Gowin SDRAM Controller

User Guide

IPUG279-1.3E,12/14/2018

Disclaimer

GOWINSEMI®, LittleBee®, Arora™, and the GOWINSEMI logos are trademarks of GOWINSEMI and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders, as described at www.gowinsemi.com.cn. 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. All information in this document should be treated as preliminary. 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
12/14/2017	1.3E	Initial version published.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

The purpose of this Gowin SDRAM Controller User Guide is to help users to quickly understand the features and functionality of the GOWINSEMI SDRAM controller by providing an overview of the associated functions, architecture view, principles, signal definition, parameters, GUI, and interface timing.

1.2 Supported Products

The information presented in this guide applies to the following products:

1. GW1N series of FPGA products: GW1N-4
2. GW1NR series of FPGA products: GW1NR-4
3. GW2A series of FPGA products: GW2A-18
4. GW2AR series of FPGA products: GW2AR-18

1.3 Related Documents

The latest user guides are available on the Gowin Semi website. Refer to the related documents at www.gowinsemi.com:

1. GW1N FPGA Products Data Sheet
2. GW1NR FPGA Products Data Sheet
3. GW2A FPGA Products Data Sheet
4. GW2AR FPGA Products Data Sheet
5. Gowin Software User Guide

1.4 Abbreviations and Terminology

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

Table 1-1: Abbreviations and Terminology

Abbreviations and Terminology	Full Name/Meaning
FPGA	Field Programmable Gate Array
SDRAM	Synchronous Dynamic Random Access Memory (RAM)
I/O	Input/Output
CL	CAS latency
tRP	PRECHARGE command period
tRFC	AUTO REFRESH period
tMRD	LOAD MODE REGISTER command to ACTIVE or REFRESH command period
tRCD	ACTIVE-to-READ or WRITE delay
tWR	WRITE recovery time

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

E-mail: support@gowinsemi.com

Tel: +86 755 8262 0391

2Function

2.1 Introduction

SDRAM is widely used for its high cost-performance. As SDRAM has strict timing requirements and complex logic control, a dedicated controller is required to implement the initialization, data read/write, refresh, and pre-charge functions.

The SDRAM controller can meet the needs of FPGA users by reducing the complexity of system development and decreasing product development time. The SDRAM is located between the SDRAM controller and the user logic modules. It offers a user-friendly interface and can read from or write to SDRAM.

The Gowin YunYuan software can be used with two different SDRAM controllers per the users' needs. The first is an external SDRAM controller for GW1N-4, GW1A-18, etc. This controller should be configured according to the external SDRAM. The second is an embedded SDRAM controller for GW2AR-18, GW1NR-4, etc., and this SDRAM IP can be generated directly without the need to set parameters.

Users can utilize the Gowin YunYuan software to generate the SDRAM controller, configure the parameters, place and route, etc.

2.2 Features

2.2.1 External SDRAM Controller

- Configurable width of SDRAM data, address, row/column, etc.
- Configurable SDRAM refresh times, etc.
- Configurable timing parameters.
- Configurable SDRAM working clock cycle.
- Can support any length read/write operation within the range of 1~Page.
- Supports auto-refresh.
- Supports self-refresh.
- Supports power-down.

- Supports industry-standard SDRAM.
- Fully synthesizable.

2.2.2 Embedded SDRAM Controller

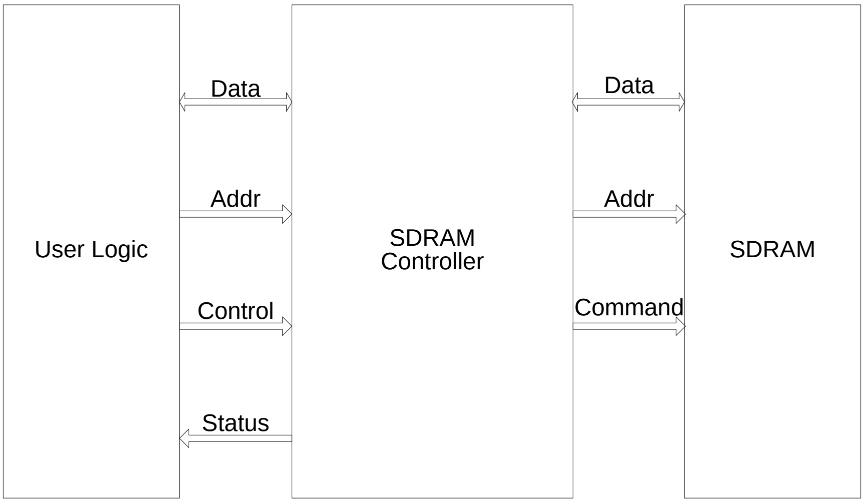
- Displays SDRAM interface parameter values and timing parameters.
- Supports any length read/write operation within the range of 1~Page.
- Supports auto-refresh.
- Supports self-refresh.
- Supports power-down.
- Supports industry-standard SDRAM.
- Fully synthesizable.
- GW1NR-4 supports two kinds of embedded controller: 32-bit data width and 16-bit data width.

3 Signal Definition

3.1 Introduction

The SDRAM controller is located between the SDRAM and the user logic modules, as shown in Figure 3-1. The signals that are available on the user logic side predominantly consist of data signal, address signal, control signal, status signal, etc. The signals on the SDRAM side mainly include data signal, address signal, and command signal.

Figure 3-1: System Architecture



3.2 SDRAM Signals

Table 3-1: Description of SDRAM Signals

No.	Signal Name	I/O	Description	Remarks
1	O_sdram_clk	I	Clock	The I/O direction of all signals takes the SDRAM as reference.
2	O_sdram_cke	I	Clock enable	
3	O_sdram_cs_n	I	Chip select	
4	O_sdram_cas_n	I	Column address strobe	
5	O_sdram_ras_n	I	Row address strobe	
6	O_sdram_wen_n	I	Write enable	
7	O_sdram_dqm	I	Data mask control	
8	O_sdram_addr	I	Address	
9	O_sdram_ba	I	Bank address	
10	IO_sdram_dq	I/O	Data	

3.3 User Signals

Table 3-2: Description of User Signals

No.	Signal Name	I/O	Description	Remarks
1	I_sdrc_rst_n	I	Reset signal, active low	The I/O direction of all signals takes the SDRAM as reference.
2	I_sdrc_clk	I	SDRAM controller working clock	
3	I_sdram_clk	I	SDRAM working clock	
4	I_sdrc_selfrefresh	I	Self-refresh control 1: self-refresh enable 0: self-refresh disable	
5	I_sdrc_power_down	I	Lower power consumption control 1: lower power consumption enable 0: lower power consumption disable	
6	I_sdrc_wr_n	I	Write enable Active-low, one clock cycle of pulse width	
7	I_sdrc_rd_n	I	Read enable Active-low, one clock cycle of pulse width	
8	I_sdrc_addr	I	Address	
9	I_sdrc_dqm	I	Data mask control	
10	I_sdrc_data_len	I	Read/write data length Take I_sdrc_data_len[7:0] for instance, the port input range is 0~255, and the corresponding valid read/write length range is 1~256, and so on.	
11	I_sdrc_data	I	Write data	
12	O_sdrc_data	O	Read data	
13	O_sdrc_init_done	O	Power-on initialization indication 1: done 0: undone	
14	O_sdrc_busy_n	O	Idle/busy controller indication User logic can issue read/write operation when the controller is idle. 1: idle 0: busy	
15	O_sdrc_rd_valid	O	Active data reading indication, high active, and aligns with valid data	
16	O_sdrc_wrd_ack	O	Read/write request response, active high After receiving the read/write request, the SDRAM controller sends the signal after two clock delays, one clock cycle width	

4GUI Parameters

4.1 Introduction

Users can use the SDRAM controller GUI to set the SDRAM interface data, address width, etc., and the timing value of the SDRAM controller.

For the external SDRAM controller, users can set the parameters on the "Customize IP" page of the Gowin YunYuan software according to the value of the detailed external SDRAM parameters.

For the embedded SDRAM controller, users do not need to set the parameters because the SDRAM controller is embedded in FPGA devices and the parameters are fixed.

4.2 External SDRAM GUI Parameters

Table 4-1: Interface Parameters

No.	Name	Value Range	Default Value	Description
1	Data Width	8/16/32/64	32	SDRAM data bus width
2	Bank Width	1/2	2	SDRAM BANK address width
3	Row Width	11/12/13/14	11	SDRAM row address width
4	Column Width	8/9/10/11/12	8	SDRAM column address width

Table 4-2: Time Delay Parameter

No.	Name	Range	Default Value	Description	Remark
1	Clock Period	≥4, ≤100	10	Working clock cycle of SDRAM controller, in ns.	Used for auto-refresh.
2	Refresh Period	≥1	64000000	Auto-refresh cycle of SDRAM, in ns.	The value is determined by the specific chip parameters.
3	Refresh Times	≥1	4096	Auto-refresh times of SDRAM.	
4	CL Period	1/2/3	3	CAS latency.	The specific value is determined by the specific chip parameters. The value is the number of the controller clock.
5	tRP Period	≥1	3	PRECHARGE command period.	
6	tRFC Period	≥1	9	AUTO REFRESH period.	
7	tMRD Period	≥1	3	LOAD MODE REGISTER command to ACTIVE or REFRESH command period.	
8	tRCD Period	≥1	3	ACTIVE-to-READ or WRITE delay.	
9	tWR Period	≥1	3	WRITE recovery time.	

