Gowin PCIe SGDMA IP

User Guide

IPUG1527-1.0E, 01/30/2026

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
01/30/2026	1.0E	Initial version published.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

Gowin PCIe SGDMA IP user guide aims to help users quickly learn the features, functions and usage of Gowin PCIe SGDMA IP. The software screenshots and the supported products listed in this manual are based on Gowin Software 1.9.12.01 (64-bit). As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

1.2 Related Documents

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

- DS981, GW5AT series of FPGA Products Data Sheet
- DS1103, GW5A series of FPGA Products Data Sheet
- DS1239, GW5AST series of FPGA Products Data Sheet
- DS1105, GW5AS series of FPGA Products Data Sheet
- DS1108, GW5AR series of FPGA Products Data Sheet
- DS1118, GW5ART series of FPGA Products Data Sheet
- IPUG1020, Gowin PCI Express Controller IP User Guide
- SUG100, Gowin Software User Guide

1.3 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
C2H	Card to Host (Data transfer from the Card to the Host)
Card	The executor of DMA transfers, which in this manual specifically refers to the FPGA design that integrates this IP
Descriptor	Data attributes generated by the Host and stored in memory

Terminology and Abbreviations	Meaning
DMA	Direct Memory Access
H2C	Host to Card (Data transfer from the Host to the Card)
Host	The initiator of DMA transfers, typically referring to a CPU or MCU system
IP	Intellectual Property
LUT	Look-up Table
Memory	A storage unit in a CPU or MCU system, generally referring to memory
Packet	In this manual, it specifically refers to a sequence of data transmitted or received through the AXI Stream interface, starting when the AXI VALID signal is asserted to 1 and ending when the LAST signal is asserted to 1

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 using the information provided below.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Overview

This document describes the functions and usage of Gowin PCIe SGDMA IP, which enables a PCIe card to directly access system memory. Gowin PCIe SGDMA IP provides users with a set of mechanisms and interfaces that allow data transfers between peripherals and system memory without CPU intervention.

Table 2-1 Gowin PCIe SGDMA IP Overview

Gowin PCIe SGDMA IP	
Logic Resource	See Table 3-1.
Delivered Doc.	
Design Files	Verilog (encryption)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software

Note!

For the devices supported, you can click here to get the information.

3 Features and Performance

3.1 Features

- Supports Scatter-Gather DMA (SGDMA).
- Supports up to one H2C channel and one C2H channel.
- Within a single channel, different streams can be configured with different tags, enabling multiple streams to be transmitted over one channel.
- The SGDMA controller uses BAR0.
- The user interface uses BAR2; the host can directly read and write user-defined registers via PCIe BAR2.
- Supports 64-bit addressing.
- Supports AXI-Stream user interface.
- Supports scatter-gather descriptor lists with unlimited list size.
- Supports channel status write-back mode.

3.2 Operating Frequency

The operating frequency of Gowin PCIe SGDMA IP is the same as that of Gowin PCI Express Controller IP. Users can flexibly configure the frequency according to data bandwidth requirements; typically, it is set to 100 MHz.

3.3 Resource Utilization

Table 3-1 shows the resource utilization. For the resource utilization of other devices, please refer to later release information.

Table 3-1 Resource Utilization

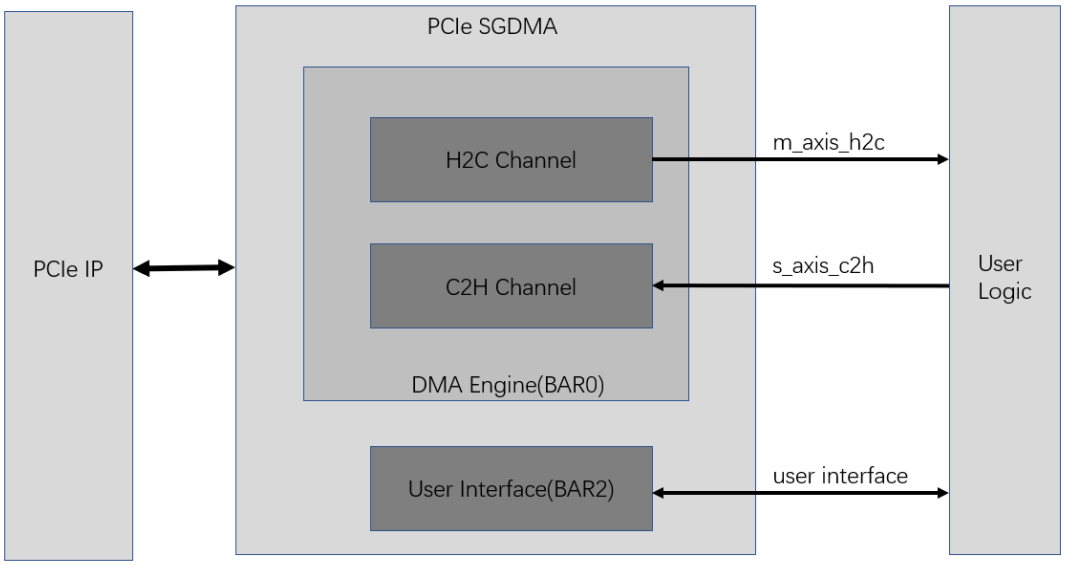
Option	LUTs	REGs	BSRAM	Device Series
1*H2C+1*C2H	7584	9710	32	GW5AT-60
1*H2C	3810	6230	16	GW5AT-60
1*C2H	5750	6623	16	GW5AT-60

4 Functional Description

4.1 Structure

The basic architecture of the Gowin PCIe SGDMA IP is shown in the Figure 4-1. It mainly consists of the H2C Channel, C2H Channel, and User Interface modules. The User Design in the Figure 4-1 refers to the User Design in the FPGA.

Figure 4-1 Gowin PCIe SGDMA IP Structure Diagram



- H2C Channel: Transfers data from memory to user logic.
- C2H Channel: Transfers data from user logic to memory.
- User Interface: Provides a direct interface to the user logic, allowing the host to access user logic through BAR2.

4.2 Functional Description

4.2.1 SGDMA Operation

SGDMA is used to transfer data between host memory and the card. To perform this operation, the host allocates buffer space in memory to create descriptors, which provide SGDMA with the attributes of the data to be transferred in memory. After SGDMA fetches the descriptors from memory, it transfers the corresponding data from host memory to the card, or transfers data from the card to the corresponding locations in host memory.

4.2.2 Stream Mode

Figure 4-2 illustrates the H2C operation flow in AXI Stream mode.

