

Gowin Software Quick Start Guide

SUG918-2.0E,2025-02-28

Copyright © 2025 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is a trademark of Guangdong Gowin Semiconductor Corporation and is registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description			
05/07/2020	1.0E	Initial version published.			
09/07/2020	1.1E	 RTL schematic added. File encryption added. Tcl command added. 			
10/21/2020	1.1.1E	Use GowinSynthesis as an example to describe synthesis.			
06/10/2021	1.2E	Synplify Pro removed.MIPI IP in the design modified.			
11/02/2021	1.3E	Some descriptions updated.			
07/28/2022	1.4E	Modified the design to FIFO HS and updated the relevant descriptions.			
12/20/2022	1.5E	 The function of viewing the schematic diagram of the netlist after synthesis added. Some figures updated. 			
05/25/2023	1.5.1E	 Figure 3-15 Synthesis Configuration and Figure 3-34 Jointly Debugging with GAO updated. The description of 3.10.1 Configuration updated. 			
08/18/2023	1.6E	The descriptions of timing optimization removed.			
11/30/2023	1.7E	 create_project, import_files, and run close commands added. Some screenshots in Chapter 3 Quick Start updated. 			
12/29/2023	1.7.1E	 The title of Figure 3-3 updated to FIFO HS Configuration. Minor description changes. 			
02/02/2024	1.7.2E	Some screenshots in Chapter 3 Quick Start updated.			
06/28/2024	1.8E	 Some screenshots in Chapter 3 Quick Start updated. Descriptions of Section 4.2 Tcl Quick Start updated. 			
08/09/2024	1.9E	 GVIO configuration added. The device information of FIFO_HS project updated. Some screenshots updated. 			
02/28/2025	2.0E	Descriptions of Section 4.2 Tcl Quick Start updated.			

Contents

Contentsi
List of Figuresiii
List of Tablesv
1 About This Guide1
1.1 Purpose
1.2 Related Documents
1.3 Terminology and Abbreviations1
1.4 Support and Feedback
2 Introduction
2.1 Design Flow Introduction
2.2 Quick Started Design Introduction
3 Quick Start
3.1 Create a New Project
3.1.1 Create a New Project5
3.1.2 Generate FIFO HS IP
3.1.3 Load File7
3.2 RTL Schematic
3.3 GAO Configuration8
3.3.1 Create Standard Mode GAO Config File8
3.3.2 Configure Standard Mode GAO9
3.4 GVIO Configuration 11
3.4.1 Create GVIO Configuration File 11
3.4.2 Configuration Options 11
3.5 Use GowinSynthesis to Synthesize12
3.5.1 Configuration
3.5.2 Synthesize
3.6 View Schematic Diagram of the Netlist after Synthesis
3.7 Physical Constraints
3.7.1 Create New Physical Constraints15
3.7.2 Modify Physical Constraints
3.8 Timing Constraint

	3.8.1 Create New Timing Constraints	. 17
	3.8.2 Modify Timing Constraints	. 19
	3.9 GPA Configuration	. 19
	3.9.1 Create GPA Config File	. 19
	3.9.2 Configure GPA	. 19
	3.10 Place & Route	. 22
	3.10.1 Configuration	. 22
	3.10.2 Run PnR	. 23
	3.11 Download Bitstream	. 24
	3.12 Debugging with GVIO and Data Acquisition with GAO	. 25
	3.13 Output Files	. 27
	3.13.1 Place & Route Report	. 27
	3.13.2 Ports and Pins Report	. 27
	3.13.3 Timing Report	. 28
	3.13.4 Power Analysis Report	. 28
	3.14 File Encryption	. 29
	3.14.1 Source File Encryption	. 29
	3.14.2 Simulation File Encryption	. 30
4 -	Гс!	32
	4.1 Tcl Execution	. 32
	4.1.1 Execution Using Tcl Editing Window	. 32
	4.1.2 Execution Using Tcl Command Line	. 32
	4.2 Tcl Quick Start	. 33

List of Figures

Figure 2-1 Open Example Project	4
Figure 3-1 Create a New Project	. 5
Figure 3-2 Project Directory	. 6
Figure 3-3 FIFO HS Configuration	. 6
Figure 3-4 FIFO HS IP Directory	. 7
Figure 3-5 Design Window	. 7
Figure 3-6 Load Files	. 8
Figure 3-7 Create GAO Config File	. 9
Figure 3-8 GAO Setting	. 9
Figure 3-9 Trigger Options Configuration	. 10
Figure 3-10 Capture Options Configuration	. 10
Figure 3-11 GAO Config Files	. 10
Figure 3-12 Create GVIO Configuration File	. 11
Figure 3-13 GVIO Configuration Window	. 12
Figure 3-14 GVIO Configuration File	. 12
Figure 3-15 Synthesis Configuration	. 13
Figure 3-16 Attributes and Instructions of GowinSynthesis	. 13
Figure 3-17 Synthesis Completed	. 14
Figure 3-18 gwsynthesis Directory	. 14
Figure 3-19 GAO Directory	. 15
Figure 3-20 I/O Constraints	. 16
Figure 3-21 Physical Constraints Display	. 16
Figure 3-22 Clock Constraints	. 17
Figure 3-23 Timing Report Constraint	. 18
Figure 3-24 Timing Constraints Display	. 18
Figure 3-25 Create GPA Config File	. 19
Figure 3-26 General Setting Configuration	. 20
Figure 3-27 Rate Setting Configuration	. 21
Figure 3-28 Clock Setting Configuration	. 21
Figure 3-29 GPA Config Files	. 22
Figure 3-30 Place & Route Configuration	. 23
Figure 3-31 Place & Route Completed	. 23

Figure 3-32 PnR Directory	24
Figure 3-33 Programmer	24
Figure 3-34 Jointly Debugging with GAO and GVIO	25
Figure 3-35 Jointly Debugging and Sampling with GAO and GVIO when gvio_test=0	26
Figure 3-36 Jointly Debugging and Sampling with GAO and GVIO when gvio_test=1	26
Figure 3-37 Place & Route Report	27
Figure 3-38 Ports & Pins Report	27
Figure 3-39 Timing Report	28
Figure 3-40 Power Analysis Report	28
Figure 3-41 Hierarchy Window	29
Figure 3-42 Pack User Design Dialog Box	30
Figure 4-1 Tcl Editing Window	32
Figure 4-2 Tcl Command Line Example	33
Figure 4-3 Tcl Script File Example	33

List of Tables

Table 1-1 Terminology and Abbreviations	1
---	---

1 About This Guide

1.1 Purpose

This manual uses FIFO HS as an example to introduce Gowin Software and aims to help you get familiar with the usage and improve the design efficiency.

1.2 Related Documents

You can find the related documents at <u>www.gowinsemi.com</u>:

- <u>SUG100, Gowin Software User Guide</u>
- <u>SUG935, Gowin Design Physical Constraints User Guide</u>
- <u>SUG940, Gowin Design Timing Constraints User Guide</u>
- <u>SUG114, Gowin Analyzer Oscilloscope User Guide</u>
- SUG282, Gowin Power Analyzer User Guide
- <u>SUG502, Gowin Programmer User Guide</u>
- <u>SUG550, GowinSynthesis User Guide</u>
- SUG755, Gowin HDL Schematic Viewer User Guide
- <u>SUG1018, Arora V Design Physical Constraints User Guide</u>
- <u>SUG1189, Gowin Virtual Input Output User Guide</u>

1.3 Terminology and Abbreviations

Table 1-1 shows the abbreviations and terminology used in this manual.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
AO Core	Analysis Oscilloscope Core
BSRAM	Block Static Random Access Memory
DFF	D Flip-Flop
FloorPlanner	FloorPlanner
FPGA	Field Programmable Gate Array

Terminology and Abbreviations	Meaning
GAO	Gowin Analyzer Oscilloscope
GPA	Gowin Power Analyzer
GVIO	Gowin Virtual Input Output
I/O	Input/Output
IP Core	Intellectual Property Core
PnR	Place & Route
RTL	Register Transfer Level
Tcl	Tool Command Language

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Introduction

2.1 Design Flow Introduction

Gowin Software is available in Windows and Linux. It supports GUI running mode and commands running mode. Take the GUI running mode in Windows and FIFO HS design as an instance to introduce quick start of Gowin Software.

