Gowin FPGA

Design Guide

SUG113-1.1E, 7/26/2018

Disclaimer

Revision History

Date	Version	Description
7/26/2018	1.1E	Initial version published.

Contents

Contents i
1 About This Guide 1
1.1 Purpose 1
1.2 Supported Products 1
1.3 Related Documents 1
1.4 Abbreviations and Terminology 2
1.5 Support and Feedback 2
2 HDL Coding Guidelines 3
2.1 General HDL Coding Requirements 3
2.1.1 Coding for Hierarchical Synthesis 3
2.1.2 Pipeline Design Requirements 4
2.1.3 Comparing If-Then-Else and Case Statements 4
2.1.4 Avoiding Unintentional Latches 4
2.1.5 Global Reset and Local Reset 4
2.1.6 Clock Enable 4
2.1.7 Multiplexer 5
2.1.8 Bidirectional Buffers 5
2.1.9 Cross Clock Domains 5
2.1.10 HDL Coding for Distributed and BlockMemory 5
2.1.11 DSP Coding 6
2.1.12 Resource Sharing 6
2.2 Finite State Machine Guidelines 6
2.2.1 General Description 7
2.2.2 State Encoding Methods for StateMachines 7
2.2.3 Initialization and Default State for Safe State Machine 7
2.2.4 Full Case and Parallel Case Specification 7
2.3 Gowin FPGA Hardware Features 7
2.3.1 I/O Logic 7
2.3.2 DSP 8
2.3.3 Distributed and Block Memory 8
2.4 Lower Power 8

2.5 Coding to Avoid Simulation/Synthesis Mismatches 8
2.5.1 Sensitivity Lists 8
2.5.2 Blocking/Nonblocking Assignments 8
2.5.3 Signal Fan-Out 8
3 Design Planning 10
3.1 Design Planning in Gowin YunYuan Software 10
3.1.1 Constraints Flow in Gowin YunYuan Software 11
3.1.2 Design Planning Tool in Gowin YunYuan Software 13
3.2 Pins Distribution 14
3.2.1 Pins Distribution Constraints 14
3.2.2 Pins Distribution - Package View 15
3.2.3 Pins Distribution Rules 15
3.2.4 Pin Migration 16
3.2.5 Pins DRC 16
3.3 Clock Assignment 16
3.3.1 Clock Resources 16
3.3.2 Clock Resources Assignment Rules 16
3.3.3 Clock Resources Assignment Flow in Gowin YunYuan Software 17
3.4 Floor Planning 17
3.4.1 Floor Planning Definition 17
3.4.2 Floor Planning Constraints 17
3.4.3 Design Performance Enhancement Strategies 18
3.4.4 Special Floor Planning Considerations 18
4 Timing Closure 19
4.1 Timing Constraint 20
4.2 Potential Reasons for Timing Issues 21
4.2.1 Synthesis Report 21
4.2.2 Timing Report 22
4.2.3 Total Negative Slack 23
4.2.4 Minimum Pulse Width 23
4.2.5 High Fanout Nets 24
4.2.6 Route Congestions 24
4.2.7 Timing Constraints Report 24
4.2.8 Clock Latency 24
4.2.9 IO Delay 24
4.3 Timing Closure Steps 24
4.3.1 Front-end Timing Closure Strategies 24
4.3.2 Backend Timing Closure Strategies 26
4.4 The Influence of Gowin Devices Architecture on Timing 26

4.4.1 Block Memory Routing Differences 26
4.4.2 DSP Routing Differences 26
4.4.3 Different LUT Pins' Delays 26
4.4.4 The Enable Pin on CFU Registers 26

List of Figures

Figure 2-1 Gowin YunYuan Software IP Core Generator - Memory 6
Figure 2-2 Gowin YunYuan Software IP Core Generator - DSP 6
Figure 3-1 RTL View 11
Figure 3-2 Technology View 11
Figure 33 Physical Constraints Editor View 12
Figure 34 Package View 15
Figure 3-5 Global Clock Constraints 17
Figure 41 Timing Constraints Editor View 20
Figure 4-2 Timing Report 22

1 About This Guide

1.1 Purpose

This manual provides an overview of coding guidelines, design planning, and timing closure, which are vital when developing successful FPGA designs.

Coding style has a significant impact on how an FPGA design is implemented and ultimately how it performs. Although many popular synthesis tools have improved optimization algorithms, it is still necessary for designers to follow a certain coding style and employ the synthesis tools in a way that achieves the optimal result for a given FPGA architecture.

Design planning helps users target the design to the chosen FPGA device and balance the associated area and speed requirements in a manner that takes full advantage of the functionality of Gowin devices.

Timing closure can ensure that a user design meets a specific timing requirement. This chapter describes timing requirements, timing constraints, and timing optimization.

1.2 Supported Products

The information in this guide applies to the following products:

1. GW1N series of FPGA products: GW1N-1, GW1N-2, GW1N-4, GW1N-6, GW1N-9.
2. GW1NR series of FPGA products: GW1NR-4, GW1NR-9
3. GW2A series of FPGA products: GW2A-18 and GW2A-55;
4. GW2AR series of FPGA products: GW2AR-18

1.3 Related Documents

The latest user guides are available on GOWINSEMI Website. You can find the related documents at www.gowinsemi.com.cn.

Gowin YunYuan Software User Guide

1.4 Abbreviations and Terminology

The abbreviations and terminology used in this manual are outlined in the table below.

Abbreviations and Terminology	Name
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
FSM	Finite State Machine
DSP	Digital Signal Processing
DRC	Design Rule Check
RTL	Register Transfer Level

1.5 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 using the information provided below.

Website: www.gowinsemi.com.cn/

E-mail: support@gowinsemi.com

+Tel: +86 755 8262 0391

2HDL Coding Guidelines

2.1 General HDL Coding Requirements

2.1.1 Coding for Hierarchical Synthesis

Complex system designs require a hierarchical approach as opposed to the use of single modules. A hierarchical coded design can be synthesized as a whole or have each module synthesized separately. When the design is synthesized as a whole, it can be synthesized as either a flat module or multiple hierarchical modules.