Figure 4-2 H2C AXI Stream Flow Chart

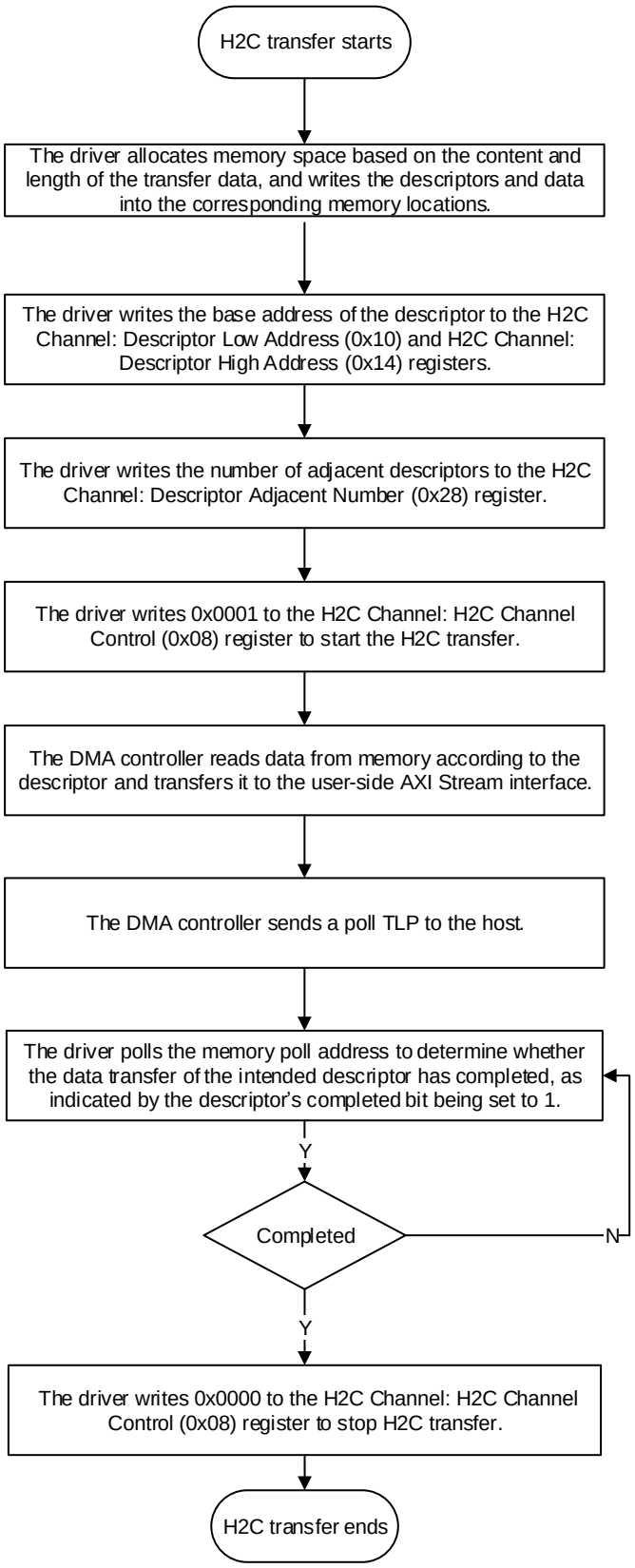
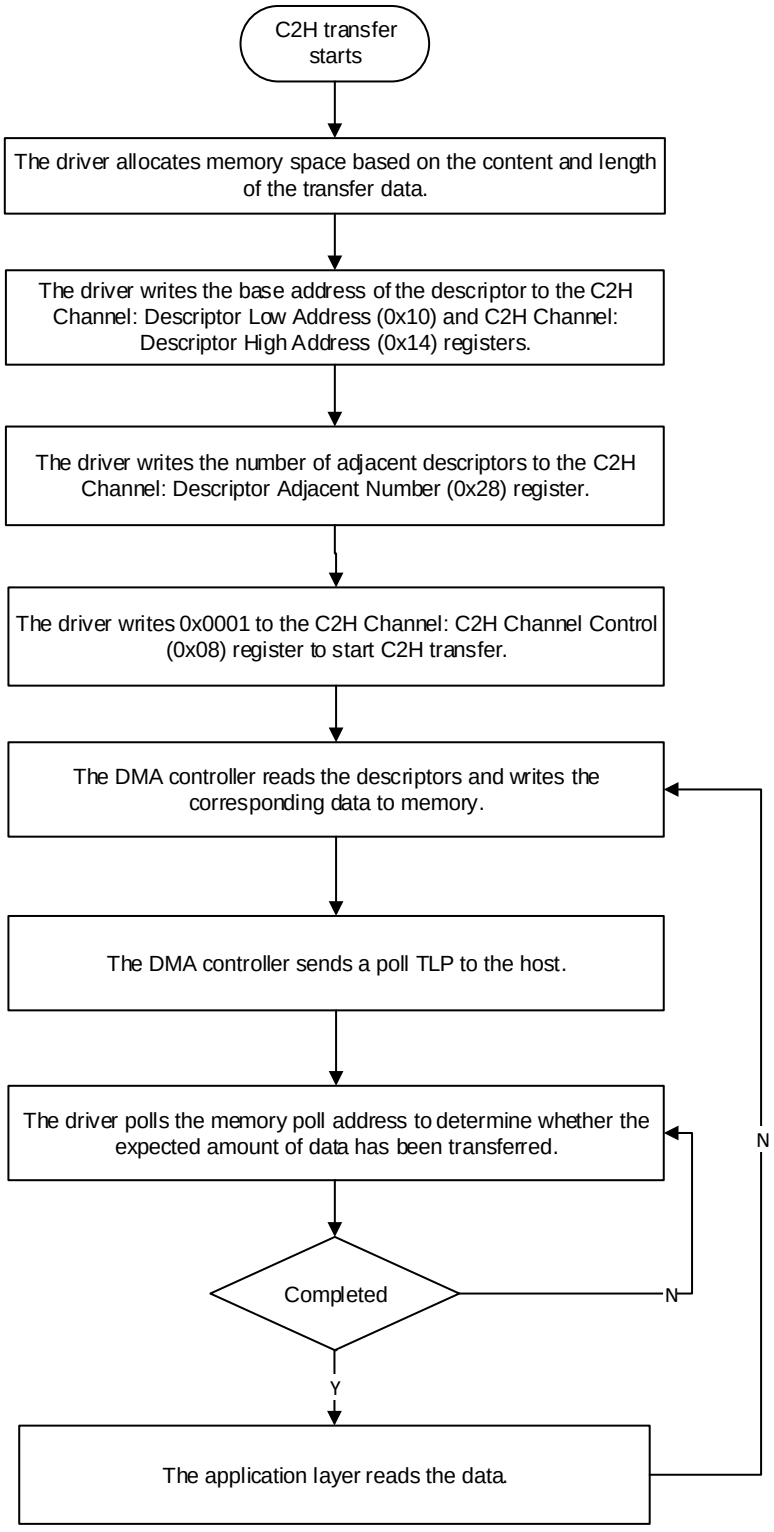


Figure 4-3 illustrates the C2H operation flow in AXI Stream mode.

