write）或 RBW

(Read before Write) 模式，但 SP 模式下 32 bit 及 36 bit 端口数据位宽禁止直接设置读写模式为 TW 和 RBW 模式，需要设置为默认的 NW 模式，具体支持情况如下表：

表 5-4 SP 模式读写模式设置支持情况

数据端口位宽	是否支持	读写模式实现方式
1 (18K/9K)	支持	GTP 中可直接设置模式参数为 SP 模式，18K 及 9K 的 18 bit 及以下位宽下读写模式可直接设置为 NW、TW、RBW
2 (18K/9K)	支持	
4 (18K/9K)	支持	
8 (18K/9K)	支持	
9 (18K/9K)	支持	
16 (18K/9K)	支持	
18 (18K/9K)	支持	
32 (18K/9K)	不支持	GTP 中 SP 模式的 32/36 bit 位宽禁止直接设置读写模式，需要使用默认的 NW；TW、RBW 实现需要额外配置，具体方法见下一段
36 (18K/9K)	不支持	

DRM 禁止通过直接配置实现 32/36 bit SP 模式下的 TW 和 RBW 模式，需要额外的配置，具体实现方式如下：

- a. Port A 和 Port B 同时配置为 16/18 bit DP 模式（分别针对 32/36 bit SP）；
- b. Port A 和 Port B 配置为相同的写模式；其他配置位也需要相同；
- c. Port A 和 Port B 的 CE, CS, WE, ORCE, RST, CLK 等控制信号分别并联起来；
- d. DIA、DIB 并联作为数据输入；DOA、DOB 并联作为数据输出；
- e. Port A 和 Port B 地址输入端口需同时接地址输入；ADDRA[4]和 ADDR[4]需分别接 0 或者 1（相反），如下图所示：

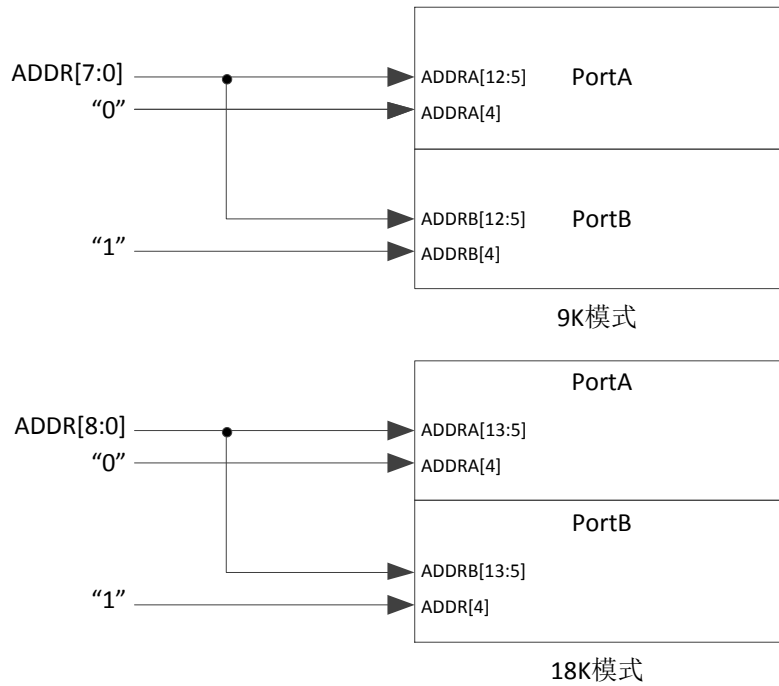


图 5-2 SP 模式的 32/36 bit 模式下 TW 和 RBW 模式地址线配置

- f. BWEA、BWEB 信号的连接需要分别参考两个端口 DP 16/18 bit 模式下的连接。

5.5 字节使能功能

SP 模式的字节使能功能与 SDP 模式类似，详见[字节使能功能](#)。

5.6 内部寄存器

SP 模式的内部寄存器和 DP 模式相同，详见[内部寄存器](#)。

5.7 复位信号

SP 模式的复位信号和 DP 模式相同，详见[复位信号](#)。

5.8 全局信号时序

SP 模式的全局信号时序和 DP 模式相同，详见[全局信号时序](#)。

5.9 应用示例

5.9.1 单个 DRM 配置

本小节举例说明了单个 18K DRM 的 GTP 配置步骤,用户还可以通过 Pango Design Suite 软件内嵌的 IP Compiler 工具直接生成 DRM 的 IP, 详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例为 SP 模式, 读写端口配置为 512x32, 使能输出寄存器, 读写模式 NW。9K DRM 的配置步骤与之类似。

对单个 18K DRM 按如下步骤进行配置:

a. 按下表所述对 DRM 的参数进行配置:

表 5-5 DRM 参数配置

参数名称	配置值	说明
DATA_WIDTH_A	32	将 A 端口配置为 512x32 模式
DATA_WIDTH_B		将 B 端口配置为 512x32 模式
WRITE_MODE_A	"NORMAL_WRITE"	将 A 端口读写模式配置为 NW
WRITE_MODE_B		将 B 端口读写模式配置为 NW
DOA_REG	1	使能 A 端口输出寄存器
DOB_REG		使能 B 端口输出寄存器
RAM_MODE	"SINGLE_PORT"	将 DRM 配置为单口模式

b. 按下表所示对 DRM 的端口进行连接:

表 5-6 DRM 端口连接

端口名称	连接信号	说明
ADDRA[13:0]	{addr[8:0],5'b11111}	A/B 端口地址接相同地址信号, 将输入地址信号 addr[8:0]连接到 ADDRA[13:5], ADDRA[4:0]接高电平, ADDR B 连接相同, 详细地址连接说明见 地址和数据端口 Mapping
ADDRB[13:0]		
ADDRA_HOLD	1'b0	不使用 A/B 端口地址锁存功能, 接低电平, 如果使用地址锁存功能 A/B 端口需接相同信号
ADDRB_HOLD		
DIA[35:0]	{1'b0, di[15:8], 1'b0, di[7:0]}	A/B 端口拼接实现 32bit 写数据端口, DIA 输入低 16bit 写数据 di[15:0], 详细数据端口连接说明见 地址和数据端口 Mapping
DIB[35:0]	{1'b0, di[31:24], 1'b0, di[23:16]}	A/B 端口拼接实现 32bit 写数据端口, DIB 输入高 16bit 写数据 di[31:16], DIB[8]、DIB[17]为字节附加信息位, 详见 字节附加信息位
CSA[2:0]	3'b0	未使用地址扩展功能, 接低电平, 如果使用地址扩展功能 A/B 端口需接相同信号
CSB[2:0]		
DOA[35:0]	{1'bz, do[15:8], 1'bz, do[7:0]}	A/B 端口拼接实现 32bit 读数据端口, DOA 输出低 16bit 读数据 do[15:0], DOA[8]、DOA[17]则悬空

