



Gowin 先进先出队列(FIFO) 用户指南

UG105-1.06,2020-06-15

版权所有© 2020 广东高云半导体科技股份有限公司

未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

版本信息

日期	版本	说明
2016/08/12	1.00	初始版本。
2016/10/27	1.01	适用 GW2AR 系列 FPGA 产品。
2017/05/25	1.02	更新 FIFO/FIFO_SC 相关配置信息。
2018/04/20	1.03	<ul style="list-style-type: none">● 更新 GW2AR-55K 和 GW2AR-18K IP 性能和资源使用；● 新增 FWFT 功能介绍。
2018/07/17	1.04	<ul style="list-style-type: none">● 更新 GW2AR-55K 和 GW2AR-18K IP 性能和资源使用；● 新增 GW1N-4 性能和资源使用；● 使用新的资源统计方式；
2019/03/28	1.05	更新部分时序图。
2020/06/15	1.06	<ul style="list-style-type: none">● 更新 IP 配置图；● 更新表 5-1 注描述。

目录

目录	i
图目录.....	iii
表目录.....	iv
1 关于本手册	1
1.1 手册内容	1
1.2 相关文档.....	1
1.3 术语、缩略语	1
1.4 技术支持与反馈.....	2
2 FIFO 概述.....	3
2.1 FIFO 简介	3
2.2 FIFO/FIFO_SC IP 简介	4
2.3 FIFO/FIFO_SC IP 功能及特征.....	5
2.3.1 FIFO IP 功能及特征	5
2.3.2 FIFO_SC IP 功能及特征.....	6
2.4 FIFO 最大频率和资源利用率	7
2.5 FIFO_SC 最大频率和资源利用率.....	11
3 端口描述	15
3.1 FIFO 端口	15
3.2 FIFO_SC 端口	16
4 时序说明	17
4.1 FIFO 信号时序	17
4.2 FIFO_SC 信号时序	19
5 FIFO/FIFO_SC IP 配置.....	22
5.1 FIFO IP 配置.....	22
5.2 FIFO_SC IP 配置.....	25

6 参考设计	30
6.1 FIFO 参考设计	30
6.2 FIFO_SC 参考设计	31
6.3 参考设计应用	31

图目录

图 2-1 异步 FIFO 结构示意图	4
图 2-2 FIFO_SC 结构示意图	4
图 4-1 写操作示例	17
图 4-2 读操作示例	18
图 4-3 FIFO 满标志和将满标志时序图	18
图 4-4 FIFO 空标志和将空标志时序图	19
图 4-5 写操作示例	19
图 4-6 读操作示例	20
图 4-7 FIFO 满标志和将满标志时序图	20
图 4-8 FIFO 空标志和将空标志时序图	21
图 5-1 FIFO IP 配置界面	22
图 5-2 FIFO_SC IP 配置页面	26
图 6-1 FIFO/FIFO_SC 参考设计实例结构	30

表目录

表 1-1 术语、缩略语	1
表 2-1 FIFO/ FIFO_SC IP 概览.....	5
表 2-2 未配置可选特性的异步 FIFO.....	7
表 2-3 已配置可选特性的异步 FIFO.....	9
表 2-4 未配置可选特性的 FIFO_SC.....	12
表 2-5 已配置可选特性的 FIFO_SC.....	13
表 3-1 FIFO 的 IO 端口列表.....	15
表 3-2 FIFO_SC 的 IO 端口列表.....	16
表 5-1 FIFO IP 配置界面参数	23
表 5-2 FIFO IP 配置界面参数	26
表 6-1 FIFO 参考设计端口定义.....	31
表 6-2 FIFO_SC 参考设计端口定义.....	31

1 关于本手册

1.1 手册内容

本手册主要描述 FIFO 的功能特性、端口、时序、相关配置及参考设计，旨在帮助用户快速了解 Gowin FIFO IP 的产品特点及使用方法。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可以下载、查看以下相关文档：

1. [DS100](#)，GW1N 系列产品数据手册
2. [DS117](#)，GW1NR 系列产品数据手册
3. [DS102](#)，GW2A 系列产品数据手册
4. [DS226](#)，GW2AR 系列产品数据手册

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
FIFO	First Input First Output	先进先出队列
IP	Intellectual Property	知识产权
RAM	Random Access Memory	随机存取存储器
LUT	Look-up Table	查找表
GSR	Global System Reset	全局系统复位
ECC	Error Correcting Code	错误检查和纠正

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 FIFO 概述

2.1 FIFO 简介

FIFO 即 First In First Out 的缩写，是一种先进先出列队，外围控制逻辑负责对 FIFO 进行读写。此外，FIFO 还提供许多握手信号用于和外围控制逻辑交互。

当 FIFO 不满并且写使能信号有效时，数据在时钟上升沿被写入 FIFO 存储队列。满标志表示 FIFO 已满且将不再执行更多的写操作。

当 FIFO 不空并且读使能信号有效时，数据在时钟上升沿时从 FIFO 存储队列读出。空标志表示 FIFO 已空且将不再执行更多的读操作。

不合法的请求不会影响 FIFO 本身。如在 FIFO 为空时发起读请求或者在 FIFO 为满时发起写请求，该类操作不会影响 FIFO 中的数据，但相应操作会被忽略且上溢和下溢标志会置位。用户通过监控这两个信号来判断是否非法操作。同时，水位信号直接指示 FIFO 中有多少有效数据，用户也可设置不同阈值来自定义空满标志。

FIFO 分为同步 FIFO (FIFO_SC) 和异步 FIFO(FIFO)，其结构示意图分别如图 2-1、图 2-2 所示。

图 2-1 异步 FIFO 结构示意图

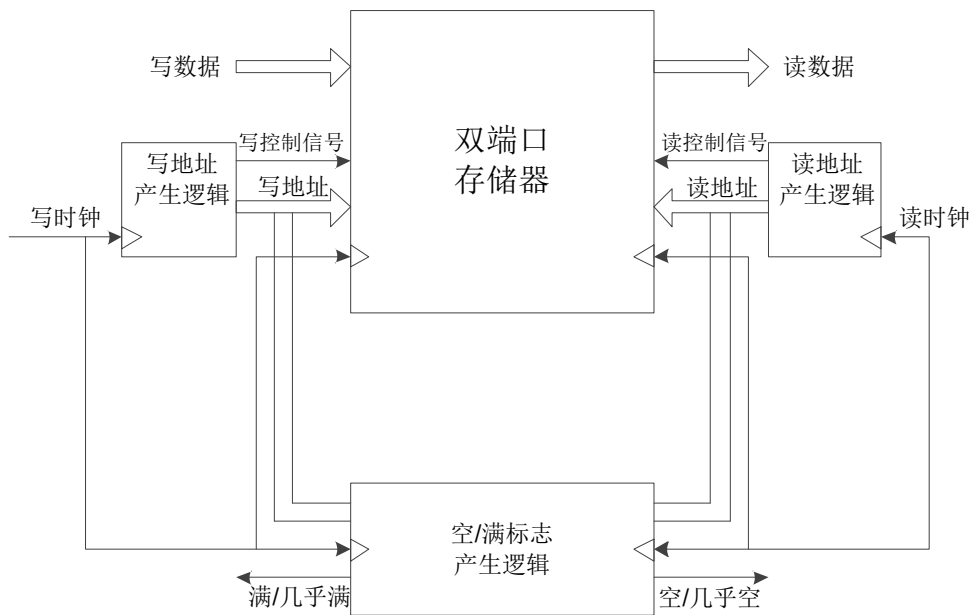
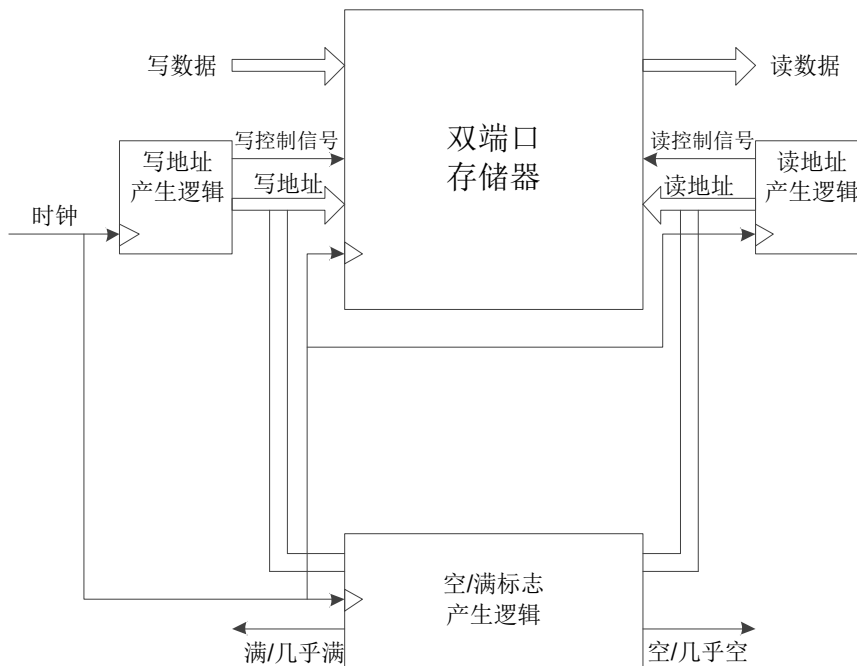