Figure 4-3 C2H AXI Stream Flow Chart



4.2.3 Descriptors

Descriptors are used to describe the source, destination, and length of a DMA transfer. The descriptor list is created by the driver and stored in host memory. After the host starts a DMA transfer, the DMA engine begins reading the descriptors and performs memory read or write operations according to the descriptor contents.

Each descriptor is 32 bytes in size, and every descriptor must be aligned on a 32-byte boundary. A descriptor block is a chain composed of multiple contiguous descriptors. A descriptor block must not cross a 4 KB address boundary. Therefore, a single descriptor block can contain up to 4KB/32 = 128 descriptors. Multiple descriptor blocks form a descriptor chain. The Stop control bit of the last descriptor in the descriptor chain is set to 1, indicating the end of the descriptor chain.

The address of the initial descriptor block is configured through registers 0x10 and 0x14, and the length of the initial descriptor block is configured through register 0x28. Starting from the second descriptor block, its base address and length are determined by the last descriptor of the previous descriptor block. The descriptor format is shown in Table 4-1.

Table 4-1 Descriptor Format

Offset	Field	Bit	Description
0x00	Stop	0	Set to 1 if this descriptor is the last descriptor in the descriptor chain.
	Eop	1	Set to 1 if this descriptor is the last descriptor of a packet (used in Stream interface).
	Completed	2	Set to 1 to indicate that after this descriptor's data transfer is completed, the number of completed descriptors should be written back to the memory poll address.
	Adj Desc Num	14:8	Number of adjacent descriptors following this descriptor within the same descriptor block.
0x04	Length	31:0	Data length in memory described by the descriptor for DMA transfer (in bytes). For all descriptors except the last descriptor of a packet, the length must be a multiple of 4.
0x08	Src Low Addr	31:0	Lower 32 bits of the source address. In C2H Stream mode, this represents the lower 32 bits of the descriptor's write-back address.
0x0C	Src High Addr	31:0	High 32 bits of the source address. In C2H Stream mode, this represents the high 32 bits of the descriptor's write-back address.
0x10	Dst Low Addr	31:0	Lower 32 bits of the destination address. In H2C Stream mode, this represents the lower 32 bits of the overhead field.
0x14	Dst High Addr	31:0	High 32 bits of the destination address. In H2C Stream mode, this represents the high 32 bits of the overhead field.
0x18	Next Desc Low Addr	31:0	Lower 32 bits of the next descriptor address.
0x1C	Next Desc High Addr	31:0	High 32 bits of the next descriptor address.

4.2.4 Overhead Fields

Overhead fields refer to the extra data that DMA transfers between memory and the Card in addition to the normal payload data for each

packet. These fields allow users to define custom private fields for each packet. Users can choose whether to use overhead fields or not.

H2C Overhead

During H2C transfers, the DMA controller transmits the overhead fields from the descriptors together with the Stream data to the user side.

C2H Overhead

During C2H transfers, the DMA controller transfers the received user-side overhead fields to memory.

4.2.5 BAR Assignment

The PCIe SGDMA reserves BAR0 exclusively for DMA controller access and it cannot be used for other functions.

BAR2 is exposed to the user, allowing the Host to directly access user-defined registers through BAR2.

4.2.6 Poll Mode

The SGDMA supports Poll Mode. When Poll Mode is enabled, after a descriptor with the completed bit set to 1 finishes its transfer, the SGDMA writes the number of completed descriptors to the corresponding memory address.

4.3 Port Description

4.3.1 Clock

The SGDMA clock is provided through the IP clk pin and must be sourced from the same clock as PCIE_Controller_Top_pcie_tl_clk_i of Gowin PCI Express Controller IP.

4.3.2 Reset

The SGDMA reset is provided through the IP pcie_rstn pin and may share the same reset source as PCIE_Controller_Top_pcie_rstn_i of Gowin PCI Express Controller IP.

4.3.3 Connecting to Gowin PCI Express Controller IP

Gowin PCIe SGDMA IP does not include Gowin PCI Express Controller IP. When using this IP, users need to generate Gowin PCI Express Controller IP separately and connect it to the corresponding pins of the SGDMA IP to enable full functionality.

Users should connect Gowin PCIe SGDMA IP and the Gowin PCI Express Controller IP according to the mapping shown in the table below. For details on generating and instantiating Gowin PCI Express Controller IP, refer to *IPUG1020, Gowin PCI Express Controller IP User Guide*.

Table 4-2 Connection Mapping Between Gowin PCIe SGDMA IP and Gowin PCI Express Controller IP

Gowin PCIe SGDMA IP	PCI Express Controller IP
pcie_tl_rx_sop	PCIE_Controller_Top_pcie_tl_rx_sop_o

Gowin PCIe SGDMA IP	PCI Express Controller IP
pcie_tl_rx_eop	PCIE_Controller_Top_pcie_tl_rx_eop_o
pcie_tl_rx_data	PCIE_Controller_Top_pcie_tl_rx_data_o
pcie_tl_rx_valid	PCIE_Controller_Top_pcie_tl_rx_valid_o
pcie_tl_rx_bardec	PCIE_Controller_Top_pcie_tl_rx_bardec_o
pcie_tl_rx_err	PCIE_Controller_Top_pcie_tl_rx_err_o
pcie_tl_rx_wait	PCIE_Controller_Top_pcie_tl_rx_wait_i
pcie_tl_rx_masknp	PCIE_Controller_Top_pcie_tl_rx_masknp_i
pcie_tl_tx_sop	PCIE_Controller_Top_pcie_tl_tx_sop_i
pcie_tl_tx_eop	PCIE_Controller_Top_pcie_tl_tx_eop_i
pcie_tl_tx_data	PCIE_Controller_Top_pcie_tl_tx_data_i
pcie_tl_tx_valid	PCIE_Controller_Top_pcie_tl_tx_valid_i
pcie_tl_tx_wait	PCIE_Controller_Top_pcie_tl_tx_wait_o
pcie_tl_int_status	PCIE_Controller_Top_pcie_tl_int_status_i
pcie_tl_int_req	PCIE_Controller_Top_pcie_tl_int_req_i
pcie_tl_int_msinum	PCIE_Controller_Top_pcie_tl_int_msinum_i
pcie_tl_int_ack	PCIE_Controller_Top_pcie_tl_int_ack_o
pcie_tl_drp_clk	PCIE_Controller_Top_pcie_tl_drp_clk_o
pcie_tl_drp_addr	PCIE_Controller_Top_pcie_tl_drp_addr_i
pcie_tl_drp_wr	PCIE_Controller_Top_pcie_tl_drp_wr_i
pcie_tl_drp_wrdata	PCIE_Controller_Top_pcie_tl_drp_wrdata_i
pcie_tl_drp_strb	PCIE_Controller_Top_pcie_tl_drp_strb_i
pcie_tl_drp_rd	PCIE_Controller_Top_pcie_tl_drp_rd_i
pcie_tl_drp_ready	PCIE_Controller_Top_pcie_tl_drp_ready_o
pcie_tl_drp_rd_valid	PCIE_Controller_Top_pcie_tl_drp_rd_valid_o
pcie_tl_drp_rddata	PCIE_Controller_Top_pcie_tl_drp_rddata_o
pcie_tl_drp_resp	PCIE_Controller_Top_pcie_tl_drp_resp_o
pcie_ltssm	PCIE_Controller_Top_pcie_ltssm_o
pcie_linkup	PCIE_Controller_Top_pcie_linkup_o
pcie_tl_cfg_busdev	PCIE_Controller_Top_pcie_tl_cfg_busdev_o