端口名称	连接信号	说明
DOB[35:0]	{1'bz, do[31:24], 1'bz, do[23:16]}	A/B 端口拼接实现 32bit 读数据端口，DOB 输出高 16bit 读数据 do[31:15]，DOB[8]、DOB[17]则悬空

c. 读写端口其余信号：将读写端口的时钟、时钟使能、输出寄存器时钟使能、数据寄存器复位、写使能分别连接到相应的 GTP 相应 A/B 端口上，上述信号 A/B 端口都需相同连接；

d. 其余不使用的参数：其余不使用的参数不进行设置，使用默认值；

e. 其余不使用的端口：其余不使用的端口悬空（不使用也必须进行连接的端口已在上文描述）。

配置后的 GTP 如下所示：

```
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 多个 DRM 级联配置

SP 模式多个 18K DRM 级联与 DP 模式相同，详见[多个 DRM 级联配置](#)。

6 ROM 模式

6.1 模式介绍

RAM 的端口模式由参数 RAM_MODE 决定，当参数 RAM_MODE 的值为"ROM"时，RAM 进入 ROM 模式。本文将详细介绍 18K 的 ROM，9K 与 18K 结构与功能基本一致，仅位宽不同。

ROM模式支持只支持数据读出，不支持数据写入。

6.2 数据端口

DRM 可以配置为 ROM，通过配置接口可以初始化 ROM 内容。ROM 模式对 9K 及 18K 模式的端口只读。如下图所示为 ROM 模式的结构图。

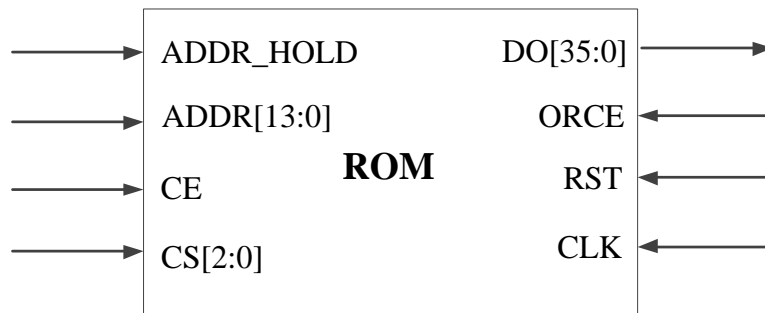


图 6-1 ROM 模式数据端口

下表罗列了 ROM 模式下的端口名称和端口的描述。

表 6-1 ROM 端口名称及描述

端口名称	方向	描述	端口名称	方向	描述
ADDR_HOLD	输入	地址锁存信号	DO	输出	读数据输出
ADDR	输入	地址输入	ORCE	输入	输出寄存器时钟使能
CE	输入	输入寄存器时钟使能	RST	输入	输出寄存器复位
CS	输入	地址扩展	CLK	输入	端口时钟

注：1.ROM 模式的 x32/x36 位宽下，DI 由 A/B 端口的 DIA/DIB 拼接而成，DO 同理，详见[地址和数据端口 Mapping](#)；
2.9K DRM 无 CSA/CSB 端口，不支持地址扩展，如需 DRM 级联进行地址深度扩展，建议使用 18K DRM。

6.3 位宽组合

RAM 的端口位宽由 GTP 中的参数 DATA_WIDTH_A/DATA_WIDTH_B 决定，例如，当参数 DATA_WIDTH_A 的值为 4 时，A 端口数据位宽被设置为 4 bit。ROM 模式的 A、B

端口需设置为相同数据位宽。

如下表所示为 18K DRM 模式下 ROM 模式允许的位宽。

表 6-2 18K DRM 模式 ROM 模式列表

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

注：√表示支持的位宽组合；

如下表所示为 9K DRM 模式下 ROM 模式允许的位宽。

表 6-3 9K DRM 模式 ROM 模式列表

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

注：√表示支持的位宽组合；

6.4 读写操作

ROM模式下，DRM不支持写操作，读操作的时序与DP模式相同，详见[读写操作](#)。

6.5 内部寄存器

ROM模式的内部寄存器和DP模式相同，详见[内部寄存器](#)。

6.6 复位信号

ROM模式的复位信号和DP模式相同，详见[复位信号](#)。

6.7 全局信号时序

ROM模式的全局信号时序和DP模式相同，详见[全局信号时序](#)。

6.8 应用示例

6.8.1 单个 DRM 配置

本小节举例说明了单个 18K DRM 的 GTP 配置步骤，用户还可以通过 Pango Design Suite

软件内嵌的 IP Compiler 工具直接生成 DRM 的 IP，详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例为 ROM 模式，读写端口配置为 1Kx18，使能输出寄存器。9K DRM 的配置步骤与之类似。

对单个 18K DRM 按如下步骤进行配置：

a. 按下表所述对 DRM 的参数进行配置：

表 6-4 DRM 参数配置

参数名称	配置值	说明
DATA_WIDTH_A	18	将 A 端口配置为 1Kx18 模式
DATA_WIDTH_B		将 B 端口配置为 1Kx18 模式
DOA_REG	1	使能 A 端口输出寄存器
DOB_REG		使能 B 端口输出寄存器
RAM_MODE	"ROM"	将 DRM 配置为 ROM 模式
INIT_00	{18'h0000f, 18'h00003, 18'h00002, 18'h00001, 18'h00000}	将 ROM 的初始值参数设置为对应的地址值,即 INIT_00 的 1-18bit 对应地址 0, 设置为 0; 19-36bit 对应地址 1 则设置为 1, 以此类推, 具体初始化参数映射规则详见 初始化配置参数映射 ;
.....	
INIT_3F	{18'h003ff, 18'h003f3, 18'h003f2, 18'h003f1, 18'h003f0}	

b. 按下表所示对 DRM 的端口进行连接：

表 6-5 DRM 端口连接

端口名称	连接信号	说明
ADDRA[13:0]	{addr[9:0],4'b1111}	A/B 端口地址接相同地址信号, 将输入地址信号 addr[9:0]连接到 ADDRA[13:4], ADDRA[3:0]接高电平, ADDR B 连接相同, 详细地址连接说明见 地址和数据端口 Mapping
ADDRB[13:0]		
ADDRA_HOLD	1'b0	不使用 A/B 端口地址锁存功能, 接低电平, 如果使用地址锁存功能 A/B 端口需要接相同信号
ADDRB_HOLD		
DIA[17:0]	悬空	ROM 不支持写操作, 端口 DIA/DIB 悬空
DIB[17:0]		
CSA[2:0]	3'b0	未使用地址扩展功能, 接低电平, 如果使用地址控制功能 A/B 端口需要接相同信号
CSB[2:0]		
DOA[17:0]	do[17:0]	将输出数据信号 do[17:0]连接到 DOA[17:0]
DOB[17:0]	悬空	未进行数据拼接, DOB 悬空
WEA	1'b0	ROM 不支持写操作, 将 WEA/WEB 连接到低电平
WEB		

c. 读写端口其余信号：将读写端口的时钟、时钟使能、输出寄存器时钟使能、数据寄

寄存器复位分别连接到相应的 GTP 相应 A/B 端口上，上述信号 A/B 端口都需相同连接；

d. 其余不使用的参数：其余不使用的参数不进行设置，使用默认值；

e. 其余不使用的端口：其余不使用的端口悬空（不使用也必须进行连接的端口已在上文描述）。

配置后的 GTP 如下所示：