图 2-2 FIFO_SC 结构示意图



2.2 FIFO/FIFO_SC IP 简介

Gowin FIFO IP 分为同步 FIFO (FIFO_SC) IP 和异步 FIFO (FIFO) IP。

- 同步 FIFO 的读端口和写端口信号均由一个时钟域控制；
- 异步 FIFO 的读写端口信号由两个独立的时钟域控制。

Gowin IP Core Generator 编译器用于生成单周期读写，端口可配置的 FIFO IP。端口配置是指可根据需要生成不同数据位宽和数据深度的 FIFO IP。

表 2-1 FIFO/ FIFO_SC IP 概览

FIFO 与 FIFO_SC IP	
IP 核应用	
芯片支持	<ul style="list-style-type: none"> ● GW1N 系列 ● GW1NR 系列 ● GW1NS 系列 ● GW1NSR 系列 ● GW1NZ 系列 ● GW2A 系列 ● GW2AR 系列
逻辑资源	不同配置，不同数据位宽和数据深度下资源不同。
交付文件	-
设计文件	Verilog (encrypted)
相关文档	<ul style="list-style-type: none"> ● GW1N 系列 FPGA 产品数据手册 ● GW1NR 系列 FPGA 产品数据手册 ● GW1NS 系列 FPGA 产品数据手册 ● GW1NSR 系列 FPGA 产品数据手册 ● GW1NZ 系列 FPGA 产品数据手册 ● GW2A 系列 FPGA 产品数据手册 ● GW2AR 系列 FPGA 产品数据手册
测试设计流程	-
综合软件	Synplify_Pro
应用软件	Gowin 云源软件

2.3 FIFO/FIFO_SC IP 功能及特征

2.3.1 FIFO IP 功能及特征

Gowin 异步 FIFO IP 可完成异步时钟域不同位宽的数据传递和缓存功能，根据用户的不同需求配置不同的输出控制信号和数据结构。

主要特性如下：

- 异步 FIFO 内部存储结构的实现类型可配置，包括：Block RAM、Distributed RAM 和 LUT；
- 写数据深度可配置，深度值是 2^n ，最大为 65536；
- 写数据位宽可配置，大小为 1-256 bit；
- 读数据的深度可配置，深度值为 2^n ，最大为 65536；

- 读数据位宽=写数据深度 x 写数据位宽/读数据深度，不可配置。

注！

上述公式须能够整除，这也限制了读数据深度的最大值；

- 读写数据数目的输出可配置，可以选择输出读写数据的数目，也可以选择不输出；
- 复位功能可配置，可以选择不使用复位（由 GSR 完成复位），可以选择一个复位，也可以选择读写分别复位；
- 标志信号输出可选，可以选择输出半空、半满标志，也可以选择不输出半空、半满标志；
- 若选择输出半空、半满标志，则半空、半满的阈值可以进行配置，可以配置为：静态的单个常量阈值、静态的双常量阈值、动态的单个输入阈值、动态的双输入阈值；
- ECC 校验功能可选。当异步 FIFO 内部的存储结构由 Block RAM 实现，且读写数据位宽相等并且小于等于 64bit 时，ECC 校验才可选；
- 输出寄存器功能可配置。若选择输出寄存器功能，则读使能（RdEn）控制可选。若选择读使能控制，则输出寄存器受 RdEn 控制，最后一个数据不能输出。若选择输出寄存器功能，不选择读使能控制，则读数据输出晚一个周期，最后一个数据会输出。
- 支持 First-Word Fall-Through 功能。

2.3.2 FIFO_SC IP 功能及特征

GOWIN 同步 FIFO_SC IP 可完成相同时钟域下，相同位宽和深度数据的传递和缓存，根据用户的不同需求，可配置输出不同控制信号。

主要特性如下：

- FIFO_SC 内部存储结构的实现类型可配置，包括：Block RAM、Distributed RAM 和 LUT；
- 写数据深度可配置，深度值是 2^n ，最大为 65536；
- 写数据位宽可配置，大小为 1-256 bit；
- 读数据位宽和深度与写数据位宽和深度相等；
- 写数据数目的输出可配置，可以选择输出写数据的数目，也可以选择不输出；
- 标志信号输出可选，可以选择输出半空、半满标志，也可以选择不输出半空、半满标志；
- 若选择输出半空、半满标志，则半空、半满的阈值可以进行配置，可以配置为：静态的单个常量阈值、静态的双常量阈值、动态的单个输入阈值、动态的双输入阈值；
- ECC 校验功能可选。当 FIFO_SC 内部的存储结构由 Block RAM 实现，

且读写数据位宽相等并且小于等于 64bit 时，ECC 校验才可选；

- 输出寄存器功能可配置。若选择输出寄存器功能，则读使能（RdEn）控制可选。若选择读使能控制，则输出寄存器受 RdEn 控制，最后一个数据不能输出。若选择输出寄存器功能，不选择读使能控制，则读数据输出晚一个周期，最后一个数据会输出。
- 支持 First-Word Fall-Through 功能。

2.4 FIFO 最大频率和资源利用率

Gowin 异步 FIFO 由 Verilog 语言实现。FIFO 的最大频率主要依赖于配置的数据深度位宽和配置功能与所使用的器件速度等级。使用不同的器件、密度、速度和等级时，其性能和资源利用情况可能不同。

以 GW2A-55 系列 FPGA 为例，FIFO 资源利用和性能情况如表 2-2 与表 2-3 所示。

表 2-2 未配置可选特性的异步 FIFO

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况			
				CFU Logics	FFs	Block Ram	Distributed Ram
Block Ram	4096 x 16	GW2A-55-484	Wr_clk=120.396; Rd_clk=109.567	251	124	4	0
	512 x 16		Wr_clk=133.227; Rd_clk=116.194	181	96	1	0
	64 x 16		Wr_clk=148.443; Rd_clk=165.963	138	68	1	0
Distributed Ram	2048 x 16	GW2A-55-484	Wr_clk=98.085; Rd_clk=94.967	5098	126	0	512
	512 x 16		Wr_clk=119.934; Rd_clk=80.903	1361	108	0	128
	64 x 16		Wr_clk=146.482;	302	88	0	16

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况			
				CFU Logics	FFs	Block Ram	Distributed Ram
			Rd_clk=176.987				
LUT	64 x 16	GW2A-55-484	Wr_clk=144.51; Rd_clk=118.609	770	1105	0	0
Block Ram	4096 x 16	GW2A-18-484	Wr_clk=120.907; Rd_clk=104.532	251	124	4	0
	512 x 16		Wr_clk=140.547; Rd_clk=116.818	181	96	1	0
	64 x 16		Wr_clk=165.234; Rd_clk=168.747	138	68	1	0
Distributed Ram	1024x 16	GW2A-18-484	Wr_clk=119.613; Rd_clk=97.107	2588	117	0	256
	512 x 16		Wr_clk=123.976; Rd_clk=92.429	1361	108	0	128
	64 x 16		Wr_clk=155.54; Rd_clk=175.921	302	88	0	16
LUT	64 x 16	GW2A-18-484	Wr_clk=139.892; Rd_clk=117.022	770	1105	0	0
Block Ram	4096 x 16	GW1N-4-144 GW1N-4-144 GW1N-4-144	Wr_clk=71.261; Rd_clk=64.502	229	127	4	0

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况			
				CFU Logics	FFs	Block Ram	Distributed Ram
	512 x 16		Wr_clk=71.247; Rd_clk=66.017	176	94	1	0
	64 x 16		Wr_clk=81.447; Rd_clk=76.353	144	70	1	0
Distributed Ram	4096 x 16	GW1N-4-144 GW1N-4-144 GW1N-4-144	不支持该类型				
	512 x 16						
	64 x 16						
LUT	64 x 16	GW1N-4-144	Wr_clk=84.709; Rd_clk=57.022	763	1105	0	0

注!