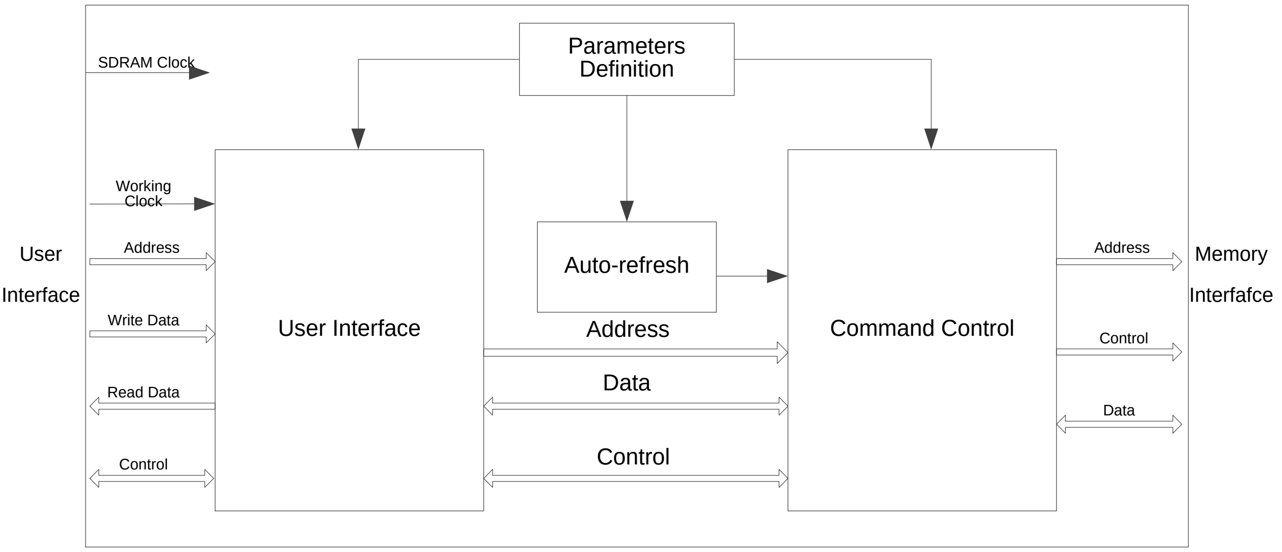
5 Principle

5.1 Introduction

The SDRAM controller offers parameter configuration, auto-refresh, user interfaces, and command control. The auto-refresh, data read/write, etc., functions are achieved through communication with the other modules.

5.2 Principle Diagram

Figure 5-1: Principle Diagram



5.3 SDRAM Commands

The common SDRAM commands include no operation, active, read operation, write operation, pre-charge, auto-refresh, configuration mode register, etc. Table 5-1 provides an overview of the operation and definition of the commands that are available.

Table 5-1: Definition of the SDRAM Commands

Command Name	CS	RAS	CAS	WE
Command Inhibit	H	X	X	X
NOP	L	H	H	H
Active	L	L	H	H
Read	L	H	L	H
Write	L	H	L	L
Burst Terminate	L	H	H	L
Pre-charge	L	L	H	L
Auto Refresh or Self Refresh	L	L	L	H
Configuration Mode Register	L	L	L	L
Write Enable	X	X	X	X
Write Inhibit	X	X	X	X

Notes!

- X: Any; L: Low level; H: High level
- CS: chip select; RAS: row selection; CAS: column selection; WE: enable

5.4 Initialization

The SDRAM must be initialized using the SDRAM controller before being powered up. After power is applied to the VDD and VDDQ (simultaneously) and the clock is stable, the SDRAM requires a 100us delay prior to issuing any commands other than the INHIBIT or NOP commands. Following that, two auto-refresh cycles must be performed. After the auto-refresh cycles are complete, the SDRAM is ready for mode register programming. The mode register is programmed via the 0/1 signals from the address line. After programming the mode register, the user will need to wait at least tMRD time. The SDRAM will enter the normal working state.

5.5 Refresh

SDRAM refresh options consist of self-refresh and auto-refresh.

- Self-refresh operation: When the self-refresh command is initiated and CKE is set to low, self-refresh mode is engaged. SDRAM saves data via self-refresh. To exit the self-refresh mode, set CEK to high when the self-refresh command is initiated.
- Auto-refresh operation: The auto-refresh command is used to refresh the parameters according to different chips. The SDRAM controller will send auto-refresh commands periodically to trigger the SDRAM to refresh and retain the data.

5.6 Read Operation

After initialization, if the SDRAM controller needs to address the array in one bank, confirm the row first and activate it before confirming the column.

The specific flow is as follows:

1. Activate the row address of the bank from which to read data.
2. Send the read command after the tRCD delay and simultaneously write the corresponding column address.
3. The SDRAM will begin to output the data after CL.
4. After reading the date, pre-charge the corresponding bank (or all banks). Perform the other operations after the tRP period.

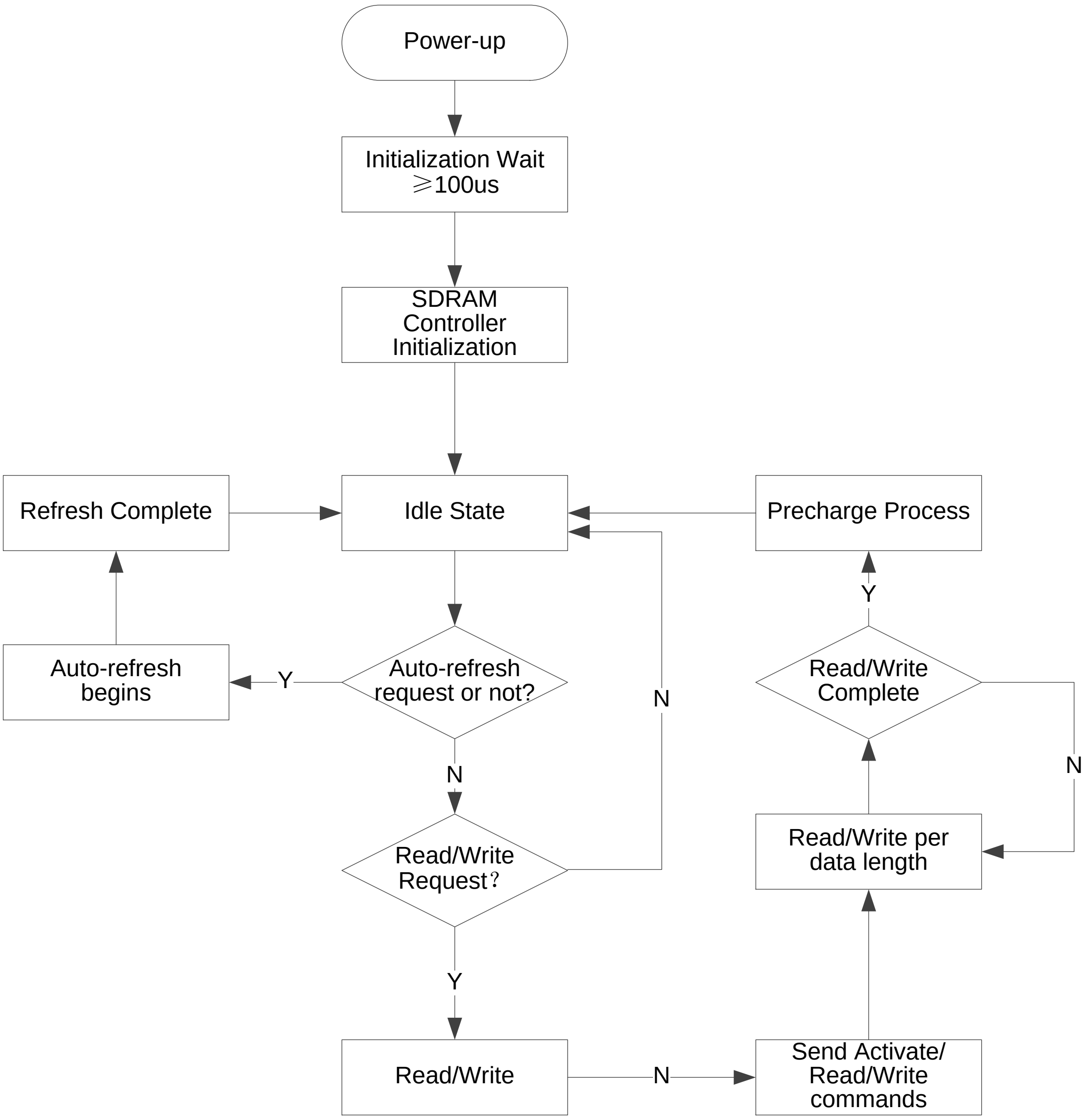
5.7 Write Operation

The flow of the write operation is the same as that of the read operation. Please refer to Section 5.6 Read Operation for further details.

5.8 Working Flow

5.8.1 Read/Write Flow

Figure 5-2: Read/Write Working Flow


Power-up
Initialization Wait ⩾100us
SDRAM Controller Initialization
Refresh Complete
Idle State
Precharge Process
Auto-refresh begins
Y
Auto-refresh request or not?
N
Y
Read/Write Complete
N
N
Read/Write Request？
Read/Write per data length
Y
Read/Write
N
Send Activate/ Read/Write commands

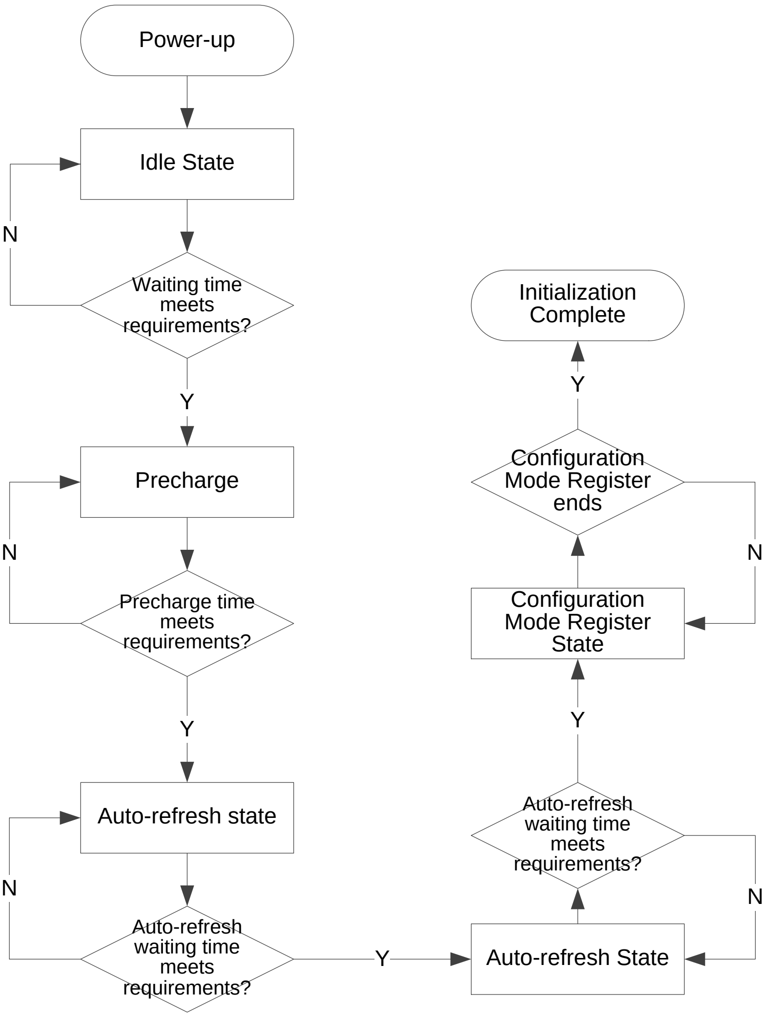
The specific read/write flow is as follows:

1. Initialization wait, go to Step (2) after the preset waiting time.
2. Start SDRAM initialization, and progress to Step (3).
3. Put the SDRAM controller in an idle state.
4. Verify whether there is auto-refresh request; if yes, auto-refresh before progressing to Step (3); if not, go to Step (5).
5. Verify whether the interfaces on the user side have received a read/write control request; if yes, start the read-write data address and length processing first and then send the read/write request before progressing to Step (6); if no, return to Step (3).

