Gowin SPI Nor Flash Interface IP

User Guide

IPUG945-1.3E,02/22/2023

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
08/31/2020	1.0E	Initial version published.
11/04/2020	1.1E	● The generated files names and ports updated. ● Some figures updated.
03/04/2021	1.1.1E	Supported devices added.
03/15/2022	1.1.2E	● Supported devices updated. ● Register list updated. ● The description of activeFlash added.
12/06/2022	1.2E	● The note in 3.2 Port List added. ● Register descriptions updated.
02/22/2023	1.3E	● The descriptions of activeFlash updated. ● The reference design updated.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

The purpose of Gowin SPI Nor Flash Interface IP User Guide is to help you to learn the features and usage of Gowin SPI Nor Flash Interface IP by providing descriptions of functions, GUI, reference design, etc.

1.2 Related Documents

You can find the related documents at www.gowinsemi.com/en:

- DS117, GW1NR series of FPGA Products Data Sheet
- DS861, GW1NSR series of FPGA Products Data Sheet
- DS881, GW1NSER series of FPGA Products Data Sheet
- DS961, GW2ANR series of FPGA Products Data Sheet
- DS976, GW2AN-55 Data Sheet
- DS971, GW2AN-18X & 9X Data Sheet
- SUG100, Gowin Software User Guide

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
AHB	Advanced High performance Bus
FIFO	First Input First Output
FPGA	Field Programmable Gate Array
SPI	Serial Peripheral Interface

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/en

E-mail: support@gowinsemi.com

2 Overview

2.1 Overview

Nor Flash is a non-volatile flash technology, and it can be divided into parallel Nor Flash where the address and data lines are directly connected to the processor and serial Nor Flash where they are connected to the processor through the SPI interface. Due to its simple interface and small package, SPI Nor Flash is widely used in audio and video equipment, mobile phones, mobile peripherals, office equipment and industrial control equipment.

Gowin has designed a SPI Nor Flash Interface IP that provides a common command interface for you to interconnect with the SPI Nor Flash chip to fulfill their memory access needs.

Table 2-1 Gowin SPI Nor Flash Interface IP

Gowin SPI Nor Flash Interface IP	
Logic Resource	Please refer to Table 2-2 and Table 2-3.
Delivered Doc.	
Design Files	Verilog (encrypted)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software (V 1.9.7Beta and above)

Note!

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

2.2 Features

- Supports AHB bus configuration interface
- Supports standard SPI interface
- Supports memory-mapped access (read-only)
- Supports configurable SPI SCLK
- Supports configurable TX/RX FIFO depth

- Supports Standard and Lite modes

2.3 Resource Utilization

Gowin SPI NOR Flash Interface IP can be implemented by Verilog language. Its performance and resource utilization may vary when the design is employed in different devices, or at different density, speed, or grade.

Take Gowin GW1NSR-4C series of FPGA as an instance, the resource utilization of Gowin SPI Nor Flash Interface IP is as shown in Table 2-2 and Table 2-3.

Table 2-2 Resource Utilization in Standard Mode

Device Series	Speed Grade	Name	Resource Utilization	Remarks
GW1NSR-4C	-6	LUT	831	-
		REG	371	
		BSRAM	2	

Table 2-3 Resource Utilization in Lite Mode

Device Series	Speed Grade	Name	Resource Utilization	Remarks
GW1NSR-4C	-6	LUT	433	-
		REG	292	

3 Functional Description

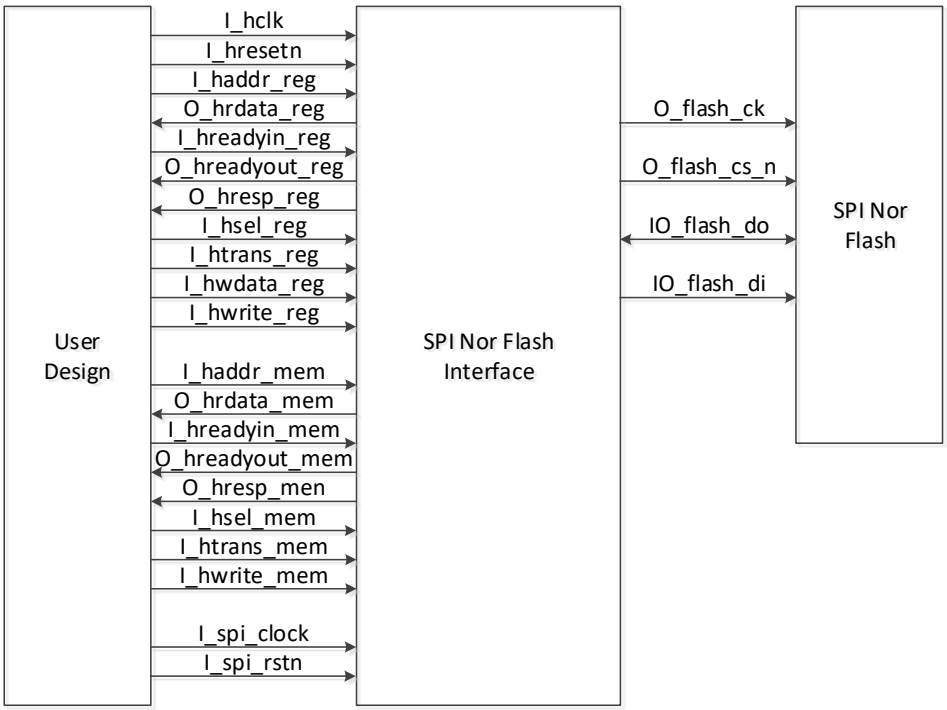
3.1 System Block Diagram

Gowin SPI Nor Flash Interface IP is a serial peripheral device interface controller IP that acts as a master device and connects to SPI Nor Flash devices. SPI transfer format and timing can be configured via the AHB bus interface. The data transmitted by SPI can be in memory-mapped read-only mode or register read-write mode via the AHB bus interface.

The format of the SPI transfer via register read/write mode includes commands, addresses and data segments. A list of corresponding registers provided by the IP can transfer commands, addresses and data via the AHB bus.

For the SPI transfer for memory-mapped read-only mode，data in the corresponding address space can be acquired simply by transferring the address through the AHB bus.

Figure 3-1 System Block Diagram



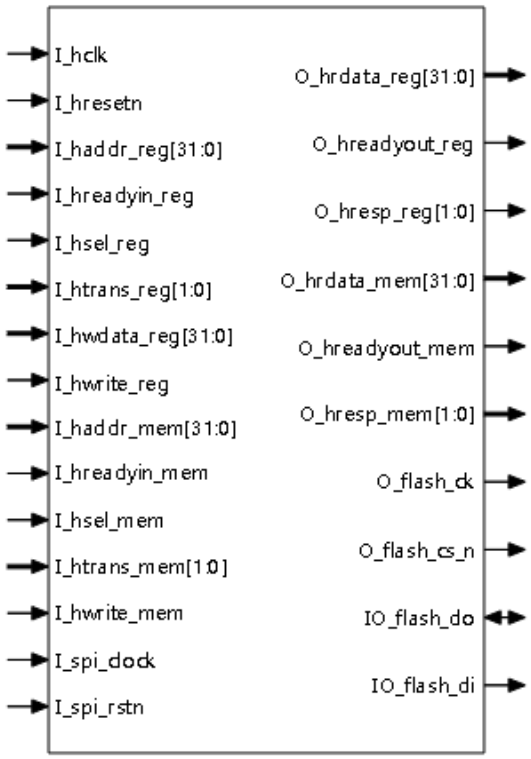
3.2 Port List

Note!

This IP is only used for Gowin devices which integrate SPI Nor Flash, and it just needs to define the Flash ports as O_flash_ck, O_flash_cs_n, IO_flash_do, IO_flash_di in the top module of the project; there is no need to constrain the pins in the constraint file, and Gowin Software will automatically connect to related pins of the integrated SPI Nor Flash.

The IO port of the Gowin SPI Nor Flash Interface IP is shown in Figure 3-2.

Figure 3-2 Gowin SPI Nor Flash Interface IP IO Port Diagram



Ports vary slightly depending on the parameters.

The details of IO ports of Gowin SPI Nor Flash Interface IP are shown in Table 3-1.

Table 3-1 Gowin SPI Nor Flash Interface IP IO Port List

No.	Signal Name	I/O	Description	Notes
1	I_hclk	I	AHB bus working clock	The I/O direction of all the signals takes SPI Nor Flash Interface IP as reference.
2	I_hresetn	I	AHB bus reset, active-low	
3	I_haddr_reg	I	The address of AHB reg mode	
4	O_hrdata_reg	O	Read data in AHB reg mode	
5	I_hreadyin_reg	I	Master device ready signal in AHB reg mode	
6	O_hreadyout_reg	O	Slave device ready signal in AHB reg mode	
7	O_hresp_reg	O	Transfer response in AHB reg mode	
8	I_hsel_reg	I	Slave device selection signal in AHB reg mode	
9	I_htrans_reg	I	Transfer type in AHB reg mode	
10	I_hwdata_reg	I	Write data in AHB reg mode	
11	I_hwrite_reg	I	Read/write signal in AHB reg mode. 1 for write; 0 for read.	
12	I_haddr_mem	I	The address of AHB mem mode Only valid in memory-mapped mode.	