4.3.4 DMA Controller Stream Interface

The Stream interfaces include the H2C Stream interface and the C2H Stream interface. During H2C transfers, the DMA controller reads data from system memory and outputs it to the user logic through the H2C Stream interface. During C2H transfers, the user logic sends data to the DMA controller through the C2H Stream interface, and the DMA controller then writes the data into system memory. If the overhead feature is enabled, the overhead data must be processed together with the Stream data.

H2C Stream Interface

When both m_axis_h2c_tvalid and m_axis_h2c_tready are asserted high, the data on m_axis_h2c_tdata is received by the user logic. When m_axis_h2c_tlast is asserted high, it indicates the last data beat of a

packet. m_axis_h2c_tkeep indicates the valid bytes within m_axis_h2c_tdata. The bits of m_axis_h2c_tkeep correspond one-to-one from least significant bit to most significant bit with the bytes of m_axis_h2c_tdata from the lowest byte to the highest byte. Except for the data beat where m_axis_h2c_tlast is asserted high, m_axis_h2c_tkeep is set to all ones.

C2H Stream Interface

When both s_axis_c2h_tvalid and s_axis_c2h_tready are asserted to 1, the data on s_axis_c2h_tdata is received by the DMA controller. When s_axis_c2h_tlast is asserted to 1, it indicates the last data beat of a packet. s_axis_c2h_tkeep indicates the valid bytes within s_axis_c2h_tdata. The bits of s_axis_c2h_tkeep correspond one-to-one from least significant bit to most significant bit with the bytes of s_axis_c2h_tdata from the lowest byte to the highest byte. Except for the data beat where s_axis_c2h_tlast is asserted to 1, s_axis_c2h_tkeep is set to all ones.

4.3.5 Overhead Interface

H2C

When m_axis_h2c_tvalid is asserted to 1, h2c_overhead is valid. Its value represents the overhead information contained in the current data descriptor and remains unchanged while m_axis_h2c_tvalid is continuously asserted as 1. The user can sample the h2c_overhead data at any point where m_axis_h2c_tvalid is asserted to 1.

C2H

When c2h_overhead_valid is asserted to 1, the c2h_overhead_data will be captured by the DMA controller and will affect the overhead for all packets transmitted after the cycle in which c2h_overhead_valid is asserted to 1. It is recommended to assert c2h_overhead_valid for one cycle during the inter-packet gap, while outputting the overhead value on c2h_overhead_data. This overhead value will apply to all subsequent packets until the next overhead update.

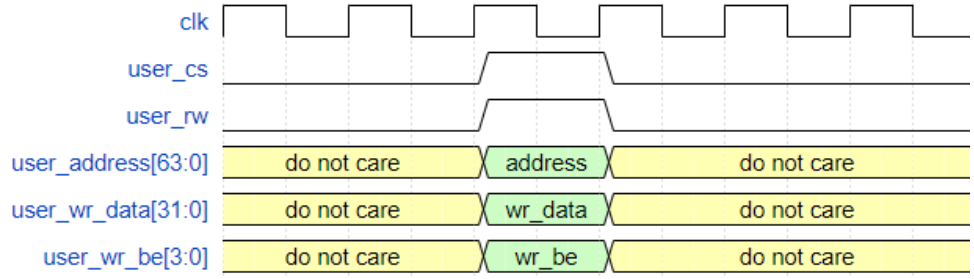
4.3.6 User BAR2 Interface

Gowin PCIe SGDMA IP reserves BAR2 as the user private register interface. The host can access user private registers by accessing the BAR2 address space. When the host accesses BAR2 addresses, the user can interact with the host through the interfaces of Gowin PCIe SGDMA IP with the prefix user_. Each BAR2 access initiated by the host supports a maximum TLP payload size of 4 bytes.

Write Operation

The figure below shows the write timing of the user BAR2 interface. When user_cs = 1 and user_rw = 1, it indicates that the host is performing a BAR2 write operation. The write address, write data, and write byte enable signals are valid in the same clock cycle.

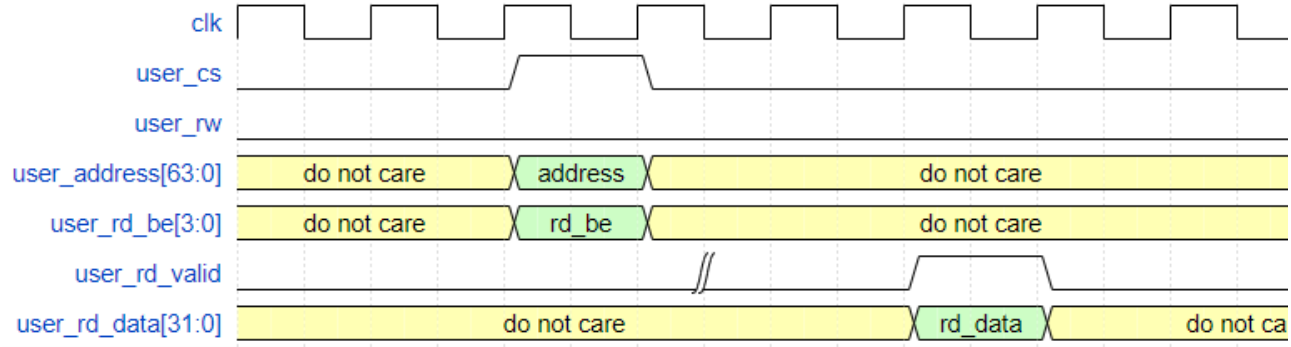
Figure 4-4 Write Timing of the User BAR2 Interface



Read Operation

The figure below shows the read timing of the user BAR2 interface. When user_cs = 1 and user_rw = 0, it indicates that the host is performing a BAR2 read operation. The read address and read byte enable signals are valid in the same clock cycle. After user_cs is asserted to 1, the user must assert user_rd_valid within several clock cycles and drive the return data on user_rd_data to respond to the host read request. Note that upon detecting a read operation, whether the operation is expected by the user or not (including zero-length reads), the user must respond with user_rd_valid=1. Otherwise, the controller will enter a state where it is blocked from receiving NP TLP, preventing the reception of subsequent NP TLPs.