```
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 多个 DRM 级联配置

ROM 模式多个 18K DRM 级联与 DP 模式相同，详见[多个 DRM 级联配置](#)。

7 FIFO 模式

7.1 模式介绍

DRM_FIFO 由 DRM 和 DRM_FIFO_CTRL 模块组成，DTM_FIFO_CTRL 模块包括：读写指针生成、状态标志生成逻辑。FIFO 的顶层结构框图如下：

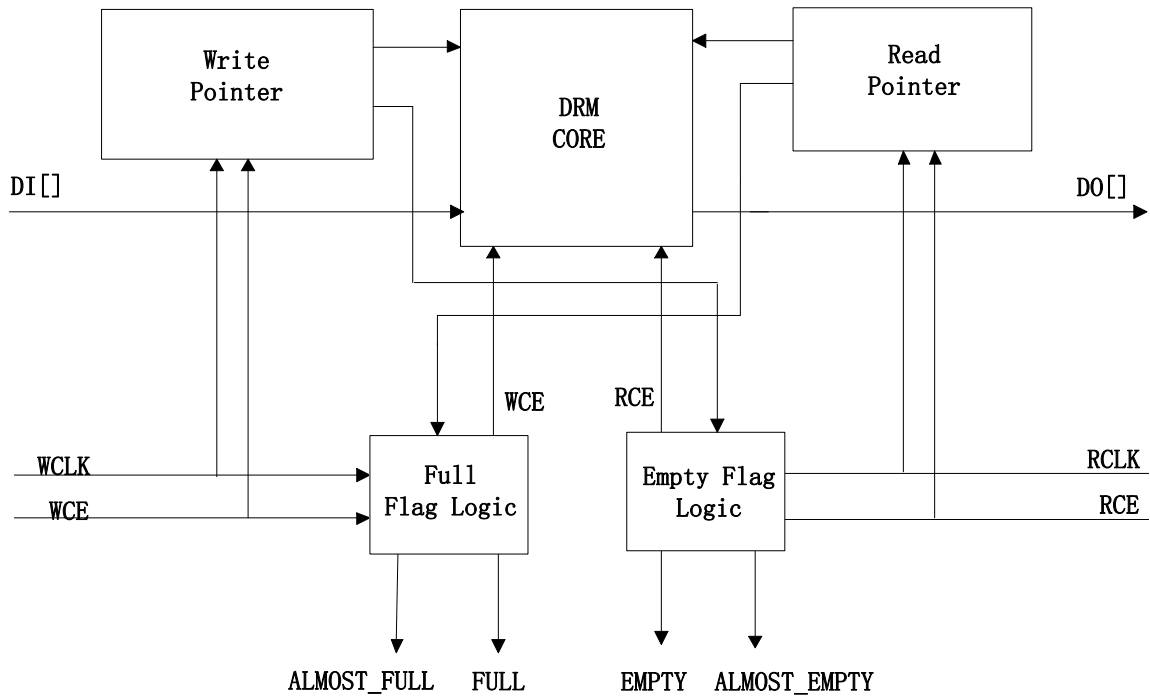


图 7-1 FIFO 顶层结构框图

用户可通过例化 GTP_DRM 配置为 SDP 模式再通过用户逻辑实现 DRM_FIFO_CTRL 模块或直接例化 GTP_FIFO18K 来实现 DRM_FIFO。

GTP_FIFO 分为同步和异步 FIFO，由参数 SYNC_FIFO 决定，当参数 SYNC_FIFO 的值为"TRUE"时，进入同步 FIFO 模式；当参数 SYNC_FIFO 的值为"FALSE"时，进入异步 FIFO 模式。此模式下，异步 FIFO 读写时钟域相互独立。写操作同步于写时钟而读操作同步于读时钟。

异步 FIFO 模式下，当写使能有效，且 FIFO 不为满时，在写时钟的上升沿向 FIFO 写入数据；当读使能有效，且 FIFO 不为空时，在读时钟的上升沿从 FIFO 读出数据。当 FIFO 写满时产生满信号，满信号同步于写时钟域。当 FIFO 读空时产生空信号，空信号同步于读时钟域。

同步 FIFO 模式下，读写时钟为同一个时钟。当写使能有效，且 FIFO 不为满时，在时钟的上升沿向 FIFO 写入数据；当读使能有效，且 FIFO 不为空时，在时钟的上升沿从 FIFO

读出数据。当 FIFO 写满时产生满信号，当 FIFO 读空时产生空信号。

DRM_FIFO 位宽模式详见附录。

7.2 数据端口

7.2.1 异步 FIFO（不含丢包重发功能）

此模式下数据流如下图所示：

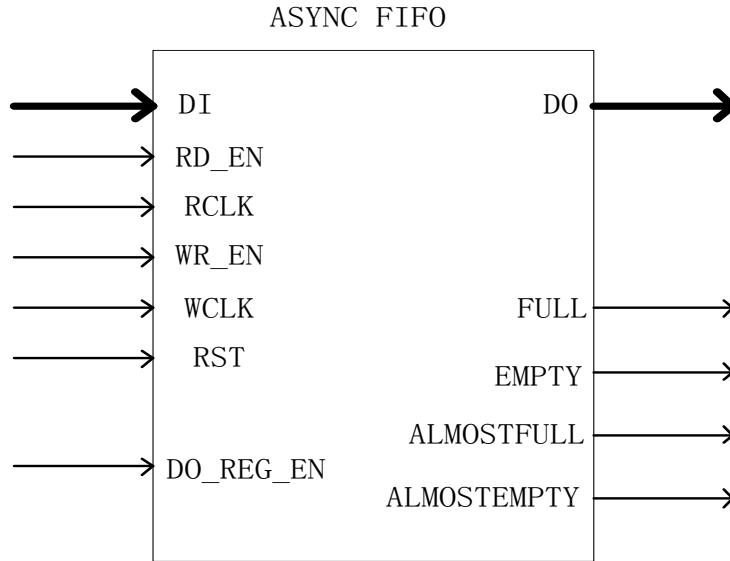


图 7-2 异步 FIFO 模式（不含丢包重发功能）

表 7-1 异步 FIFO 模式（不含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

7.2.2 异步 FIFO（支持丢包功能）

丢包功能：异步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。丢包功能可以单独使能，wr_eop 为输入信号。