No.	Signal Name	I/O	Description	Notes
13	O_hrdata_mem	O	Read data in AHB mem mode Only valid in memory-mapped mode.	
14	I_hreadyin_mem	I	Master device ready signal in AHB mem mode Only valid in memory-mapped mode.	
15	O_hreadyout_mem	O	Slave device ready signal in AHB mem mode Only valid in memory-mapped mode.	
16	O_hresp_mem	O	Transfer response in AHB mem mode Only valid in memory-mapped mode.	
17	I_hsel_mem	I	Slave device selection signal in AHB mem mode Only valid in memory-mapped mode.	
18	I_htrans_mem	I	Transfer type in AHB mem mode Only valid in memory-mapped mode.	
19	I_hwrite_mem	I	Read/write signal in AHB mem mode. 0 for Read, and no write. Only valid in memory-mapped mode.	
20	I_spi_clock	I	SPI Flash reference clock	
21	I_spi_rstn	I	SPI Flash reset, active-low.	
22	O_flash_ck	O	SPI Flash clock signal	
23	O_flash_cs_n	O	SPI Flash chip select signal	
24	IO_flash_do	I/O	SPI Flash data signal MISO	
25	IO_flash_di	O	SPI Flash data signal MOSI	

3.3 Parameter Configuration

Table 3-2 Gowin SPI Nor Flash Interface IP Parameters

No.	Name	Range	Default	Description
1	Operation Mode	Standard/Lite	Standard	● Standard supports the standard mode. ● Lite supports the lite mode.
2	Memory-mapped Read Support	Yes/No	No	It denotes if memory-mapped read mode is supported, data is read-only.
3	SPI Clock Divider	0~128	0	It denotes the ratio of SCLK frequency to source clock frequency.

No.	Name	Range	Default	Description
				When the SPI Clock Divider value is N and N is a different value, the correspondence is as follows: 0: Fsclk= Fsource_clk 1~128: Fsclk = Fsource_clk/(N*2)
4	TX FIFO Depth	2/4/8/16/32/64/128	4	RX FIFO depth
5	RX FIFO Depth	2/4/8/16/32/64/128	4	TX FIFO depth

3.4 Register Description

Table 3-3 Register List Summary

Address Offset	Register	Description
0x00~0x0C	-	Reserved
0x10	TransFmt	SPI transfer format register
0x20	TransCtrl	SPI transfer control register
0x24	Cmd	SPI command register
0x28	Addr	SPI address register
0x2C	Data	SPI data register
0x30	Ctrl	SPI control register
0x34	Status	SPI status register
0x38	IntrEn	SPI interrupt enable register
0x3C	IntrSt	SPI interrupt status register
0x40	Timing	SPI interface timing register
0x44~0x78	-	Reserved
0x7C	Config	Register configuration

Note!

Register read/write type:

- RO: Read-only
- RW: Read/Write
- W1C: Read and write 1 to clear

3.4.1 SPI Transfer Format Register

Table 3-4 SPI Transfer Format Register

Name	Bit	Type	Description	Default
Reserved	31:18	-	-	0x0
AddrLen	17:16	RW	Address length (byte) 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	0x2
Reserved	15:13	-	-	0x0

Name	Bit	Type	Description	Default
DataLen	12:8	RW	Data length (bit) The actual data length is (DataLen+1) bits.	0x07
DataMerge	7	RW	Data merge mode enable, valid only when DataLen = 0x07. 0x1: 4 bytes transfer 0x0: Only the lowest byte transfer	0x1
Reserved	6:4	-	-	0x0
LSB	3	RW	Data transfer sequence 0x0: MSB priority 0x1: LSB priority	0x0
Reserved	2:0	-	-	0x0

Note!

Before the register operation starts, set the SPI transfer format register (0x10) value to 0x00020780.

3.4.2 SPI Transfer Control Register (0x20)

Table 3-5 SPI Transfer Control Register

Name	Bit	Type	Description	Default
Reserved	31	-	-	0x0
CmdEn	30	RW	SPI command segment enable 0x0: Disable 0x1: Enable	0x0
AddrEn	29	RW	SPI Address segment enable 0x0: Disable 0x1: Enable	0x0
Reserved	28	-	-	0x0
TransMode	27:24	RW	Transfer mode 0x0: Write and read simultaneously 0x1: Write-only 0x2：Read-only 0x3: Write and read 0x4: Read and write 0x5: Write, Dummy and read 0x6: Read, Dummy and write 0x7: No data (CmdEn or AddrEn must be enabled) 0x8: Dummy and write 0x9: Dummy and read 0xa~0xf: Reserved	0x0
DualQuad	23:22	RW	SPI data segment format 0x0: Single mode 0x1: Reserved 0x2: Reserved 0x3: Reserved	0x0

Name	Bit	Type	Description	Default
TokenEn	21	RW	Token byte transfer enable The address segment is appended with 1 byte during SPI read operation, and the value is selected from TokenValue. 0x0: Disable 0x1: Enable	0x0
WrTranCnt	20:12	RW	Number of write data units. ● WrTranCnt indicates the number of data units transferred from the data register to the SPI bus. The actual quantity is (WrTranCnt+1). ● WrTranCnt only works when the TransMode is 0, 1, 3, 4, 5, 6 or 8. ● For TransMode 0, WrTranCnt must be equal to RdTranCnt.	0x0
TokenValue	11	RW	Token byte value 0x0: The value is 0x00. 0x1: The value is 0x69.	0x0
DummyCnt	10:9	RW	Number of Dummy data ● The actual number is DummyCnt+1. ● The number of Dummy cycles at SPI interface is (DummyCnt+1)*((DataLen+1)/SPI IO width. ● The data pin is high resistance during Dummy. ● DummyCnt is valid only when TransMode is 5,6,8,9. ● Dummy cycle settings in common transfer formats are shown in Table 3-6.	0x0
RdTranCnt	8:0	RW	Number of read data units. ● RdTranCnt indicates the number of data units received from the SPI bus and stored in the data register. The actual quantity is (RdTranCnt+1). ● RdTranCnt only works when the TransMode is 0, 2, 3, 4, 5, 6 or 9. ● For TransMode 0, WrTranCnt must be equal to RdTranCnt.	0x0

Table 3-6 Dummy Cycle Settings in Common Transfer Formats

Transfer Format	DummyCnt+1	DataLen+1	DualQuad	#Dummy Cycles on the SPI Interface
Dummy Cycle	1	8	Single	8

3.4.3 SPI Command Register (0x24)

Table 3-7 SPI Command Register

Name	Bit	Type	Description	Default
Reserved	31:8	-	-	0x0
CMD	7:0	RW	SPI Command	0x0

3.4.4 SPI Address Register (0x28)

Table 3-8 SPI Address Register

Name	Bit	Type	Description	Default
ADDR	31:0	RW	SPI address	0x0

3.4.5 SPI Data Register (0x2C)

Table 3-9 SPI Data Register

Name	Bit	Type	Description	Default
DATA	31:0	RW	SPI transmits or receives data ● For write operations, data is written to the TX FIFO; low byte first transfers. If the TX FIFO is full and the SPIActive bit of the status register is 1, the hready signal is pulled down to insert the wait state. ● For read operations, read data from RX FIFO; low byte first transfers. If the RX FIFO is empty and the SPIActive bit of the status register is 1, the hready signal is pulled down to insert the wait state. ● When the TX FIFO is empty, the SPI transfer will hold until more data is written to the TX FIFO; when the RX FIFO is full, the SPI transfer will hold until the RX FIFO has space to receive data. ● In write operations, if more data than WrTranCnt is written to the TX FIFO, the excess data goes to the next transfer or is cleared by resetting the TX FIFO.	0x0

3.4.6 SPI Control Register (0x30)

Table 3-10 SPI Control Register

Name	Bit	Type	Description	Default
Reserved	31:3	-	-	0x0
TXFIFORST	2	RW	TX FIFO reset Write 1 to reset; clear automatically after the reset operation is completed.	0x0
RXFIFORST	1	RW	RX FIFO reset Write 1 to reset; clear automatically after the reset operation is completed.	0x0

Name	Bit	Type	Description	Default
SPIRST	0	RW	SPI Reset Write 1 to reset; clear automatically after the reset operation is completed.	0x0

Note!

Lite mode is not supported.

3.4.7 SPI Status Register (0x34)

Table 3-11 SPI Status Register

Name	Bit	Type	Description	Default
Reserved	31:30	-	-	0x0
TXNUM[7:6]	29:28	RO	Number of valid data in TX FIFO [7:6]	0x0
Reserved	27:26	-	-	0x0
RXNUM[7:6]	25:24	RO	Number of valid data in RX FIFO [7:6]	0x0
TXFULL	23	RO	The flag of TX FIFO full.	0x0
TXEMPTY	22	RO	The flag of TX FIFO empty.	0x1
TXNUM[5:0]	21:16	RO	Number of valid data in TX FIFO [5:0]	0x0
RXFULL	15	RO	The flag of RX FIFO full	0x0
RXEMPTY	14	RO	The flag of RX FIFO empty	0x1
RXNUM[5:0]	13:8	RO	Number of valid data in RX FIFO [5:0]	0x0
Reserved	7:1	-	-	0x0
SPIActive	0	RO	The flag of SPI register active The SPIActive goes to 1 when the SPI command register is written, and to 0 when the transfer is completed.	0x0

3.4.8 SPI Interrupt Enable Register (0x38)

Table 3-12 SPI Interrupt Enable Register

Name	Bit	Type	Description	Default
Reserved	31:5	-	-	0x0
EndIntEn	4	RW	SPI transfer end interrupt enable Controls whether an interrupt is triggered when SPI transfer ends.	0x0
Reserved	3:0	-	-	0x0

Note!

This register does not support Lite mode.