The design uses FloorPlanner to add physical constraints, uses Timing Constraints Editor to add timing constraints, uses GAO to add GAO config file and to capture data, GPA to add GPA config file, and Programmer to download bitstream.

2.2 Quick Started Design Introduction

FIFO HS IP can complete the data transmission and buffering with different bit widths in the asynchronous clock domain, and configure different output control signals and data structures according to your requirements.

The whole design provides clock for FIFO HS through port, provides reset signal, enable signal and input data through logic, and finally uses GAO to collect data to verify the correctness of FIFO HS.

The design has been added to the sample project FIFO_HS, which can be quickly created by clicking "Start Page > Open Example Project...", as shown in Figure 2-1. Creating a project through the example will skip the previous steps and go directly to placement and routing, and the subsequent process. If you want to be familiar with the use of Gowin Software step by step, you can operate according to the guidelines in the document. The source files, constraint files and configuration files involved in the design are consistent with those in the sample project. You can save the files in the sample project for later use.

🐝 G	OWIN FPGA Designer - [St	tart Page]	• •	,							_		×
💡 Eil	e <u>E</u> dit <u>P</u> roject <u>T</u> ools	<u>W</u> indow <u>H</u> e	lp										_ 8 ×
		5		🖾 🔮 🔒									
Rec	ent Projects:		Quick Start	. Open Pr) oject Op	en Example Pr	oject						
	💡 Open Example Project	:									?	×	
	Example Project Name	Project Descr	iption	Source Type	Target Device	Target Devi	Target Part Number	CST	SDC	GAO	GPA	^	
	game	Sample game	using state	Verilog	GW1N-4	D	GW1N-UV4PG256C6	Y	Y	Y	N		
¥	сри	Achieve state j	ump, and reali	System Verilog	GW1N-2	С	GW1N-UV2LQ144XC	Y	Y	Y	Y		
Mes	vga_text_generator	A basic VGA te	ext generator f	VHDL	GW1N-9	с	GW1N-LV9LQ100C6/I5	Y	Y	Y	N		D' X
	can	Can design, ci	phertext source	Verilog	GW2A-18	с	GW2A-LV18PG256C8	Y	Y	Y	N		
	FIFO_HS	FIFO HS IP de	sign, ciphertext	Verilog	GW1N-9	с	GW1N-LV9LQ144C6/I5	Y	Y	Y	Y		
	<										>	Ť	
	Create In: E:\												
									OK		Cance		
Con	sole Message												

Figure 2-1 Open Example Project

3 Quick Start

3.1 Create a New Project

3.1.1 Create a New Project

Open Gowin Software and click "Start Page > Quick Start > New Project" to create a new project named as FIFO_HS. The device selected is as shown in Figure 3-1.

- Series: GW1N
- Device: GW1N-9
- Device Version: C
- Package: LQFP144
- Speed: C6/I5
- Part Number: GW1N-LV9LQ144C6/I5

Click "Next" until the project creation completed. For the details, please refer to <u>SUG100, Gowin Software User Guide</u>.

Figure 3-1 Create a New Project

🐳 Project Wizard									>	×
Project Name Select Device Summary	Select Device Specify a target device for your Filter	· projec	t							
	Series:	GW1	N		\sim	Package:	LQFP144		~	
	Device:	GW1	N-9		\sim	Speed:	C6/I5		\sim	
	Device Version: ≵no version number is initial version	С			~					
	Part Number		Device	D	evice	Version	Package	Speed	Voltage	
	GW1N-LV9LQ144C6/I5	G	W1N-9	с			LQFP144	C6/I5	LV	
	GW1N-UV9LQ144C6/I5	G	W1N-9	C			LQFP144	C6/I5	UV	
	٢								>	
							< <u>B</u> ack	<u>N</u> ext >	Cancel	

After the project is created, the impl and src folders are generated

under the project creation path, as shown in Figure 3-2. impl contains synthesis and PnR files and src contains the source files.

Figure 3-2 Project Directory

Name	Date modified	Туре	Size
impl	5/31/2022 15:54	File folder	
src	5/31/2022 15:54	File folder	
🐳 FIFO_HS.gprj	5/31/2022 15:43	GPRJ File	1 KB
FIFO_HS.gprj.user	5/31/2022 15:51	USER File	4 KB

3.1.2 Generate FIFO HS IP

Click "Tools > IP Core Generator" to open the IP Core Generator window. Double-click "Memory Control > FIFO > FIFO HS" to open the IP Customization dialog box to configure as required. The FIFO HS configuration in this design is shown in Figure 3-3. Then click "OK" to generate FIFO HS IP.

Figure 3-3 FIFO HS Configuration

🐝 IP Customization		? ;
FIFO HS		en e
		General Device: GW1N-9 Device: GW1N-LV9LQ144C6/15 Language: Verilog File Name: FIFO_HS Create In: Ex/FIFO_HS\src\FIFO_HS
Data[31:0]	Ole30	Options Output Registers Selected Controled by RdEn Write Depth: 1024 Write Data Width: 32 ÷ (1~256)
→ WrGlk	Empty	Read Depth: 512 V Read Data Width: 64 (1~256) FIFO Implementation BSRAM SSRAM REG
→ Rd Clk	Full	Read Mode
→ Rd En	Rn u m[9:0] 🔶	Standard FIFO O First-Word Fall-Through Data Number A Read Data Num(Synthronized with Read Clk). Write Data Num(Synthronized with Write Clk)
→ WrReset	Almost_Empty	En_Reset
Ri Reset	Almost_Ful	Imag Conton Imag Conton Imag Conton Imag Conton Set: Imag Conton Imag Conton Clear: Imag Conton Imag Conton Imag Conton Imag Conton Imag Conton Imag Conton Image Conton Image
	e	OK Cancel

After generation, IP design files and simulation files are generated under the IP creation path, as shown in Figure 3-4.

• .v file is an IP design file, encrypted.

- _tmp.v is an IP design template file.
- .vo file is an IP simulation model file, unencrypted.
- .ipc file is an IP configuration file. The user can load the file to modify the configuration.
- temp contains the files required to generate the IP.
- The doc, model, sim, and tb contain the simulation files: readme text, simulation model, simulation script, and testbench.

Note!

- For Gowin Software 1.9.8.06 and later versions, if VHDL is selected as the Language during IP generation, .vho file will be generated under the IP creation path, which is an IP simulation model file in plaintext.
- At present, for some IPs, the created path still generates doc, model, sim, and tb folders, indicating readme text, simulation model, simulation script, and testbench simulation file. The IP directory is subject to IP Core Generator in use.

Figure 3-4 FIFO HS IP Directory

Name	Date modified	Туре	Size
d temp	5/31/2022 15:54	File folder	
📔 FIFO_HS.ipc	5/30/2022 16:59	IPC File	1 KB
📔 FIFO_HS.v	5/30/2022 16:59	V File	59 KB
FIFO_HS.vo	5/30/2022 16:59	VO File	60 KB
FIFO_HS_tmp.v	5/30/2022 16:59	V File	1 KB

After FIFO HS IP is generated, the Design window is as shown in Figure 3-5.