6. Send Activate, Read/Write commands to SDRAM. Proceed to Step (7).
7. Read/Write according to the data length.
8. Verify whether the read/write operation has ended or not; if yes, proceed to Step (9); if not, return to Step (7).
9. Precharge and return to Step (3) after precharging is complete.

5.8.2 Initialization Flow

Figure 5-3: Initialization Flow


Power-up
Idle State
N
Waiting time meets requirements?
Y
Precharge
N
Precharge time meets requirements?
Y
Auto-refresh state
N
Auto-refresh waiting time meets requirements?
Y
Auto-refresh State
N
Auto-refresh waiting time meets requirements?
Y
Configuration Mode Register State
N
Configuration Mode Register ends
Y
Initialization Complete

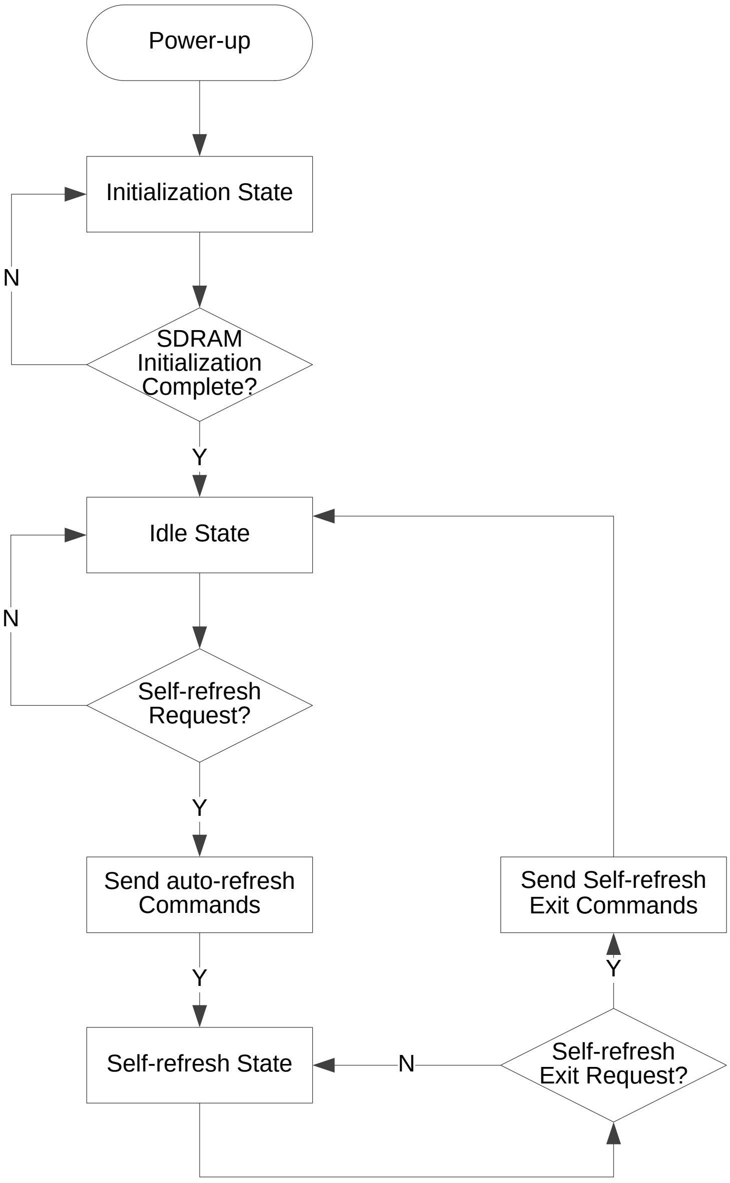
The detailed initialization flow is as follows:

1. After power-up, the SDRAM controller enters an idle state; progress to Step (2) after the preset initialization waiting time, or keep waiting.
2. Precharge and send a pre-charging command; progress to Step (3) after the preset waiting time, or keep waiting.
3. Auto-refresh and send an auto-refresh command; progress to Step (4) after the preset waiting time, or keep waiting.

4. Auto-refresh and send an auto-refresh command; progress to Step (5) after the preset waiting time, or keep waiting.
5. Enters the configuration mode register state and send configuration mode register command; progress to Step (6) after the preset waiting time, or keep waiting.
6. Initialization is complete.

5.8.3 Auto-refresh Flow

Figure 5-4: Auto-refresh Flow


Power-up
Initialization State
N
SDRAM Initialization Complete?
Y
Idle State
N
Self-refresh Request?
Y
Send auto-refresh Commands
Y
Self-refresh State
N
Self-refresh Exit Request?
Y
Send Self-refresh Exit Commands

The auto-refresh flow process is as follows:

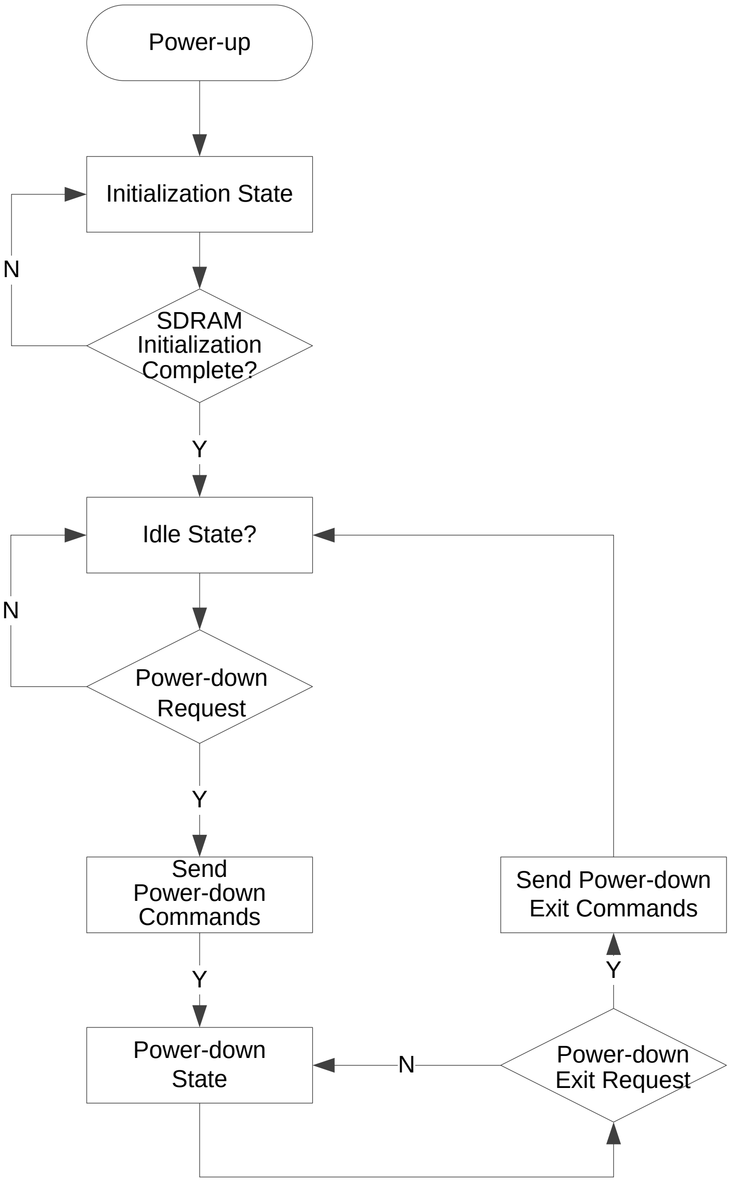
1. After power-up, the SDRAM controller enters a waiting state.
2. Verify whether the initialization is complete or not; if yes, progress to

Step (3); if no, keep waiting.

3. Verify whether there is an auto-refresh request; if yes, proceed to Step (4).
4. Send an auto-refresh command; progress to Step (5).
5. The xxx enters the auto-refresh state. Verify whether there is auto-refresh exiting request; if yes, progress to Step (6).
6. Send an auto-refresh exiting command, and then return to the idle state.

5.8.4 Power-down Flow

Figure 5 5: Power-down Flow



The power-down flow is as follows:

1. After power-up, the SDRAM controller enters a waiting state.

2. Verify whether the initialization is complete or not; if yes, go to Step (3); if no, keep waiting.
3. Verify whether there is power-down request; if yes, proceed to Step (4).
4. Send a power-down command. Proceed to Step (5).
5. XX enters the power-down state. Verify whether there is a power-down exiting request; if yes, proceed to Step (6).
6. Send a power-down exiting command, and then return to the idle state.

6Examples and Explanation

6.1 Introduction

This chapter described the use of the Gowin YunYuan Software interface to configure parameters and generate a SDRAM controller module according to user requirements. Application cautions are also described.