3.4.9 SPI Interrupt Status Register (0x3C)

Table 3-13 SPI Interrupt Status Register

Name	Bit	Type	Description	Default
Reserved	31:5	-	-	0x0
EndInt	4	W1C	SPI transfer end interrupt This bit is set to 1 when SPI transfer end interrupt is triggered.	0x0
Reserved	3:0	-	-	0x0

3.4.10 SPI Interface Timing Register (0x40)

Table 3-14 SPI Interface Timing Register

Name	Bit	Type	Description	Default
Reserved	31:12	-	-	0x0
Reserved	11:8	-	-	0x2
SCLK_DIV	7:0	RW	SPI interface clock to source clock frequency ratio, depending on configuration. ● When the SCLK_DIV value is in the range 0~127, SCLK frequency = source clock frequency/((SCLK_DIV+1)*2) ● When the SCLK_DIV value is 255, the SCLK frequency is equal to the source clock spi_clock frequency.	0xff

Note!

Lite mode is not supported.

3.4.11 Register Configuration (0x7C)

Table 3-15 Register Configuration

Name	bit	Type	Description	Default
Reserved	31:8	-	-	0x0
TxFIFOSize	7:4	RO	TX FIFO depth depends on configuration 0x0:2 words 0x1:4 words 0x2:8 words 0x3:16 words 0x4:32 words 0x5:64 words 0x6:128 words	0x1
RXFIFOSize	3:0	RO	RX FIFO depth depends on configuration 0x0:2 words 0x1:4 words 0x2:8 words 0x3:16 words 0x4:32 words 0x5:64 words 0x6:128 words	0x1

Note!

This register does not support Lite mode.

3.5 Hardware Configuration Options

3.5.1 TX FIFO depth

The depth of TX FIFO is determined by the IP core generator, corresponding to TXFIFOSize in the configuration register (0x7C). The depth can be set to 2, 4, 8, 16, 32, 64 and 128. The default depth value is 4.

Lite mode depth is not configurable and fixed to 1.

3.5.2 RX FIFO depth

The depth of RX FIFO is determined by the IP core generator, corresponding to RX FIFOSize in the configuration register (0x7C). The depth can be set to 2, 4, 8, 16, 32, 64 and 128. The default depth value is 4.

Lite mode depth is not configurable and is fixed at 1.

3.5.3 SPI Interface Clock Parameters

The SPI interface clock frequency is determined by the IP core generator, corresponding to the SCLK_DIV in the SPI interface timing register (0x40). For example:

- When the SPI Clock Divider is 0 and the SCLK_DIV value is 255.
- When the SPI Clock Divider is 1 to 128 and the SCLK_DIV value is 0 to 127.

The Lite mode SCLK clock is not configurable and the clock frequency is fixed to hclk.

3.5.4 Memory-Mapped Read

IP provides an interface for memory-mapped reads. The IP core generator is used to determine if this interface is supported.

Lite mode does not support memory-mapped read.

3.6 Read and Write Operation Flow

In this manual, take SPI NOR Flash integrated in Gowin devices as an instance to illustrate the data read/write operation steps of SPI Nor Flash Interface IP. This IP only supports standard SPI mode, so Dual SPI mode and Quad SPI mode are not supported. The list of SPI NOR Flash commands (excluding Dual SPI, Quad SPI modes) are as follows.

Table 3-16 List of SPI NOR Flash Commands (Excluding Dual SPI, Quad SPI Modes)

Command Name	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Write Enable	06H	–	–	–	–	–	–
Write Disable	04H	–	–	–	–	–	–
Read Status Register-1	05H	(S7-S0)	(cont.)	–	–	–	–
Read Status Register-2	35H	(S15-S8)	(cont.)	–	–	–	–
Read Status Register-3	15H	(S23-S16)	(cont.)	–	–	–	–

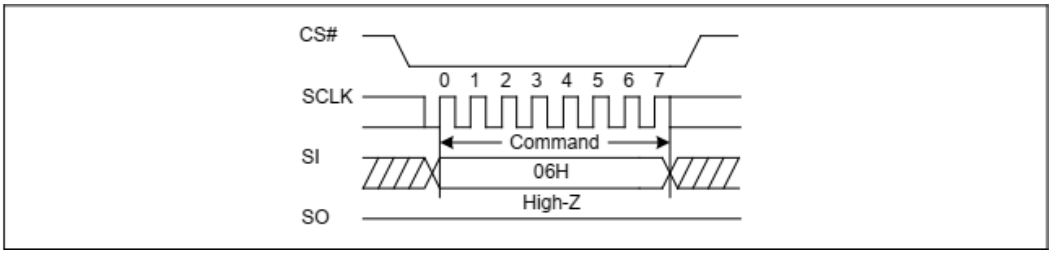
Command Name	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Write Status Register-1	01H	S7-S0	–	–	–	–	–
Write Status Register-2	31H	S15-S8	–	–	–	–	–
Write Status Register-3	11H	S23-S16	–	–	–	–	–
Read Data	03H	A23-A16	A15-A8	A7-A0	(D7-D0)	(cont.)	–
Page Program	02H	A23-A16	A15-A8	A7-A0	D7-D0	Next Byte	–
Sector Erase	20H	A23-A16	A15-A8	A7-A0	–	–	–
Block Erase(32K)	52H	A23-A16	A15-A8	A7-A0	–	–	–
Block Erase(64K)	D8H	A23-A16	A15-A8	A7-A0	–	–	–
Chip Erase	C7/60H	–	–	–	–	–	–
Read Manufacturer/ Device ID	90H	00H	00H	00H	(MID7-MID0)	(DID7-DID0)	(cont.)
Read Identification	9FH	(M7-M0)	(JDID15-JDID8)	(JDID7-JDID0)	(cont.)	–	–
Read Unique ID	4BH	00H	00H	00H	dummy	(UID7-UID0)	(cont.)

3.6.1 Write Enable (WREN) (06H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000006.
   CMD = 0x06

Figure 3-3 Timing Diagram of Write Enable

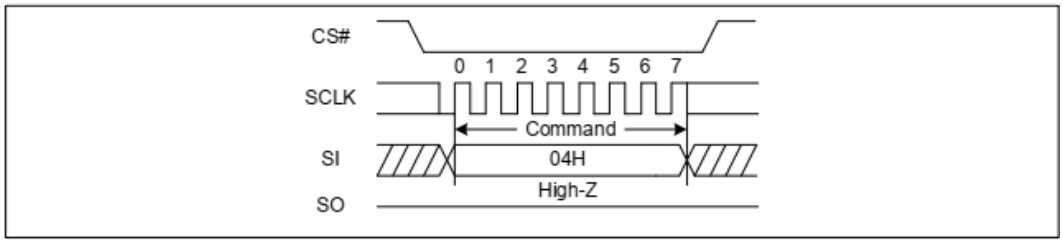


3.6.2 Write Disable (WRDI) (04H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000004.
   CMD = 0x04

Figure 3-4 Timing Diagram of Write Disable

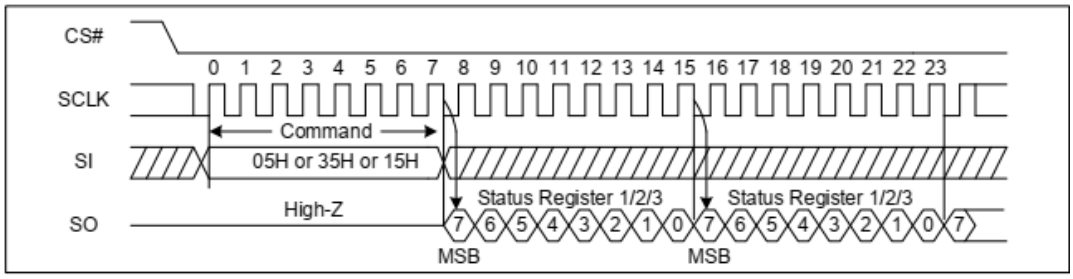


3.6.3 Read Status Register (RDSR) (05H/ 35H/ 15H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x42000000.
   a) CmdEn = 1
   b) TransMode = 0x2 (Read-only)
2. Set SPI command register (0x24) to 0x00000005.
   CMD = 0x05
3. Read SPI data register (0x2C)

Figure 3-5 Timing Diagram of Read Status Register

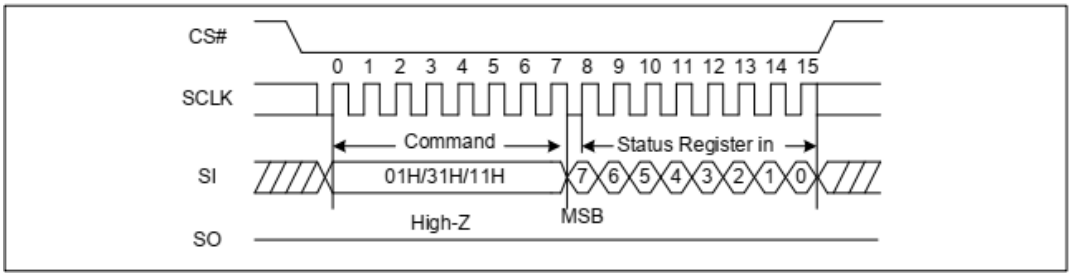


3.6.4 Write Status Register (WRSR) (01H/ 31H/ 11H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x41000000.
   a) CmdEn = 1
   b) TransMode = 0x1 (Write-only)
2. Set SPI data register (0x2C) to 0x00000000.
   DATA = 0x00000000 (Close Protect Bit)
3. Set SPI command register (0x24) to 0x00000001.
   CMD = 0x01

Figure 3-6 Timing Diagram of Write Status Register



3.6.5 Read Data Bytes (READ) (03H) Operating Process

The process of consecutively reading 16 bytes is as follows:

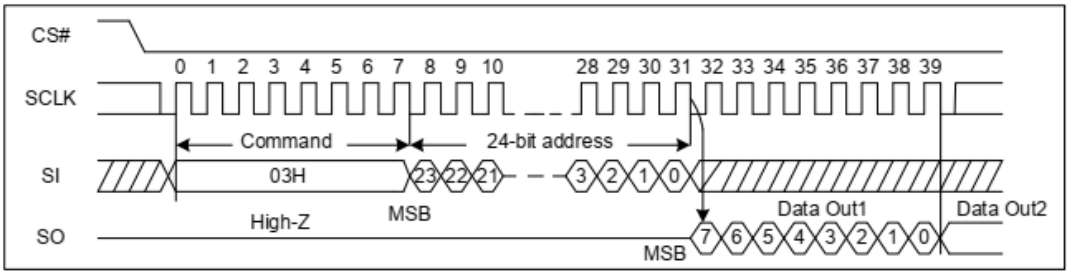
1. Set SPI transfer control register (0x20) to 0x6200000f.
   a) CmdEn = 1
   b) AddrEn = 1
   c) TransMode = 0x2 (Read-only)
   d) RdTranCnt = 15 (Total number of transfer - 1)
2. Set SPI control register (0x30) to 0x00000002.
   RXFIFORST = 1
3. Set SPI Address register (0x28) to 0x00000000.
   ADDR = 0x00000000
4. Set SPI command register (0x24) to 0x00000003.
   CMD = 0x03
5. Read SPI data register (0x2C)

Read up to 4 bytes at a time; more than 4 bytes, read this register in multiple times.