Figure 3-5 Design Window



3.1.3 Load File

In order to test FIFO HS, some design files need to be loaded or created, as shown in Figure 3-6. For the steps to load files, you can see Section <u>2.2 Quick Started Design Introduction</u>.

Design			ð	×		
Y 🧰 FIF	FIFO_HS - [E:\FIFO_HS\FIFO_HS.gprj]					
🔍 🔍	GW1N-LV9 Verilog File	LQ144C6/I5 s				
src\FIFO_HS\FIFO_HS.v						
src\rstn_gen.v						
src\test_fifo.v						
Design	Process	Hierarchy				

Figure 3-6 Load Files

3.2 RTL Schematic

After the source file is loaded, you can view the RTL design schematic by clicking "Tools > Schematic Viewer > RTL Design Viewer" to help you better understand the RTL logic. For details, see <u>SUG755, Gowin HDL</u> <u>Schematic Viewer User Guide</u>.

3.3 GAO Configuration

Gowin Software supports two signal capture sources: RTL-level signal capture and post-synthesis netlist-level signal capture; GAO config. file can be created after the source files are created or loaded at the RTL level, and GAO config. file can be created after the synthesis is completed at the post-synthesis netlist level. GAO config. file can be used to capture data and verify the the design. In addition, Gowin Software provides Standard Mode GAO and Lite Mode GAO. For the usage, see <u>SUG114, Gowin</u> <u>Analyzer Oscilloscope User Guide</u>.

This design uses RTL-level signal capture and Standard Mode GAO as an instance.

3.3.1 Create Standard Mode GAO Config File

Select "Design > New File..." to open "New" dialog box, and select "GAO Config File" in "New", as shown in Figure 3-7. Click "OK".

🐝 New	?	Х
Physical Constraints File		^
📑 Timing Constraints File		
📑 GowinSynthesis Constraints File		
📑 User Flash Initialization File		
📑 GAO Config File		
GVIO Config File		
GPA Config File		~
Create a GAO Config File.		
ОК	Can	cel

Figure 3-7 Create GAO Config File

Select "For RTL Design" in Type, and "Standard" in Mode, as shown in Figure 3-8. Click "Next". The file name is FIFO HS. Then click "Next" until finished.

Figure 3-8 GAO Setting

🗞 New GAO Wizard		×
🔿 GAO Setting	GAO Setting	
GAO Configure File	Туре	
Summary	I For RTL Design	
	○ For Post-Synthesis Netlist	
	Mode	
	Standard	
	⊖ Lite	
	For RTL Design, analyse rtl design.	
	For Post-Synthesis Netlist, analyse post-synthesis netlist.	
	<u>N</u> ext > Can	cel

3.3.2 Configure Standard Mode GAO

After file created, you can configure the number of AO cores, trigger options and capture options. The trigger options include match unit, trigger port, match type and expressions; The capture options include sample clock, capture, capture utilization and capture signals. In this design the number of AO cores is 1 and the trigger options and capture options configuration are shown in Figure 3-9 and Figure 3-10.

Ao Core	Core 0									
Core 0	Trigger Options Captu	re Options								
	Trigger Ports	Match	Units						Expressions	
	✓ Trigger Port 0	Mat	tch Unit	Trigger Port	Match Type	Function	Counter	Value	 Static 	🔿 Dyn
	rst_n Trigger Port 1		M0	Trigger 0	Basic w/edges	==	Disabled	R	M0	
	Trigger Port 2		M1	NONE	Basic	= =	Disabled			
	Trigger Port 3 Trigger Port 4		M2	NONE	Basic	= =	Disabled			
	Trigger Port 5		M3	NONE	Basic	= =	Disabled			
	Trigger Port 6 Trigger Port 7		M4	NONE	Basic	= =	Disabled			
	Trigger Port 8		M5	NONE	Basic	= =	Disabled			
	Trigger Port 9		M6	NONE	Basic	= =	Disabled			
	Trigger Port 11		M7	NONE	Basic	==	Disabled			
	Trigger Port 12		M8	NONE	Basic	==	Disabled			
	Trigger Port 14		M9	NONE	Basic	==	Disabled			
	Trigger Port 15		M10	NONE	Basic	= =	Disabled			

Figure 3-9 Trigger Options Configuration



Ao Core	Core 0
Core 0	Trigger Options Capture Options Sample Clock Clock: Sample On:

After configuration, click "Save" to finish and the design window is as shown in Figure 3-11.

Figure 3-11 GAO Config Files



3.4 GVIO Configuration

Gowin Virtual Input Output (GVIO) can monitor and drive internal FPGA signals in real-time. When jointly debugging with the online logic analyzer Gowin Analyzer Oscilloscope (GAO), GVIO provides a more powerful debugging environment. This debugging environment can generate internal signal stimuli and obtain logic responses through the GAO tool, aiming to help users quickly perform system analysis and fault localization, thereby improving design efficiency. For detailed usage of GVIO, <u>SUG1189, Gowin Virtual Input Output User Guide</u>.

3.4.1 Create GVIO Configuration File

Select "Design > New File..." in Gowin Softare. In the "New" dialog box that appears, choose to create a new "GVIO Config File," as shown in Figure 3-12. Click "OK," define the file name as FIFO_HS, and the file path defaults to the src folder under the project. Click "OK" again to complete the creation of the GVIO configuration file.

New	Physical Constraints File	?	×
	Physical Constraints File		
			^
	Timing Constraints File		
	GowinSynthesis Constraints File		
	User Flash Initialization File		
	GAO Config File		
	GVIO Config File		
	GPA Config File		
	Memory Initialization File		
			¥
Create a C	GVIO Config File.		

```
Figure 3-12 Create GVIO Configuration File
```

3.4.2 Configuration Options

Double-click the configuration file (.gvio) in the "Design" view. The GVIO Config window will pop up. The GVIO configuration window includes the GVIO Core view for configuring the number of AO cores and their corresponding signal configuration view. The core signal configuration view consists of "Probe In" view for configuring sampling signals and "Probe Out" view for configuring stimulus signals. In this design, the number of AO core is set to 1. The configuration of the sampling and stimulus signals is shown in Figure 3-13.

GVIO Core	gvio_0	
gvio_0	Probe Ports Probe In Probe In rst_n w_data_d[31:0] Add Remove	Probe Out Ports Initial Value v Probe Out0 gvio_test 0 Add Remove Output Probe Synchronizes with User Clock

Figure 3-13 GVIO Configuration Window

After the configuration is completed, click "Save" on the toolbar to complete the GVIO configuration file. The GVIO configuration file will be displayed in the "Design" window, as shown in Figure 3-14.

Figure 3-14 GVIO Configuration File

Design	8	×			
✓					
😡 GW1N-LV9LQ144C6/I5					
🗸 📂 Verilog Files					
src\FIFO_HS\FIFO_HS.v					
src\rstn_gen.v					
src\test_fifo.v					
✓ 2 GAO Config Files					
src\FIFO_HS.rao					
🗸 📂 GVIO Config Files					
src\FIFO_HS.gvio					
Design Process Hierarchy					

3.5 Use GowinSynthesis to Synthesize

3.5.1 Configuration

Select "Process > Synthesize (right-click) > Configuration" to open "Configuration" dialog box. For details, see <u>SUG550, GowinSynthesis User</u> <u>Guide</u>.

The top module/entity is test_fifo, as shown in Figure 3-15.