此模式下数据流如下图所示：

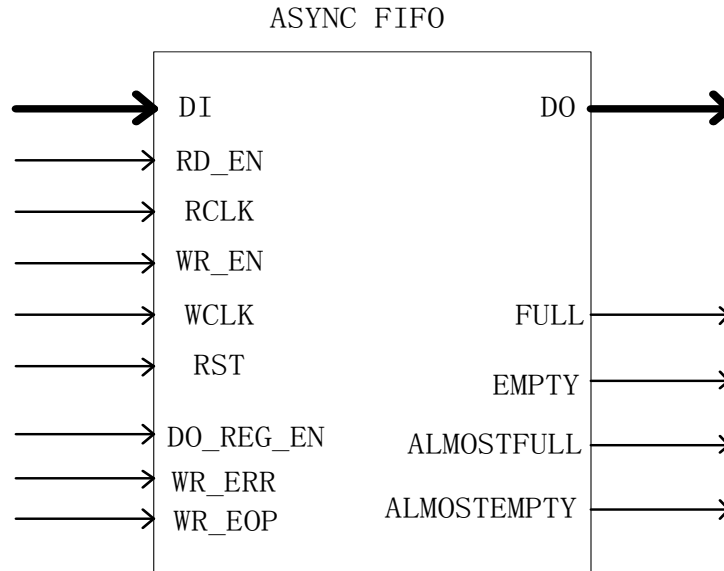


图 7-3 异步 FIFO 模式（支持丢包功能）

表 7-2 异步 FIFO 模式（支持丢包功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
DO[N:0]	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

7.2.3 异步 FIFO（包含丢包重发功能）

丢包功能：异步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。

重发功能：当异步 FIFO 读时钟检测到 rd_nak 信号，读地址退回到数据包包头所在的地址。

丢包功能和重发功能需要同时使能。wr_eop 输入连接到数据输入 DI。

此模式下数据流如下图所示：

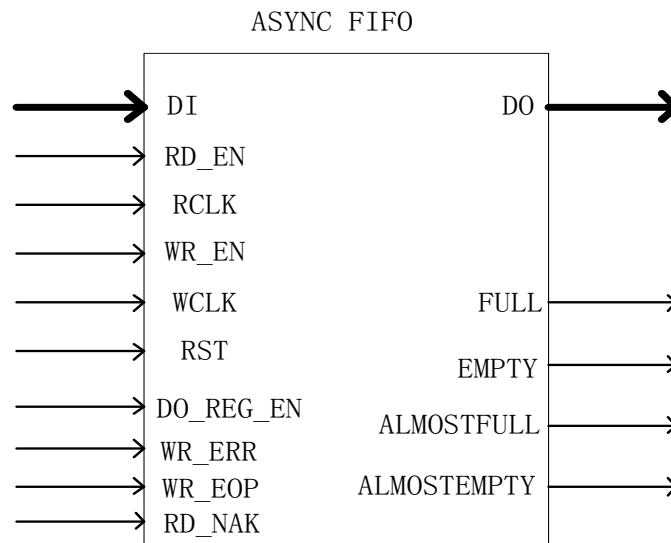


图 7-4 异步 FIFO 模式（包含丢包重发功能）

表 7-3 异步 FIFO 模式（包含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
RD_NAK	输入	读端口重发指示
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志

端口名	输入/输出	描述
ALMOSTEMPTY	输出	将空标志

7.2.4 同步 FIFO（不含丢包重发功能）

同步 FIFO 读写时钟为同一个时钟。

当写使能有效，且 FIFO 不为满时，在时钟的上升沿向 FIFO 写入数据。

当读使能有效，且 FIFO 不为空时，在时钟的上升沿从 FIFO 读出数据。

当 FIFO 写满时产生满信号，当 FIFO 读空时产生空信号。

此模式下数据流如下图所示：

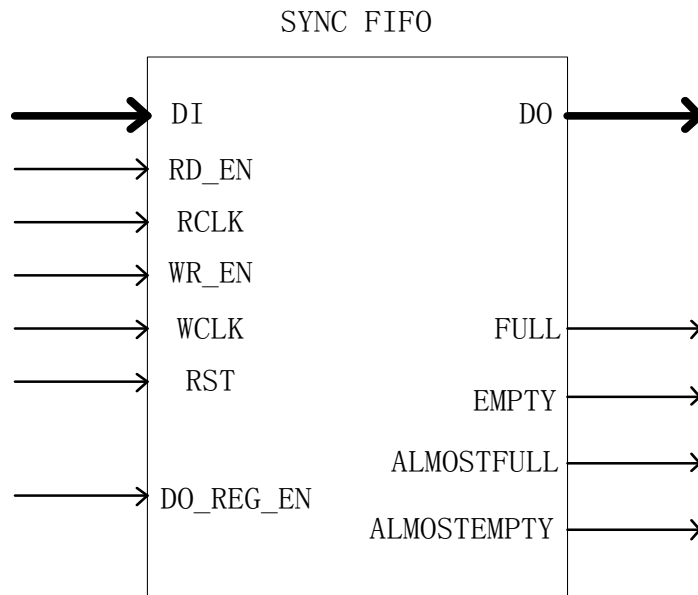


图 7-5 同步 FIFO 模式（不含丢包重发功能）

表 7-4 同步 FIFO 模式（不含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

7.2.5 同步 FIFO（支持丢包功能）

丢包功能：同步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。丢包功能可以单独使能，wr_eop 为输入信号。