Note!

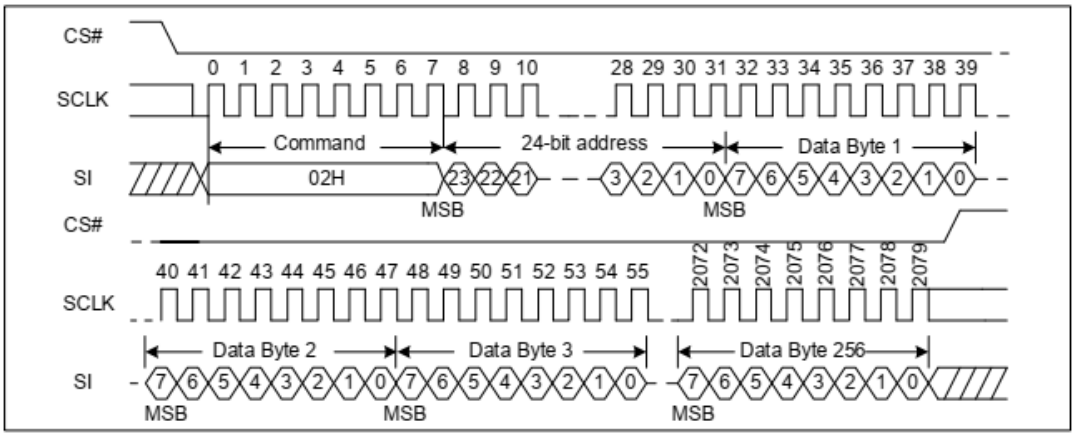
Lite mode reads up to 4 bytes consecutively.

Figure 3-7 Timing Diagram of Read Data Bytes



3.6.6 Page Program (PP) (02H) Operation Procedure

Figure 3-8 Timing Diagram of Page Program



To illustrate the specific process of writing 16 bytes consecutively.

There are two main steps, the first is to erase the data and second is to write the data, with the following commands.

1. Erase data
   a) Set SPI transfer control register (0x20) to 0x47000000
      i. CmdEn = 1
      ii. TransMode = 0x7 (no data)
   b) Set SPI command register (0x24) to 0x00000006.
      CMD = 0x06
   c) Set SPI transfer control register (0x20) to 0x67000000.
      i. CmdEn = 1
      ii. AddrEn = 1
      iii. TransMode = 0x7 (no data)
   d) Set SPI address register (0x28) to 0x00000000.
      ADDR = 0x00000000
   e) Set SPI command register (0x24) to 0x00000020.
      CMD = 0x20 (sector erase)
2. Write data
   a) Set SPI transfer control register (0x20) to 0x47000000.
      i. CmdEn = 1
      ii. TransMode = 0x7 (no data)
   b) Set SPI command register (0x24) to 0x00000006.
      CMD = 0x06
   c) Set SPI transfer control register (0x20) to 0x6100f000.
      i. CmdEn = 1
      ii. AddrEn = 1
      iii. TransMode = 0x1 (write-only)
      iv. WrTranCnt = 15 (Total number of transfer - 1)
   d) Set SPI control register (0x30) to 0x00000004.
      TXFIFORST = 1
   e) Set SPI interrupt enable register (0x38) to 0x00000010.
      EndIntEn = 1
   f) Set SPI data register (0x2C) to 0x33221100.
      DATA = 0x33221100 (1~4 bytes)
   g) Set SPI data register (0x2C) to 0x77665544.
      DATA = 0x77665544 (5~8 bytes)

h) Set SPI data register (0x2C) to 0x33221100

DATA = 0xbbaa9988 (9~12 bytes)

i) Set SPI data register (0x2C) to 0x77665544.

DATA = 0xffeeddcc (13~16 bytes)

j) Set SPI address register (0x28) to 0x00000000.

ADDR = 0x00000000

k) Set SPI command register (0x24) to 0x00000002.

CMD = 0x02

l) Read SPI interrupt status register (0x3C)

See if EndInt is 1. If it is 1, the transfer ends.

m) Set SPI interrupt status register (0x3C) to 0x00000010.

EndInt= 1 (Write 1 to clear)

Note!

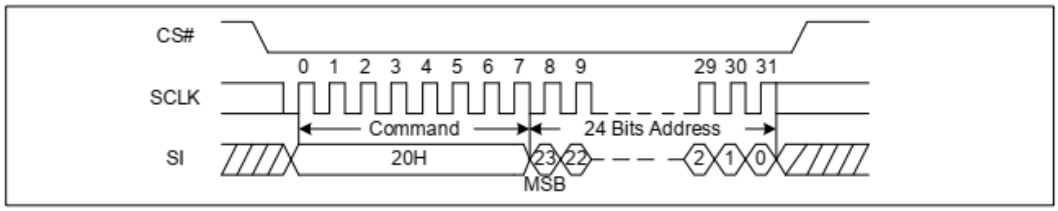
- The erase command sets the corresponding area of Flash data to 1. There are 4 erase commands, including sector erase, 32KB block erase, 64KB block erase, and chip erase. If there are other valid data in the erased area, the operator needs to read the other data in the erased area before writing, and then write back in time.
- Write data cannot exceed TX FIFO depth.
- Lite mode Write up to 4 bytes consecutively.

3.6.7 Sector Erase (SE) (20H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000006.
   CMD = 0x06
3. Set SPI transfer control register (0x20) to 0x67000000.
   a) CmdEn = 1
   b) AddrEn = 1
   c) TransMode = 0x7 (no data)
4. Set SPI address register (0x28) to 0x00000000.
   ADDR = 0x00000000
5. Set SPI command register (0x24) to 0x00000020.
   CMD = 0x20 (sector erase)

Figure 3-9 Timing Diagram of Sector Erase

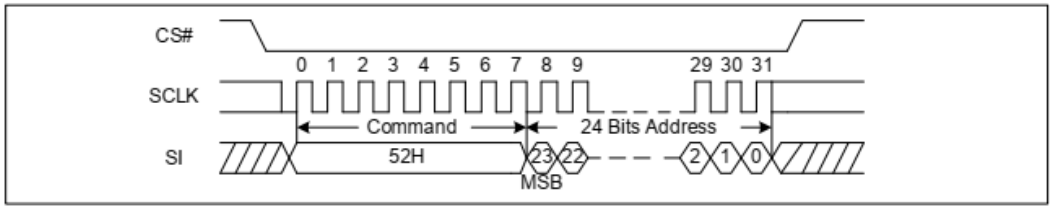


3.6.8 32KB Block Erase (BE) (52H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000006.
   CMD = 0x06
3. Set SPI transfer control register (0x20) to 0x67000000.
   a) CmdEn = 1
   b) AddrEn = 1
   c) TransMode = 0x7 (no data)
4. Set SPI address register (0x28) to 0x00000000.
   ADDR = 0x00000000
5. Set SPI command register (0x24) to 0x00000052.
   CMD = 0x52 (32KB block erase)

Figure 3-10 Timing Diagram of 32KB Block Erase



3.6.9 64KB Block Erase (BE) (D8H) Operating Process

The specific process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000006.
   CMD = 0x06
3. Set SPI transfer control register (0x20) to 0x67000000.
   a) CmdEn = 1
   b) AddrEn = 1

c) TransMode = 0x7 (no data)

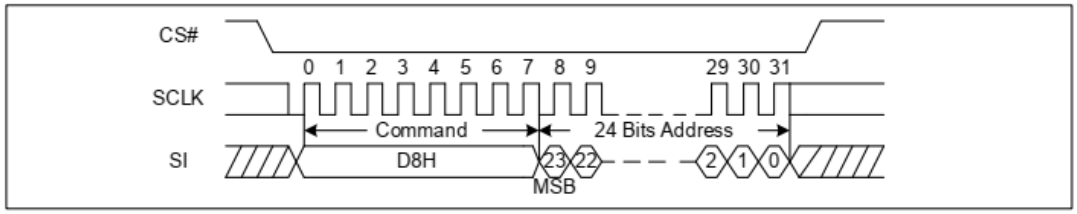
4. Set SPI address register (0x28) to 0x00000000.