Each methodology has its associated advantages and disadvantages. Hierarchical coding is preferable for complex system designs because designs in smaller blocks are easier to keep track of and reduce the design period by allowing the reuse of design modules. However, if the design mapping into the FPGA is not optimal across hierarchical boundaries, it can lead to higher device utilization and lower design performance. Users can overcome this disadvantage by utilizing an effective design hierarchy. Here are some tips for building hierarchical structures.

- The top level should only be used to call the submodule. It is better to implement control logic in submodules.
- Any I/O instantiations should be at the top level of the hierarchy.
- All input, output, or bidirectional pins instantiations should be at the top level of the hierarchy.
- The tristate statement for bidirectional pins should be written at the top module.
- Ensure that all the module output signals use registers. The advantages of doing this are as follows:
  - Combinational logic and registers are in one module, which helps to overcome the lack of no cross-module synthesis.
  - Place relative logics in one module. This facilitates resource sharing and optimizes key paths.
  - Split Irrelevant logics into different modules. You can use different optimization strategies, such as speed first or area first.

2.1.2 Pipeline Design Requirements

Pipeline design can improve design performance by restructuring a long data path into several levels of logic and breaking it up over multiple clock cycles. Pipeline structure is an effective way to improve data path speed; however, special care must be taken due to the additional data path latency.

2.1.3 Comparing If-Then-Else and Case Statements

The if-then-else statement generates priority-encoded logic, whereas the case statement implements balanced logic. If-then-else statements can contain a set of different expressions, but a case statement can contain only one expression. If-then-else statements and case statements are equivalent if the conditions are mutually exclusive.

2.1.4 Avoiding Unintentional Latches

FPGA users should avoid using latches. The synthesis tools can build them out with feedback loops, which will increase the design area and result in performance degradation and problems with static timing analysis by introducing combinatorial feedback loops.

Synthesis tools infer latches from incomplete conditional expressions; for example, an if-then-else statement without an else clause or a case statement without a default clause. To avoid unintentional latches, specify all conditions explicitly.

2.1.5 Global Reset and Local Reset

A global set/rest (GSR) network is built into Gowin devices. There is a direct connection to the core logic. It can be used as an asynchronous/synchronous set or asynchronous/synchronous reset. The registers in CFU and I/O can be individually configured to use the GSR. The global reset resource provides a convenient mechanism by which design components can be reset without using any general routing resources. There are two primary ways to take advantage of the GSR hardware resource in your design: Use the GSR to reset all components on your FPGA or use the reset with maximum fanout as the global reset to save routing resources. Local reset usually has smaller fan-out. It is recommended to use common routing as a reset signal.

2.1.6 Clock Enable

Gating clocks is not encouraged in FPGA designs because it can cause timing issues, such asunexpected clock skews. The CLS structure in Gowin devices contains dedicated clock enable signals. Users can use clock enable as a better alternative to achieve the same functions without worrying about timing issues. You should take the following into consideration when using the clock enable functionality of Gowin devices.

- Clock enable is only supported by flip-flops, not latches.
- Each CLS in one CFU share one clock enable signal.
- All flip-flops have a positive clock enable input.
- The clock-enable input has higher priority than the synchronous set/reset.

2.1.7 Multiplexer

The flexible configurations of LUTs within CLS can realize 2-to-1, 3-to-1, 4-to-1, or 5-to-1 multiplexers, etc. You can create larger multiplexers by programming multiple four-input LUTs.

2.1.8 Bidirectional Buffers

Using bidirectional buffers allows for fewer device pins, controlling output enable, and reducing power consumption. You can disable automatic I/O insertion in your synthesis tool and thenmanually instantiate the I/O pads for specific pins, as needed.

2.1.9 Cross Clock Domains

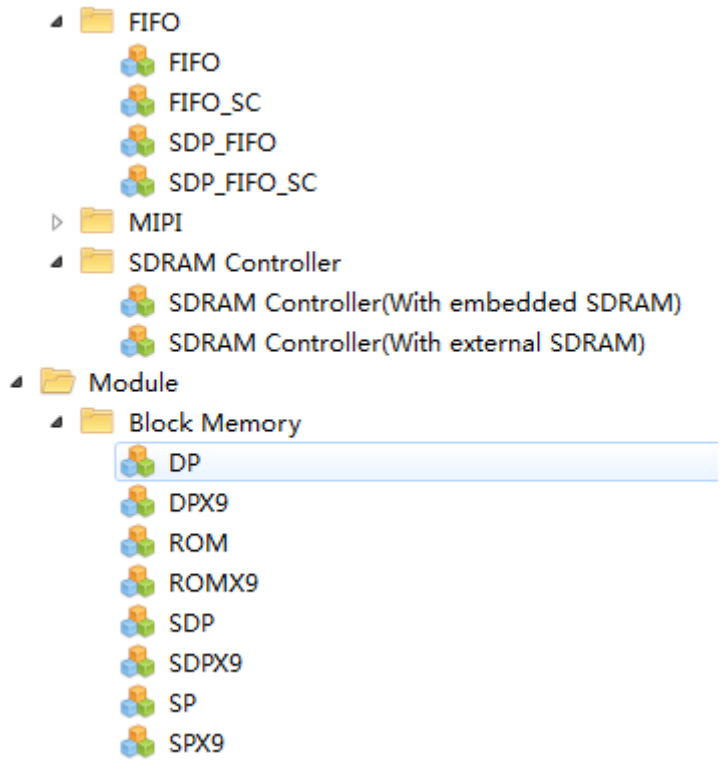
When passing data from one clock domain to another, special care must betaken to ensure that metastability issues do not arise as a result of set-up andhold timing violations. For single signals, it is suggested to use a double register structure to eliminate metastability. For data buses in different clock domains, asynchronous FIFOs are recommended.

2.1.10 HDL Coding for Distributed and BlockMemory

Although an RTL description is portable and the coding is straightforward, different coding may generate different synthesis results.

For Gowin devices, it's recommended to use the IP Core Generator in Gowin Yunyuan software to generate block memory and distributed memory. Gowin devices support multiple memory structures, including dual-port RAM, single port RAM, semi dual-port RAM, ROM, synchronous FIFO and asynchronous FIFO, as shown in Figure 2-1.