When you instantiate an embedded SDRAM controller in your design, the I/O ports connected with the SDRAM controller in the top design module should have the same name as that of the embedded SDRAM controller. Gowin YunYuan software will perform place and route automatically according to the designed I/O port name, which facilitates smooth communication between the SDRAM controller and the SDRAM.

Examples of the operation flow of the two controllers are described below.

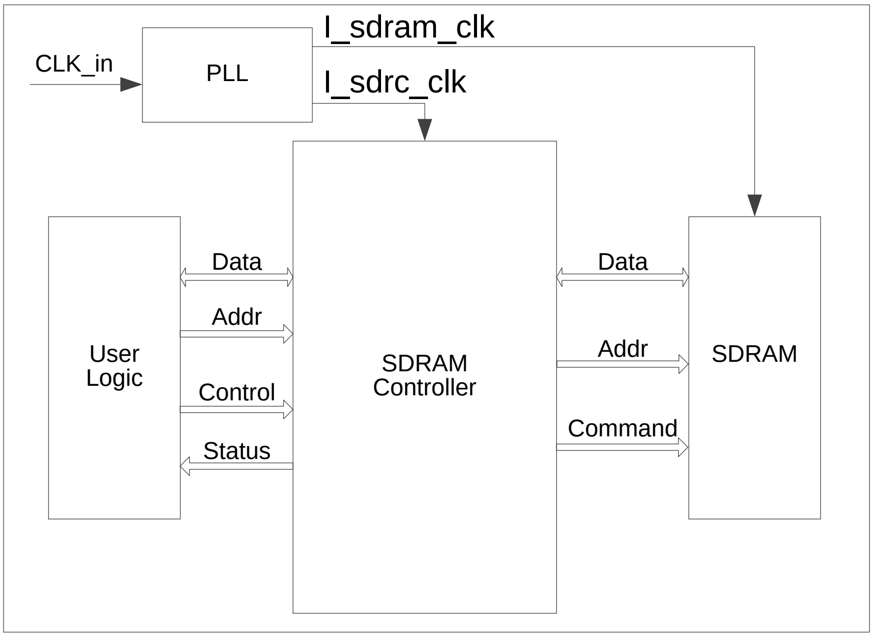
6.2 Application

The SDRAM controller application mainly includes PLL, user logic, SDRAM controller, and SDRAM. Figure 6-1 shows the SDRAM application diagram.

As an internal clock resource of FPGA, PLL can generate SDRAM working clock I_sdram_clk and SDRAM controller working clock I_sdrc_clk via frequency multiplication and division. When the SDRAM controller works at a high speed, the phase of I_sdrc_clk needs to be adjusted to meet the setup/hold time of the SDRAM side read/write signals.

All input/output signals of the user logic interface align with the rising edge of I_sdrc_clk.

Figure 6-1: Application Diagram



Notes!

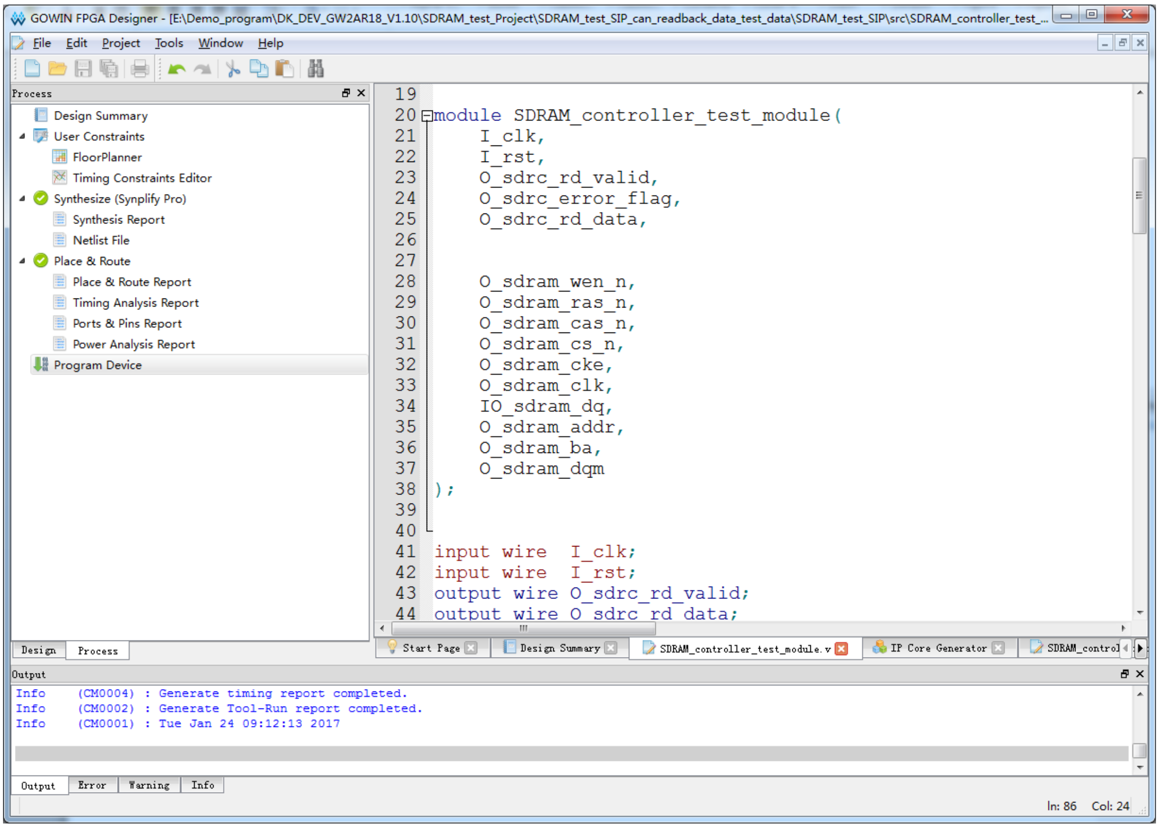
- When the SDRAM controller works at a high speed, you can adjust the phase of I_sdram_clk to meet the setup/hold time of the SDRAM side read/write signals.
- For external SDRAM, the input/output DFF of the SDRAM side can be constrained to IOLOGIC DFF to improve the transmission performance.
- Considering internal delay and PCB routing, etc., when the SDRAM controller works at a high speed, if reading data has one cycle delay in excess of the active indicator, you need to delay the active indicator for one cycle so that it is aligned with the active data.
- For embedded SDRAM, the signals on the SDRAM side occur on Module ports after the SDRAM controller is called. These signals should be led to the TOP layer and retain the same names. These signals do not need constraints in .cst files.
- When the GW1NR-4 32-bit width controller is called, the clock frequency of I_sdrc_clk must be half that of I_sdram_clk frequency; I_sdrc_clk and I_sdram_clk have the same clock frequency in other cases.

6.3 Example of an Embedded SDRAM Controller

6.3.1 Open an Existing Project

Start the Gowin YunYuan Software and open an existing project, as shown in Figure 6-2.

Figure 6-2: Open an Existing Project



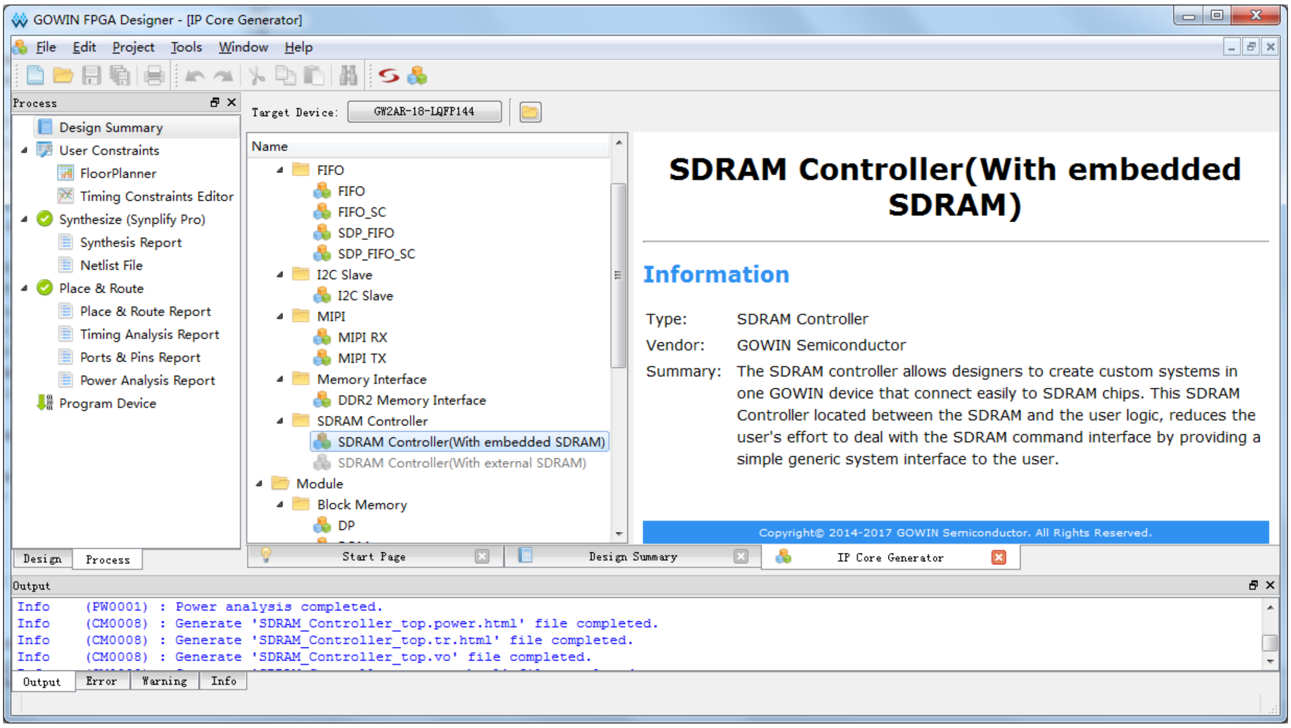
```

module SDRAM_controller_test_module(
    I_clk,
    I_rst,
    O_sdrc_rd_valid,
    O_sdrc_error_flag,
    O_sdrc_rd_data,


    O_sdram_wen_n,
    O_sdram_ras_n,
    O_sdram_cas_n,
    O_sdram_cs_n,
    O_sdram_cke,
    O_sdram_clk,
    IO_sdram_dq,
    O_sdram_addr,
    O_sdram_ba,
    O_sdram_dqm
);


input wire  I_clk;
input wire  I_rst;
output wire O_sdrc_rd_valid;
output wire O_sdrc_rd_data;
```

```
Info    (CM0004) : Generate timing report completed.
Info    (CM0002) : Generate Tool-Run report completed.
Info    (CM0001) : Tue Jan 24 09:12:13 2017
```

6.3.2 SDRAM Controller Generation

Click “Tools > IP Core Generator”. The window displayed in Figure 6-3 will appear.