Synthesize	
Synthesis Tool:	GowinSynthesis
Top Module/Entity:	test_fifo
Include Path:	
TclPre	
GowinSynthesis	
Verilog Language:	System Verilog 2017 🔻
Looplimit:	2000
Show All Warnin	gs
Disable Insert Pa	ad
Ram R/W Check	:

Figure 3-15 Synthesis Configuration

In addition, you can add some attributes and instructions to the source file to control synthesis. For the details, see <u>SUG550, GowinSynthesis</u> <u>User Guide</u>. As shown in Figure 3-16, in this design, a specific net is retained without optimization during the synthesis by using the/* synthesis syn_keep=1 */ attribute.

Figure 3-16 Attributes and Instructions of GowinSynthesis

0			•
67	reg	[1:0]	ALT_CNT_d;
68	reg	[7:0]	rand_num;
69	reg	[9:0]	rand cnt;
70	reg	[11:0]	start_rdmck;
71	reg		fifo empty d;
72	wire	[WRSIZE-1:0]] w_data_d/* synthesis syn_keep=1 */;
73	wire		load;
74	wire	[RDSIZE-1:0]	r_data;
75	wire	[WNSIZE:0]	w_num;
76	wire	[RNSIZE:0]	r_num;
77	wire		fifo_full;
78	wire		fifo empty;
79	wire		fifo_alempty;
80	//test	state machine	e

3.5.2 Synthesize

After synthesis configuration, you can start to synthesize.

Double-click "Synthesize" in Process window to synthesize, as shown in Figure 3-17. When the icon changes to ", you can double-click Synthesis Report to view the report and double-click Netlist File to view the netlist file.

Process 🗗 🗙					
📳 Design Summary					
🗸 🎲 User Constraints					
🔢 FloorPlanner					
🔀 Timing Constraints Editor					
🗸 🧭 Synthesize					
Synthesis Report					
Netlist File					
✓ Place & Route					
Place & Route Report					
Timing Analysis Report					
Ports & Pins Report					
Power Analysis Report					
↓ ¹⁰ Programmer					
Design Process Hierarchy					

Figure 3-17 Synthesis Completed

After synthesis, the gwsynthesis folder is generated under the \impl path. The folder contains all the files and folders generated in synthesis, as shown in Figure 3-18.

Figure 3-18 gwsynthesis Directory

Name	Date modified	Туре	Size
RTL_GAO	5/31/2022 15:54	File folder	
📔 FIFO_HS.log	5/31/2022 15:51	LOG File	6 KB
FIFO_HS.prj	5/31/2022 15:50	PRJ File	2 KB
📔 FIFO_HS.vg	5/31/2022 15:51	VG File	454 KB
FIFO_HS_syn.rpt.html	5/31/2022 15:51	360 se HTML Doc	29 KB
FIFO_HS_syn_resource.html	5/31/2022 15:51	360 se HTML Doc	3 KB
FIFO_HS_syn_rsc.xml	5/31/2022 15:51	XML Document	1 KB

If the project contains the GAO config file, after PnR, RTL_GAO folder is generated under the project creation path \impl\gwsynthesis, as shown in Figure 3-18, and this folder contains all the files generated by the RTL GAO synthesis as shown in Figure 3-19.

- ao_0 contains the parameter files of the AO core.
- ao_control contains the parameter files of the control AO core.
- gao.v is the netlist file GAO post-synthesis, encrypted.
- gw_gao_top.v is the top file of GAO, connecting ao, ao_control and jtag modules.
- The other files are generated during GAO synthesis.

Figure 3-19 GAO Directory

Name	Date modified	Туре	Size
ao_0	5/31/2022 15:54	File folder	
ao_control	5/31/2022 15:54	File folder	
🧾 gw_gao_top.v	5/31/2022 15:54	V File	6 KB

3.6 View Schematic Diagram of the Netlist after Synthesis

After completing the synthesis, you can view the schematic diagram of the entire design through the menu bar "Tools > Schematic Viewer > Post-Synthesis Netlist Viewer" to help you better understand the logic of the design after synthesis. For more details, see <u>SUG755-1.2.1E_Gowin</u> <u>HDL Schematic Viewer User Guide</u>.

3.7 Physical Constraints

After synthesis, you can use FloorPlanner or write manually to add physical constraints. In this design, FloorPlanner is selected. For more details, please refer to the <u>SUG935, Gowin Design Physical Constraints</u> <u>User Guide</u> and <u>SUG1018, Arora V Design Physical Constraints User</u> <u>Guide</u>.

3.7.1 Create New Physical Constraints

Click "Process > User Constraints > FloorPlanner" to open FloorPlanner, which supports I/O, Primitive, and Group physical constraints. This design only adds I/O constraints and uses it as an instance.

You can create I/O constraints in I/O Constraints window. Drag the port row to be constrained in the Netlist or I/O Constraints window to a specific location in the Package View or Chip Array view. After finished, the port location displays in the IOB, as shown in Figure 3-20.





After constraints finished, click "Save" to generate physical constraints files as shown in Figure 3-21.

Figure 3-21 Physical Constraints Display



In PnR, if there is no physical constraints file, the PnR will be automatically performed. If there is a physical constraint file, the PnR will be performed according to the physical constraints file.

3.7.2 Modify Physical Constraints

After physical constraints files generated, you can modify the constraints by FloorPlanner. Click "Save" to finish.

3.8 Timing Constraint

After synthesis, you can use Timing Constraints Editor or write manually to add timing constraints. In this design, Timing Constraints Editor is selected. For more details, please refer to <u>SUG940, Gowin Design</u> <u>Timing Constraints User Guide</u>.

3.8.1 Create New Timing Constraints

Click "Process > User Constraints > Timing Constrains Editor" to open Timing Constrains Editor, which supports clock, I/O and timing report constraints. This design adds clock and timing report constraints and uses them as instances.

Clock Constraints

Select "Clocks" under "Timing Constraints", right-click in the blank space on the right and select "Create Clock". This will open the "Create Clock" dialog, as shown in Figure 3-22. And create the following constraints:

- Clock name: clk
- Period: 20
- Frequency: 50
- Rising: 0
- Falling: 10
- Source Object: get_ports {clk}

Figure 3-22 Clock Constraints

🐳 Create Cl	ock					?	×
Clock name:	clk						
Waveform							
Period:	20	ns					
Frequency:	50	MHz					
Rising:	0	ns		-			
Falling:	10	ns	0	10		20	_
Objects: [g	et_ports {clk}]] Add
					OK	Cance	<u>1</u>

The design uses GAO, so the clock tck_pad_i is created in the same way as clk. The relationship between clk and tck_pad_i is an asynchronous

clock. If you do not want to use Gowin Software to analyze this relationship, you can create a clock group constraint through the timing constraint editor.

Timing Report Constraint

Select "Timing Constraints > Report > Report Timing", right-click in the blank space on the right and select "Create Report". In the popped-up "Report Timing" dialog, configure the parameters; the setup paths for clk to clk are reported, limiting the number of paths to 100, as shown in Figure 3-23.

Figure 3-23 Tir	ning Report	Constraint
-----------------	-------------	------------

🐳 Report Timing					? ×
Clocks From clock: ▼ clk To clock: ▼ clk					~ ~
Objects From: Through: To: Analysis Type Setup) Hold	Rec	overy	O Removal	
Path Max Paths: 100 Max Common Paths: Module Instance:	M:	in Logic Level: ax Logic Level:		OK	Cancel

After constraints is finished, click "Save" to generate timing constraints, as shown in Figure 3-24.

Figure 3-24 Timing Constraints Display



In PnR, if there is no timing constraints file, the PnR will be automatically performed. If there is a timing constraint file, the PnR will be performed according to the timing constraints file.

3.8.2 Modify Timing Constraints

After timing constraints files are generated, you can modify the constraints by Timing Constrains Editor. Click "Save" to finish.