Figure 2-1 Gowin YunYuan Software IP Core Generator - Memory

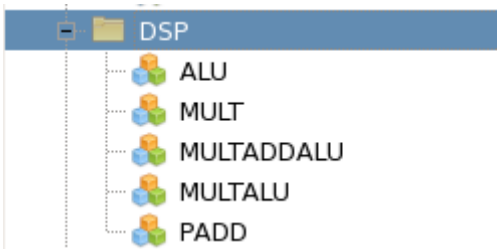


2.1.11 DSP Coding

Although an RTL description is portable and the coding is straightforward, different coding may generate different synthesis results.

For Gowin devices, it's recommended to use the IP Core Generator in Gowin Yunyuan software to generate DSP. Gowin devices support multiple DSP structures, including MULT, MULTALU, MULTADDALU, and PADD.

Figure 2-2 Gowin YunYuan Software IP Core Generator - DSP



2.1.12 Resource Sharing

Resource sharing is generally the default for most synthesis tools. It helps to conserve resources; however, it can create longer routes and add to routing congestion. If timing problems are caused, turn the global application of resource sharing off, and use attributes on specific modules /*synthesis syn_sharing = "on" */.

2.2 Finite State Machine Guidelines

A finite state machine advances from the current state to the next state

at the clock edge. This section discusses the methods and strategies for state machine encoding.

2.2.1 General Description

There are two ways to implement finite state machine. One is to process state-jump and state output simultaneously in one process; the other is to process state-jump and state output in two independent processes respectively. It is recommended to use the second way. It is easier to be read and modified, and it will not cause extra delay for outputting the state directly without registering.

2.2.2 State Encoding Methods for StateMachines

There are several ways to encode a state machine, including binary, one-hot encoding, and gray code. State machines with binary code and gray code have minimal numbers of flip-flops and wide combinatorial logics, whereas one-hot encoding is exactly the opposite.

The greatest advantage of one-hot encoding is that only one bit is required for state comparing. As a result, it decreases the decoding logic Although one-hot encoding uses more bits, i.e. more flip-flops for same states, its encoding circuit saves an equivalent area to offset the area consumed by flip-flops.

For small state machines, less than five states, binary code and gray code are typically the defaults. For larger state machines of five states or greater states, one-hot is the default.

2.2.3 Initialization and Default State for Safe State Machine

A state machine must be initialized to a valid state after power-up. You can initialize it during power-up or by including a reset operation to bring it to a known state. In the same manner, a state machine should have a default state to ensure that the state machine does not go into an invalid state. This could happen if all the possible combinations have been clearly defined in the design source code.

2.2.4 Full Case and Parallel Case Specification

RTL has attributes to define the default state. Full case attribute can be used to ensure no illegal state. Parallel case attributes can be used to ensure that all the statements in a case statement are mutually exclusive and only one case can be true at a time.

2.3 Gowin FPGA Hardware Features

2.3.1 I/O Logic

IO logic supports Deserializer, Serializer, delay control, and byte alignment, mainly used for high-speed data transmission. IO Logic supports basic mode, SDR mode, DDR mode, QDR mode, eightfold data

rate mode, and tenfold data rate mode, etc. You can use Gowin IO logic according to your design requirements.

2.3.2 DSP

Gowin DSP supports 9-bit and18-bit signed and unsigned multiplier, multiplication and accumulation, 54-bit ALU, barrel shifter, and pipeline and bypass registers.

2.3.3 Distributed and Block Memory

GOWINSEMI FPGA products provide abundant block static random access memories, B-SRAM for short. Data width and address depth can be configured for each B-SRAM.Each B-SRAM can be configured with up to 18Kbits. Each B-SRAM has two ports, port A and port B. Either port can be a read or write port. They have independent clocks, addresses, data, and control signals, but theyshare the same storage memory. Each B-SRAM has four operation modes: Single Port, Dual Port, Semi Dual Port, and ROM.

Distributed memory can be configured as a single port random memory with depth of 16 and width of 1/2/4 bits, semi dual port random memory with a depth of 16 and width of 1/2/4 bits, and 16 bits x 1 ROM16.

2.4 Lower Power

Optimize an area to reduce logic usage and routing lengths, and then to reduce power. It is recommended to use the IP Core Generator in Gowin YunYuan software to call the basic cells in Gowin devices for the most power-efficient (least area and resources) implementation. Eliminate known glitches for power reasons, reduce I/O toggle rate, and enter into the sleep state to reduce system power.

2.5 Coding to Avoid Simulation/Synthesis Mismatches

Certain coding styles can lead to synthesis result that differs from simulations. This is caused by error info. that are ignored by a simulator and can not be passed to the synthesis tool. This can usually be detected by running BKM check; as such, the Gowin coding style is recommended.

2.5.1 Sensitivity Lists

Sensitivity lists must contain all input and output signals to avoid mismatches between simulation and synthesized logic.

2.5.2 Blocking/Nonblocking Assignments

Use blocking assignments to generatecombinational logic; Use nonblocking assignments to generate sequential logic.

2.5.3 Signal Fan-Out

Signal fan-out control is designed to maintain reasonable

post-synthesis fan-outs. The synthesis tool reduces signal fan-out by duplicating circuits. Use syn_maxfan for specific signals to acquire better timing closure. Gowin FPGA device architectures are designed to handle high fan-out clock signal with dedicated clock and handle high fan-out reset signal with dedicated global reset network. However, synthesis tools tend to replicate logic. This type of logic replication occupies more resources in the devices and makes performance checking more difficult. Use syn_maxfan flexibly based on actual conditions to avoid the generation of many logic replication.

3 Design Planning

FPGA designs mainly include the following two phases:

1. Define the design functions and architecture, choose an FPGA device, and start writing the RTL code.
2. Target your design to the chosen FPGA device and fully utilize the chosen device.

Each of the two phases described above affects the other. This chapter focuses on the second phase and explains how you can fully utilize the functionality and features provided by Gowin devices.

3.1 Design Planning in Gowin YunYuan Software

Design planning is not mandatory for all designs, but it will be beneficial to most designs, especially in the case of medium or large designs that involve high resource utilization and/or a tight timing requirements. For these designs, design planning can help reduce potential placement and routing problems or timing issues, and it can increase the possibilities for design reuse and migration.