- “-”表示暂不支持，等待后续更新；
- FIFO 在其他高云 FPGA 上的应用验证会陆续推出。

表 2-3 已配置可选特性的异步 FIFO

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况				配置项		
				CFU Logics	FFs	Block Ram	Distributed Ram	Data Num.	Control Flag	ECC
Block Ram	4096 x 16	GW2A-55-48	Wr_clk=102.33; Rd_clk=104.918	376	157	6	0	√	√	√
	512 x 16	4	Wr_clk=140.366; Rd_clk=107.101	302	149	1	0	√	√	√

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况				配置项		
				CFU Logics	FFs	Block Ram	Distributed Ram	Data Num.	Control Flag	ECC
	64 x 16		Wr_clk=151.163; Rd_clk=148.65	242	112	1	0	√	√	√
Distributed Ram	2048 x 16	GW2A-55-484	Wr_clk=95.643; Rd_clk=88.557	5106	170	0	512	√	√	×
	512 x 16		Wr_clk=117.17; Rd_clk=84.097	1373	148	0	128	√	√	×
	64 x 16		Wr_clk=141.822; Rd_clk=143.365	312	116	0	16	√	√	×
LUT	64 x 16	GW2A-55-484	Wr_clk=151.829; Rd_clk=125.791	731	1139	0	0	√	√	×
Block Ram	4096 x 16	GW2A-18-484	Wr_clk=99.288; Rd_clk=114.648	376	157	6	0	√	√	√
	512 x 16		Wr_clk=122.775; Rd_clk=110.636	302	149	1	0	√	√	√
	64 x 16		Wr_clk=148.711; Rd_clk=173.416	242	112	1	0	√	√	√
Distributed Ram	1024 x 16	GW2A-18-484	Wr_clk=128.221; Rd_clk=76.014	2600	159	0	256	√	√	×
	512 x 16		Wr_clk=119.455; Rd_clk=87.461	1373	148	0	128	√	√	×

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况				配置项		
				CFU Logics	FFs	Block Ram	Distributed Ram	Data Num.	Control Flag	ECC
	64 x 16		Wr_clk=146.078; Rd_clk=156.31	312	116	0	16	√	√	×
LUT	64 x 16	GW2A-18-484	Wr_clk=142.922; Rd_clk=106.383	731	1139	0	0	√	√	×
Block Ram	4096 x 16	GW1N-4-144 GW1N-4-144	Wr_clk=65.951; Rd_clk=49.374	385	157	6	0	√	√	√
	512 x 16	GW1N-4-144	Wr_clk=77.302; Rd_clk=64.869	286	129	1	0	√	√	√
	64 x 16		Wr_clk=75.821; Rd_clk=75.619	266	90	1	0	√	√	√
Distributed Ram	4096 x 16	GW1N-4-144	不支持该类型							
	512 x 16	GW1N-4-144								
	64 x 16	GW1N-4-144								
LUT	64 x 16	GW1N-4-144	Wr_clk=84.263; Rd_clk=80.362	717	1139	0	0	√	√	×

注!

- “-”表示暂不支持，等待后续更新；
- FIFO 在其他高云 FPGA 上的应用验证会陆续推出。

2.5 FIFO_SC 最大频率和资源利用率

Gowin 同步 FIFO 由 Verilog 语言实现。FIFO_SC 的最大频率主要依赖于配置的数据深度位宽和配置功能与所使用的器件速度等级。使用不同的器

件、密度、速度和等级时，其性能和资源利用情况可能不同。

以高云的 GW2A-55 系列 FPGA 为例，FIFO_SC 资源利用和性能情况如表 2-4 与表 2-5 所示。

表 2-4 未配置可选特性的 FIFO_SC

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况			
				CFU Logics	FFs	Block Ram	Distributed Ram
Block Ram	4096 x 16	GW2A-55-484	131.757	91	42	4	0
	512 x 16		131.803	58	33	1	0
	64 x 16		151.448	65	29	1	0
Distributed Ram	4096 x 16	GW2A-55-484	82.58	4943	56	0	512
	512 x 16		77.827	1260	50	0	128
	64 x 16		128.136	230	48	0	16
LUT	64 x 16	GW2A-55-484	100.266	639	1064	0	0
Block Ram	4096 x 16	GW2A-18-484	123.377	91	42	4	0
	512 x 16		135.391	58	33	1	0
	64 x 16		147.623	65	29	1	0
Distributed Ram	1024 x 16	GW2A-18-484	91.559	2473	53	0	256
	512 x 16		83.219	1260	50	0	128
	64 x 16		135.602	230	48	0	16
LUT	64 x 16	GW2A-18-484	99.671	639	1064	0	0
Block Ram	4096 x 16	GW1N-4-144	77.268	91	42	4	0
	512 x 16	GW1N-4-144	75.283	57	33	1	0
	64 x 16	GW1N-4-144	70.792	73	30	1	0
Distributed Ram	4096 x 16	GW1N-4-144	不支持该类型				
	512 x 16	GW1N-4-144					
	64 x 16	GW1N-4-144					
LUT	64 x 16	GW1N-4-144	56.535	640	1065	0	0

注!

- “-”表示暂不支持，等待后续更新；
- FIFO_SC 在其他高云 FPGA 上的应用验证会陆续推出。

表 2-5 已配置可选特性的 FIFO_SC

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况				配置项		
				CFU Logics	FFs	Block Ram	Distributed Ram	Data Num.	Control Flag	ECC
Block Ram	4096 x 16	GW2A-55-48	121.158	187	77	6	0	√	√	√
	512 x 16	4	133.801	154	67	1	0	√	√	√
	64 x 16		147.675	168	60	1	0	√	√	√
Distributed Ram	2048 x 16	GW2A-55-48	82.237	4943	84	0	512	√	√	×
	512 x 16	4	91.368	1260	76	0	128	√	√	×
	64 x 16		125.148	230	71	0	16	√	√	×
LUT	64 x 16	GW2A-55-48 4	101.459	639	108 7	0	0	√	√	×
Block Ram	4096 x 16	GW2A-18-484	123.42 132.574 150.944	187	77	6	0	√	√	√
	512 x 16			154	67	1	0	√	√	√
	64 x 16			168	60	1	0	√	√	√
Distributed Ram	1024 x 16	GW2A-18-484	96.128 80.138	2473	80	0	256	√	√	×
	512 x 16		137.403	1260	76	0	128	√	√	×
	64 x 16			230	71	0	16	√	√	×
LUT	64 x 16	GW2A-18-484	106.956	639	108 7	0	0	√	√	×
Block Ram	4096 x 16	GW1N-4-144	71.363	201	69	6	0	√	√	√
	512 x 16		65.051	168	67	1	0	√	√	√
	64 x 16		81.128	185	61	1	0	√	√	√

FIFO Type	Depth x Width	FPGA Family	Performance (MHz)	资源利用情况				配置项		
				CFU Logics	FFs	Block Ram	Distributed Ram	Data Num.	Control Flag	ECC
Distributed Ram	4096 x 16	GW1N-4-144	不支持该类型							
	512 x 16									
	64 x 16									
LUT	64 x 16	GW1N-4-144	67.522	640	1088	0	0	√	√	×

注！

- “-” 表示暂不支持，等待后续更新；
- FIFO_SC 在其他高云 FPGA 上的应用验证会陆续推出。

3 端口描述

3.1 FIFO 端口

FIFO 的 IO 端口描述如表 3-1 所示。

表 3-1 FIFO 的 IO 端口列表

信号名	位宽	方向	初始状态	是否可选	描述
Data	[WDSIZE-1:0]	输入	-	否	写入数据
WrClk	1	输入	-	否	写时钟
RdClk	1	输入	-	否	读时钟
WrEn	1	输入	-	否	写使能
RdEn	1	输入	-	否	读使能
Reset	1	输入	-	是	复位，高电平有效
WrReset	1	输入	-	是	写复位，高电平有效
RdReset	1	输入	-	是	读复位，高电平有效
AlmostEmptySetTh	[RASIZE-1:0]	输入	-	是	半空标志置 1 阈值
AlmostEmptyClrTh	[RASIZE-1:0]	输入	-	是	半空标志清 0 阈值
AlmostEmptyTh	[RASIZE-1:0]	输入	-	是	半空标志置 1 阈值
AlmostFullSetTh	[ASIZE-1:0]	输入	-	是	半满标志置 1 阈值
AlmostFullClrTh	[ASIZE-1:0]	输入	-	是	半满标志清 0 阈值
AlmostFullTh	[ASIZE-1:0]	输入	-	是	半满标志置 1 阈值
Q	[RDSIZE-1:0]	输出	-	否	读出数据
Empty	1	输出	1	否	空标志
Full	1	输出	0	否	满标志
Wnum	[ASIZE: 0]	输出	0	是	写入数据数目
Rnum	[RASIZE: 0]	输出	0	是	可读数据数目