此模式下数据流如下图所示：

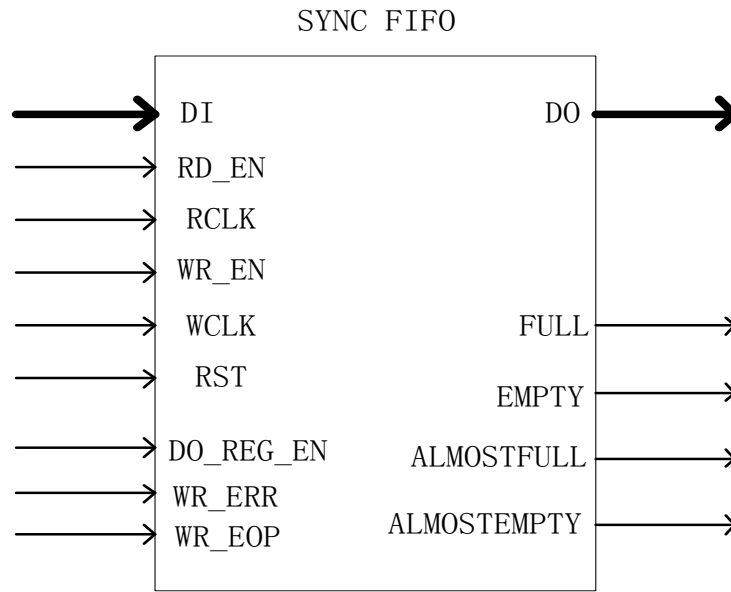


图 7-6 同步 FIFO 模式（支持丢包功能）

表 7-5 同步 FIFO 模式（支持丢包功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

7.2.6 同步 FIFO（包含丢包重发功能）

丢包功能：FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。

重发功能：当同步 FIFO 读时钟检测到 rd_nak 信号，读地址退回到数据包包头所在的地址。

丢包功能和重发功能需要同时使能。wr_eop 输入连接到数据输入 DI。

此模式下数据流如下图所示：

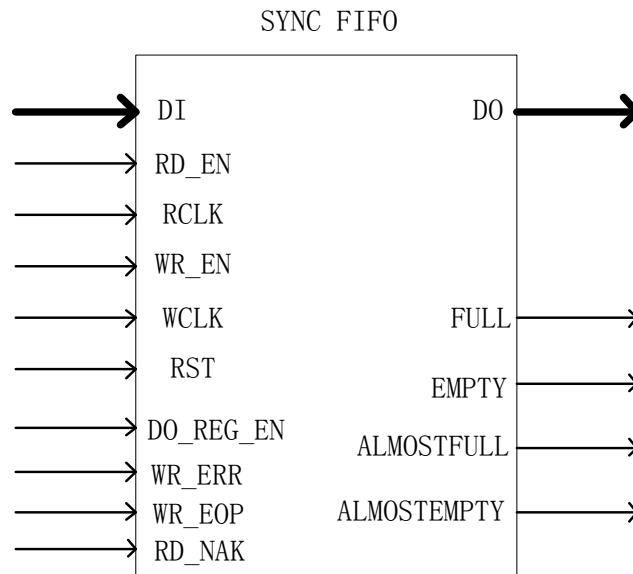


图 7-7 同步 FIFO（包含丢包重发功能）

表 7-6 同步 FIFO（包含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
RD_NAK	输入	读端口重发指示
DO[N:0]	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志

端口名	输入/输出	描述
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

7.3 位宽组合

FIFO 的端口位宽由 GTP 中的参数 DATA_WIDTH 决定，例如，当参数 DATA_WIDTH 的值为 4 时，端口数据位宽被设置为 4 bit。GTP_FIFO 不支持混合位宽。

如下表所示为 18K GTP_FIFO 模式允许的位宽。

表 7-7 18K GTP_FIFO 模式允许的位宽

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

注：√表示支持的位宽组合。

7.4 读写操作

7.4.1 写入空 FIFO 时序

EMPTY 信号指示 FIFO 为空，当 WR_EN 有效并且有效数据写入后，同步 FIFO 时，EMPTY 信号在 1 个 RCLK 时钟周期后置“0”；写入异步 FIFO 时，EMPTY 信号在 2 个 RCLK 时钟周期后置零。当 WR_EN 继续有效，ALMOST_EMPTY 信号根据 ALMOST_EMPTY_OFFSET 的配置延迟置零。