In Gowin YunYuan software, design planning starts when the synthesis has been completed successfully and before placement and routing. CST files contain all the design planning constraints required to guide backend placement and routing. If design planning is modified, i.e. CST files are modified, the design returns to the stage after synthesis before placement and routing. For a detailed overview of CST operation, please refer to the *Gowin Design and Constraint User Guide.*

3.1.1 Constraints Flow in Gowin YunYuan Software

Gowin YunYuan software allows users to set constraints for post-synthesis ports, netlist, registers, and instances. The CST file is the main input for defining design planning.

A VM file is generated after synthesis for user design. It includes all the design logic info.. Users can check the pre-synthesis and post-synthesis logic units by consulting the RTL View or Technology View in Synplify Pro, as shown in Figure 3-1 and Figure 3-2.

Figure 3-1 RTL View

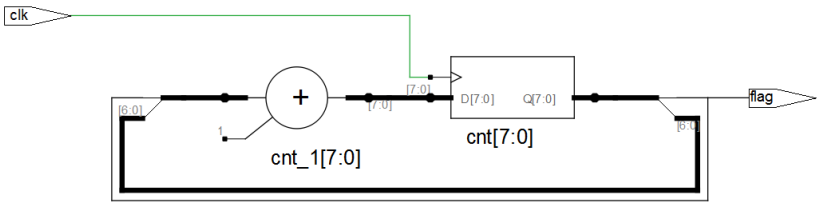
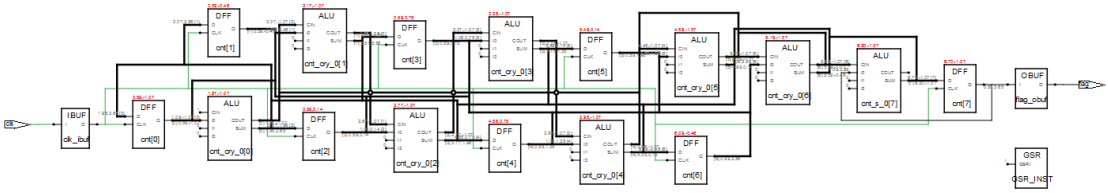
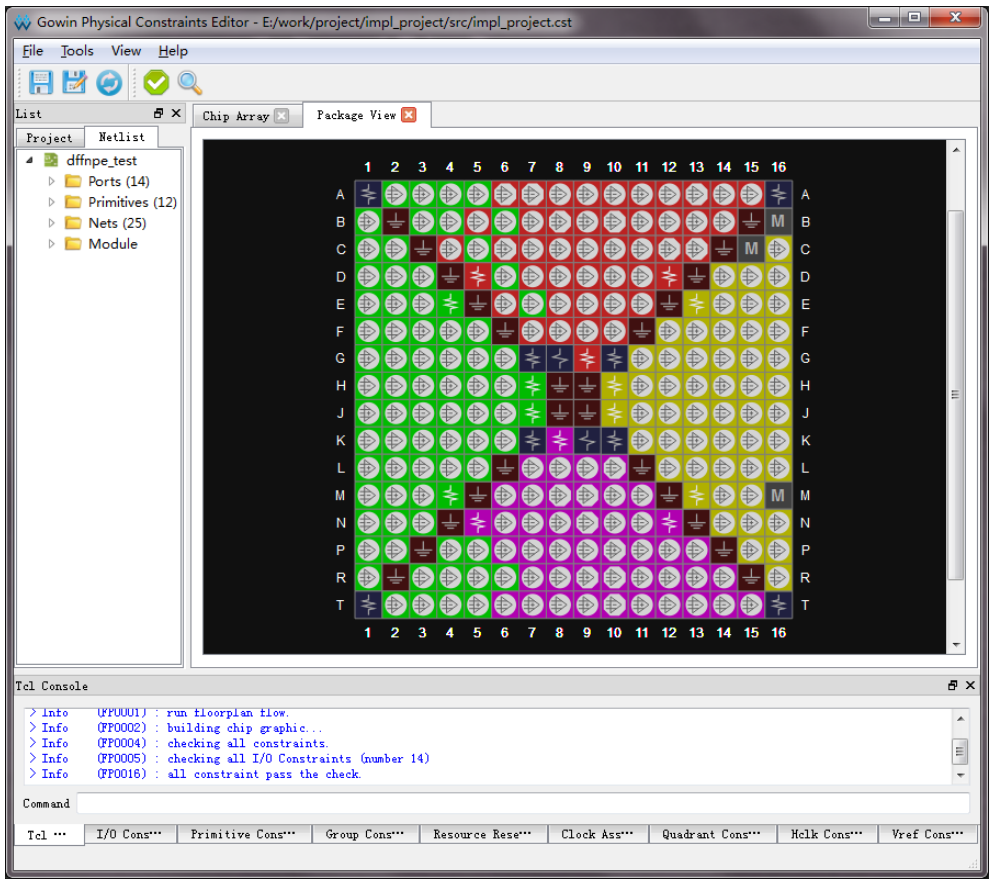


Figure 3-2 Technology View



Backend placement and routing software acquire users constraints by reading the CST file. If there is a syntax error or an illegal constraints error, the backend placement and routing software will exit directly. You can generate CST files by text editing or using the Physical Constraints Editor . Text editing is only recommended for advanced users; it is recommended to use the Physical Constraints Editor to generate CST file for error checking, as shown in Figure 33.

Figure 33 Physical Constraints Editor View



3.1.2 Design Planning Tool in Gowin YunYuan Software

Gowin YunYuan software includes a Physical Constraints Editor for design planning. The functions of this editor are as follows:

- View all the design elements that you can manage;
- View the hardware resources that are available for the chosen device;
- Assign specific design elements to specific FPGA resources;
- Identify any potential constraints problems through DRC check;
- Support drag and drop to define constraints.