信号名	位宽	方向	初始状态	是否可选	描述
Almost_Empty	1	输出	1	是	半空标志
Almost_Full	1	输出	0	是	半满标志
ERROR	2	输出	0	是	ECC 校验输出

3.2 FIFO_SC 端口

FIFO_SC 的 IO 端口描述如表 3-2 所示。

表 3-2 FIFO_SC 的 IO 端口列表

信号名	位宽	方向	初始状态	是否可选	描述
Data	[DSIZE-1:0]	输入	-	否	写入数据
Clk	1	输入	-	否	写时钟
WrEn	1	输入	-	否	写使能
RdEn	1	输入	-	否	读使能
Reset	1	输入	-	是	复位，高电平有效
AlmostEmptySetTh	[ASIZE-1:0]	输入	-	是	半空标志置 1 阈值
AlmostEmptyClrTh	[ASIZE-1:0]	输入	-	是	半空标志清 0 阈值
AlmostEmptyTh	[ASIZE-1:0]	输入	-	是	半空标志置 1 阈值
AlmostFullSetTh	[ASIZE-1:0]	输入	-	是	半满标志置 1 阈值
AlmostFullClrTh	[ASIZE-1:0]	输入	-	是	半满标志清 0 阈值
AlmostFullTh	[ASIZE-1:0]	输入	-	是	半满标志置 1 阈值
Q	[DSIZE-1:0]	输出	-	否	读出数据
Empty	1	输出	1	否	空标志
Full	1	输出	0	否	满标志
Wnum	[ASIZE: 0]	输出	0	是	写数据数目
Almost_Empty	1	输出	1	是	半空标志
Almost_Full	1	输出	0	是	半满标志
ERROR	2	输出	0	是	ECC 校验输出