写入空 FIFO 时序图如下图所示：

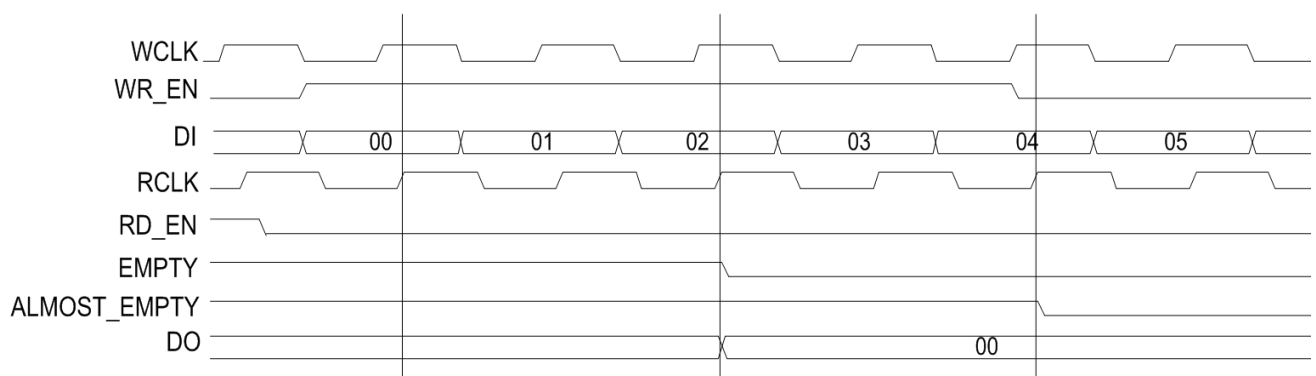


图 7-8 写入空 FIFO 时序图

7.4.2 写入将满 FIFO 时序

FIFO 将满时，根据 ALMOST_FULL_OFFSET 的配置，ALMOST_FULL 将提前置 1。

当 FIFO FULL 时，写指针不再增加。

写入将满 FIFO 时序图如下图所示：

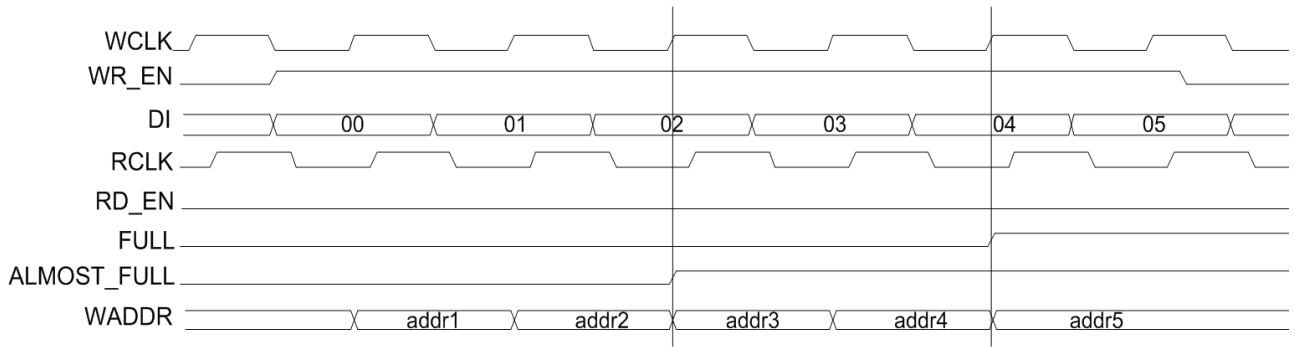


图 7-9 写入将满 FIFO 时序图

7.4.3 满 FIFO 读出时序

FULL 信号指示 FIFO 为满，当 RD_EN 有效读出数据后，FULL 信号在 1 到 2 个 WCLK 时钟周期内置零。当 RD_EN 继续有效，ALMOST_FULL 信号根据 ALMOST_FULL_OFFSET 的配置延迟置零。从满 FIFO 读出时序图如下图所示：

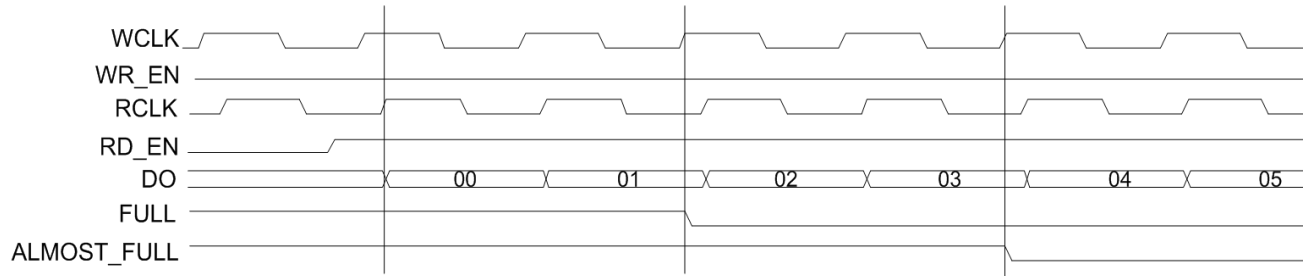


图 7-10 从满 FIFO 读出时序图

7.4.4 将空 FIFO 读出时序

FIFO 将空时，根据 ALMOST_EMPTY_OFFSET 的配置，ALMOST_EMPTY 提前置 1。

当 FIFO EMPTY 时，读指针不再增加。从将空 FIFO 读出时序图如下图所示：

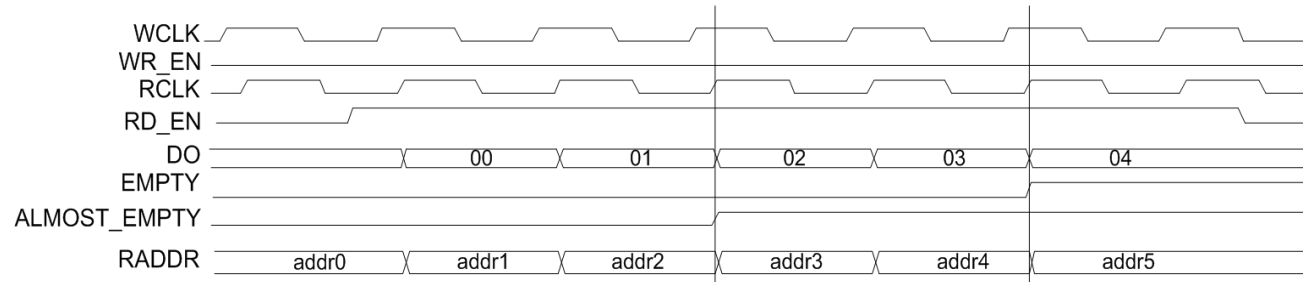


图 7-11 从将空 FIFO 读出时序

7.4.5 写丢包时序

当包尾指示信号 WR_EOP 置“1”并且 WR_ERR 数据包错误指示有效时，写指针在下一个 wclk 周期退回数据包包头所在位置。写丢包时序图如下图所示：

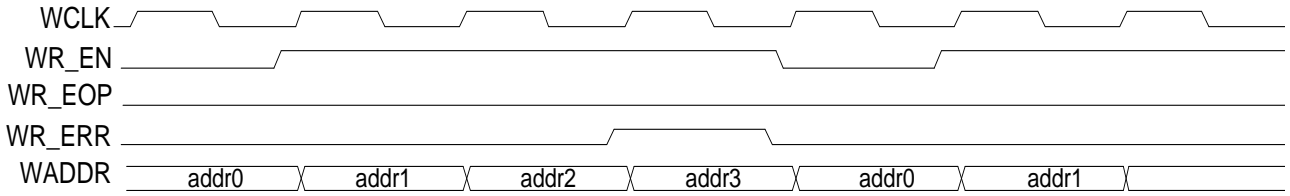


图 7-12 写丢包时序

7.4.6 读重发时序

当包尾指示信号 WR_EOP 置“1”时标记上一组数据包包尾地址，读指针在 RD_NAK 信号置“1”后退回到当前数据包包头所在地址。

写重发时序如下图所示：

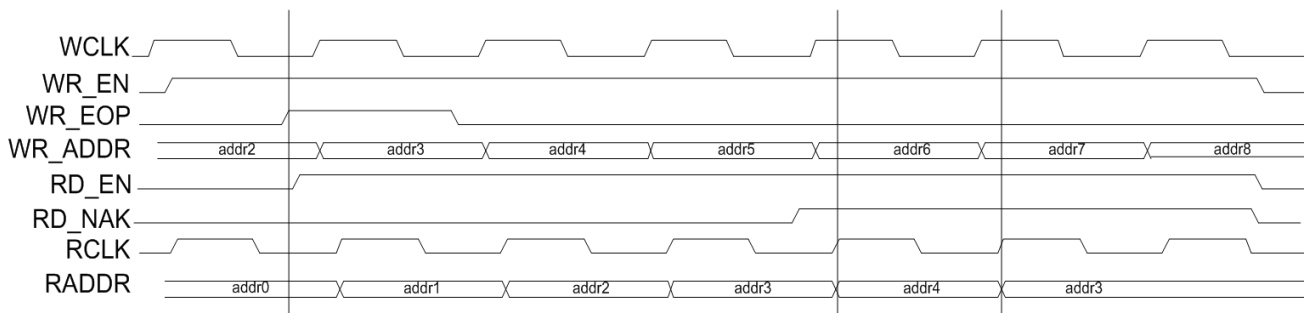


图 7-13 读重发时序

7.4.7 支持丢包功能下写入空 FIFO

EMPTY 信号指示 FIFO 为空，当 WR_EN 有效，并且第一个 WR_EOP 包尾信号有效 (WR_ERR 等于 0)，EMPTY 信号在 RCLK 时钟域内置“0”。