3.2 Pins Distribution

Pins planning is the process of defining your FPGA I/O protocols and locations on the chosen device. It is based on your actual design and the chosen device.The pin planning process involves the following steps:

1. Assign your design ports to specific I/O locations.
2. Define the I/O protocol and otherI/O characteristics.
3. Check your assignments for legal usage.
4. Exchange your pin assignments with other parties according to PCB and circuit diagrams, if necessary.

3.2.1 Pins Distribution Constraints

The physical constraints editor that is integrated into Gowin Yunyuan software supports SystemIO attributes setting. The physical constraints editor is used to generate the .cst file. For further details, please refer to the information below.

- Lock the physical location for SystemIO:

  IO_LOC "xxx" H4 exclusive

- Set SystemIO standard:

  IO_PORT "xxx" IO_TYPE=LVCMOS18D

- Set driver strength for output pins or IO pins:

  IO_PORT "xxx" DRIVE=12

- Set pull up/down modes, such as UP, DOWN, KEEPER, and NONE:

  IO_PORT "xxx" PULL_MODE=DOWN

- Set reference voltage for SystemIO. The reference voltage can be from external pins or internal reference voltage generator:

  IO_PORT "xxx" VREF=VREF1_LOAD

- Set the amount of hysteresis for the input pins or bidirectional I/O pins. The value is NONE, H2L, L2H, HIGH from small to large in sequence:

  IO_PORT "xxx" HYSTERESIS=L2H

- Open drain is available for both output and bidirectional IO pins. The values are ON and OFF;

  IO_PORT "xxx" OPEN_DRAIN=ON

- Specify slew rate for output pins or bidirectional IO pins. SLOW: low noise model FAST: high speed mode:

  IO_PORT "xxx" SLEW_RATE=SLOW

- Set terminal matched resistance for single port signal. The values are OFF and 100 ohms:

  IO_PORT "xxx" SINGLE_RESISTOR=ON

- Set terminal matched resistance for the differential signal. The values

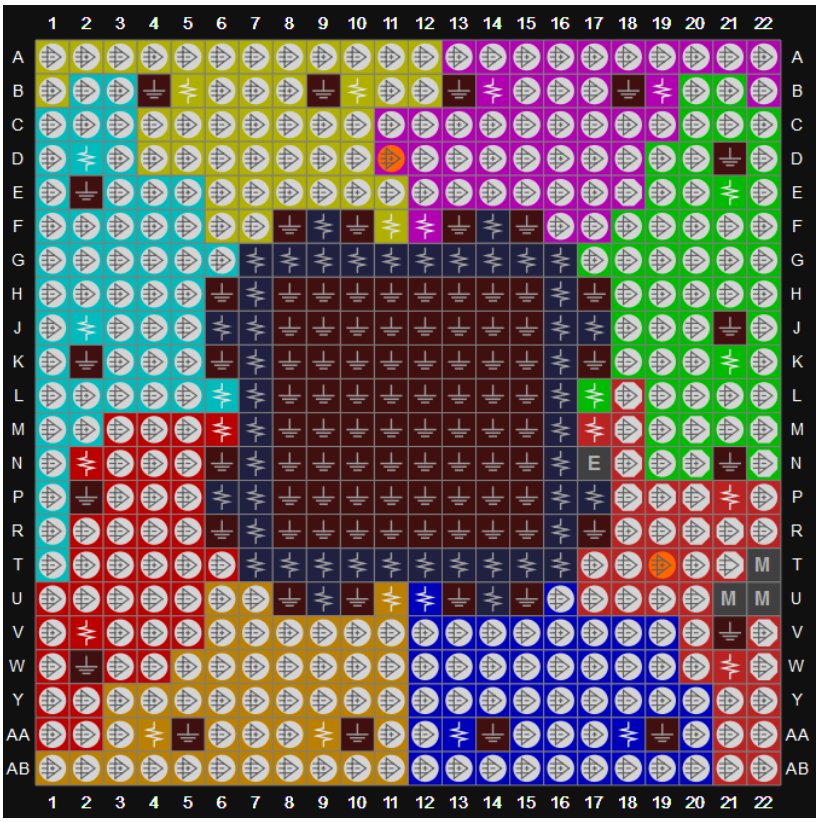
are OFF and 100 ohms:

IO_PORT "xxx" Diff_RESISTOR=ON

3.2.2 Pins Distribution - Package View

The FloorPlanne Package view is located on the right of the FloorPlanner and displays the locations of the pins in the selected package, as shown in Figure 34. The figure shows the available pin numbers, location info., and pin reuse info. When you hovers the mouse over a position, the I/O information of the location will be shown, including I/O type, bank, and LVDS. Different colors in the figure represent different bank, and different symbols have different meanings. You can refer to the detailed information by placing the mouse in the symbol position.

Figure 34 Package View



Note!

Right click to check enable ‘MODE’ pins on the right of the package view. The grey M for dedicated pins will become an icon for common pins.

3.2.3 Pins Distribution Rules

- Distribute dedicated pins first, such as clock input, phase-locked loop input, DDR, etc.
- Distribute the common pins instead of specific pins to the specific BANK. This helps the backend routing tools to achieve an optimum result.
- Check if duplicated pins conflict with the device programming modes.
- Run a DRC check, and find and correct the potential errors in time.

3.2.4 Pin Migration

For the device with the same package, users might want to migrate the design to to a device with a larger capacity for further function extension or to a device with a smaller capacity to reduce cost. The pin assignments should stay the same or changes should be minimal to avoid PCB redesign. Gowin YunYuan software provides a pin migration feature. You can view incompatible pins in the Physical Constraints Editor and FloorPlanner. Use LOC_RESERVE to disable these pins.

3.2.5 Pins DRC

Gowin YunYuan software provides I/O design rule checking (DRC), which enables you to validate I/O placements and attribute assignments. DRC works in two modes: Real-time check and on-demand check. DRC can catch many common errors and prevent incorrect pin usage at an early stage of the design process. The final Place&Route report gives the final pin assignment.

3.3 Clock Assignment

3.3.1 Clock Resources

The Gowin clock includes a global clock resource, high-speed clock resource, and quadrant clock resource.