Figure 4-5 Read Timing of the User BAR2 Interface



user_zero_read is a flag indicating a zero-length read from the host to BAR2. If the host initiates a zero-length read to BAR2, this signal is asserted when user_cs = 1.

user_address

Gowin PCIe SGDMA IP exposes the full 64-bit BAR2 address to the user logic. Users can select the required bit width for user_address based on the allocated BAR space size for BAR2 when instantiating the PCI Express Controller IP. For example, if the user configures the BAR2 Size in Gowin PCI Express Controller IP to 8K, only the lower 13 bits of user_address need to be considered, i.e., user_address[12:0]. The upper address bits correspond to the system-assigned BAR2 base address and do not need to be handled by the user.

4.3.7 Port List

The following table lists the I/O ports of Gowin PCIe SGDMA IP. Except for clk and pcie_rstn, all signals are synchronous to clk.

Table 4-3 Gowin PCIe SGDMA IP I/O Port List

Signal	I/O	Data Width	Description
Reset			
pcie_rstn	input	1	Reset signal input
Clock			
clk	input	1	Must be sourced from the same clock as PCIE_Controller_Top_pcie_tl_clk_i of Gowin PCI Express Controller IP.
Connect to PCI Express Controller IP			
pcie_tl_rx_sop	input	1	Refer to Section 4.3.3Connecting to Gowin PCI Express Controller IP for connection details.
pcie_tl_rx_eop	input	1	
pcie_tl_rx_data	input	256	
pcie_tl_rx_valid	input	8	
pcie_tl_rx_bardec	input	6	
pcie_tl_rx_err	input	8	
pcie_tl_rx_wait	output	1	
pcie_tl_rx_masknp	output	1	
pcie_tl_tx_sop	output	1	
pcie_tl_tx_eop	output	1	
pcie_tl_tx_data	output	256	
pcie_tl_tx_valid	output	8	
pcie_tl_tx_wait	input	1	
pcie_tl_int_status	output	1	
pcie_tl_int_req	output	1	
pcie_tl_int_msinum	output	5	
pcie_tl_int_ack	input	1	
pcie_tl_drp_clk	input	1	
pcie_tl_drp_addr	output	24	
pcie_tl_drp_wr	output	1	
pcie_tl_drp_wrdata	output	32	
pcie_tl_drp_strb	output	8	
pcie_tl_drp_rd	output	1	
pcie_tl_drp_ready	input	1	
pcie_tl_drp_rd_valid	input	1	
pcie_tl_drp_rddata	input	32	
pcie_tl_drp_resp	input	1	
pcie_ltssm	input	5	
pcie_linkup	input	1	
pcie_tl_cfg_busdev	input	13	

Signal	I/O	Data Width	Description
H2C Interface			
m_axis_h2c_tready	input	1	When the user is ready to receive data, assert this signal to logic 1. When both m_axis_h2c_tready and m_axis_h2c_tvalid are asserted to 1, the SGDMA determines that the data has been received by the user. When m_axis_h2c_tvalid is set to 1, if the user sets m_axis_h2c_tready to 0, m_axis_h2c_tvalid will remain at 1 until m_axis_h2c_tready returns to 1.
m_axis_h2c_tvalid	output	1	Asserted to logic 1 when the SGDMA outputs valid data.
m_axis_h2c_tdata	output	256	H2C data output from the SGDMA. Data is received by the user when both m_axis_h2c_tready and m_axis_h2c_tvalid are asserted to logic 1.
m_axis_h2c_tlast	output	1	Asserted to logic 1 by the SGDMA during the last cycle of a packet.
m_axis_h2c_tuser	output	32	Always driven to 0.
m_axis_h2c_tkeep	output	32	Byte-valid indicator. When the data corresponds to the last cycle of a packet, this signal indicates which bytes in the final cycle are valid. The bits of this signal, from the least significant bit to the most significant bit, correspond one-to-one to the bytes of m_axis_h2c_tdata from the lowest byte to the highest byte. The bits set to 1 in this signal must start at bit 0 and be contiguous. For non-last cycles, this signal must be asserted to all ones.
h2c_overhead	output	64	Overhead field output.
h2c_run	output	1	Transfer-in-progress indicator. When this signal is asserted to logic 1, it indicates that the host is initiating an H2C transfer. Otherwise, it is 0.
C2H Interface			
s_axis_c2h_tready	output	1	This signal is asserted to logic 1 when the SGDMA is ready to receive data. When both s_axis_c2h_tready and s_axis_c2h_tvalid are asserted to 1, the SGDMA receives s_axis_c2h_tdata. This signal is synchronous to clk.
s_axis_c2h_tvalid	input	1	The user asserts this signal to logic 1 when s_axis_c2h_tdata is valid.
s_axis_c2h_tlast	input	1	This signal is asserted to logic 1 by the user when the data corresponds to the last cycle of a packet.
s_axis_c2h_tdata	input	256	The user sends valid data. When both s_axis_c2h_tready and s_axis_c2h_tvalid are asserted to logic 1, the data is received by the SGDMA.
s_axis_c2h_tuser	input	32	Must be tied to 0.