3.9 GPA Configuration

After synthesis, you can create a GPA config file to analyze power. For the usage, please refer to <u>SUG282, Gowin Power Analyzer User Guide</u>.

3.9.1 Create GPA Config File

Select "Design > New File..." to open "New" dialog box, and select "GPA Config File" in "New", as shown in Figure 3-25. Click "OK". The file name is FIFO_HS and the file is under src by default. Then click "OK" to finish.

0		8		
₩	New		?	×
		GowinSynthesis Constraints File Jser Flash Initialization File GAO Config File		^
		GVIO Config File GPA Config File Memory Initialization File		
G	OWIN Pow	ver Analyzer Config File.		~
		ОК	Canc	el

Figure 3-25 Create GPA Config File

3.9.2 Configure GPA

After GPA config file is created, configure General Setting, Rate Setting and Clock Setting.

- General Setting includes the parameters of device, package, speed grade, temperature grade, thermal impedance, and voltage.
- Rate Setting is used to configure signal transition rate. You can set transition rate of IO or Net, or use the default value.
- Clock Setting is used to configure clock and enable features of BSRAM, I/O and DFF.

General Setting

In this design, the general setting is configured as follows: commercial

temperature, 25° C ambient temperature, no heat sink, VCCX 3.3V and VCC 1.2V, as shown in Figure 3-26.

Figure 3-26 General Setting Configuration

Invironment		
Junction Tem	iperature: 25.408°C	
Ambient Tempe	rature: 25.000°C	
Custom Thet	a JA: 25.000°C/W	
Heat Sink		
● None ○	Low Profile O Medium Profile O High Profile O Custom	
Air-flow:	0 ~ (LFM)	
Custom Theta	SA: 25.000°C/W ↓	
Board Therma	al Model	
None	○ Custom ○ Typical	
Board Tempe	rature: 25.000℃ 💂	
Custom Theta	JB: 25.000°C/W ♀	
Voltage		
VCC: 1 200V		
1.2000		

Rate Setting

In this design, the transition rate of clk is 50% and the remaining signals use the default value 12.5%, as shown in Figure 3-27.

		-			
General Setting	Rate Setting	Clock Setting			
Net Rate			VCD File		
۰ %	○ transition/s	🛨 🗙	Instance	File Name	File Type
Name	•	Value			
clk	50.00%	5			
			Filter glitch on V Default Rate Setting Default Rate urad f	CD file	₽ X
٢		>	Default Nate used Default Rate used Default Value: 12	I for remaining signals	
**	FIF	O_HS.gpa	×		

Figure 3-27 Rate Setting Configuration

Clock Setting

In this design, the clock is created in the timing analysis, and the rest are not set, as shown in Figure 3-28.

Figure 3-28 Clock Setting Configuration

Clock	Setting Clock Setting						10		
Global Enable: 100.00	0 ≑					🕂 🖌	Name	Out Enable	Load Capacity
Clock Name clk	Clock Enable	Quad1	Quad2	Quad3	Quad4				
B-SRAM									
Clock Enable: 100.00 Name Clock	Read Probability: 10	0.00 🗘 Write Proba obability WriteA	ability: 100.00 🗘 Probability Clock	<b enable="" p<="" readb="" td=""><td>robability WriteB P</td><td>robability</td><td>Name</td><td></td><td>Value</td>	robability WriteB P	robability	Name		Value
**	FI	FO_HS.gpa		×			1		

After configuration, click "Save" to finish and the design window is as shown in Figure 3-29.



Figure 3-29 GPA Config Files

In PnR, if there is no GPA config file, the PnR will be automatically performed. If there is a GPA config file, the PnR will be performed according to the GPA config file.

3.10 Place & Route

After synthesis and the creation of physical constraints files, timing constraints file, GPA config file as required, you can start PnR.

3.10.1 Configuration

Select "Process > Place & Route (right-click) > Configuration" to open "Configuration" dialog box to configure Place & Route and Bitstream. For the details, see <u>SUG100, Gowin Software User Guide</u>.

In this design, "Generate SDF File", "Generate Post-Place File" and "Generate Post-PNR Verilog Simulation Model File" in "General" option are configured to True. "Place output register to IOB" in "Plcae" option is configured to False, and the rest options use default values, as shown in Figure 3-30.

Place & Route	
Category: All ~ Label	Reset all to defau Value
Generate SDF File	True
Generate IBIS File	False
Generate Post-Place File	True
Generate Post-PnR Verilog Simulation Model File	True
Generate Post-PnR VHDL Simulation Model File	False
Generate Plain Text Timing Report	False
Promote Physical Constraint Warning to Error	True
Show All Warnings	False
Report Auto-Placed IO Information	False
	Place & Route Category: All Label Generate SDF File Generate IBIS File Generate Post-Place File Generate Post-PnR Verilog Simulation Model File Generate Post-PnR VHDL Simulation Model File Generate Plain Text Timing Report Promote Physical Constraint Warning to Error Show All Warnings Report Auto-Placed IO Information

Figure 3-30 Place & Route Configuration

3.10.2 Run PnR

After configuration, you can run PnR.

Double-click Place & Route in Process window to start PnR based on physical constraints and GAO configuration, start timing analysis based on timing constraints, and start power analysis based on power analysis configuration. After PnR, the icon before the Place & Route changes to "



After finishing PnR, the pnr folder is generated under the project creation path \impl, as shown in Figure 3-32. The folder contains all the

files generated in PnR, including the bitstream file, the netlist file after PnR, and the output reports. For the details, refer to <u>3.13 Output Files</u>.

Figure 3-32 PnR Directory

Name	Date modified	Туре	Size
🥁 ao_0.fs	5/31/2022 15:51	FS File	1,732 KB
📓 cmd.do	5/31/2022 15:51	DO File	1 KB
📓 device.cfg	5/31/2022 15:51	CFG File	1 KB
FIFO_HS.db	5/31/2022 15:51	Data Base File	43 KB
🛃 FIFO_HS.log	5/31/2022 15:51	LOG File	2 KB
🖻 FIFO_HS.pin.html	5/31/2022 15:51	360 se HTML Doc	35 KB
FIFO_HS.posp	5/31/2022 15:51	POSP File	1 KB
FIFO_HS.power.html	5/31/2022 15:51	360 se HTML Doc	8 KB
FIFO_HS.rpt.html	5/31/2022 15:51	360 se HTML Doc	40 KB
FIFO_HS.rpt.txt	5/31/2022 15:51	TXT File	29 KB
🛃 FIFO_HS.sdf	5/31/2022 15:51	SDF File	2,321 KB
FIFO_HS.timing_paths	5/31/2022 15:51	TIMING_PATHS File	32 KB
FIFO_HS.tr.html	5/31/2022 15:51	360 se HTML Doc	1 KB
FIFO_HS.vo	5/31/2022 15:51	VO File	561 KB
FIFO_HS_tr_cata.html	5/31/2022 15:51	360 se HTML Doc	8 KB
FIFO_HS_tr_content.html	5/31/2022 15:51	360 se HTML Doc	844 KB

3.11 Download Bitstream

Run Place & route to generate the bitstream file and download it with Programmer to verify the design. For the usage, please see <u>SUG502</u>, <u>Gowin Programmer User Guide</u>.

Select "Process > Program Device (double-click)" to open Programmer, and the programmer automatically identifies the bitstream file. After the development board is ready, click "Program/Configure" to download the bitstream to the development board. Figure 3-33 shows the completion of the bitstream download.