写丢包功能下写入空 FIFO 时序图如下图所示：



图 7-14 写丢包模式下写入空 FIFO 时序图

7.4.8 丢包重发功能下写入将满 FIFO

FULL 信号指示 FIFO 写满，当读端口读出 WR_EOP 所在读地址的数据且 RD_NAK 为“0”时，表示读出包数据有效。当读出数据未满一个包，并且写指针指向最后一个读出有效包尾地址时，会产生 FULL 信号。

丢包重发模式下写入将满 FIFO 时序图如下图所示：

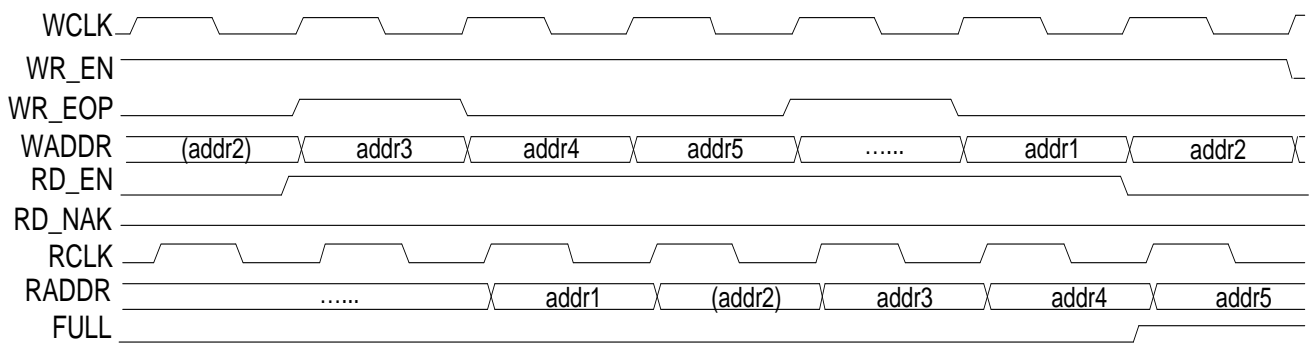


图 7-15 丢包重发模式下写入将满 FIFO 时序图

7.5 内部寄存器

FIFO 模式的内部寄存器和 DP 模式相同，详见[内部寄存器](#)。

7.6 复位信号

FIFO 模式的复位信号和 DP 模式相同，详见[复位信号](#)。

7.7 全局信号时序

FIFO 模式的全局信号时序和 DP 模式相同，详见[全局信号时序](#)。

7.8 应用示例

本小节举例说明了单个 18K GTP_FIFO 的配置步骤，用户还可以通过 Pango Design Suite 软件内嵌的 IP Compiler 工具直接生成基于 GTP_DRM 的 FIFO 的 IP，详见 IP Compiler 工具内的《Logos 系列 DRM RAM/FIFO IP 用户指南》。

示例为异步 FIFO 模式，端口数据位宽配置为 1Kx18，使能输出寄存器。

对单个 18K FIFO 按如下步骤进行配置：

a. 按下表所述对 FIFO 的参数进行配置：

表 7-8 FIFO 参数配置

参数名称	配置值	说明
DATA_WIDTH	18	将端口配置为 1Kx18 模式
DO_REG	1	使能端口输出寄存器
SYNC_FIFO	"FALSE"	异步 FIFO 模式
USE_EMPTY	1	空标志位使能
USE_FULL	1	满标志位使能
ALMOST_EMPTY_OFFSET	'd4	将空信号设置为 FIFO 内部储存数据个数小于等于 4 个时置 1
ALMOST_FULL_OFFSET	'd1020	将满信号设置为 FIFO 内部储存数据个数大于等于 1020 个时置 1

b. 按下表所示对 FIFO 的端口进行连接：

表 7-9 FIFO 端口连接

端口名称	连接信号	说明
DI[35:0]	di[17:0]	将输入数据信号 di[17:0] 连接到 DI[17:0]，高位 DI 端口悬空或接低电平
DO[35:0]	do[17:0]	将输入数据信号 do[17:0] 连接到 DO[17:0]，高位 DO 端口悬空

c. 读写端口其余信号：将读写端口的时钟、读写时钟使能、输出寄存器时钟使能、数据寄存器复位、标志信号分别连接到相应的 GTP 相应读写端口上；

d. 其余不使用的参数：其余不使用的参数不进行设置，使用默认值；

e. 其余不使用的端口：其余不使用的端口悬空（不使用也必须进行连接的端口已在上述步骤描述）。

配置后的 GTP 如下所示：

```
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
);
```

8 附录

8.1 地址和数据端口 Mapping

如表 8-1、表 8-2 所示分别为 18K 和 9K 模块存储器模式下，对应的地址和数据端口 Mapping。

表 8-1 18K DRM 模式下存储器对应的地址和数据端口 Mapping

组合模式	A 端口地址	B 端口地址	A 端口数据输入	B 端口数据输入	A 端口数据输出	B 端口数据输出
18K 的 DRM 存储器模式						
16Kx1	ADDRA[13:0]	ADDRB[13:0]	DIA[0]	DIB[0]	DOA[X],X=16~9,7~0 中任意一位	DOB[X],X=16~9,7~0 中任意一位
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]或者 DOA[16:9]	DOB[7:0]或者 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]或者 DOA[17:9]	DOB[8:0]或者 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]}

表 8-2 9K DRM 模式下存储器对应的地址和数据端口 Mapping

组合模式	A 端口地址	B 端口地址	A 端口数据输入	B 端口数据输入	A 端口数据输出	B 端口数据输出
DRM 端口模式(9K 模块 0)						
8Kx1	ADDRA[12:0]	ADDRB[12:0]	DIA[0]	DIB[0]	DOA[X],X=16~9,7~0 中任意一位	DOB[X],X=16~9,7~0 中任意一位
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]或者 DOA[16:9]	DOB[7:0]或者 DOB[16:9]
1Kx9	ADDRA[12:3]	ADDRB[12:3]	DIA[8:0]	DIB[8:0]	DOA[8:0]或者 DOA[17:9]	DOB[8:0]或者 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]}