Figure 6-3: IP Core Generator Page



Target Device: GW2AR-18-LQFP144

SDRAM Controller(With embedded SDRAM)

Information

Type: SDRAM Controller

Vendor: GOWIN Semiconductor

Summary: The SDRAM controller allows designers to create custom systems in one GOWIN device that connect easily to SDRAM chips. This SDRAM Controller located between the SDRAM and the user logic, reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the user.

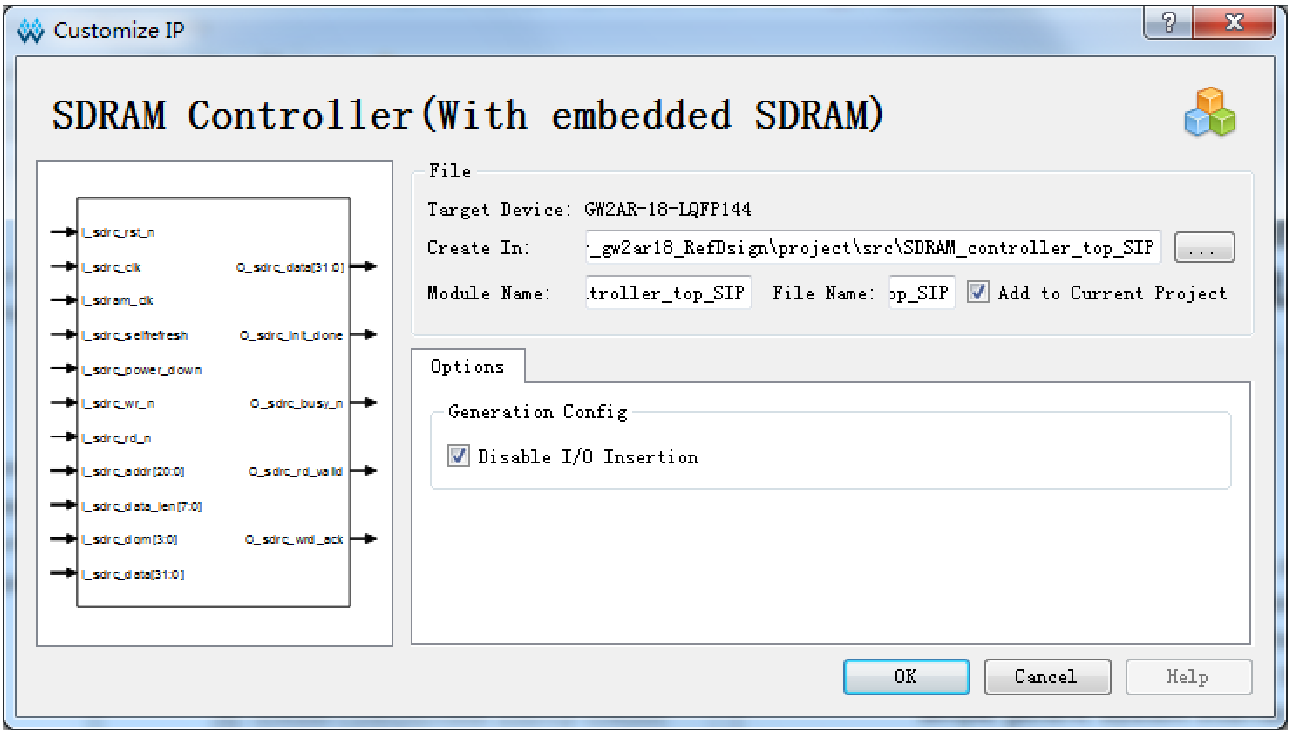
Copyright© 2014-2017 GOWIN Semiconductor. All Rights Reserved.

Output

```
Info    (PW0001) : Power analysis completed.
Info    (CM0008) : Generate 'SDRAM_Controller_top.power.html' file completed.
Info    (CM0008) : Generate 'SDRAM_Controller_top.tr.html' file completed.
Info    (CM0008) : Generate 'SDRAM_Controller_top.vo' file completed.
```

Select GW2AR-18-LQFP144, and then double-click "SDRAM Controller (With embedded SDRAM)" to open the "Customize IP" page, as shown in Figure 6-4. Accept the default settings displayed on this page and click "OK" to generate the SDRAM_controller_top_SIP.

Figure 6-4: SDRAM Controller Interface



SDRAM Controller(With embedded SDRAM)

File

Target Device: GW2AR-18-LQFP144

Create In: _gw2ar18_RefDsign\project\src\SDRAM_controller_top_SIP

Module Name: troller_top_SIP File Name: op_SIP ☑ Add to Current Project

Options

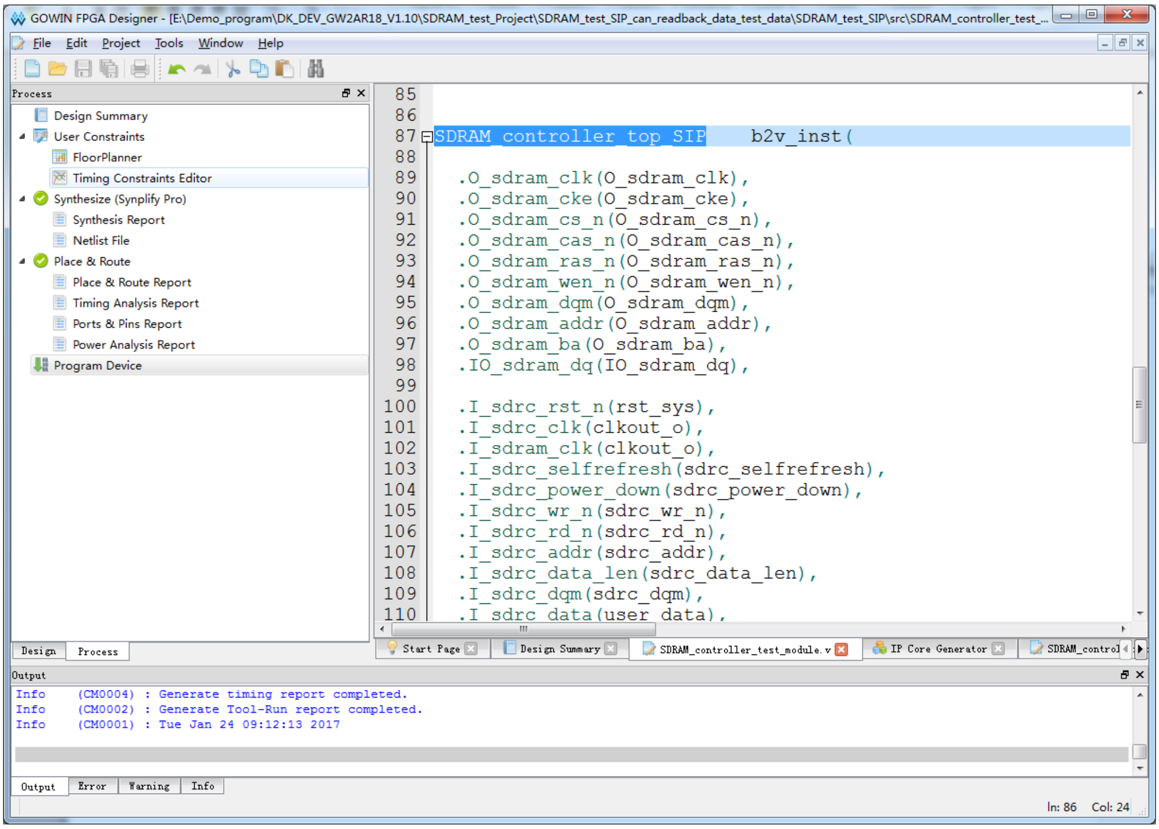
Generation Config

☑ Disable I/O Insertion

6.3.3 SDRAM Controller Instantiation

Instantiate the module "SDRAM_controller_top_SIP” in a project, as shown in Figure 6-5.

Figure 6-5: SDRAM_controller_top_SIP Instantiation



```


SDRAM_controller_top_SIP     b2v_inst(

  .O_sdram_clk(O_sdram_clk),
  .O_sdram_cke(O_sdram_cke),
  .O_sdram_cs_n(O_sdram_cs_n),
  .O_sdram_cas_n(O_sdram_cas_n),
  .O_sdram_ras_n(O_sdram_ras_n),
  .O_sdram_wen_n(O_sdram_wen_n),
  .O_sdram_dqm(O_sdram_dqm),
  .O_sdram_addr(O_sdram_addr),
  .O_sdram_ba(O_sdram_ba),
  .IO_sdram_dq(IO_sdram_dq),

  .I_sdrc_rst_n(rst_sys),
  .I_sdrc_clk(clkout_o),
  .I_sdram_clk(clkout_o),
  .I_sdrc_selfrefresh(sdrc_selfrefresh),
  .I_sdrc_power_down(sdrc_power_down),
  .I_sdrc_wr_n(sdrc_wr_n),
  .I_sdrc_rd_n(sdrc_rd_n),
  .I_sdrc_addr(sdrc_addr),
  .I_sdrc_data_len(sdrc_data_len),
  .I_sdrc_dqm(sdrc_dqm),
  .I_sdrc_data(user_data),
```

```
Info     (CM0004) : Generate timing report completed.
Info     (CM0002) : Generate Tool-Run report completed.
Info     (CM0001) : Tue Jan 24 09:12:13 2017
```

6.3.4 Bitstream File Generation

After inputting the necessary constraints, begin to synthesize, P&R, and then generate the bitstream files.