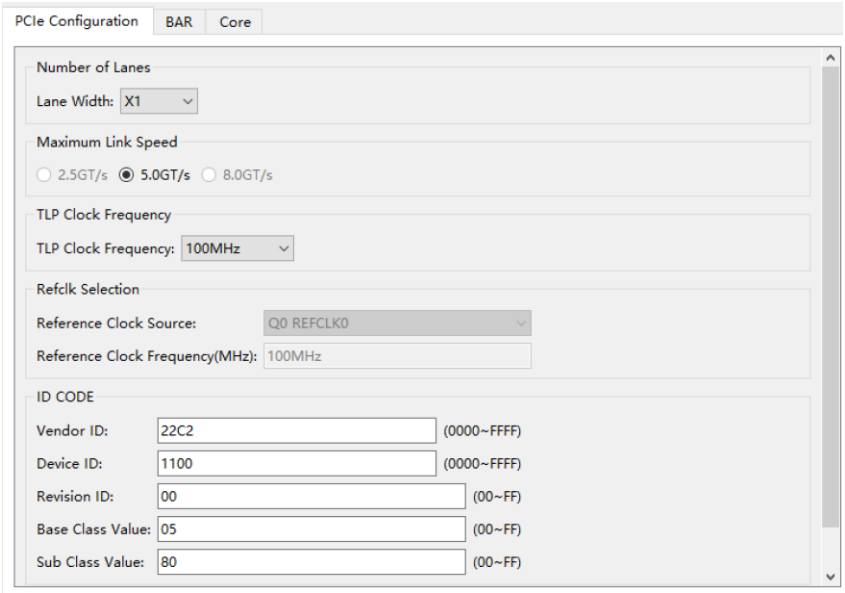
Signal	I/O	Data Width	Description
s_axis_c2h_tkeep	input	32	Byte-valid indicator. When the data corresponds to the last cycle of a packet, this signal indicates which bytes in the final cycle are valid. The bits of this signal, from the least significant bit to the most significant bit, correspond one-to-one to the bytes of s_axis_c2h_tdata from the lowest byte to the highest byte. The bits set to 1 in this signal must start at bit 0 and be contiguous. For non-last cycles, this signal must be asserted to all ones.
c2h_overhead_valid	input	1	Valid indication for c2h_overhead_data.
c2h_overhead_data	input	64	Overhead field input.
c2h_run	output	1	Transfer-in-progress indicator. When this signal is asserted to logic 1, it indicates that the host is initiating a C2H transfer. Otherwise, it is 0.
Bar2 Interface			
user_cs	output	1	User chip selected, active-high.
user_address	output	64	User address
user_rw	output	1	User read/write indicator ● 0: read operation; ● 1: write operation
user_wr_data	output	32	User write data
user_wr_be	output	4	User write byte enable
user_rd_be	output	4	User read byte enable
user_rd_valid	input	1	User read data valid
user_rd_data	input	32	User read data
user_zero_read	output	1	User zero-length read indicator, active high.

5 Gowin PCI Express Controller IP Configuration

Gowin PCIe SGDMA IP does not include Gowin PCI Express Controller IP. Users must generate Gowin PCI Express Controller IP according to the configuration described below, and connect the Gowin PCIe SGDMA IP to Gowin PCI Express Controller IP as described in Section 4.3.3 Connecting to Gowin PCI Express Controller IP.

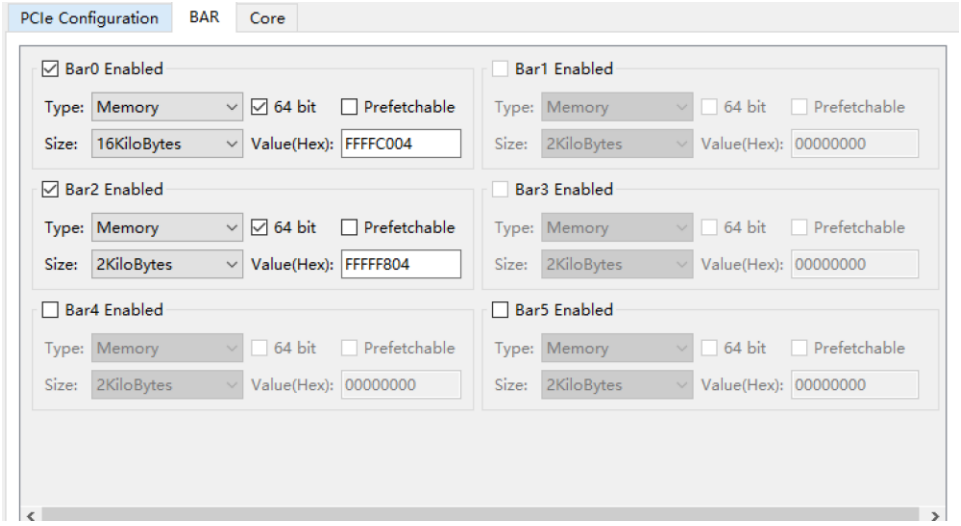
The figure below shows the PCIe Configuration page of Gowin PCI Express Controller IP. The options on this page can be configured by the user according to specific application requirements.

Figure 5-1 PCIe Configuration



The figure below shows the BAR configuration page of Gowin PCI Express Controller IP. BAR0 must be enabled and configured as 64 bits with a size of 16 KiloBytes. BAR2 must be enabled and configured as 64 bits, and its size can be configured according to user requirements. BAR4 does not need to be enabled.

Figure 5-2 BAR Configuration



The figure below shows the Core configuration page of Gowin PCI Express Controller IP. The "Max Payload Size" can be configured by the user according to system requirements; Gowin PCIe SGDMA IP supports a maximum payload size of 4096 bytes. The "Enable MSI Capability" option must be enabled.

Figure 5-3 Core Configuration

6 Register

6.1 Register Format

Gowin PCIe SGDMA IP uses a total of 16 KB BAR0 address space. The register allocation is shown in the table below.

Table 6-1 Register Allocation

Address Bits				Field
13	12	11~8	7~0	–
0	0	Channel ID	Byte offset	H2C register
0	1	Channel ID	Byte offset	C2H register
1	0	4’b0001	Byte offset	Control register

Note!

The Channel ID supports only 4'b0000.

6.2 Register Definition

The access types are defined as follows:

- RO: Read-only
- RW: Read/Write
- W1S: Write 1 to Set
- W1C: Write 1 to Clear
- RW1C: Read, Write 1 to Clear
- RC: Read to Clear
- WO: Write Only

6.2.1 H2C/C2H Register

Channel Identifier(0x00)

Table 6-2 Channel Identifier(0x00) Register

Bit Index	Access	Description
31: 24	RO	H2C/C2H register version. Current version is 0.
23: 20	RO	Reserved, 0.
19: 16	RO	0000: H2C; 0001: C2H

Bit Index	Access	Description
15: 12	RO	Reserved, 0.
11: 8	RO	Channel ID
7: 1	RO	Reserved, 0.
0	RO	User interface mode: Current version is 1: Stream interface

H2C Channel Control(0x04)

Table 6-3 H2C Channel Control(0x04) Register

Bit Index	Access	Description
31: 5	RO	Reserved
4: 2	RW	Indicates whether the corresponding status bit at the same bit position in the Status (0x30) register is enabled If enabled, the corresponding bit in the status register will be set to 1 when the relevant event occurs. 1: Enable 0: Disable
1	RW	poll mode 1: Enable 0: Disable
0	RW	Start H2C/C2H transfer. Write 1 to start the transfer; Write 0 to stop the transfer.