- The dedicated global clock resource is distributed across the whole chip. It is also quadrant-based.
- High-speed clocks span through left, right, top or bottom side of a device. They are mainly designed for high-speed I/O interfaces with high fanout capability.
- A global clock can be split into two quadrant clocks (GW1N, GW1NR series) or four quadrant clocks of TOPLEFT, TOPRIGHT, BOTTOMLEFT, BOTTOMRIGHT (GW2A, GW2AR series). Quadrant clocks provide Dynamic Clock Selection (DCS) feature.

3.3.2 Clock Resources Assignment Rules

Clock frequency and clock fan-out are the main concerns when assigning clock resources assignment. The total number of available clock resources in the Gowin FPGA’s is also a deciding factor.

In general, dedicated resources give better timing results because of the minimized relative time delay. The routing resources saved also ease routing congestions in highly congested designs. In rare cases, you might use the general routing resources as a clock. Because general routing will increase the time delay, it should only be used in the designs that are characterized by low speed and low fanout. Here are the general rules for clock resource assignment:

- Determine the clock numbers and the fanout for each clock in your design.

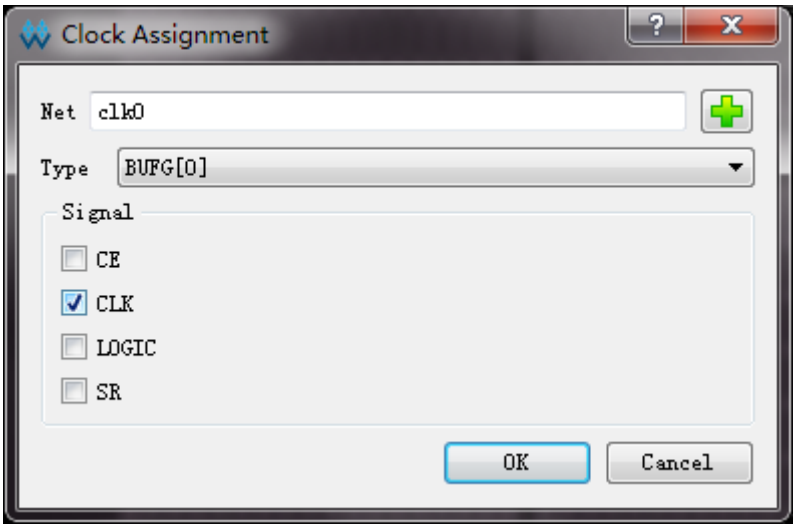
- Determine the clock resources provided by the target device.
- Determine the speed requirement for each clock.
- Assign the high-speed and high-fanout clock as the global clock.
- If the number of global clocks is less than the clock number in your design, use quadrant clocks.
- Use high-speed clock resources for high-speed interfaces with high fanout.

3.3.3 Clock Resources Assignment Flow in Gowin YunYuan Software

NET_LOC "xxx" BUFG0 = CLK | CE | SR | LOGIC

Assign the clock signal to the dedicated global clock network; BUFG0~BUFG7 is the eight global clock supported by Gowin devices. CLK | CE | SR | LOGIC indicates the constraint object. CLK is clock; CE is clock enable; SR is synchronous reset, and LOGIC is a logic device, as shown in Figure 3-5.

Figure 3-5 Global Clock Constraints



3.4 Floor Planning

3.4.1 Floor Planning Definition

Floor Planning is the logical partitioning of design elements, which results in a change in the design's physical placement or implementation. Floor Planning is accomplished by specifying FPGA location preferences.

For Gowin devices, floor Planning can improve the design performance, especially for the designs with good structures. Floor Planning provides a combination of automation and user control for design reuse and modular, hierarchical, and incremental design flows.

3.4.2 Floor Planning Constraints

INS_LOC "cnt[5]" R2C2

Specify the specific object to the specific CFU location.

GROUP hh = { "cnt_Z[1]" "cnt_Z[2]" "cnt_Z[3]" "cnt_Z[4]" "cnt_Z[5]" "cnt_Z[6]" "cnt_Z[7]" }

GRP_LOC hh R[3:6]C[4:6]

Group specific objects and specify them to the precise regional location.

3.4.3 Design Performance Enhancement Strategies

- Define regions based on design hierarchy.
- Define regions based on critical paths.
- Define regions based on input/output signals with high fanout.
- Define logic modules and optimize modules individually using different enhancement strategies.

3.4.4 Special Floor Planning Considerations

- Block RAM can be placed alone. Do not group block RAMs.
- Larger logic groups need starting position and relative size.
- Do not group carry chains and bus grouping.
- Do not group supplemental logic.

4 Timing Closure

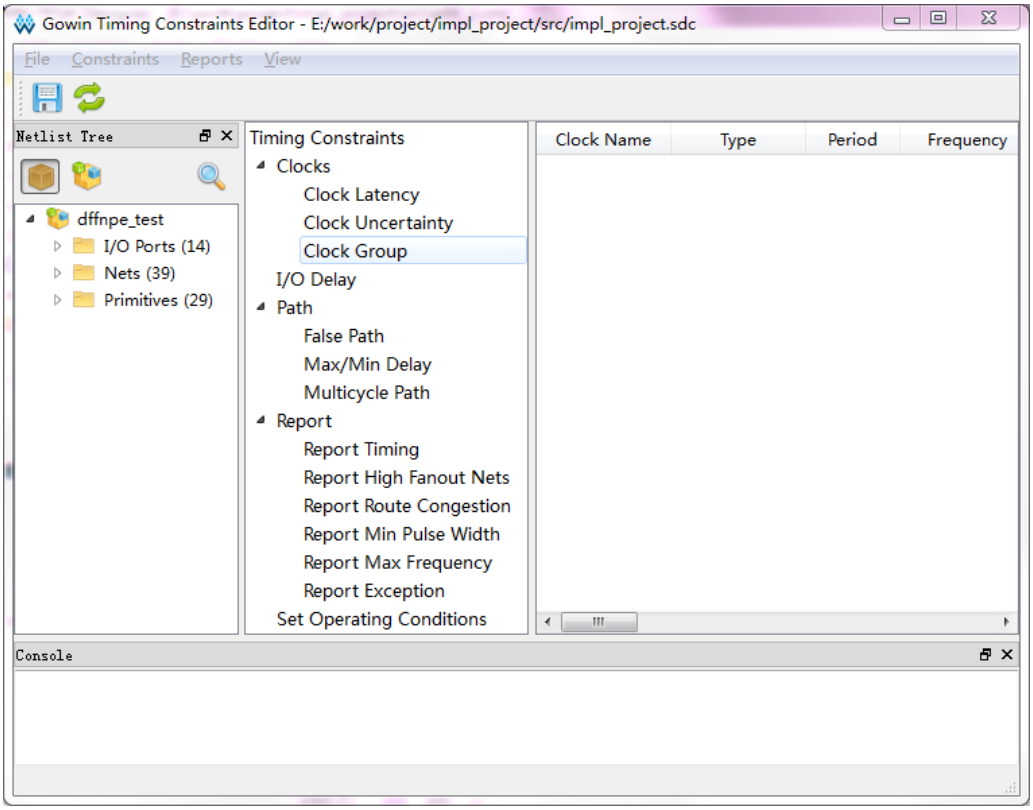
Every design has to run at a certain speed based on the design requirement. There are generally three types of speed requirement in an FPGA design: Timing, throughput, and latency. Throughput and latency are mutually exclusive. High throughput usually means more pipelining, which increases the latency, while low latency usually requires longer combinatorial paths, which removes pipelines and can reduce the throughput. Therefore, there is a requirement to balance throughput and latency as a priority.