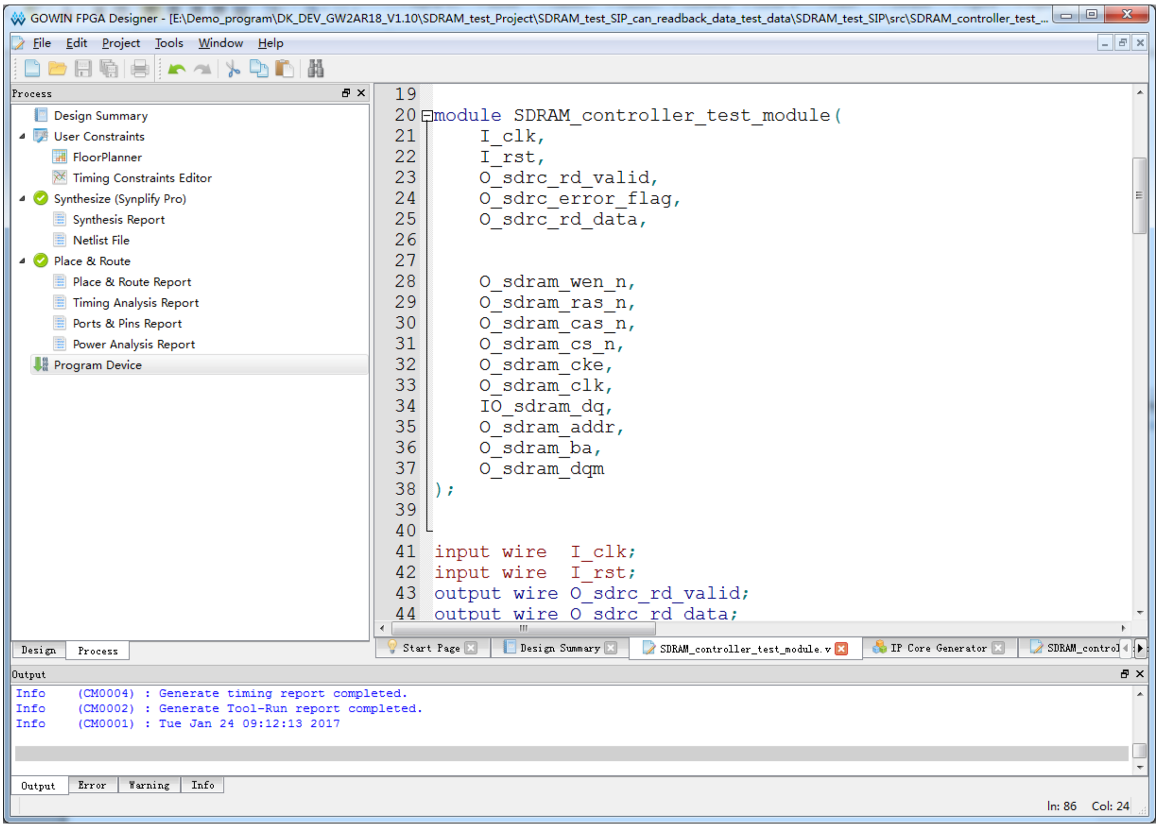
Load the bitstream files to the development board or test board using the Gowin download cable. You can also observe the SDRAM read/write operation via the test interface.

6.4 Example of an External SDRAM Controller

6.4.1 Open an Existing Project

Start the Gowin YunYuan Software and open an existing project, as shown in Figure 6-6.

Figure 6-6: Open an Existing Project



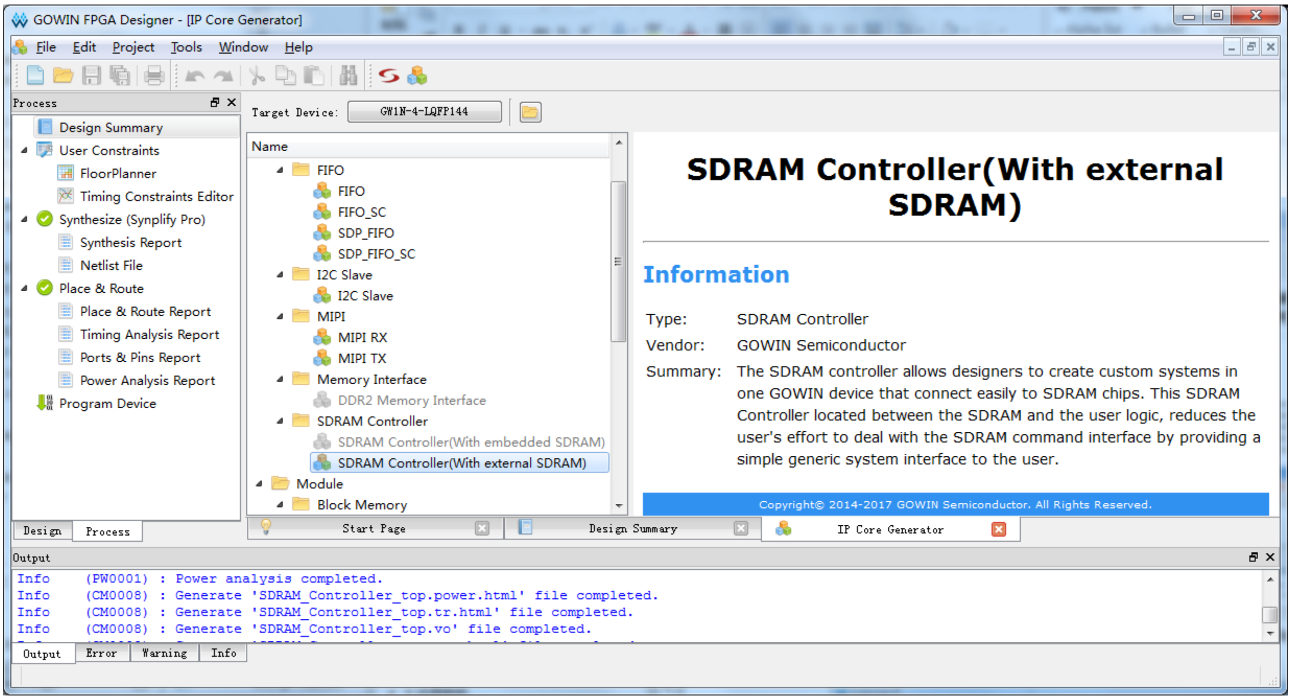
```
module SDRAM_controller_test_module(
    I_clk,
    I_rst,
    O_sdrc_rd_valid,
    O_sdrc_error_flag,
    O_sdrc_rd_data,


    O_sdram_wen_n,
    O_sdram_ras_n,
    O_sdram_cas_n,
    O_sdram_cs_n,
    O_sdram_cke,
    O_sdram_clk,
    IO_sdram_dq,
    O_sdram_addr,
    O_sdram_ba,
    O_sdram_dqm
);


input wire  I_clk;
input wire  I_rst;
output wire O_sdrc_rd_valid;
output wire O_sdrc_rd_data;
```

```
Info    (CM0004) : Generate timing report completed.
Info    (CM0002) : Generate Tool-Run report completed.
Info    (CM0001) : Tue Jan 24 09:12:13 2017
```

6.4.2 SDRAM Controller Generation

Click “Tools > IP Core Generator”. The window displayed in Figure 6-7 will appear.

Figure 6-7: IP Core Generator Page



SDRAM Controller(With external SDRAM)

Information

Type: SDRAM Controller

Vendor: GOWIN Semiconductor

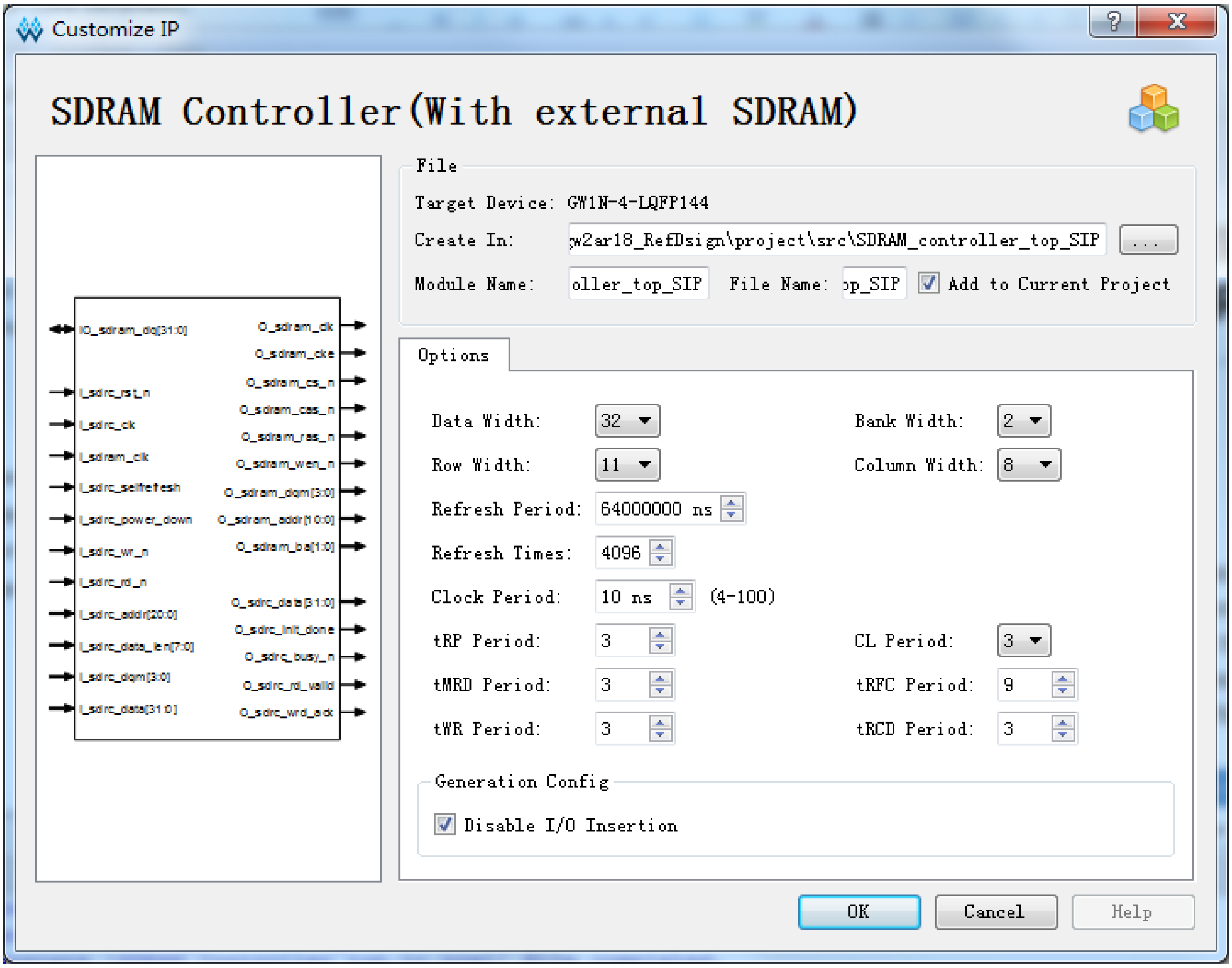
Summary: The SDRAM controller allows designers to create custom systems in one GOWIN device that connect easily to SDRAM chips. This SDRAM Controller located between the SDRAM and the user logic, reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the user.