4 时序说明

本章主要介绍 FIFO 和 FIFO_SC 在读写操作时信号的时序情况。

在实际应用中，用户可能会根据自己需求配置不同的输出控制信号，读写控制时机也会各不相同，本章以最常用且最重要的输入、输出、空满、半空及半满信号为例。

4.1 FIFO 信号时序

图 4-1 为写操作示例，以 16 数据深度为例，FIFO 不满时，写使能拉高，将数据写入至 Data；FIFO 为满状态时，写使能将被屏蔽，上溢信号拉高。

图 4-1 写操作示例

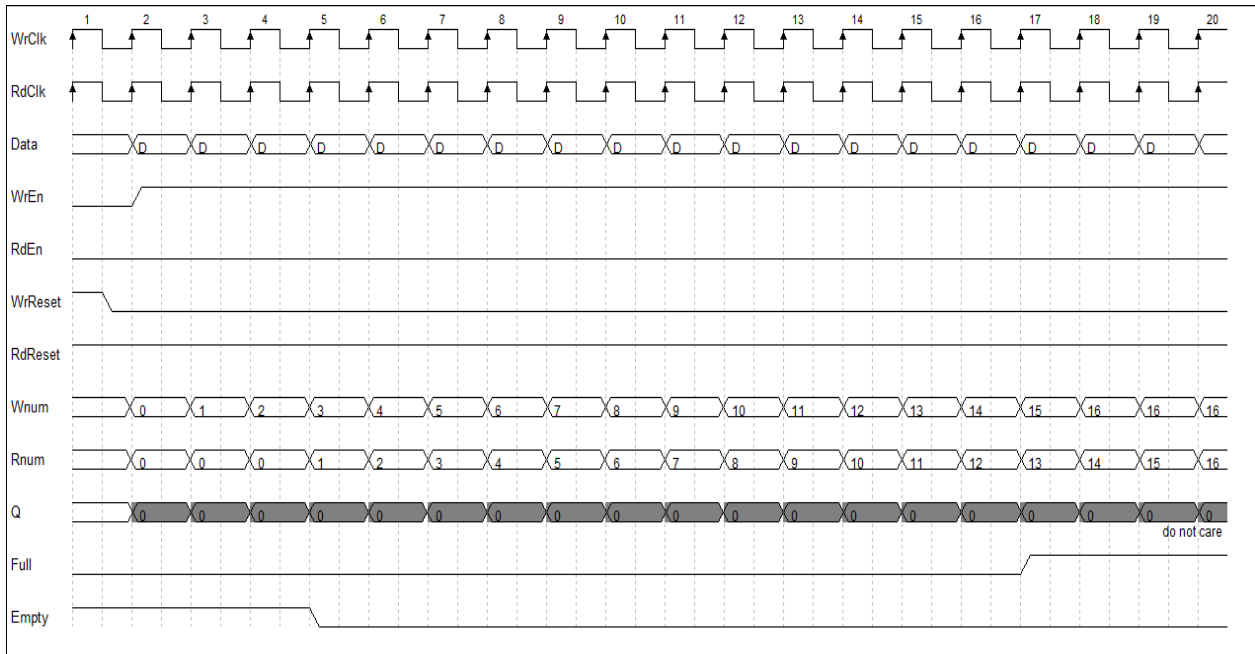


图 4-2 为读操作示例，以 16 位数据深度为例，FIFO 不为空时，读使能拉高，将数据读出至 Q。FIFO 为空时，读使能将被屏蔽，空信号拉高。

图 4-2 读操作示例

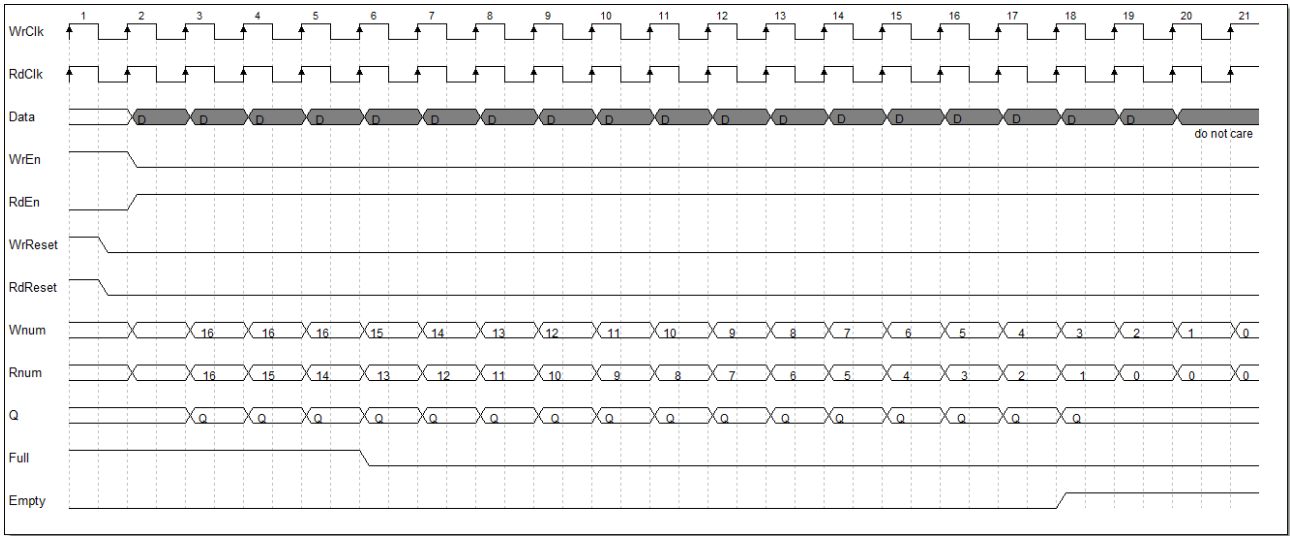


图 4-3 为 FIFO 满标志和将满标志时序图示例。在 FIFO 被写入数据后，Almost_full 信号拉高。

图 4-3 FIFO 满标志和将满标志时序图

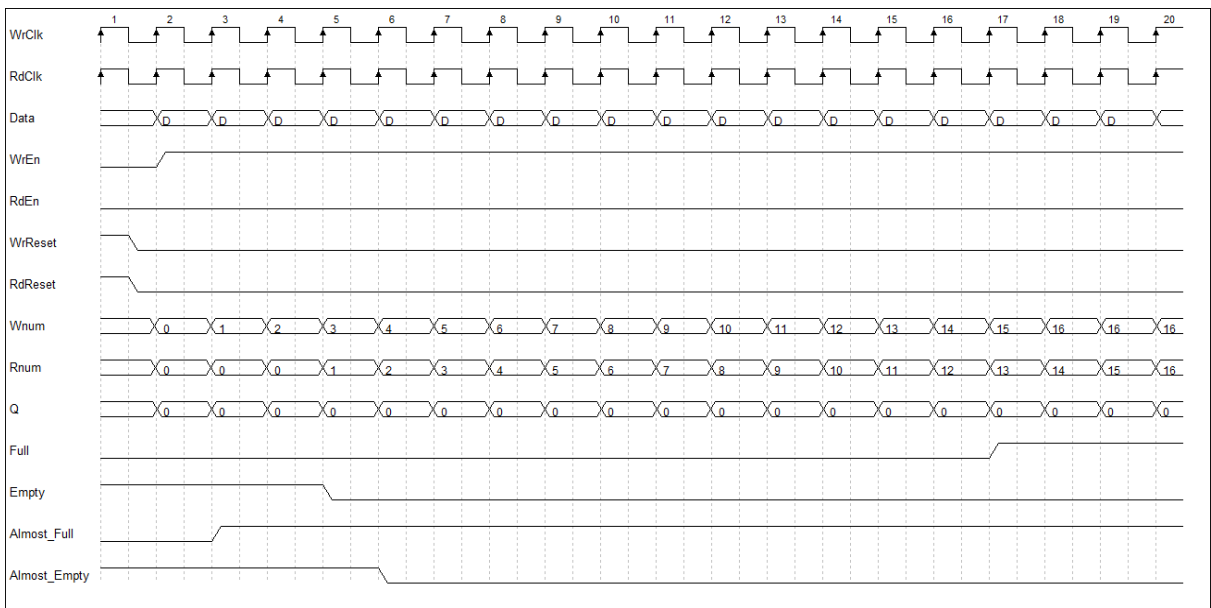
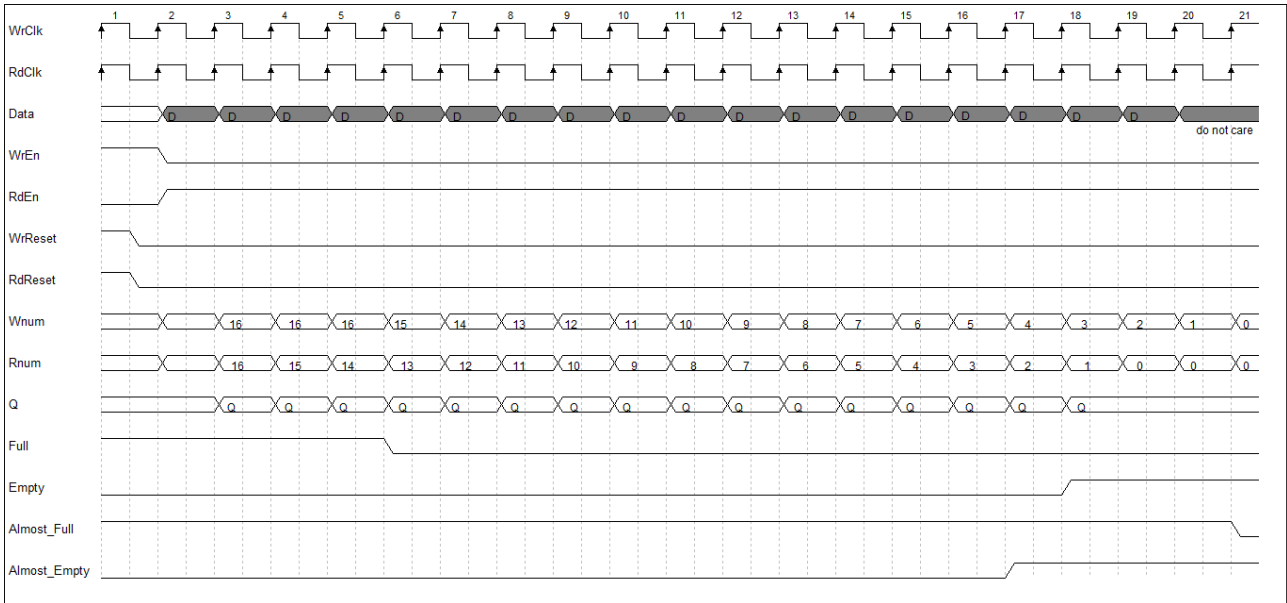