Channel Control(0x08)

Table 6-4 Channel Control(0x08) Register

Bit Index	Access	Description
31: 0	W1S	Same definition as register 0x04

Channel Control(0x0C)

Table 6-5 Channel Control(0x0C) Register

Bit Index	Access	Description
31: 0	W1C	Same definition as register 0x04

Descriptor Low Address(0x10)

Table 6-6 Descriptor Low Address(0x10) Register

Bit Index	Access	Description
31: 0	RW	Descriptor low 32-bit address

Descriptor High Address(0x14)

Table 6-7 Descriptor High Address(0x14)寄存器

Bit Index	Access	Description
31: 0	RW	Descriptor high 32-bit address

Poll Low Address(0x18)

Table 6-8 Poll Low Address(0x18) Register

Bit Index	Access	Description
31: 0	RW	Poll mode low 32-bit address

Poll High Address(0x1C)

Table 6-9 Poll High Address(0x1C) Register

Bit Index	Access	Description
31: 0	RW	Poll mode high 32-bit address

Completed Descriptor Count (0x20)

Table 6-10 Completed Descriptor Count (0x20) Register

Bit Index	Access	Description
31: 0	RO	Number of completed descriptors

Descriptor Adjacent Number(0x28)

Table 6-11 Descriptor Adjacent Number(0x28) Register

Bit Index	Access	Description
6: 0	RW	Number of adjacent descriptors following the first descriptor

Status (0x30)

Table 6-12 Status (0x30) Register

Bit Index	Access	Description
31: 5	RW1C	Reserved, 0.
4	RW1C	Set to 1 when a descriptor transfer is completed.
3	RW1C	Set to 1 when the transfer of a descriptor with completed = 1 finishes.
2	RW1C	Set to 1 when the data transfer associated with a descriptor with stop = 1 finishes.
1	RO	Poll mode enable status, synchronized with bit 1 of the Control register.
0	RO	DMA busy status, synchronized with bit 0 of the Control register.

Status (0x34)

Table 6-13 Status (0x34) Register

Bit Index	Access	Description
4: 2	RC	Same definition as register 0x30

Credit (0x4C)

Table 6-14 Credit (0x4C) Register

Bit Index	Access	Description
9: 0	RW	Credit value register, valid only for C2H. Writing this register allows the host to release a corresponding number of credits to the controller. Reading this register allows the host to obtain the remaining credits available in the controller.

6.2.2 Control register

Control Identifier(0x00)

Table 6-15 Control Identifier(0x00) Register

Bit Index	Access	Description
31: 24	RO	Control register version; the current version is 0.
23: 20	RO	Reserved, 0.
19: 16	RO	0011 indicates Control.
15: 12	RO	Reserved, 0.
11: 8	RO	Reserved, 0.
7: 0	RO	Reserved, 0.

Initial Control (0x04)

Table 6-16 Initial Control (0x04) Register

Bit Index	Access	Description
31: 1	RO	Reserved, 0.
0	WO	Writing 1 enables the SGDMA to read the PCIe hard IP configuration space. After the PCIe link is successfully established, the driver must first write this register to notify the SGDMA to read the required information from the PCIe hard IP configuration space. The driver then reads the Initial Status (0x08) register to determine whether the SGDMA has completed the read operation.

Initial Status(0x08)

Table 6-17 Initial Status(0x08) Register

Bit Index	Access	Description
31: 0	RO	0x55ff21b8: Reading of the PCIe hard IP configuration space is not yet complete. 0xaa009719: Reading of the PCIe hard IP configuration space has been completed.

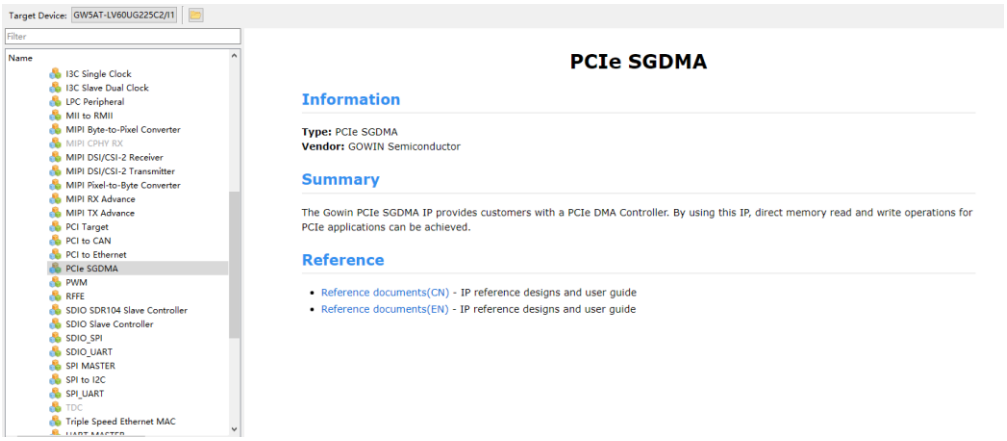
7 Interface Configuration

You can invoke and configure Gowin PCIe SGDMA IP using the IP Core Generator tool in the IDE. This section will introduce the configuration interface, the configuration process, and the configuration options.

1. Open IP Core Generator.

After the user creates a project, click the “Tools” tab in the upper-left corner, then select “IP Core Generator” from the dropdown menu to open Gowin’s IP Core Generation tool. From there, choose "PCIe SGDMA", as shown in Figure 7-1.