   ADDR = 0x00000000

5. Set SPI command register (0x24) to 0x000000D8.

   CMD = 0xD8 (64KB block erase)

Figure 3-11 Timing Diagram of 64KB Block Erase



3.6.10 Chip Erase (CE) (60/ C7H) Operating Process

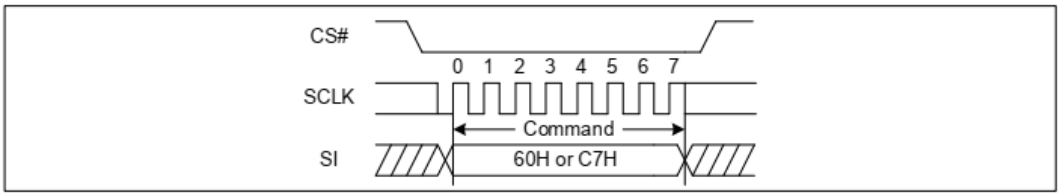
The process is as follows:

1. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
2. Set SPI command register (0x24) to 0x00000006.

   CMD = 0x06
3. Set SPI transfer control register (0x20) to 0x47000000.
   a) CmdEn = 1
   b) TransMode = 0x7 (no data)
4. Set SPI command register (0x24) to 0x00000060.

   CMD = 0x60 (chip erase)

Figure 3-12 Timing Diagram of Chip Erase



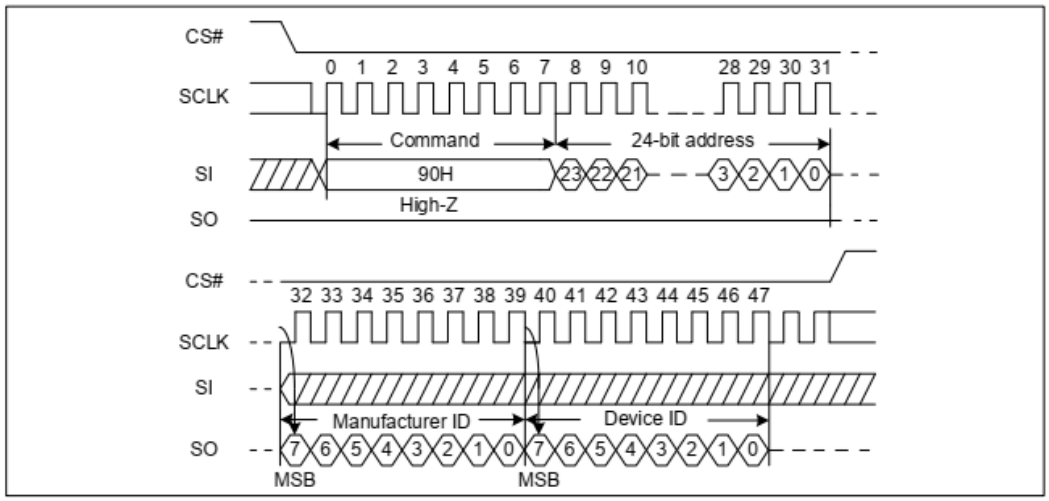
3.6.11 Read Manufacture ID/ Device ID (REMS) (90H) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x62000001.
   a) CmdEn = 1
   b) AddrEn = 1
   c) TransMode = 0x2 (read-only)
   d) RdTranCnt = 1 (Total number of transfer - 1)

2. Set SPI control register (0x30) to 0x00000002.
   RXFIFORST = 1
3. Set SPI address register (0x28) to 0x00000000.
   ADDR = 0x00000000
4. Set SPI command register (0x24) to 0x00000090.
   CMD = 0x90
5. Read SPI data register (0x2C)

Figure 3-13 Timing Diagram of Read Manufacture ID/Device ID

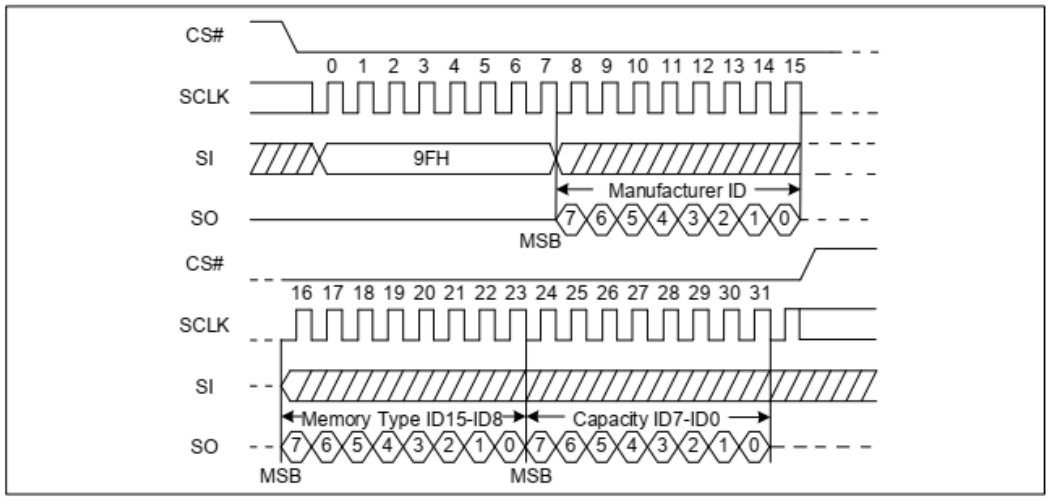


3.6.12 Read Identification (RDID) (9FH) Operating Process

The process is as follows:

1. Set SPI transfer control register (0x20) to 0x42000002.
   a) CmdEn = 1
   b) TransMode = 0x2 (Read-only)
   c) RdTranCnt = 2 (Total number of transfer - 1)
2. Set SPI control register (0x30) to 0x00000002.
   RXFIFORST = 1
3. Set SPI command register (0x24) to 0x0000009F.
   CMD = 0x9F
4. Read SPI data register (0x2C)

Figure 3-14 Timing Diagram of Read Identification



3.6.13 Read Unique ID (4BH) Operating Process

The process is as follows:

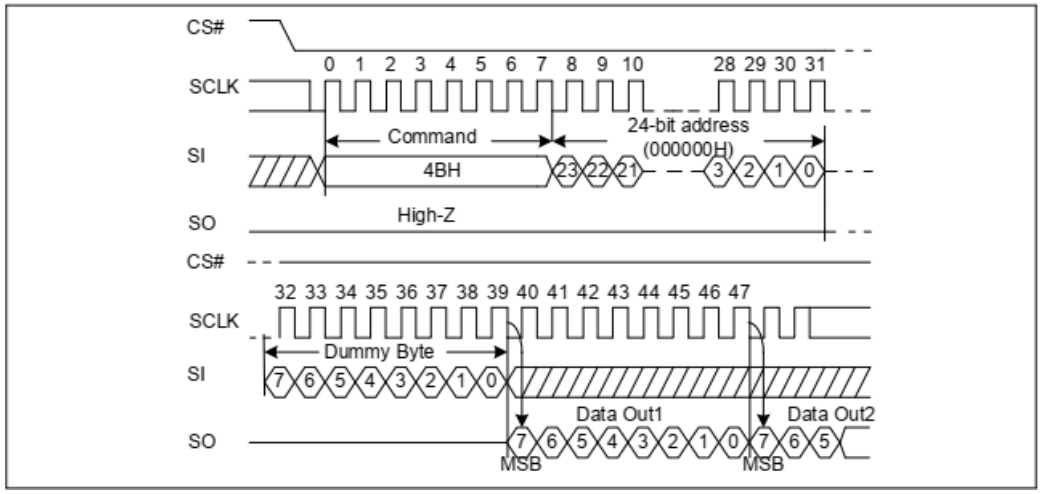
1. Set SPI transfer control register (0x20) to 0x6900000f.
   a) CmdEn = 1
   b) AddrEn = 1
   c) TransMode = 0x9 (dummy, read)
   d) RdTranCnt = 15 (Total number of transfer - 1)
2. Set SPI control register (0x30) to 0x00000002.
   RXFIFORST = 1
3. Set SPI address register (0x28) to 0x00000000.
   ADDR = 0x00000000
4. Set SPI command register (0x24) to 0x0000004B.
   CMD = 0x4B
5. Read SPI data register (0x2C)

Since Unique ID is 128 bits, the SPI data register needs to read four times.

Note!

This command is not supported in Lite mode.

Figure 3-15 Timing Diagram of Read Unique ID



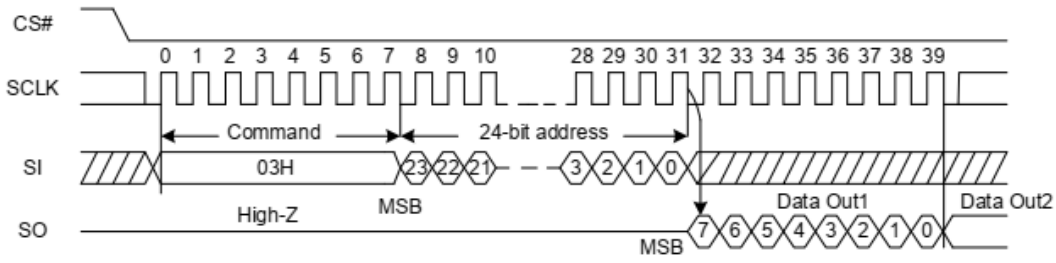
3.7 Timing Description

This section describes the timing of Gowin SPI Nor Flash Interface IP.

3.7.1 SPI Interface Timing

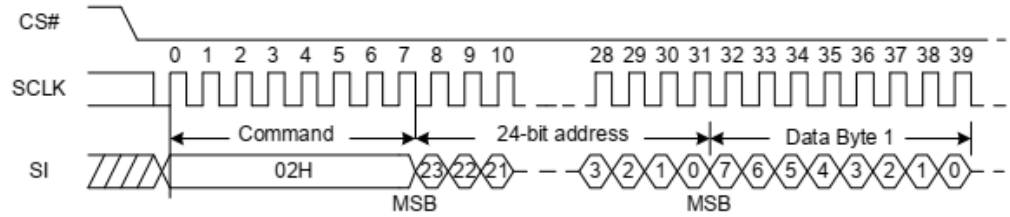
The timing diagram of SPI Flash is as shown in Figure 3-16.