Timing closure is usually the key factor that impacts the ability of a design to run at a required speed. It can be very challenging to close timing using various techniques. This chapter focuses on timing closure, and explains how to close timing in your design.

4.1 Timing Constraint

The Timing Constraints Editor in Gowin YunYuan software supports common timing constraints, as shown in Figure 41.

Figure 41 Timing Constraints Editor View



Note!

Please refer to the *Gowin Design Constraints Guide* for timing constraints syntax.

4.2 Potential Reasons for Timing Issues

4.2.1 Synthesis Report

If Synplify Pro runs at a timing driven mode, it uses timing constraints; If Synplify Pro runs at area driven mode, timing constraints will be ignored. You can set the "Frequency" attribute in Synplify Pro based on your requirements. In timing-driven mode, you need to set "Frequency"; the default value is AUTO. The default will be ignored when you use timing constraints in your design.

You can also improve your design performance using other synthesis attributes, such as resource sharing, pipelining and retiming.

Run your synthesis process with proper synthesis strategies and timing constraints. The Synthesis Report will be generated when synthesis process is finished. The report includes timing info.. You can search the key word START OF TIMING REPORT/END OF TIMING REPORT. Pay specific attention to the following:

- The summary at the top, which shows the appropriate SDC file and the default clock frequency.
- Performance summary, which gives the worst slack in the design:.
- Clock relationship, which shows register-to-register slacks. If your design includes multiple constrained clocks, they will all be included here.
- Interface information, which shows input setup and clock to output timing information and slacks..

Synplify Pro timing report is generated from the synthesis result, which does not have any placement and routing information. Therefore, to secure the highly accurate timing analysis result, run Place&Route first.

4.2.2 Timing Report

The Timing Analysis Report will be generated after performing timing analysis in Gowin YunYuan software. An example of a timing report is shown in Figure 4-2.

Figure 4-2 Timing Report

Timing Analysis Report

Setup Analysis:

Path1

Path Summary:

Slack:	8.355
Data Arrival Time:	3.363
Data Required Time:	11.718
From:	reg11_Z
To:	reg12_Z
Launch Clk:	sysclk:[R]
Latch Clk:	sysclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk(clock)
0.000	0.000	tCL	RR	1	IOL11[A]	clk1_ibuf/I
1.024	1.024	tINS	RR	2	IOL11[A]	clk1_ibuf/O
2.375	1.351	tNET	RR	1	R2C2[1][B]	reg11_Z/CLK
2.999	0.624	tC2Q	RR	1	R2C2[1][B]	reg11_Z/Q
3.363	0.364	tNET	RR	1	R2C2[1][A]	reg12_Z/D

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysclk(clock)
10.000	0.000	tCL	RR	1	IOL11[A]	clk1_ibuf/I
11.013	1.013	tINS	RR	2	IOL11[A]	clk1_ibuf/O
12.143	1.130	tNET	RR	1	R2C2[1][A]	reg12_Z/CLK
11.943	-0.200	tUnc				reg12_Z
11.718	-0.225	tSu		1	R2C2[1][A]	reg12_Z

Path Statistics:

Clock Skew:	-0.232
Setup Relationship:	10.000
Logic Level	0
Arrival Clock Path Delay:	cell: 1.024, 43.112%; route: 1.351, 56.888%
Arrival Data Path Delay:	cell: 0.000, 0.000%; route: 0.364, 36.874%; tC2Q: 0.624, 63.126%
Required Clock Path Delay:	cell: 1.013, 47.279%; route: 1.130, 52.721%

The first part of timing report shows the key path information.

- Prime Register: Refers to the register that sends data at one timing path;
- Post Register: Refers to the register that receives data at one timing path;
- Launch Edge: Refers to the clock edge at which the prime register sends data. This clock edge is also the time starting point for timing analysis.
- Latch Edge: Refers to the clock edge at which the post register receives data. This clock edge is also the time ending point for timing analysis.
- Slack: Refers to the time of the allowed Data Required Time minus the Data Arrival Time. It is the value of Data Required Time minus Data Arrival Time in this report. Note that the timing analysis tool can achieve the result with an accuracy within three decimal places of the

dollar. This might result in the third decimal place of the data that is displayed in the timing analysis report being inaccurate.

- Data Arrival Time: Refers to the time required from the beginning of the timing path to the end point.
- Data required time: Refers to the allowed time required for data arrival. The time is calculated from latch edge to latch edge arriving at post register.
- From: Refers to the start point of the timing analysis.
- To: Refers to the end point of timing analysis.
- Launch Clock: Refers to the clock that provides the launch edge.
- Latch Clock: Refers to the clock that provides latch edge.

Data Required Time minus Data Arrival Time is the timing closure slack. A positive result means timing closure while a negative result means no timing closure. The design needs to be modified to improve timing performance, or the constraints conditions need to be modified to reduce the frequency.