图 4-4 为 FIFO 空标志和将空标志时序图示例。在 Almost_empty 为高时读使能拉高导致 Empty 信号在下一个周期拉高。

图 4-4 FIFO 空标志和将空标志时序图



4.2 FIFO_SC 信号时序

图 4-5 为写操作示例，以 16 数据深度为例，FIFO 不满时，写使能拉高，数据被写入。FIFO 为满状态时，写使能将被屏蔽，满信号拉高。

图 4-5 写操作示例

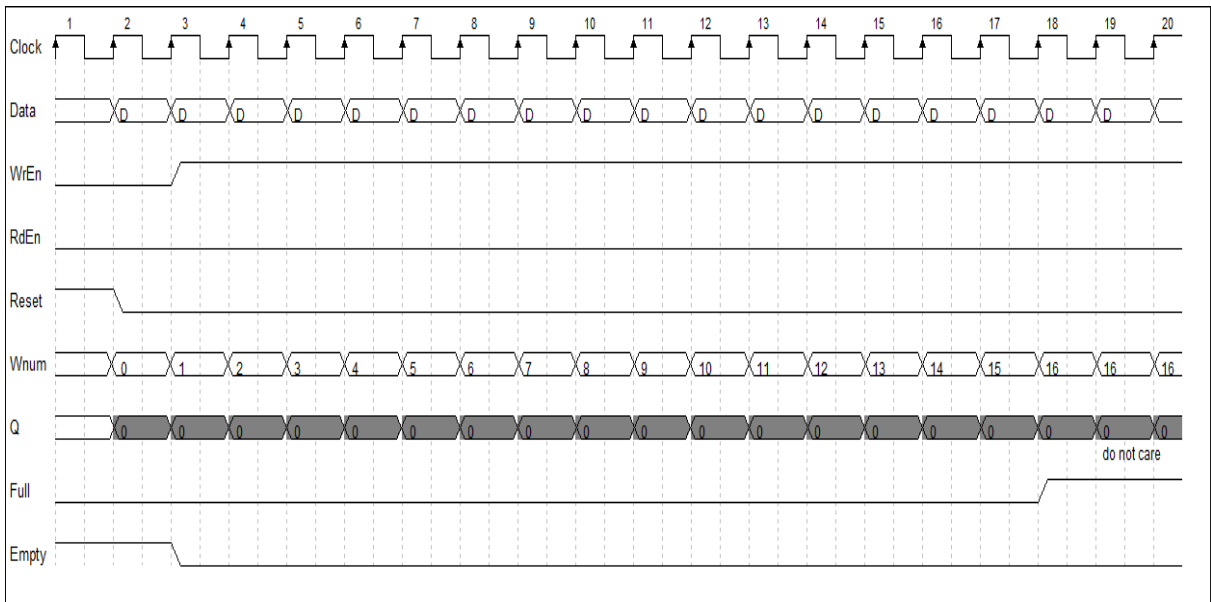


图 4-6 为读操作示例，以 16 数据深度为例，FIFO 不为空时，读使能拉高，数据被读出到 Q。FIFO 为空时，读使能将被屏蔽，空信号拉高。

图 4-6 读操作示例

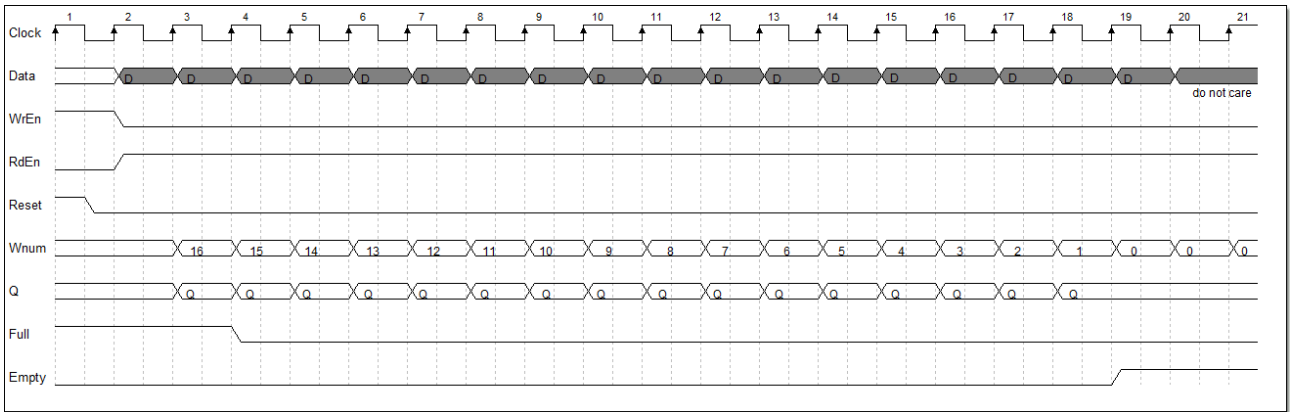


图 4-7 为 FIFO 满标志和将满标志时序图示例。在 FIFO 被写入数据后 Almost_full 拉高。

图 4-7 FIFO 满标志和将满标志时序图

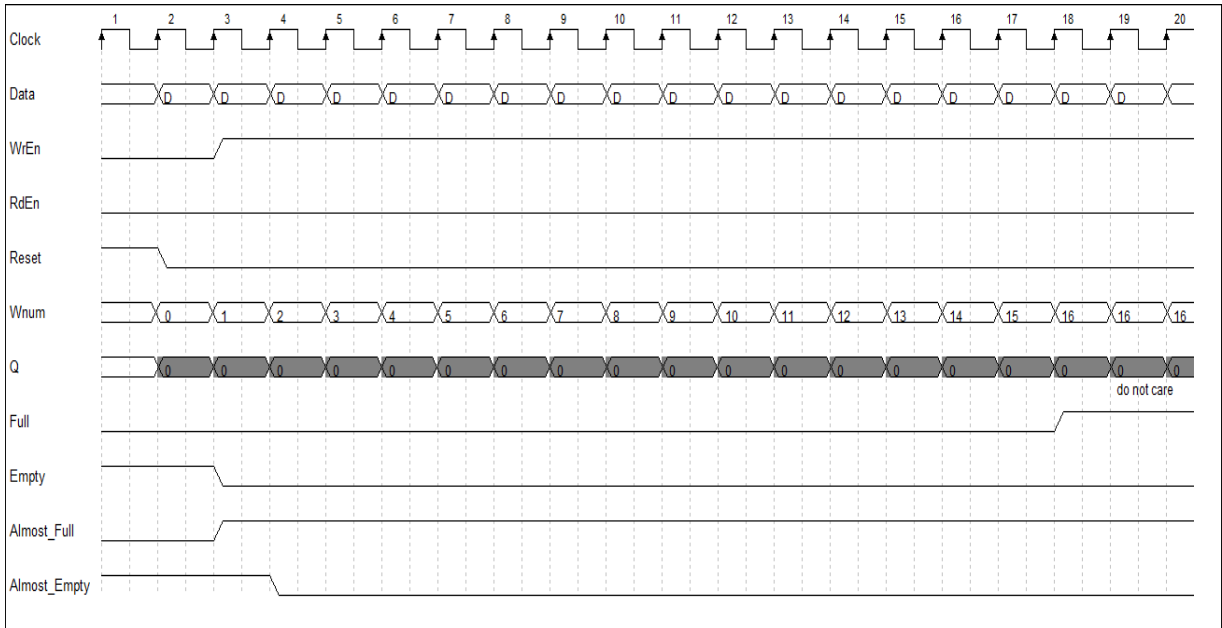
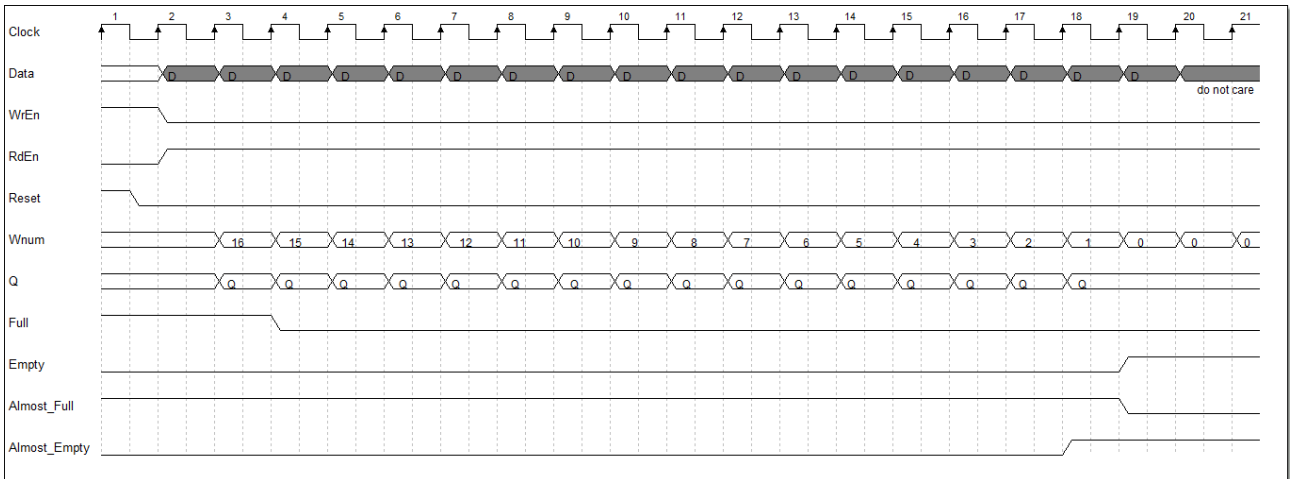


图 4-8 为 FIFO 空标志和将空标志时序图示例。在 Almost_empty 为高时读使能拉高导致 Empty 信号在下一个周期拉高。

图 4-8 FIFO 空标志和将空标志时序图



5 FIFO/FIFO_SC IP 配置

在 Gowin 云源软件界面中，用户可通过菜单栏“Tools”，启动 IP Core Generator 工具，完成调用并配置 FIFO IP 或 FIFO_SC IP。FIFO/FIFO_SC 配置界面及界面参数说明如下所述。