Figure 3-33 Programmer

👯 Go	win Programmer Version V	1.9.10 (64-bit) bu	ld 40618		-	_ ×
File E	dit Tools About					
ا 🔍	33119	m USB Cab	le Setting			
Enab	le Series	Device	Operation	FS File	User Code	IDCODE
1 🗹	GW1N	GW1N-9C	SRAM Program	E:/FIFO_HS/impl/pnr/ao_0.fs	0x0000C81E	1100481B
Output						5,
Info	Cable found: Gowin US	B Cable(FT2CH)/0	/17/null (USB location:17)			
Info	Cost 0.43 second(s)					
Info	Target Cable: Gowin US	B Cable(FT2CH)/0	/17/null@2.5MHz			
Info	Target Device: GW1N-9	C(0x1100481B)				
Info	Operation "SRAM Progr	ram" for device#1				
Info	User Code is: 0x0000C8	1E				
Info	Status Code is: 0x0003F	020				
Info	Finished.					
Info	Cost 3.47 second(s)					
Ready						

3.12 Debugging with GVIO and Data Acquisition with GAO

After the bitstream is downloaded, you can use GAO to verify the design or use GVIO to debug. For the GAO usage, see <u>SUG114, Gowin</u> <u>Analyzer Oscilloscope User Guide</u>; for the GVIO usage, see <u>SUG1189,</u> <u>Gowin Virtual Input Output User Guide</u>.

Click the GAO icon in the Gowin Software toolbar to open the GAO interface, which automatically identifies the .gao and .gvio config files, as shown in Figure 3-34.

🔲 Gowin Analyze	r Oscilloscope							-		×
Cable: Gow:	in USB Cable(FT2CH)	↓ ∨ Location:321	- D 🗘 🤇) 🗩 🔍 🔍	Q 🔍 🛶 o	∢ ⊳ ∈ ⊤ ±	±r +F Fe •	→	F Ľ	
Configuration										
Programmer										^
🗌 Enable Prog	rammer									
As Care	C 0									
Core 0	Capture									
	Storage Size: 204	18 Seg	ments Number: 1	∨ Captu	re Amount: 2048	∨ Trigg	er Position: 10	*		
	Trigger Expressio	ns								
	exp0: M0									
	Match Units									
	Match Unit	Trigger Port	Match Type	Function	Counter		Value			
	M0	Trigger 0	Basic w/edges	==	Disabled		R			J
									1	
Gvio Core										
	Gvio_0									
	¥ ¢ 🔍	Add R	emove Refresh	rate: 500	ms ∨ curres	nt rate : O se	amples/second			
	Туре		Name		Value		[Edge		
< >										

Figure 3-34 Jointly Debugging with GAO and GVIO

There are two "Start" buttons in the interface. The upper "Start" button controls the operation of GAO, and the lower "Start" button controls the operation of GVIO. GAO and GVIO can run simultaneously or independently. Take the simultaneous operation of GAO and GVIO as an example in the followings.

The gvio_test signal acts on the rst_n signal in the design, which is active high and is stimulated through GVIO. When gvio_test is low, it does not affect the FIFO HS design, as shown in Figure 3-35. When gvio_test is high, the design remains in a reset state. Click on "GAO > Configuration" in the Gowin Analyzer Oscilloscope interface, double-click "Match Units", modify the "Value" to X, and the captured waveform is as shown in Figure 3-36.

Click the "Start" icon in the GAO interface to capture data. After finishing capturing data, a window is generated to display the waveform. The window supports cursor, zoom-out and so on so as to facilitate you to analyze the data.





Figure 3-36 Jointly Debugging and Sampling with GAO and GVIO when gvio_test=1



3.13 Output Files

3.13.1 Place & Route Report

The Place & Route Report describes the resource, memory consumption, time consumption, etc. occupied by the user design, with the file extension name .rpt.html. Check the *.rpt.html file for further details.

Double-click "Place & Route Report" in the Process window to open Place & Route report, as shown in Figure 3-37.

For the details, refer to 6.2 Place & Route Report of <u>SUG100 Gowin</u> <u>Software User Guide</u>.



Process Ø ×			PnR Details	
Constraints Constrain	PonR Messages PonR Messages PonR Details Resource Resource Usage Summary IdoBat Clock Usage Summary Globat Clock Usage Summary Globat Clock Stogaes Pinout by Pont Name All Decknow Dise	Place & Route Process	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	200 2015 975 955 8 8 8
Timing Analysis Report	• All Package Pills	Total Time and Memory Usage	CPU time = 0h 0m 4s, Elapsed time = 0h 0m 4s, Peak memory usage = 311	IMB
Power Analysis Report				
		Resource Usage Sum	Resource	
		Resource Usage Sum	Resource mary: Usage	Utilization
		Resource Usage Sum	Resource mary: Usage 1000/840	Utilization
		Resource Usage Sum Resource Logic LUT,ALU,ROM16	Resource mary: 1000860 15001050	Utilization 19% -
		Resource Usage Sum Resource Logic LUT,ALU,ROM15 SSRAM(RAM16)	Resource mary: 14008540 1554(1205 LUT, 255 ALU, 0 ROM16) 6	Utilization 19% -
		Resource Usage Sum Resource Logic UT,ALU,ROM16 SSRAM(RAM16) Register	Descure 1000840 54(135 LUT, 259 ALU, 0 80016) 6 97/7843	Utilization 19% - - 15%
		Resource Usage Sum Resource Logic LUT,ALU,ROM16 SSRAH((AAN16) Register Logic Register as Latch	Resource Image 1000840 595(1305 UF, 259 AUL, 0 RDM16) 6 977(684) 1/4480	Utilization 19% 15% <15% <1%
		Resource Usage Sum Logic UTLAUROM16 SSRAH(RAM16) Register Logic Register as Latch Logic Register as FF	Resource usae 10001840 1564(1305 LUT, 259 AUL, 0 ROM15) 6 7/76443 L/4460 1/4460	Utilization 19% - 15% 25% <1%
		Resource Usage Sum Resource Logic LUT_LUU.ROM16 SSRAM(RAM16) Register Logic Register as Latch Logic Register as FF LUQ Register as Latch	Descurse 1000840 15001100 5541130 SUT, 359 ALU, 0 80M150 6 977/8440 264400 747440 7438	Utilization 19% - 25% 25% 25% 25% 25% 0%
		Resource Usage Sum Lopic USTAU(RAM15) SSRAH(RAM15) Register Logic Register as Latch Logic Register as FF I/O Register as FF	Resource tase 1000840 5564(1305 UL7 259 AUL, 0 90%16) 67/643 77/643 77/643 97/643 97/643 97/643 97/643 97/643 97/643 97/644	Utilization 39% 15% - 21% - 26% - 0% - 0% - 0% - 0% - 0% - 0% - 0% -

3.13.2 Ports and Pins Report

The Ports and Pins Report is the ports and pins files after placement. It includes port types, attributes, and locations, etc. The generated file is saved with extension name .pin.html. Check the .pin.html file for further details.

Double-click Ports & Pins Report in the Process window to open Ports & Pins Report, as shown in Figure 3-38.

For the details, refer to 6.3 Ports & Pins Report of <u>SUG100, Gowin</u> <u>Software User Guide</u>.

Figure 3-38 Ports & Pins Report



3.13.3 Timing Report

The timing report includes setup check, hold check, recovery time check, removal time check, min. clock pulse check, max. fan out path, Place & Route congestion report, etc. by default. The timing report also includes the max. frequency report.

Double-click Timing Analysis Report in the Process window to open the timing analysis report for the project, as shown in Figure 3-39.

For the details, please refer to <u>SUG940, Gowin Design Timing</u> <u>Constraints User Guide</u>.