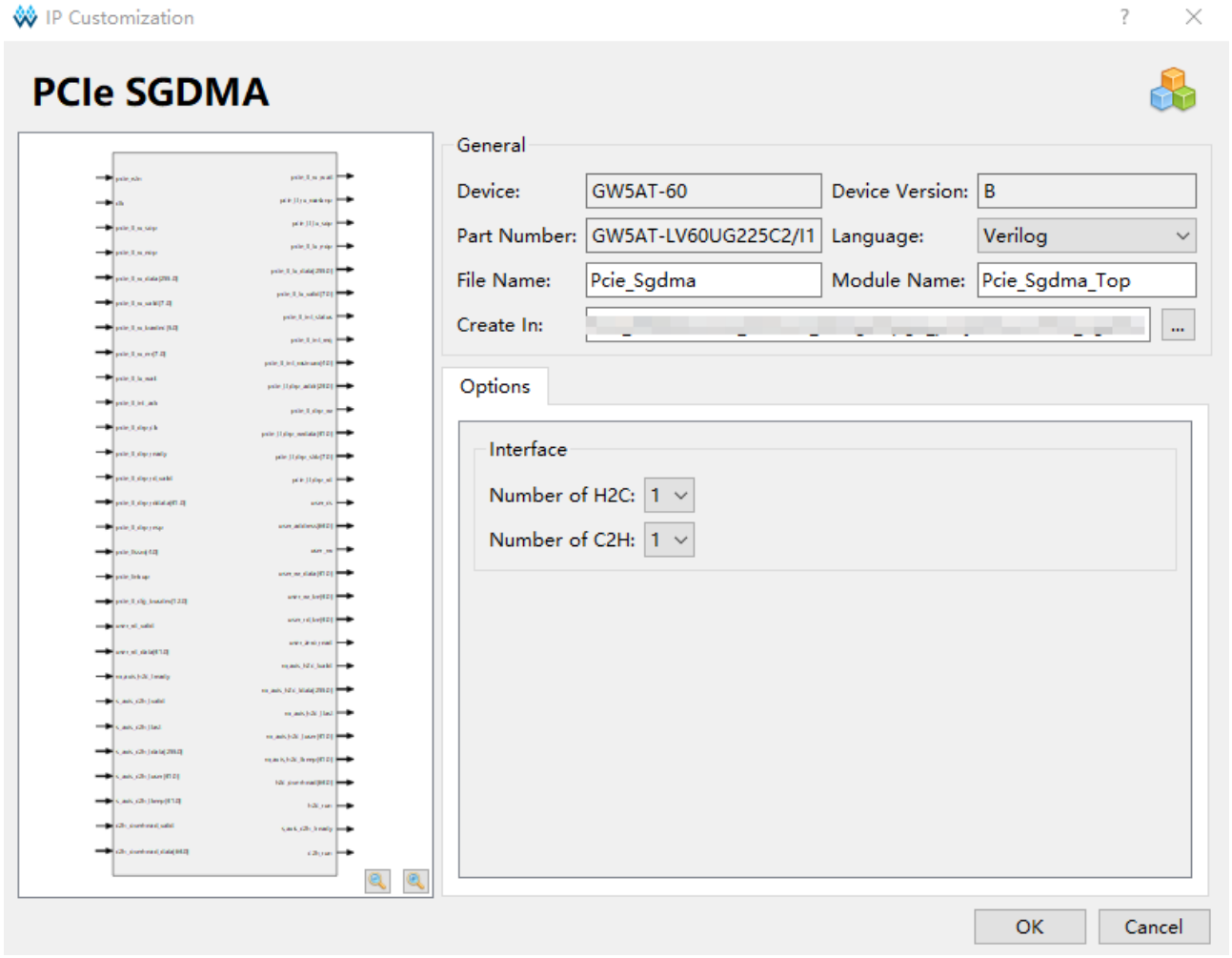
Figure 7-1 IP Core Generator



2. PCIe SGDMA IP Interface

The left of the configuration interface is Gowin PCIe SGDMA IP ports diagram, and the right is the IP configuration options, as shown in Figure 7-2.

Figure 7-2 PCIe SGDMA IP Configuration Interface



You can configure the static parameters of Gowin PCIe SGDMA IP according to the design requirements, and you can refer to Table 7-1.

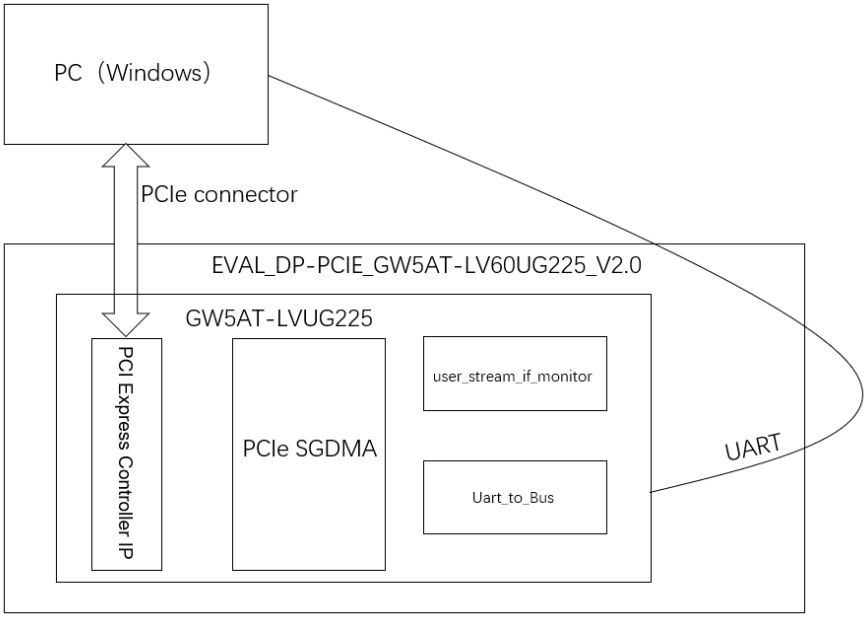
Table 7-1 Gowin PCIe SGDMA IP Parameter

Name	Description	Option
Number of H2C	Number of H2C channels	0: 0 H2C channel 1: 1 H2C channel
Number of C2H	Number of C2H channels	0: 0 C2H channel 1: 1 C2H channel

8 Reference Design

Please download Gowin PCIe SGDMA IP reference design from the official Gowin website. The reference design runs on the EVAL_DP-PCIE_GW5AT-LV60UG225_V2.0 development board. If a different development board is used, the design must be ported accordingly. The overall structure of the reference design is shown in the following figure:

Figure 8-1 Overall Structure of Reference Design



The test steps are as follows:

1. Connect the development board to the PC through the PCIe interface.
2. Download the bitstream (fs) to the development board.
3. Connect the UART interface to the host PC. TX: Development board J5 E5; RX: Development board J5 E6.
4. Power on the PC and install the driver.
5. Enter the w 0201 10 command through the serial port.
6. Enter the w 0102 1 command through the serial port.

7. Run cmd on the PC with administrator privileges.
8. Navigate to the directory xxx/build/x64/bin in the command prompt.
9. Execute the perf.exe -d h2c_0 -l 1K -t 1 command to test H2C.
10. Execute the perf.exe -d c2h_0 -l 1K -t 1 command to test C2H.
11. After the test is completed, the results will be displayed, including Packet Size, Throughput, Efficiency, and other metrics.


GOWIN
PROGRAMMING FOR THE FUTURE