5.1 FIFO IP 配置

FIFO 配置界面如图 5-1 所示。

图 5-1 FIFO IP 配置界面

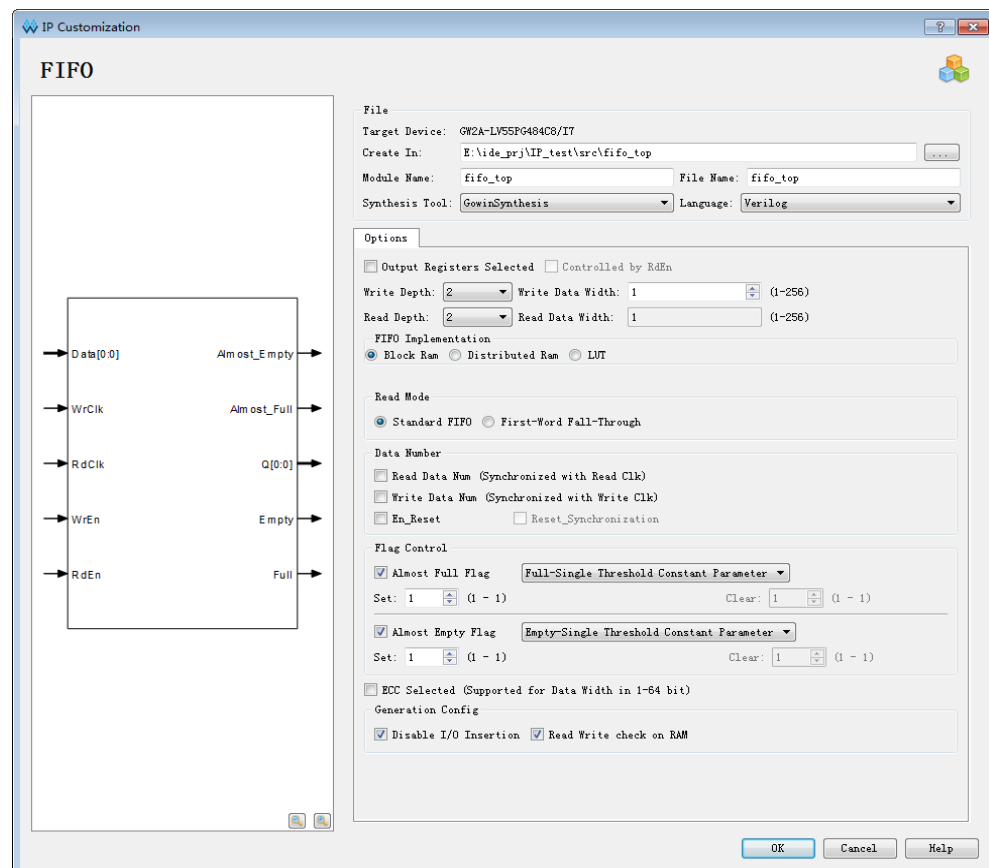


表 5-1 FIFO IP 配置界面参数

选项	参数名称		描述	备注	
File	Target Device		-	-	
	File Name		FIFO 文件名称	-	
	Module Name		FIFO 的顶层模块名称	-	
	Create In		-	FIFO IP Model 产生时可选择是否添加到当前项目	
Options	FIFO Implementation	Block RAM	配置使用 Block Ram、Distributed Ram、LUT	-	
		Distributed RAM			
		LUT			
	Write Depth		写数据深度	异步 FIFO IP 满足写入数据宽度不同于读出数据宽度，Read Data Width 自动计算等于 Write Depth x Write Data Width / Read Depth。	
	Write Data Width		写数据位宽		
	Read Depth		读数据深度		
	Read Data Width		读数据位宽		
	Read Mode	Standard FIFO		标准 FIFO	标准 FIFO 按照时序图标准读写时序，FWFT FIFO 不管是否有读使能信号，都会将写入的第一个数马上放在输出数据总线上，读使能拉高后会按顺序输出写入的其他数据。
		First-Word Fall Through		FWFT FIFO	
	Data Num	Read Data Num		写数据数目	有效时，增加输出 Wnum
		Write Data Num		读数据数目	有效时，增加输出 Rnum
		En_Reset		使能复位	使能时，读写各自复位，增加输入 WrReset、RdReset。
		Reset_Synchronization		同一个复位	En_Reset 有效后，Reset_Synchronization 可选；若选择，表示使用一个复位，增加输入 Reset。
Flag Control	Almost Full Flag		半满标志使能	有效时，增加输出 Almost_Full。	

选项	参数名称		描述	备注
	Almost Full Flag	Full-Single Threshold Constant Parameter	静态半满单常量阈值, 有效时, Set 有效	Almost Full Flag 有效: Full-Single Threshold Constant Parameter、Full-Dual Threshold Constant Parameters、Full-Single Threshold Input Parameter、Full-Dual Threshold Input Parameters 可四选一
		Full-Dual Threshold Constant Parameters	静态半满双常量阈值, 有效时, Set、Clear 有效。	
		Full-Single Threshold Input Parameter	动态半满单输入阈值, 有效时, 增加输入 AlmostFullTh。	
		Full-Dual Threshold Input Parameters	动态半满双输入阈值, 有效, 增加输入 AlmostFullSetTh、AlmostFullClrTh。	
	Set		半满置 1 阈值大小	选择单常量阈值, Set 有效
	Clear		半满清 0 阈值大小	选择双常量阈值, Set、Clear 有效
	Almost Empty Flag		半空标志使能	有效时, 增加输出 Almost_Empty。
	Almost Empty Flag	Empty-Single Threshold Constant Parameter	静态半空单常量阈值, 有效时, Set 有效。	
		Empty-Dual Threshold Constant Parameters	静态半空双常量阈值, 有效时, Set、Clear 有效。	
		Empty-Single Threshold Input Parameter	动态半空单输入阈值, 有效时, 增加输入 AlmostEmptyTh。	
		Empty-Dual Threshold Input Parameter	动态半空双输入阈值, 有效, 增加输入 AlmostEmptySetTh、AlmostEmptyClrTh。	

选项	参数名称		描述	备注
		Set	半空置 1 阈值	选择单常量阈值, Set 有效;
		Clear	半空清 0 阈值	选择双常量阈值, Set、Clear 有效。
	ECC Selected		ECC 功能	Data width 1-64bit 时有效, 增加输出 ERROR。
	Output Register Selected		输出寄存器可选	有效时, 数据延迟一个周期输出。 BSRAM 输出数据需要满足时钟到输出延时。当 Output Register 使能打开时, 这个时间自动满足; 当 Output Register 使能关闭时, 用户需满足时钟到输出延时, 避免采样到亚稳态的数据。
	Controlled by RdEn		输出受 RdEn 控制	Output Register Selected 有效时, 可选。

注!

- 开启 ECC 功能需满足数据宽度小于等于 64 位;
- 可根据需求配置不同输出控制信号和阈值, 各选项配置如表 5-1 所示。
- Read Write check on RAM 功能是防止 RAM 的读写冲突, 具体可查看 FPGA 或 Synplify Pro 的使用手册。

5.2 FIFO_SC IP 配置

FIFO_SC IP 配置界面如图 5-2 所示。

图 5-2 FIFO_SC IP 配置页面

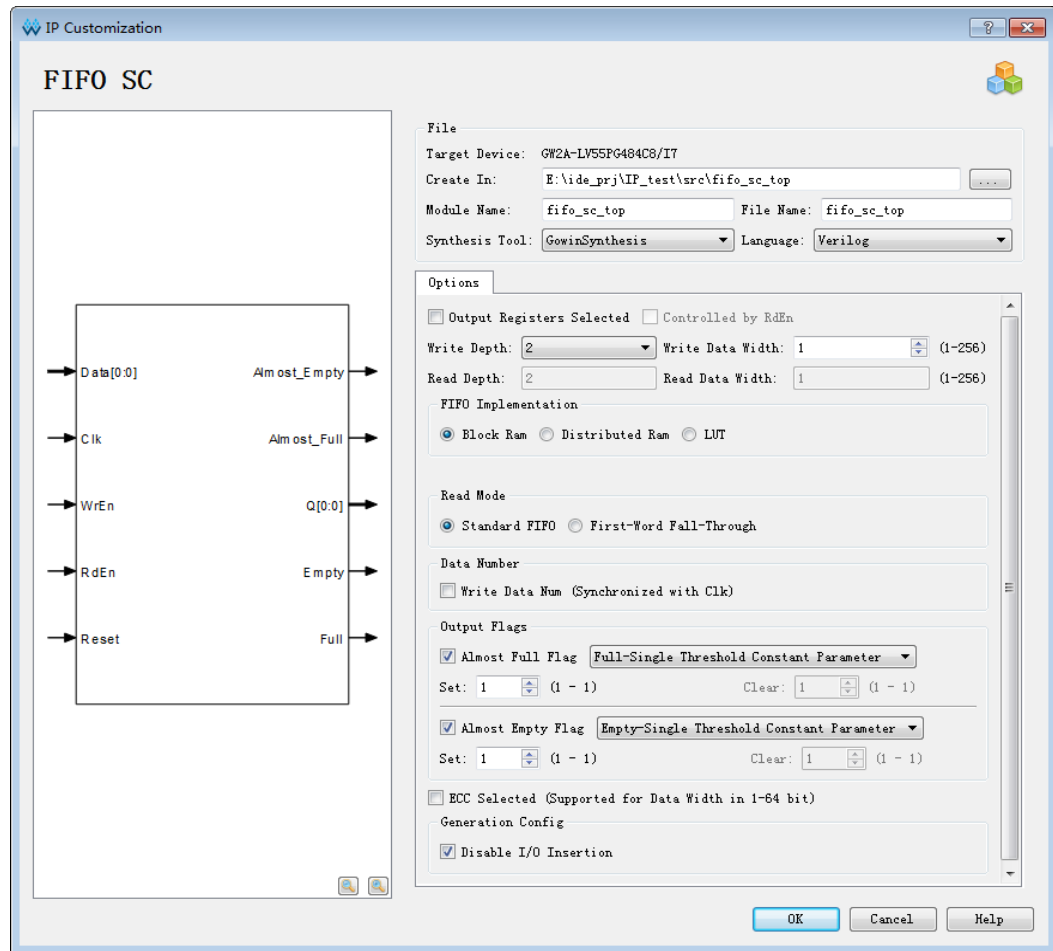


表 5-2 FIFO IP 配置界面参数

选项	参数	描述	备注
File	Target Device	-	-
	Module Name	FIFO 的顶层模块名称	-
	File Name	FIFO 文件名称	-
	Create In	-	FIFO IP Model 产生时可选择是否添加到当前项目
Options	FIFO Implementation	Block RAM	存储结构由 Block RAM 实现
		Distributed RAM	存储结构由分布式 Ram 实现
		LUT	存储结构由 Registers 实现