Figure 3-16 Timing Diagram of SPI Flash Read Interface



The timing diagram of SPI Flash is as shown in Figure 3-17.

Figure 3-17 Timing Diagram of SPI Flash Write Interface



3.7.2 AHB Interface Timing

Figure 3-18 AHB Bus Transfer Interface Timing without Waiting State

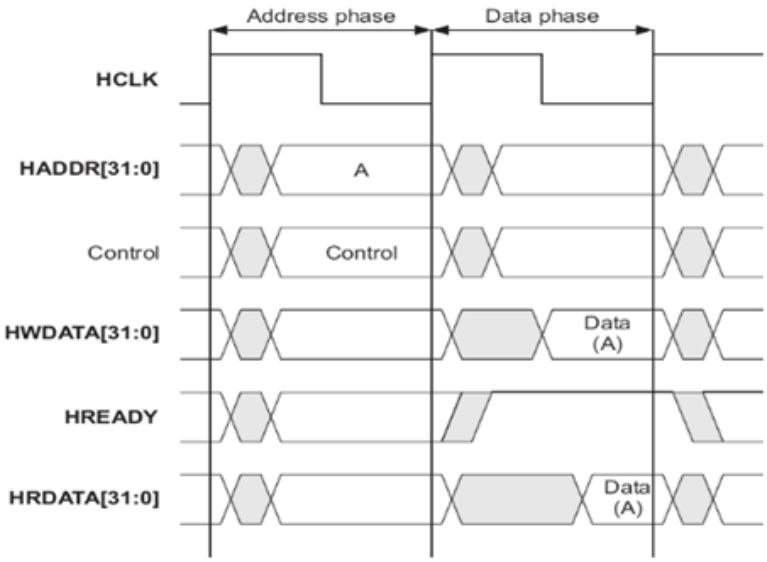
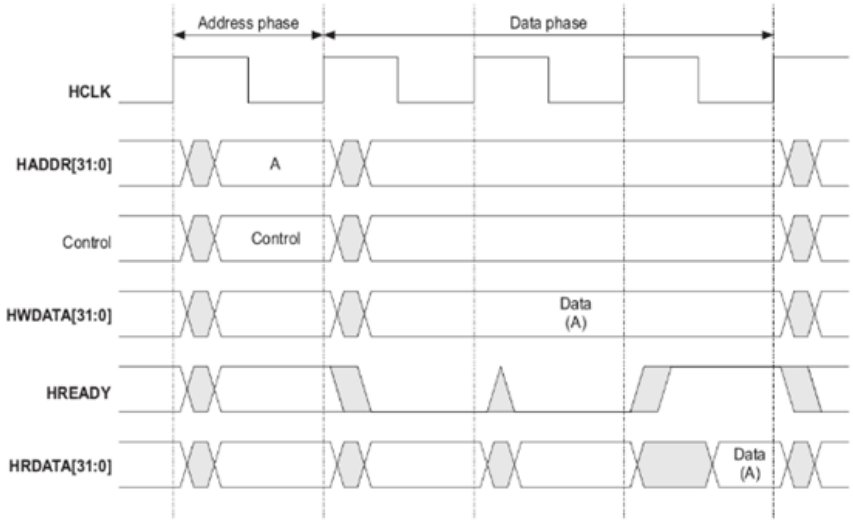


Figure 3-19 AHB Bus Transfer Interface Timing with Waiting State



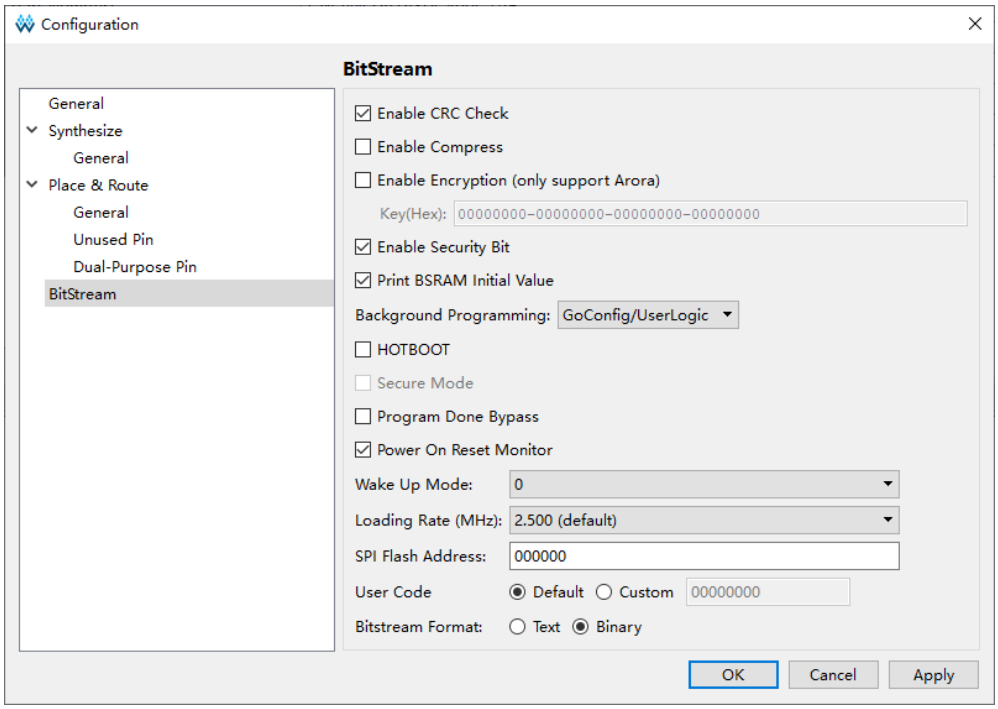
3.8 activeFlash

When the GW2AN-18X or GW2AN-9X device is selected, there are two methods for activating the embedded SPI Nor Flash if the embedded SPI Nor Flash is used.

Method One

You do need to modify the code, just select "GoConfig/UserLogic" in "Configuration > BitStream > Background Programminig" option, and method 1 is recommended.

Figure 3-20 Configuration Interface (Method One)

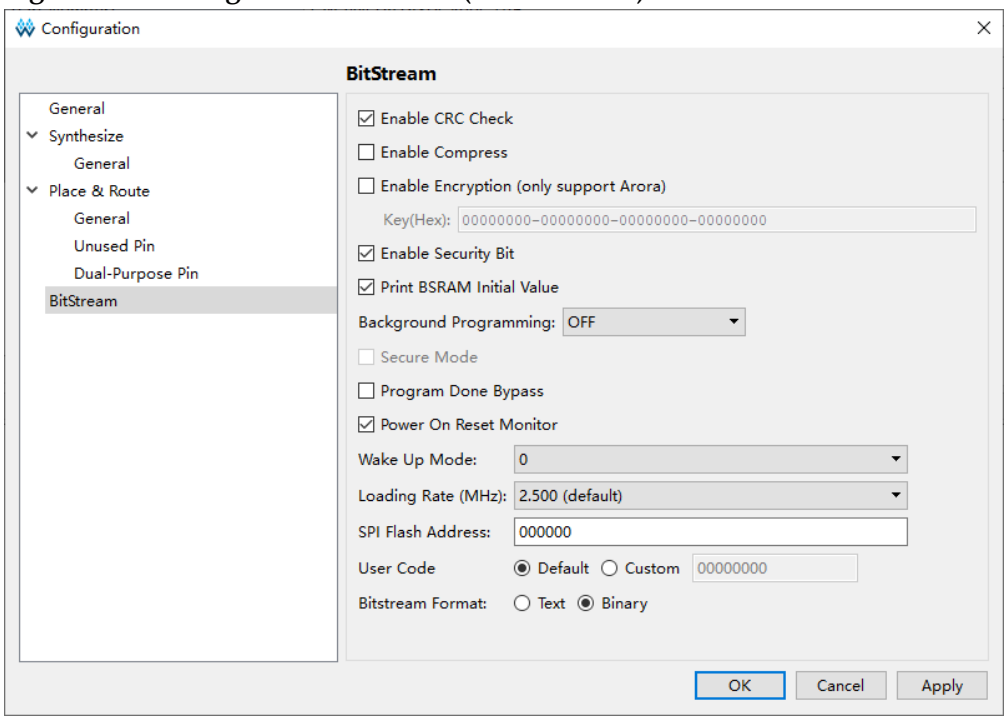


Method Two

You can modify the code to add the instantiated activeFlash module. For the details, you can see *SUG283, Gowin Primitives User Guide*.

In addition, select "OFF" in the "Configuration > BitStream > Background Programminig" option.

Figure 3-21 Configuration Interface (Method Two)



Note!