The second and third part of the clock report focuses on the Data Arrival Path and Data Required Path and lists the delay type, fanout, and location info, etc.

The fourth part of the clock report analysis the key path and also the corresponding statistics.

- Clock Skew: Refers to the time difference between the clock arriving at the prime register and the post register.
- Setup Relationship: Refers to the time relationship between the prime register sending data and the post register receiving data.
- Logic Level: Refers to the number of logic levels between two registers in the data arrival path; 0 refers to a direct connection.
- Arrival Clock Path Delay: Refers to the clock delay in the Data Arrival Path.
- Arrival Data Path Delay: Refers to the data delay in the Data Arrival Path.
- Required Clock Path Delay: Refers to the clock delay in the Data Required Path.
- Hold Check: Same as the Setup Check.

4.2.3 Total Negative Slack

Total Negative Slack is the statistics related to the negative slack at Setup Check in the timing analysis.

The number Of Negative Slack Paths is the total number of paths with no timing closure. The Total Negative Slack is the sum of Negative Slack at Setup Check.

4.2.4 Minimum Pulse Width

The minimum Pulse Width is the minimum pulse span of the timing components in the timing analysis path.

4.2.5 High Fanout Nets

High fanout nets analyze all the network fan-outs in the timing analysis path and also the worst slack and max. delay of the network.

4.2.6 Route Congestions

Route congestions analyze the routing congestion for each grid. The calculation involves dividing the used routing resources by the available routing resources.

4.2.7 Timing Constraints Report

The Timing Constraints Report describes all the clocks in the timing constraints, including the derived clock.

4.2.8 Clock Latency

Clock latency will affect setup time and hold time calculations.

Clock latency includes Clock Source Latency and Clock Network Latency.
Clock Source Latency refers to the transmission time between the clock source and the clock definition point. Clock Network Latency refers to the transmission time between the clock definition point and the register.

4.2.9 IO Delay

I/O Delay constraints contain FPGA ports delay information.

4.3 Timing Closure Steps

When using Gowin YunYuan software, you need to follow some general rules to get better timing closure.

- Obey Gowin RTL coding rules.
- Use proper constraints to guide synthesis and placement and routing.
- Compile the whole design with Gowin YunYuan software. If a synthesis timing error, this could be due to RTL coding issues, and it needs to be optimized.
- Check the synthesis report, PnR report, timing report, and Post PnR report to access the complete timing info.
- Use dedicated clock resources whenever possible, such as GCLK and HCLK.
- Analyze high fanout path, key path, and large delay path.
- Apply design planning constraints if necessary and repeat the steps above to meet timing demands.

4.3.1 Front-end Timing Closure Strategies

- Optimize the timing based on timing or area. Optimizing based on timing results has better performance at the expense of increasing

resource utilization. If the resource usage of your design is high, the extra resources required for timing optimization will affect performance by causing issues, such as, partial congestion and long routing, and result in timing issues. For low-speed design, the area should be the optimization base for synthesis.

- Add complete and reasonable timing constraints. The clock main frequency constraint is the basic and the most important constraint. If there are no frequency costraints, the system will use 200MHz as the default value. If your design has multiple clocks, you should add maxdeley or multicycle to process clock-domain signals; add set_input_delay or set_output_delay constraints for pins input and output.
- Read the synthesis timing report carefully and thoroughly. As the synthesis timing report does not contain placement and routing information, the results outlined in the report are typically better than the actual results. The actual result is usually 1/3 to 1/2 less than the report result.
- Check for Over Constraints. Sometimes you may over-constrain your design for better performance and broader margin. This may not work because Over Constraints will add resource consumption, which will affect timing.
- Use dedicated GSR resources. If your design contains high fanout reset and set signals, it is advisable that you use GSR resource to reduce routing congestion and enhance routing efficiency.
- Use Input/Output pins registers to improve pins timing. Input pins register can be used to improve input setup time; Output pins register can be used to improve clock to output time.
- Use the embedded programming delay unit to improve input hold time. Input pins register usage may result in hold time issues because of short data channel delay. Adjust the relationship between the data and clock using HCLK to meet both the setup time and hold time. In addition, Gowin devices have embedded programming delay cells. You can add a controllable time delay in the data path for reimbursed input hold time.
- Reduce fanout to improve the main frequency. It is suggested to use syn_maxfan selectively to reduce fanouts at the expense of register duplication.
- Use GCLK for clock enable whenever possible. Clock enable is usually a group of high fanout signals for driving a large number of registers. If regular routing resources are used, this may result in a huge delay. It's advisable that you use use GCLK reources.
- Use one-hot state encoding. For high-speed design, it's advisable to use one-hot state encoding. Also it will increase resource utilization rate and power consumption.
- Resource reuse. Resource reuse will increase logic levels and generate a huge delay path. The Synthesis tool usually reuses the non-key paths, but you also need to check the key paths to ensure that no timing issues are caused.
- Pipelining and retiming. This choice allows the synthesis tool to adjust the locations before and after registers to improve timing.

4.3.2 Backend Timing Closure Strategies

- Read the timing report generated after placement and routing. Pay special attention to the warnings and errors displayed in the tool, and also clock domain crossing analysis, main clock frequency, and logic progressions.
- Use multicycle or maxdelay to acquire lax timing;
- Design planning;
- Avoid under-constraints, such as no clock constraints, no pins constraints, shielding pins path, internally generated clocks, asynchronous logic, etc.
- Avoid over-constraints, such as the main frequency of the clock being higher than the actual value, unrecognizable multicycle path, and abundant constraints.

4.4 The Influence of Gowin Devices Architecture on Timing

Understanding the hardware details of Gowin devices should help you to fully utilize the hardware capability or avoid the unnecessary timing problems that result from improper use.

4.4.1 Block Memory Routing Differences

Routing from block memory to top CLS could be different to routing to bottom CLS.

4.4.2 DSP Routing Differences

TBD

4.4.3 Different LUT Pins' Delays

C and D inputs to a LUT are faster than the A and B inputs to the same LUT.

4.4.4 The Enable Pin on CFU Registers

The enable pin has a larger delay than the data pins.


GOWIN
PROGRAMMING FOR FUTURE