Copyright© 2014-2017 GOWIN Semiconductor. All Rights Reserved.

```
Info     (PW0001) : Power analysis completed.
Info     (CM0008) : Generate 'SDRAM_Controller_top.power.html' file completed.
Info     (CM0008) : Generate 'SDRAM_Controller_top.tr.html' file completed.
Info     (CM0008) : Generate 'SDRAM_Controller_top.vo' file completed.
```

Select GW1N-4-LQFP144, and then double-click on “SDRAM Controller (with external SDRAM)” to open the "Customize IP" page, as shown in Figure 6-8. Accept the default settings on this page and click “OK” to generate the SDRAM_controller_top_SIP.

Figure 6-8: SDRAM Controller Interface



SDRAM Controller(With external SDRAM)

File

Target Device: GW1N-4-LQFP144

Create In: gw2ar18_RefDsign\project\src\SDRAM_controller_top_SIP

Module Name: oller_top_SIP File Name: op_SIP ☑ Add to Current Project

Options

Data Width: 32

Bank Width: 2

Row Width: 11

Column Width: 8

Refresh Period: 64000000 ns

Refresh Times: 4096

Clock Period: 10 ns (4-100)

tRP Period: 3

CL Period: 3

tMRD Period: 3

tRFC Period: 9

tWR Period: 3

tRCD Period: 3

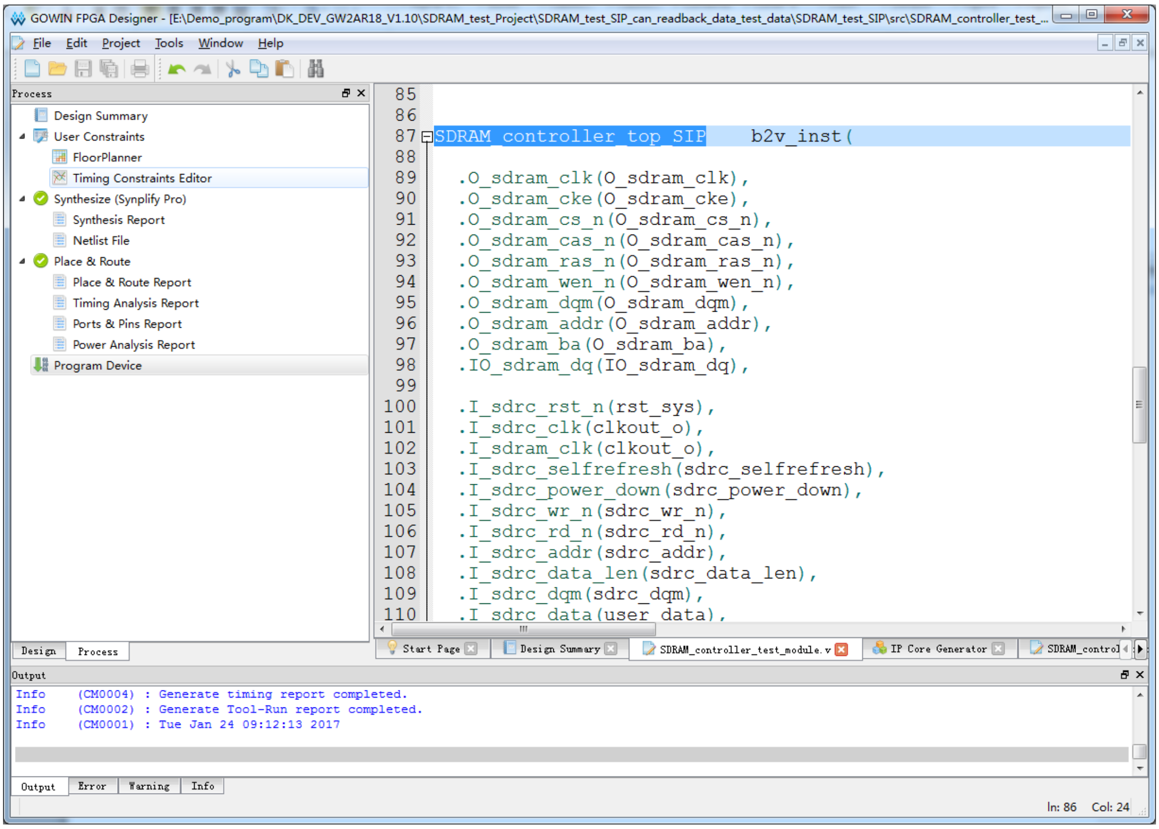
Generation Config

☑ Disable I/O Insertion

6.4.3 SDRAM Controller Instantiation

Instantiate the "SDRAM_controller_top_SIP” module in a project, as shown in Figure 6-9.

Figure 6-9: SDRAM_controller_top_SIP Instantiation



```


SDRAM_controller_top_SIP    b2v_inst(

    .O_sdram_clk(O_sdram_clk),
    .O_sdram_cke(O_sdram_cke),
    .O_sdram_cs_n(O_sdram_cs_n),
    .O_sdram_cas_n(O_sdram_cas_n),
    .O_sdram_ras_n(O_sdram_ras_n),
    .O_sdram_wen_n(O_sdram_wen_n),
    .O_sdram_dqm(O_sdram_dqm),
    .O_sdram_addr(O_sdram_addr),
    .O_sdram_ba(O_sdram_ba),
    .IO_sdram_dq(IO_sdram_dq),

    .I_sdrc_rst_n(rst_sys),
    .I_sdrc_clk(clkout_o),
    .I_sdram_clk(clkout_o),
    .I_sdrc_selfrefresh(sdrc_selfrefresh),
    .I_sdrc_power_down(sdrc_power_down),
    .I_sdrc_wr_n(sdrc_wr_n),
    .I_sdrc_rd_n(sdrc_rd_n),
    .I_sdrc_addr(sdrc_addr),
    .I_sdrc_data_len(sdrc_data_len),
    .I_sdrc_dqm(sdrc_dqm),
    .I_sdrc_data(user_data),
```

```
Info    (CM0004) : Generate timing report completed.
Info    (CM0002) : Generate Tool-Run report completed.
Info    (CM0001) : Tue Jan 24 09:12:13 2017
```

6.4.4 Bitstream File Generation

After inputting the necessary constraints, begin to synthesize, P&R, and then generate the bitstream files.

Load the bitstream files to the development board or test board using the Gowin download cable. You can also observe the SDRAM read/write operation via the test interface.

7 Interface Timing

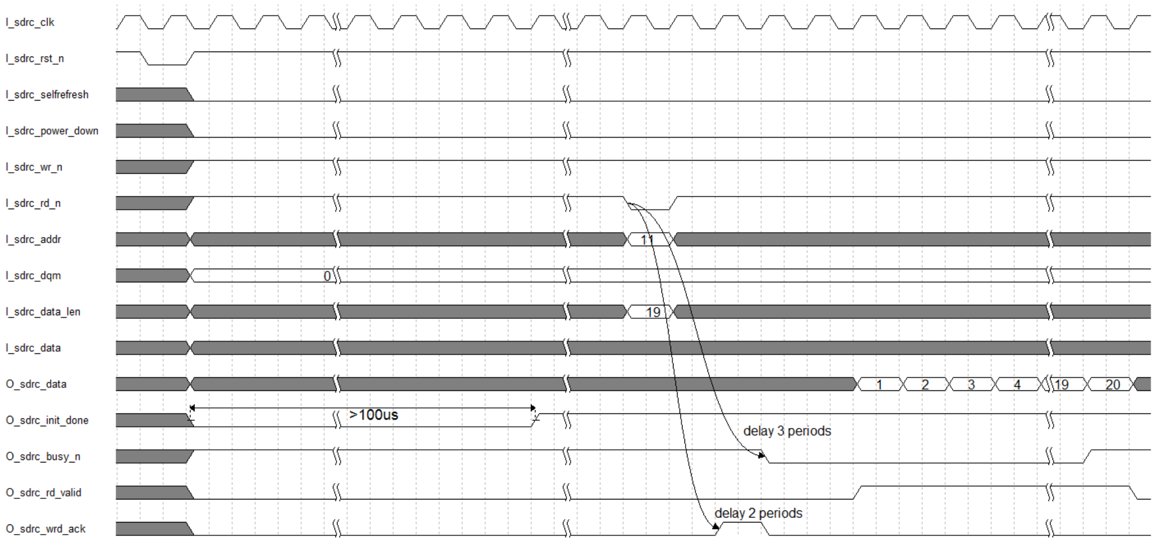
7.1 Introduction

Interface timing includes user interface timing and SDRAM interface timing. The user interface timing mainly includes read timing and write timing. All user interface input/output signals align at the rising edge of the controller working clock. The SDRAM interface timing mainly includes initialization, read timing, write timing, and auto-refresh timing. All output signals align at the rising edge of the controller working clock.