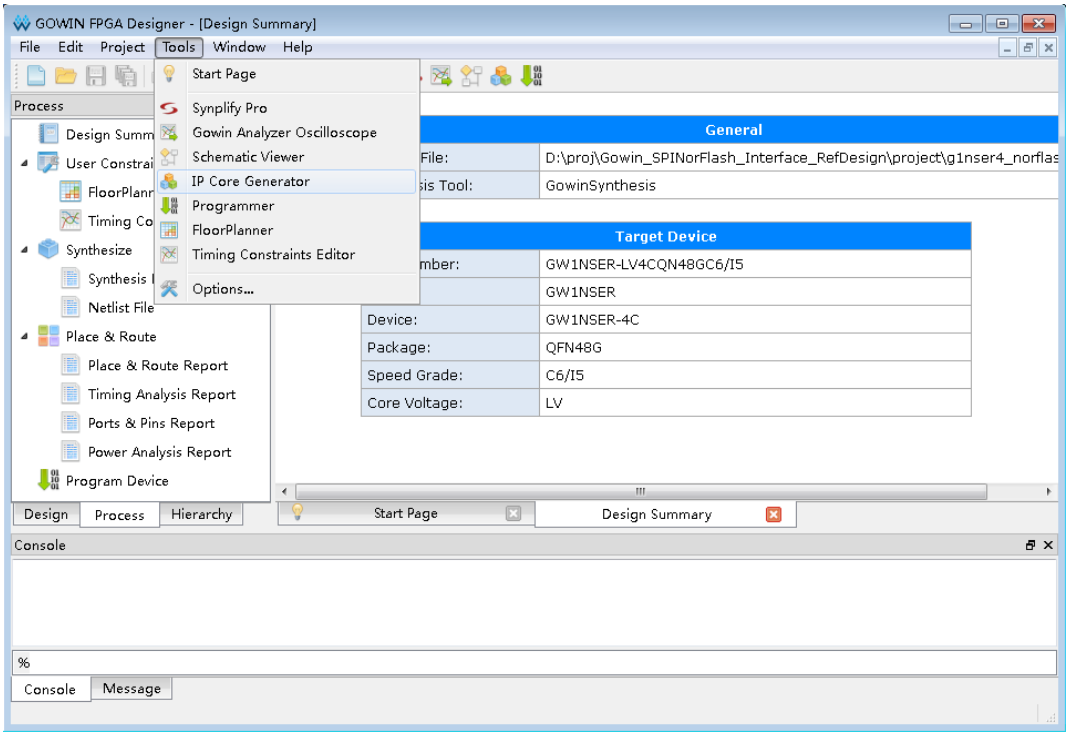
The SPI Nor Flash embedded in GW2AN-18X and GW2AN-9X devices has a size of 16 Mbit. This SPI Nor Flash itself is used for bitstream storage and occupies 896 Kbytes with the address 0x000000~0x0DFFFF, so the available space for users is 1152 Kbytes with the address 0x0E0000~0x1FFFFF. Users need to avoid operating the bitstream storage space when using it.

4 GUI

You can call and configure Gowin SPI Nor Flash Interface IP using the IP core generator tool in the IDE.

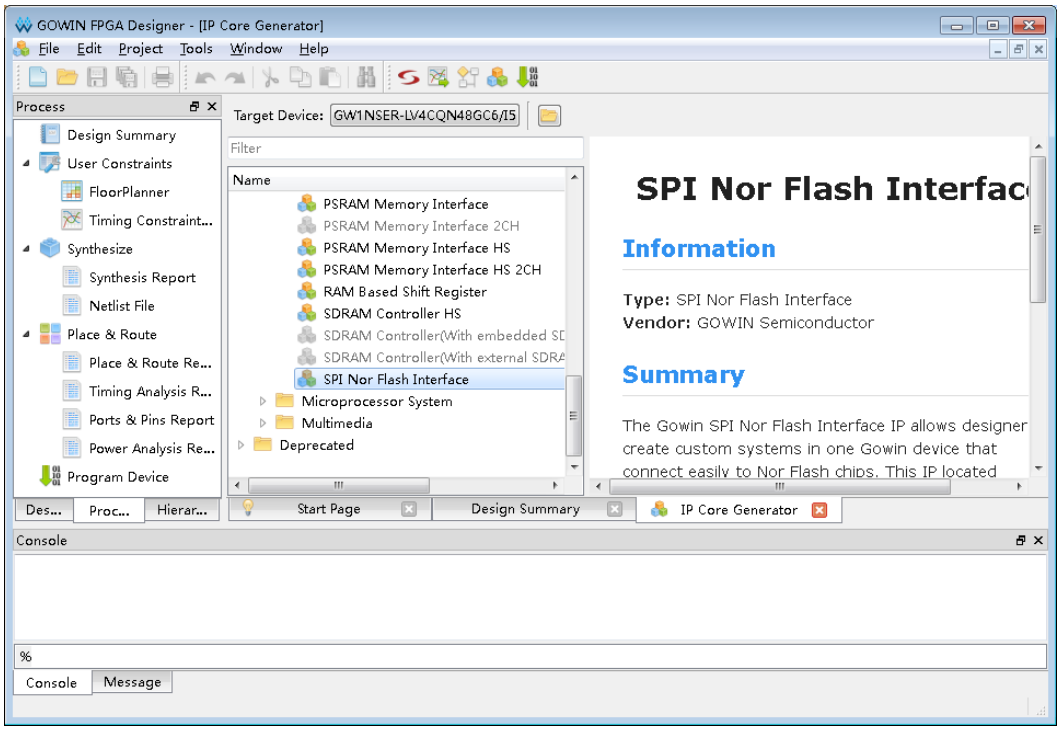
1. Open IP Core Generator
   After creating the project, click the "Tools" tab in the upper left, select and open the IP Core Generater from the drop-down list, as shown in Figure 4-1.

Figure 4-1 Open IP Core Generator



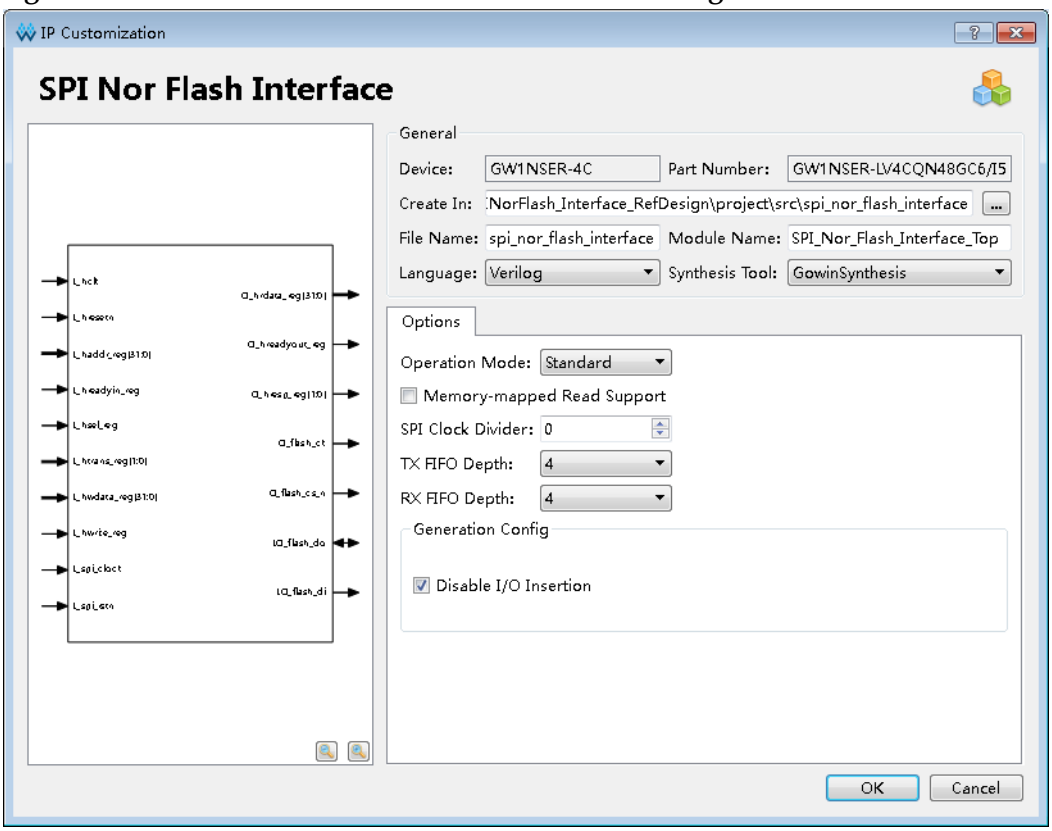
2. Open SPI Nor Flash Interface IP Core
   Click the Memory Control option, double-click SPI Nor Flash Interface. SPI Nor Flash Interface IP core opens, as shown in Figure 4-2.

Figure 4-2 Open SPI Nor Flash Interface IP Core



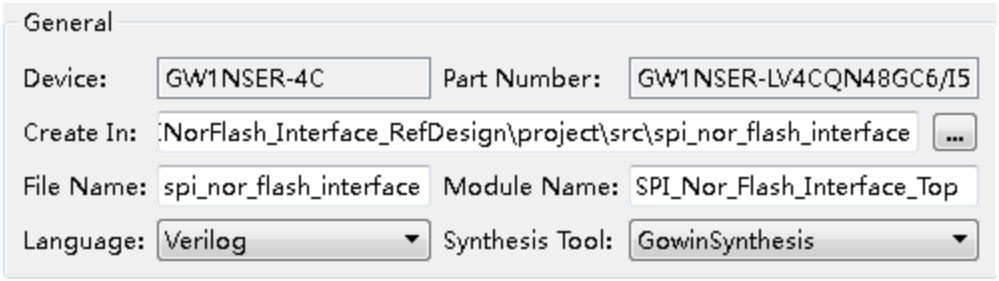
3. SPI Nor Flash Interface IP Core Port Interface
   On the left of the configuration interface is the port diagram of SPI Nor Flash Interface IP core, as shown in Figure 4-3.