选项	参数		描述	备注	
	Write Depth		写数据深度	FIFO_SC IP 要求写入数据宽度和深度与读出数据宽度和深度相同, 无需编写 Read Depth 和 Read Data Width, 自动等于 Write Depth 和 Write Data Width。	
	Write Data Width		写数据位宽		
	Read Depth		读数据深度		
	Read Data Width		读数据位宽		
	Read Mode	Standard FIFO		标准 FIFO	标准 FIFO 按照时序图标准读写时序, FWFT FIFO 不管是否有读使能信号, 都会将写入的第一个数马上放在输出数据总线上, 读使能拉高后会按顺序输出写入的其他数据。
		First-Word Fall Through		FWFT FIFO	
	Data Num	Write Data Num (Synchronized with Clk)		写数据数目	有效时, 增加输出 Rnum
	Output Flags	Almost Full Flag	Full-Single Threshold Constant Parameter	静态半满单常量阈值, 有效时, Set 有效	Almost Full Flag 有效: Full-Single Threshold Constant Parameter、Full-Dual Threshold Constant Parameters、Full-Single Threshold Input Parameter、Full-Dual Threshold Input Parameters 可四选一
			Full-Dual Threshold Constant Parameters	静态半满双常量阈值, 有效时, Set、Clear 有效	
			Full-Single Threshold Input Parameter	动态半满单输入阈值, 有效时, 增加输入 AlmostFullTh	
			Full-Dual Threshold Input Parameters	动态半满双输入阈值, 有效, 增加输入 AlmostFullSetTh、AlmostFullClrTh	
	Set	半满置 1 阈值大小	选择单常量阈值, Set 有效;		
	Clear	半满清 0 阈值大小	选择双常量阈值, Set、Clear 有效。		

选项	参数		描述	备注
	Almost Empty Flag	Empty-Single Threshold Constant Parameter	静态半空单常量阈值, 有效时, Set 有效	Almost Empty Flag 有效: Empty -Single Threshold Constant Parameter、Empty-Dual Threshold Constant Parameters、Empty-Single Threshold Input Parameter、Empty-Dual Threshold Input Parameters 可四选一
		Empty-Dual Threshold Constant Parameters	静态半空双常量阈值, 有效时, Set、Clear 有效	
		Empty-Single Threshold Input Parameter	动态半空单输入阈值, 有效时, 增加输入 AlmostEmptyTh	
		Empty-Dual Threshold Input Parameter	动态半空双输入阈值,有效, 增加输入 AlmostEmptySetTh、AlmostEmptyClrTh	
		Set	半空置 1 阈值	选择单常量阈值, Set 有效。
		Clear	半空清 0 阈值	选择双常量阈值, Set、Clear 有效。
	ECC Selected		ECC 功能	有效时, 增加输出 ERROR。
Output Register Selected		输出寄存器可选	有效时, 数据晚一个周期输出。 BSRAM 输出数据需要满足时钟到输出延时。当 Output Register 使能打开时, 这个时间自动满足; 当 Output Register 使能关闭时, 用户一定要满足时钟到输出延时, 避免采样到亚稳态的数据。	
Controlled by RdEn		输出受 RdEn 控制	Output Register Selected 有效时, 可选。	

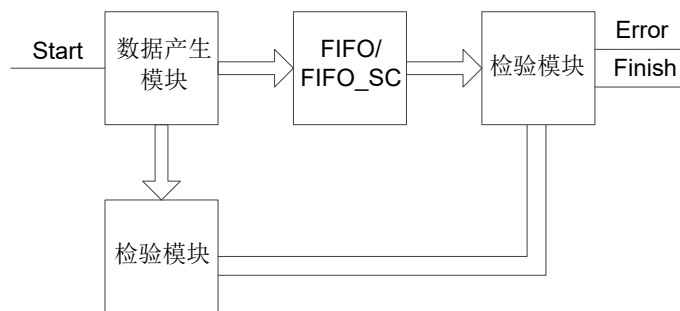
注!

- 开启 ECC 功能需满足数据宽度小于等于 64 位;
- 可根据需求配置不同输出控制信号和阈值, 各选项配置如表 5-2 所示。

6 参考设计

本节主要介绍 FIFO 和 FIFO_SC 的参考设计实例的搭建及其使用方法。FIFO 与 FIFO_SC 的设计实例基本结构基本相同，如图 6-1 所示，只是在读写位宽和读写时钟域上需要稍作更改。

图 6-1 FIFO/FIFO_SC 参考设计实例结构



在本设计实例中，有数据产生模块产生数据流，数据流进入 FIFO/FIFO_SC 后再输出，在数据产生模块和 FIFO/FIFO_SC 的 Q 端口处设计数据检验模块，以确定数据流经 FIFO/FIFO_SC 后的正确性，整个过程以“Start”键开始，完成一轮读写后 LED 点亮表示读写和检查结束，8 段数码管显示数据错误个数“Error”信号。

在参考设计中，两种 FIFO 读写位宽不同，其中：

- FIFO 采用写位宽 16 深度 1024，读位宽 8 深度 2056，读写在两个时钟域操作；
- FIFO_SC 采用读写位宽都为 16，深度 1024，读写在相同时钟域操作。

6.1 FIFO 参考设计

该部分参考设计实现了 FIFO 对异步时钟域数据的传递和缓存功能，同时实现了不同位宽的传递，将写入的 16 位数据分两次 8 位数据传出，先输出低 8 位，后输出高 8 位。顶层模块为 fifo_rf，端口定义如表 6-1 所示，配置信息如下：

- 使用 Block Ram 生成 FIFO;
- 写数据宽度 16 深度 1024, 读数据宽度 8 深度 2056;
- GW_PLL: 提供读写时钟, 写时钟 5M, 读时钟 10M;
- 输出 Rnum、Wnum、Empty、Full、Almost_Empty 和 Almost_FULLL;
- 读写分别复位。

表 6-1 FIFO 参考设计端口定义

Signal	Direction	Description
rstn	I	复位信号, 低有效
clk	I	板上 50M 时钟送入 pll
start	I	系统开始工作信号
finish	O	数据传递检测结束信号
seg	O	error 数据传到数码管管脚显示信号
seg_en	O	数码管位选

6.2 FIFO_SC 参考设计

该部分参考设计实现了 FIFO_SC 对相同时钟域数据的传递和缓存功能, 读写数据位宽和深度相同, 数据宽度 16 位深度 1024。顶层模块为 `fifosc_rf`, 端口定义如表 6-2 所示, 配置信息如下:

- 使用 Block Ram 生成 FIFO;
- 写数据宽度 16 深度 1024, 读数据宽度 16 深度 1024;
- GW_PLL: 提供读写时钟 5M;
- 输出 Wnum、Empty、Full、Almost_Empty 和 Almost_FULLL;
- 同步复位。

表 6-2 FIFO_SC 参考设计端口定义

Signal	Direction	Description
rstn	I	复位信号, 低有效
clk	I	板上 50M 时钟送入 pll
start	I	系统开始工作信号
finish	O	数据传递检测结束信号
seg	O	error 数据传到数码管管脚显示信号
seg_en	O	数码管位选

6.3 参考设计应用

通过本参考设计能够快速验证 FIFO/FIFO_SC 的功能。当该参考设计应用于板级测试时, 用户只需输入时钟、复位、`start` 信号即可快速检测 FIFO/FIFO_SC 功能。

参考设计基于 GW2A-55K 开发板，连接板上的 LED 灯和 8 端数码管可帮助用户直观了解 FIFO/FIFO_SC 输出数据是否正确和读写完成时机。