如下表 8-3、表 8-4 所示分别为 x1, x2, x4, x8, x16, x32 数据位宽和 x9, x18, x36 数据位宽的数据地址 Mapping。

表 8-3 不同位宽数据地址 Mapping (x1, x2, x4, x8, x16, x32 数据宽度)

数据位宽	最低位端口地址 (与最大数据宽度模式最低位端口地址比较)																															
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

表 8-4 不同位宽数据地址 Mapping (x9, x18, x36 数据宽度)

数据位宽	最低位端口地址 (与最大数据宽度模式最低位端口地址比较)																																															
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 初始化配置参数映射

INIT_XX 是 DRM 的初始化配置参数，它们对 DRM 进行了初始化，决定了内存的初始值，默认状态下，DRM 的初始值全为 0。GTP_DRM18K 有包含从 INIT_00 到 INIT_3F 的 64 个初始化配置参数，GTP_DRM9K 有包含从 INIT_00 到 INIT_1F 的 32 个初始化配置参数，每个配置参数是 288 位数，配置了 DRM 对应地址的 288 bits 内存。

下面的公式和表格用于确认每个 INIT_XX 映射的内存 bit 的位置（式中 YY 为十六进制数 XX 转换后的十进制数）：

$$\text{最高位} = [(YY+1)*288]-1$$

$$\text{最低位} = (YY)*288$$

表 8-5 INIT_XX 参数映射

参数	最高位	最低位
INIT_00	287	0
INIT_01	575	288
INIT_02	863	576
.....
INIT_10	4895	4608
.....
INIT_1F	9215	8928
.....
INIT_3F	18431	18144

数据位宽为 2^N 即 1/2/4/8/16/32bit 时，上表中 INIT_XX 每 9 位数据只映射低 8 位到内存；数据位宽为 9×2^N 即 9/18/36bit 时，INIT_XX 数据全部映射到内存。例如，当 DRM 配置为 8x2K 时，INIT_00[7:0]对应地址 addr 为 0 时读出数据的初始值，INIT_00[16:9]对应地址 addr 为 1 时读出数据的初始值；当 DRM 配置为 9x2K 时，INIT_00[8:0]对应地址 addr 为 0 时读出数据的初始值，INIT_00[17:9]对应地址 addr 为 1 时读出数据的初始值。

8.3 DRM 端口信号说明

1.地址总线 (ADDRA[13:0]、ADDRB[13:0]): A/B 端口读写地址, 地址总线的有效位数与读写数据位宽有关, 例如 DRM18K 配置为 18 位读写数据位宽, 则地址位宽为 10 位, 有效地址输入为 ADDRA[13:4]、ADDRB[13:4], 剩下的 ADDRA[3:0]和 ADDRb[3:0]则需要接高电平或字节使能信号 (字节使能信号复用低位地址, 使用字节使能功能时低位地址需要接字节使能信号)。详细地址映射关系可参考附录[地址和数据端口 Mapping](#)。

2.地址保持信号 (ADDRA_HOLD、ADDRB_HOLD): 该端口置高电平时对应端口地址输入保持之前的值不变, 不随输入地址的改变而改变; 该端口置低电平时对应端口地址输入才为地址总线端口输入的地址。在不使用地址保持功能时需要将该端口接至低电平。

3.数据总线 (DIA[17:0]、DIB[17:0]、DOA[17:0]、DOB[17:0]): A、B 端口的读写数据端口, 详细数据端口映射关系可参考附录[地址和数据端口 Mapping](#)。

4.地址扩展信号 (CSA[2:0]、CSB[2:0]): 多个 DRM 可以通过级联扩展的方式组合成更大的 DP RAM, SDP RAM, SPRAM, ROM 或者 FIFO。对此, DRM 的 A、B 端口各提供额外的 3 bit 地址扩展, 常用于深度扩展的应用。

5.写使能信号 (WEA、WEB): DRM 的写操作控制信号, 具体写操作时序可参考[读写操作](#)。

6.时钟相关信号 (CLKA、CLKB、CEA、CEB): DRM 的 A、B 端口时钟及其使能信号, SDP 模式下, CLKA、CEA 为写时钟及其使能信号, CLKB、CEB 为读时钟及其使能信号。

7.输出寄存器相关信号 (ORCEA、ORCEB、RSTA、RSTB): A、B 端口输出寄存器使能信号及复位信号。

8.4 字节附加信息位

在数据宽度为x9、x18、x36模式下, 对应每字节有一位附加信息存储位, 如下表所示:

表 8-6 字节附加信息位列表

端口	模式	数据端口		数据输入	
		字节	附加信息位	字节	附加信息位
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 位宽模式

FIFO在丢包重发模式下, wr_eop信号需要占用一位数据位, FIFO数据位宽需要进行如下配置:

表 8-7 FIFO 丢包重发模式地址深度与数据宽度配置列表

FIFO 地址与位宽配置	FIFO 可用数据位	wr_eop 占用数据位
2Kx9	DI[7:0]	DI[8]
1Kx18	DI[17:9],DI[7:0]	DI[8]
512x36	DI[35:27],DI[25:0]	DI[26]

含丢包重发模式的DRM_FIFO支持如下表所示的数据位宽:

表 8-8 含丢包重发模式的异步/同步 FIFO 模式数据位宽列表

		写端口		
		2Kx9	1Kx18	512x36
读端口	2Kx9	√		
	1Kx18		√	
	512x36			√

不含丢包重发模式的DRM_FIFO支持如下表所示的数据位宽:

表 8-9 不含丢包重发模式的异步/同步 FIFO 模式数据位宽列表

		写端口					
		16Kx1	8Kx2	4Kx4	2Kx9 (2Kx8)	1Kx18 (1Kx16)	512x36 (512x32)
读端口	16Kx1	√					
	8Kx2		√				
	4Kx4			√			
	2Kx9(2Kx8)				√		
	1Kx18(1Kx16)					√	
	512x36(512x32)						√

8.6 GTP_DRM9K 例化模板

```
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 例化模板

```

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 例化模板

```
GTP_FIFO18K #(
    .GRS_EN("TRUE"),
    .DATA_WIDTH(18),
    .DO_REG(0),
    .ALMOST_FULL_OFFSET('b000000000000000),
    .ALMOST_EMPTY_OFFSET('b000000000000000),
    .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
);
```

免责声明

版权声明

本文档版权归深圳市紫光同创电子有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

免责声明

1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。

2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。