Figure 3-39 Timing Report

Process Ø X	Timing Messages					Т	iming	Summ	aries	5		
🗸 📴 User Constraints	 Timing Summaries 	STA	Tool Run Summary:									
FloorPlanner	STA Tool Run Summary	Setur	Delay Model	Slow 1.14	/ 85C 0	6/15						
🕅 Timing Constraints Editor	Clock Summary	Hold	Delay Model	Fast 1.26	0C C6	/15						
🗸 🥝 Synthesize	Max Frequency Summary	Numb	ers of Paths Analyzed	2962								
Synthesis Report	Total Negative Slack Summary	Numb	ers of Endpoints Analyzed	2862								
Netlist File	Timing Details	Numb	ers of Falling Endpoints	4								
Y 🥝 Place & Route	Path Slacks Table	Numb	ers of Setup Violated Endpoints	0								
Place & Route Report	Setup Paths Table	Numt	ers of Hold Violated Endpoints	0								
Timing Analysis Report	Hold Paths Table	_										
Ports & Pins Report	Recovery Paths Table	Cloc	k Summary:									
Power Analysis Report	Removal Paths Table		Clush Manua		Town	Designal	Terrorent	Dian Dian	1 -1	Courses Mary		Objects
특ಡ Programmer	Minimum Pulse Width Table	NO.	Clock Name		Type	Period	Frequency()	(HZ) Rise	Fall	source Mas	elli.	Objects
	Timing Report By Analysis Type	A	tek and I		Dase	20.000	20.000	0.000	25.000		tak and I	
	Satur Analysis Report	4	aw and min last film avia ican b	00/054 4	Base	20,000	20.000	0.000	25.000		col_pao_r	t O/u quia icon tan/oE4 c1/E
	Hold Analysis Report	<u> </u>	gw_gao_gwo_inst_oro_gwo_icon_e	00/1104_4	base	20.000	30.000	0.000	10.000		9N_980_9N0_IIIS	Coto_0400_000_000/104_51/1
	Recovery Analysis Report	Max	Frequency Summary									
	Removal Analysis Report	N	O. Clock Name			Constra	aint		Actual Fm	аж	Logic Leve	d Entity
	Minimum Pulse Width Report	1	clk	50	000(MI	4z)		60.928(MH	z)		8	TOP
	High Fanout Nets Report	2	tck_pad_i	20	000(MI	Hz)		73.716(MH	z)		6	TOP
	Route Congestions Report											
	Timing Exceptions Report	Notin	ling paths to get frequency of gu	v_gao_gv	io_insi	_0/u_g	vio_icon_top	/n54_4:				
	Setup Analysis Report	Tota	I Negative Slack Sum	mary:								
	Hold Analysis Report		Clock Name				Analysis	ype		Endpoints TI	IS N	lumber of Endpoints
	Recovery Analysis Report	dk				Set	ID		0.000		0	
Design Process Hierarchy	💡 Start Page			Desi	n Sumr	nary					FIFO_HS.tr.ht	ml 🔳

3.13.4 Power Analysis Report

The Power Analysis Report helps you evaluate the basic power consumption of your design.

Double-click Power Analysis Report in the Process window to open the power analysis report as shown in Figure 3-40.

For the details, please refer to chapter 4 Power Analysis Report of SUG282, Gowin Power Analysis User Guide.

Figure 3-40 Power Analysis Report

Power Messages	Grade	Commercial	
 Configure Information 	Process	Typical	
Power Summary	Ambient Temperature	25.000	
 Power Information 	Use Custom Theta JA	false	
 Thermal Information 	Heat Sink	None	
 Supply Information 	Air Flow	LFM_0	
Power Details	Use Custom Theta SA	false	
 Power By Block Type 	Board Thermal Model	None	
 Power By Hierarchy 	Use Custom Theta JB	false	
Power By Clock Domain	Related Vcd File		
	Related Salf File		
	Filter Glitches	false	
	Default IO Toggle Rate	0.125	
	Default Remain Toggle Rate	0.125	
	Default Remain Toggle Rate Power Information:	o.125 Power Summary	
	Default Remain Topple Rate Power Information: Total Power (mW)	0.125 Power Summary 25.124	
	Default Remain Toggle Rate Power Information: Total Power (mW) Quiescent Power (mW)	0.125 Power Summary 25.124 9.907	
	Default Remain Topple Rate Power Information: Total Power (mW) Quiescent Power (mW) Dynamic Power (mW)	0.125 Power Summary 25.124 9.907 15.217	
	Default Remain Topgle Rate Power Information: Total Power (mW) Quiescent Power (mW) Dynamic Power (mW) Thermal Information:	0.123 Power Summary 25.124 9.907 15.217	
	Default Remain Topple Rate Power Information: Total Power (mW) Quiescent Power (mW) Dynamic Power (mW) Thermal Information: Junction Temperature	0.125 Power Summary 25.124 9.907 15.217 25.687	
	Default Remain Topple Rate Power Information: Total Power (mW) Quiescere Power (mW) Dynamic Power (mW) Thermal Information: Junction Temperature Theta JA	25.125 25.124 9.907 15.217 25.687 29.009	

3.14 File Encryption

3.14.1 Source File Encryption

When you need to encrypt and protect source files, you can encrypt the selected module and its sub modules in Hierarchy window, as shown in Figure 3-41. For details, see <u>SUG100, Gowin Software User Guide.</u>

Figure 3-41 Hierarchy Window

Hierarchy							8 ×
- + Update							
Unit	File	Register	LUT	ALU	BSRAM	SSRAM	MULTALU36X18
🗸 🎇 test_fifo	<pre>src\test_fifo.v</pre>	977(247)	1308(586)	242(184)	8(0)	6(0)	2(2)
rstn_gen(rstn_gen)	src\rstn_gen.v	17(17)	6(6)	15(15)	0 (0)	0 (0)	0 (0)
 FIFO_HS_Top(u_fifo_hs_top) ~fifo.FIFO_HS_Top(fifo_inst) 	src\FIFO_HS\FIFO_HS.v src\FIFO_HS\FIFO_HS.v	93(93)	102(102)	29(29)	2(2)	6(6)	0 (0)
Design Process Hierarchy							

Take module module test_fifo as an example to introduce the file encryption.

You can right-click test_fifo in the Hierarchy window and select "Pack User Design" in the right-click list to open the dialog box, as shown in Figure 3-42.

inguite o in i uck	boer Design Dialog Den				
🐳 Pack User Design	1			?	×
Create In:	E:\FIFO_HS\src\test_fifo_pack				
Synthesis Tool:	GowinSynthesis 🔻	Language:	Verilog		•
Target Top Module:	test_fifo				
Source Files					
			Add File	Remove	File
E:\FIFO_HS\src\tes E:\FIFO_HS\src\rst E:\FIFO_HS\src\FIF	st_fifo.v :n_gen.v [:] O_HS\FIFO_HS.v				
Output					
			Pack	Sto	р

Figure 3-42 Pack User Design Dialog Box

Select test_fifo as the top module. Click "Pack" to start encryption. The relevant information will be printed in the Output window.

After the encryption, two files are generated under the destination path (E:\FIFO_HS\src\test_fifo_pack): test_fifo_gowin.vp and test_fifo_sim.v.

- test_fifo_gowin.vp: Encrypted files that can be used by others.
- test_fifo_sim.v: Flattened synthesized plaintext netlist file that can be used for simulation.

3.14.2 Simulation File Encryption

The simulation file provided by Gowin is plaintext. In order to protect the simulation file, it can be encrypted by using a third-party simulation software, such as Modelsim and VCS, and the license of the tool needs to be obtained. Here it uses test_fifo_sim.v as an example to introduce the encryption.

Encryption by Modelsim

When using Modelsim, the steps to encrypt the simulation file are as follows:

1. Add macro `protect and `endprotect before and after the encrypted in the simulation file test_fifo_sim.v.

- 2. Run command: vlog +protect test_fifo_sim.v.
- 3. After running the command, test_fifo_sim.vp is generated in the work library, which is the encrypted file of test_fifo_sim.v that can be used for Modelsim simulation.