7.2 User Interface Timing

7.2.1 User Interface Read Operation

Figure 7-1: User Interface Read Operation

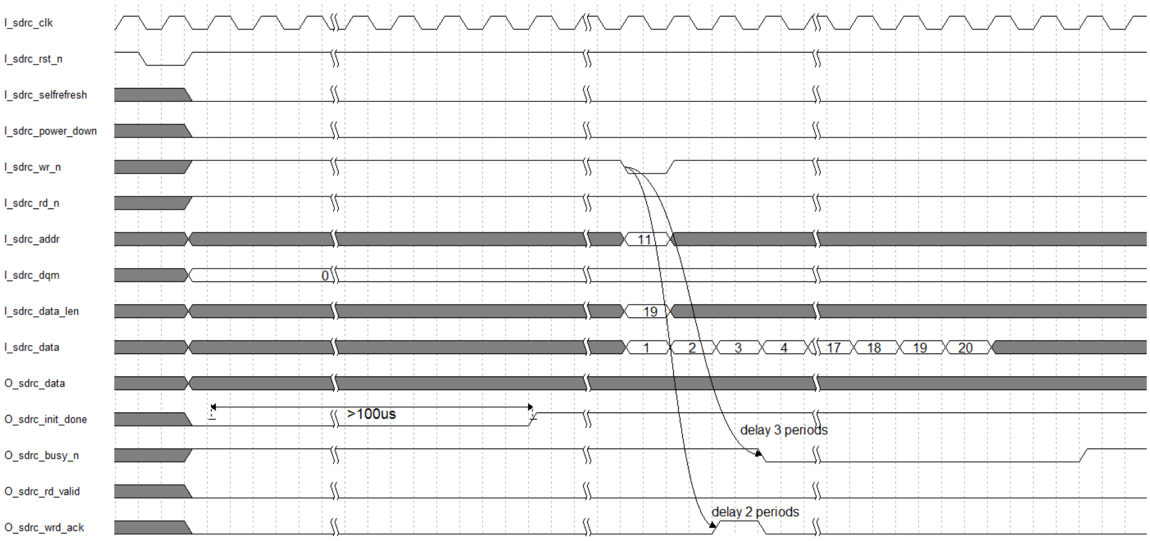


Note!

The actual data read/write length of the SDRAM controller is I_sdrc_data_len+1.

7.2.2 User Interface Write Operation

Figure 7-2: User Interface Write Operation



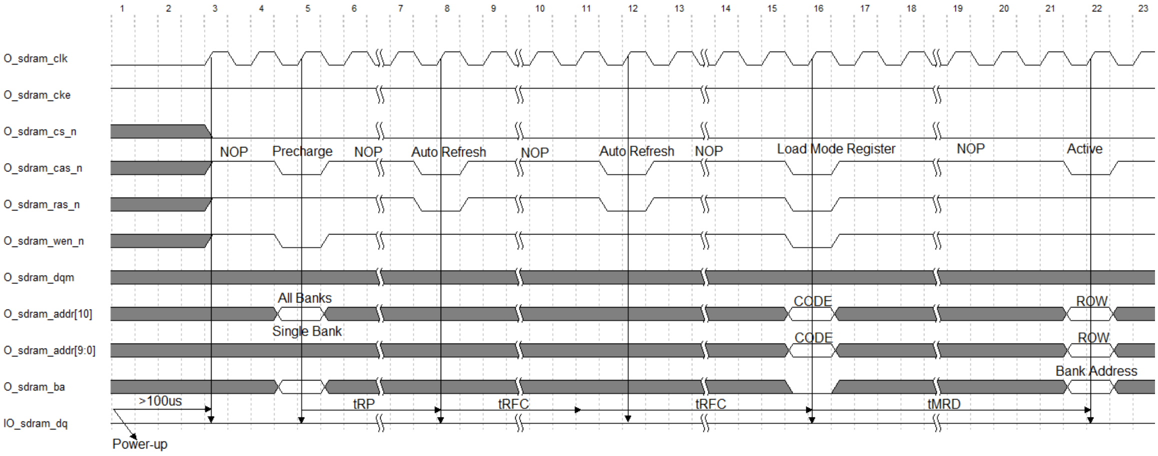
Note!

The actual data read/write length of the SDRAM controller is I_sdrc_data_len+1.

7.3 SDRAM Interface Timing

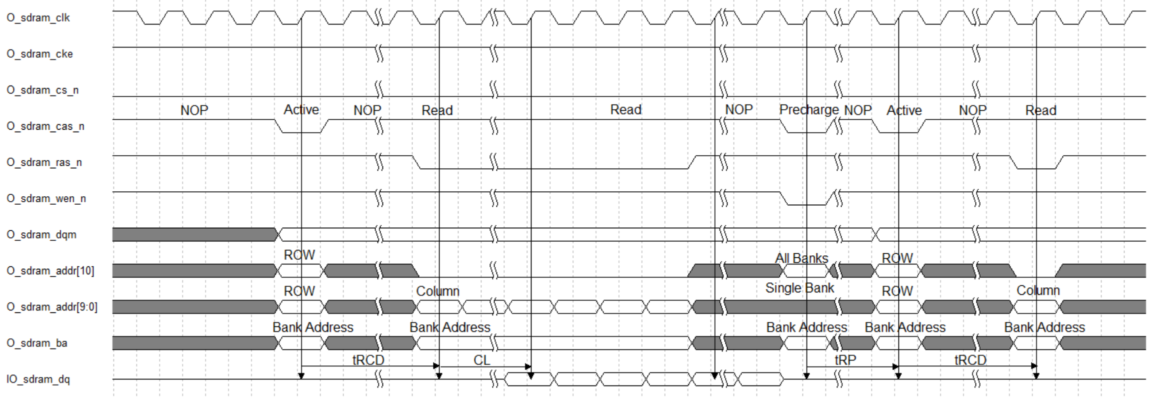
7.3.1 SDRAM Initialization

Figure 7-3: SDRAM Initialization



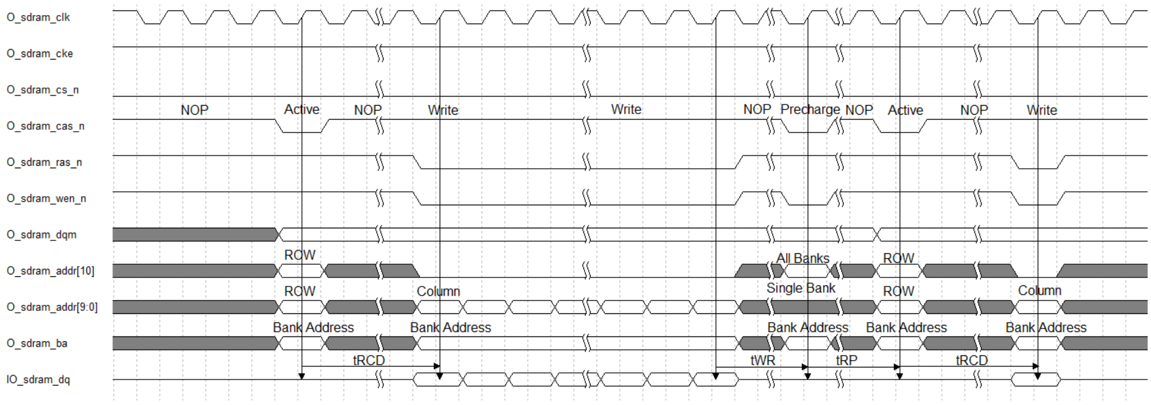
7.3.2 SDRAM Read Operation

Figure 7-4: SDRAM Read Operation



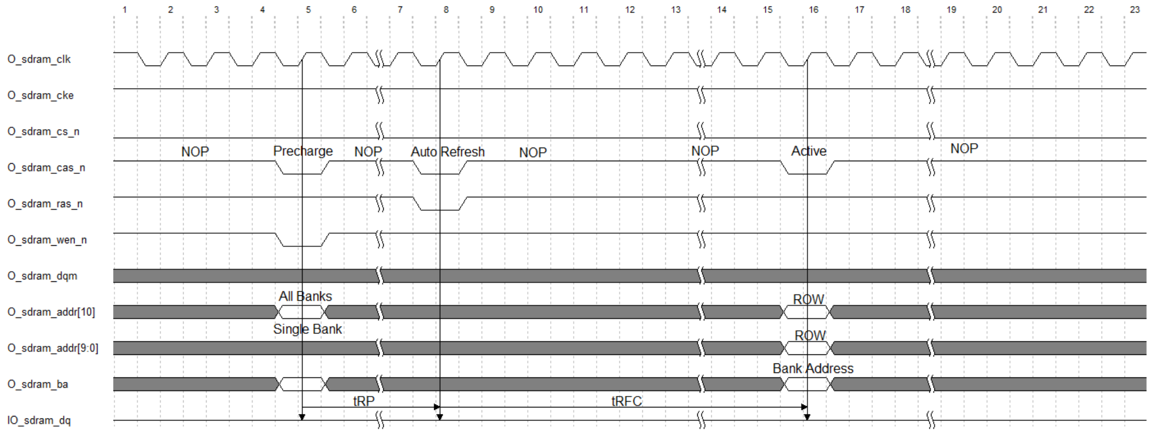
7.3.3 SDRAM Write Operation

Figure 7-5: SDRAM Write Operation



7.3.4 SDRAM Auto-refresh

Figure 7-6: SDRAM Auto-refresh



8 Performance Reference

The SDRAM controller uses Verilog. Table 8-1 shows the performance indications for different devices.

Table 8-1: SDRAM Performance Reference

Device	Programming Language	LUTs	Registers	BSRAMs	PLLs	Fmax
GW1N-4	Verilog	323	504	1	1	≥90 MHz
GW1NR-4	Verilog	323	504	1	1	≥90 MHz
GW2AR-18	Verilog	285	488	1	1	≥160 MHz
GW2A-18	Verilog	285	488	1	1	≥160 MHz

GOWIN

PROGRAMMING FOR FUTURE