Figure 4-3 SPI Nor Flash Interface IP Core Interface Diagram



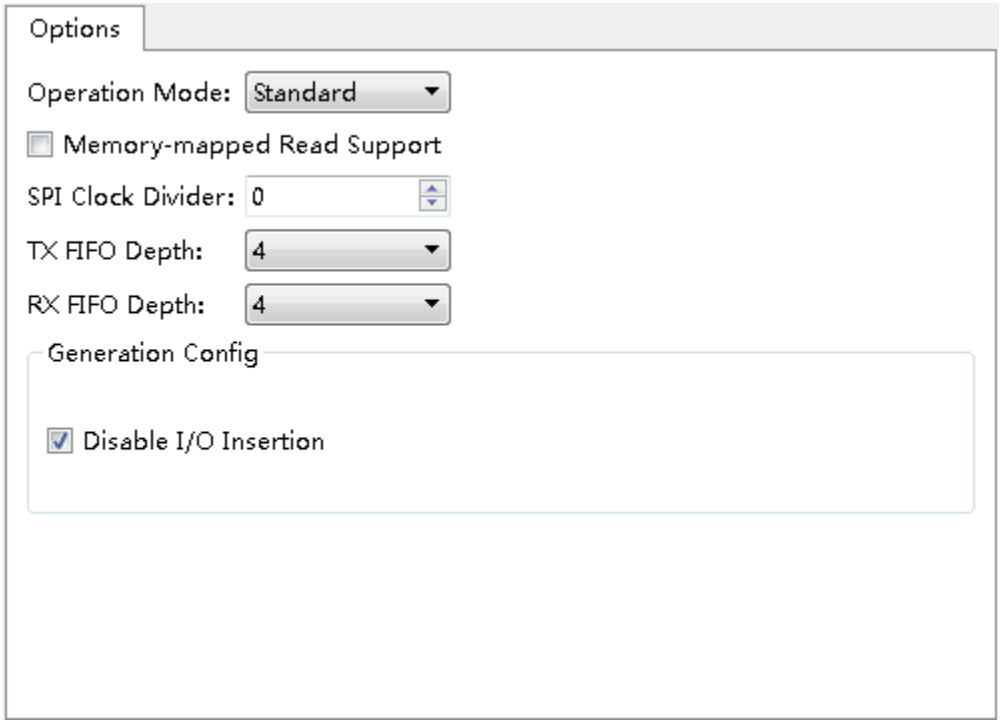
4. General Tab
   See the basic information in the upper part of GUI. Take the GW1NSER-4C chip as an example, and select the "QFN48G" package. The "Module Name" displays the top-level file name of the generated project, and the default value is "SPI_Nor_Flash_Interface_Top". You can modify the name. The folder generated by the IP core is shown in "File Name", which contains the files required by SPI Nor Flash Interface IP core, and the default is "spi_nor_flash_interface ". You can modify the path. "Creat In" shows the path of IP core file. The default is "\project path\ src\spi_nor_flash_interface". You can modify the path.

Figure 4-4 Basic Information Configuration Interface



5. Options
   You can configure clock parameters of SPI Nor Flash Interface in "Options".

Figure 4-5 Options



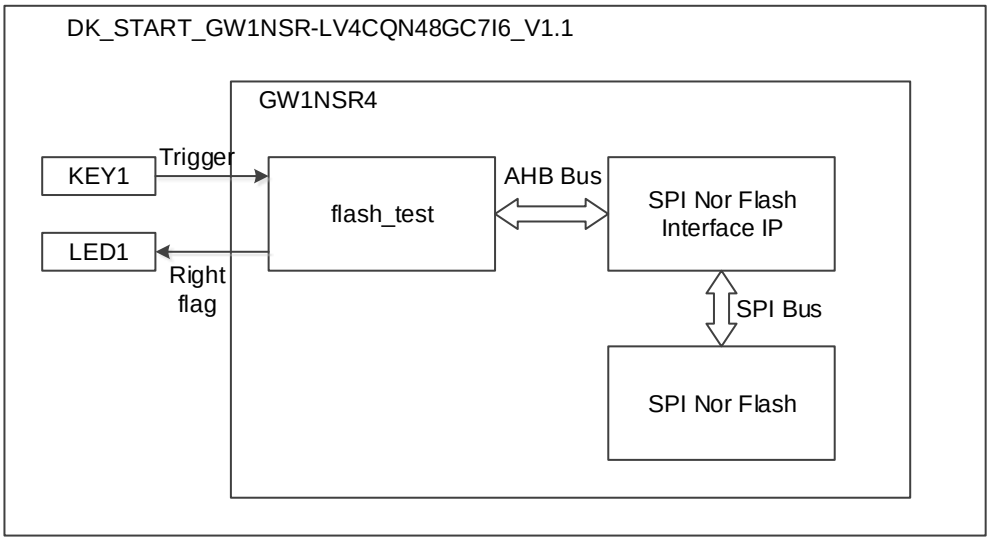
5 Reference Design

This chapter describes the usage of the reference design of SPI Nor Flash Interface IP. Please see the SPI Nor Flash Interface Reference Design for details at Gowinsemi website.

5.1 Application One

This reference design uses DK_START_GW1NSR-LV4CQN48GC7I6_V1.1 development board for an instance, the block diagram of the reference design is shown in Figure 5-1. For the details of DK_START_GW1NSR-LV4CQN48GC7I6_V1.1 development board, you can click here.

Figure 5-1 Block Diagram of Application One

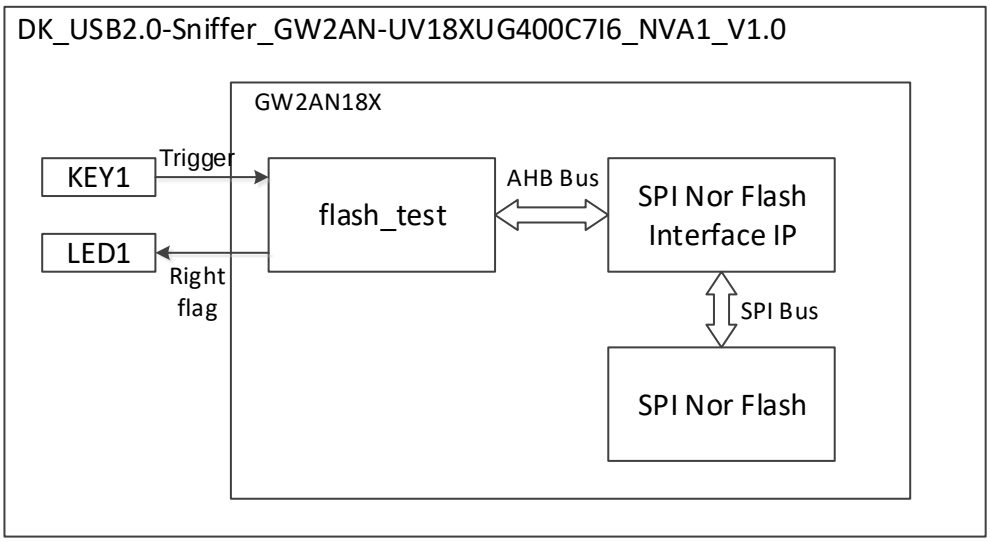


In the reference design, the flash_test module simulates the AHB bus master device; when key KEY1 is pressed, it triggers flash_test to issue write and read commands; then perform continuous write and continuous read operations to SPI Nor Flash through SPI Nor Flash Interface IP; and then compare whether the read data is correct with the write data. If it is correct, flash_test issues Right flag signal to LED1 for indication, and the light is on which indicates that the read and write are normal.

5.2 Application Two

This reference design uses DK_USB2.0-Sniffer_GW2AN-UV18XUG400C7I6_NVA1_V1.0 development board for an instance, the block diagram of the reference design is shown in Figure 5-2. For the details of DK_USB2.0-Sniffer_GW2AN-UV18XUG400C7I6_NVA1_V1.0 development board, you can click here.

Figure 5-2 Block Diagram of Application Two



In the reference design, the flash_test module simulates the AHB bus master device; when key KEY1 is pressed, it triggers flash_test to issue write and read commands; then perform continuous write and continuous read operations to SPI Nor Flash through SPI Nor Flash Interface IP; and then compare whether the read data is correct with the write data. If it is correct, flash_test issues Right flag signal to LED1 for indication, and the light is on which indicates that the read and write are normal.

6 File Delivery

The delivery files for the Gowin SPI Nor Flash Interface IP includes the document, the design source code, and the reference design.

6.1 Document

The document is listed as shown in Table 6-1.

Table 6-1 Document List

Name	Description
IPUG945, Gowin SPI Nor Flash Interface User Guide	Gowin SPI Nor Flash Interface IP User Guide, i.e., this manual.

6.2 Design Source Code (Encryption)

The Encryption Code Folder contains the RTL encryption code of Gowin SPI Nor Flash Interface IP used for the GUI, to generate the IP cores as needed.

Table 6-2 SPI Nor Flash Interface Design Source Code List

Name	Description
spi_nor_flash_interface.v	The top file of the IP core, which provides users with interface information, encrypted.

6.3 Reference Design

Gowin SPI Nor Flash Interface RefDesign 1NSR4 folder contains the netlist file, reference design, constraints file, top file and the project file of Gowin SPI Nor Flash Interface IP, etc.

Table 6-3 Gowin SPI Nor Flash Interface RefDesign 1NSR4 Folder Contents List

Name	Description
flash_test_top.v	The top module of reference design
flash_test.v	Test stimulus generation module
test_top.cst	Project physical constraints file
test_top.sdc	Project timing constraints file
gowin_pllvr	PLLVR IP folder
spi_nor_flash_interface	SPI Nor Flash Interface IP folder

Gowin SPI Nor Flash Interface RefDesign 2AN18X folder contains the netlist file, reference design, constraints file, top file and the project file of Gowin SPI Nor Flash Interface IP, etc.

Table 6-4 Gowin SPI Nor Flash Interface RefDesign 2AN18X Folder Contents List

Name	Description
flash_test_top.v	The top module of reference design
flash_test.v	Test stimulus generation module
test_top.cst	Project physical constraints file
test_top.sdc	Project timing constraints file
gowin_pllo	PLLO IP folder
spi_nor_flash_interface	SPI Nor Flash Interface IP folder


GOWIN
PROGRAMMING FOR THE FUTURE