Encryption by VCS

When using VCS, the steps to encrypt the simulation file are as follows:

- 1. Add macro `protect128 and `endprotect128 before and after the encrypted in the simulation file test_fifo_sim.v.
- 2. Run command: vcs +v2k -protect128 test_fifo_sim.v.
- 3. After running the command, test_fifo_sim.vp is generated under the current path, which is the encrypted file of test_fifo_sim.v that can be used for VCS simulation.

4 Tcl

The previous chapters introduce the way to implement the entire design process by using GUI. Gowin Software also provides tcl commands for some settings. Take FIFO HS design in Windows as an example to introduce the usage of tcl commands. For the details, see Chapter 8 Tcl Commands of <u>SUG100, Gowin Software User Guide</u>.

4.1 Tcl Execution

4.1.1 Execution Using Tcl Editing Window

At the bottom of the Console page is the tcl editing window, where you can enter the tcl commands and press Enter to run, as shown in Figure 4-1.

Figure 4-1 Tcl Editing Window

Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.power.html" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.pin.html" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.rpt.html" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.rpt.txt" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.sdf" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.vo" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.tr.html" completed
Generate file "E:\FIFO_HS\impl\pnr\FIFO_HS.posp" completed
Fri May 06 09:20:08 2022
% run pnr
Console Message

4.1.2 Execution Using Tcl Command Line

Start command: \x.x\IDE\bin\gw_sh.exe [script file] under the installation directory

The First Way: enter gw_sh.exe to start. This mode executes in the same way as the Tcl editing window, executing tcl commands one by one, as shown in Figure 4-2.

Figure 4-2 Tcl Command Line Example

•	
*** GOWIN Tc1 Command Line Console ***	
% add_file -type verilog "E:/FIFO_HS/src/test_fifo.v"	
add new file: "E:/FIFO_HS/src/test_fifo.v"	
% add_file -type verilog "E:/FIFO_HS/src/FIFO_HS/FIFO_HS.v"	
add new file: "E:/FIFO_HS/src/FIFO_HS/FIFO_HS.v"	
δ	

The Second Way: use gw_sh.exe [script file] to execute the script file, shown in Figure 4-3. Tcl script file can contain all the supported tcl commands, such as, device, design file, option, and run information. Tcl script file can be generated by handwriting or saveto command, but saveto command The tcl script file can be generated by hand or by saveto command, but the saveto command does not include the run command when generating the tcl script, so you can add the run command if needed. For tcl script details, see <u>4.2 Tcl Quick Start</u>.

Figure 4-3 Tcl Script File Example

S C:\Gowin\Gowin_V1.9.10_x64\IDE\bin> .\gw_sh.exe E:\FIF0_HS\FIF0_HS.tc1 ** GOWIN Tc1 Command Line Console *** urrent working directory: E:/tc1/FIF1_HS_tc1 winSynthesis start mning parser halyzing Verilog file 'E:\tc1\FIF1_HS_tc1\src\FIF0_HS.v' halyzing Verilog file 'E:\tc1\FIF1_HS_tc1\src\test_fif0.v' halyzing Verilog file 'E:\tc1\FIF1_HS_tc1\src\test_fif0.v' halyzing Verilog file 'E:\tc1\FIF1_HS_tc1\src\test_fif0.v' halyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' halyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
** GOWIN Tcl Command Line Console *** mrent working directory: E:/tcl/FIFI_HS_tcl wningynthesis start halyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\FIF0_HS.v' halyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rstn_gen.v' halyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fifo.v' halyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' halyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
rrent working directory: E:/tcl/FIFI_HS_tcl winSynthesis start mning parser nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\FIF0_HS.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rest_fifo.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fifo.v' nalyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
owinSynthesis start mning parser nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\FIF0_HS.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rstn_gen.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fifo.v' nalyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
nning parser nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\FIF0_HS.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rstn_gen.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fif0.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\FIF0_HS, v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rtsman, gen.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fifo.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\rstn_gen.v' nalyzing Verilog file 'E:\tcl\FIFI_HS_tcl\src\test_fifo.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\G_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\G_A0_0\gw_ao_match.v'
nalyzing Verilog file 'E:\tcl\FIF_LB_tcl\src\test_fifo.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_crc32.v' nalyzing Verilog file 'C:\Gowin\Gowin_VI.9.10_x64\IDE\data\ipcores\GA0\GW_A0_0\gw_ao_match.v'
nalyzing Verilog file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GA0\G_A0_0\gw_ao_match.v'
nalyzing included file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GAO\GW_AO_0\gw_ao_define.v'("C:\Gowin\Gowin_V1.9.10
x64\IDE\data\ipcores\GAO\GW_AO_0\gw_ao_match.v":374)
ack to file 'C:\Gowin\Gowin_V1.9.10_x64\IDE\data\ipcores\GAO\G_AO_0\gw_ao_match.v'("C:\Gowin\Gowin_V1.9.10_x64\IDE\dat
<pre>\ipcores\GAO\GW_AO_0\gw_ao_match.v":374)</pre>
halyzing included file 'E:\tcl\FIFI_HS_tcl\impl\gwsynthesis\RTL_GAO\ao_0\gw_ao_top_define.v'("C:\Gowin\Gowin_V1.9.10_x6
\IDE\data\ipcores\GAO\GW_AO_0\gw_ao_match.v":374)

4.2 Tcl Quick Start

The use of Tcl commands in command line mode is the same as that in Tcl command editing window. Taking gw_sh.exe [script file] as an example, we will introduce the usage of Tcl commands. Using this Tcl script, a new project, FIFO_HS_tcl, is created based on the existing project FIFO_HS; then load the design file from FIFO_HS into the new project FIFO_HS_tcl (FIFO_HS.v is generated and loaded into the project by the command), configure the project, and then run the flow. The content of the Tcl script is described as follows:

#create project

create_project -name FIFO_HS_tcl -dir E:/tcl -pn GW1N-LV9LQ144C6/I5 -device_version C

#create FIFO HS IP

create_ipc -name fifo_hs -module_name FIFO_HS_Top -language Verilog -file_name FIFO_HS

#set the option value of specified options

set_property -dict {CONFIG.Write_Depth 1024 CONFIG.Write_Data_Width 32 CONFIG.Read_Depth 512 CONFIG.Read_Data_Num true CONFIG.Write_Data_Num true CONFIG.En_Reset true} [get_ips FIFO_HS_Top]

#generate IP file and add to project

generate target [get ips FIFO HS Top] #import design file to FIFO HS tcl/src import files -file "E:/FIFO HS/src/rstn gen.v" import files -file "E:/FIFO HS/src/test fifo.v" import_files -file "E:/FIFO_HS/src/FIFO_HS.cst" import files -file "E:/FIFO HS/src/FIFO HS.sdc" import files -file "E:/FIFO HS/src/FIFO HS.rao" import files -file "E:/FIFO HS/src/FIFO HS.gvio" import files -file "E:/FIFO HS/src/FIFO HS.gpa" #set output base name set option -output base name FIFO HS #set global frequency set option -global freq 50.000 #set synthesis tool set option -synthesis tool gowinsynthesis #set top module set option -top module test fifo #set verilog language set option -verilog std sysv2017 #set ram r/w check set option -rw check on ram 1 #set generate sdf file set option -gen sdf 1 #set generate post-place file set option -gen posp 1 #set generate post-pnr verilog simulation model file set option -gen verilog sim netlist 1 #set place output registers to IOB set_option -oreg_in_iob 0 #set run process run all

