Gowin SPMI

User Guide

IPUG529-1.0E, 1/9/2019

Disclaimer

Revision History

Date	Version	Description
1/9/2019	1.0E	Initial version published.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

The purpose of this Gowin SPMI User Guide is to help users to quickly understand the features and usage of Gowin SPMI by providing an overview of the functions, signal definition, working principle, and GUI call, among other functionality.

1.2 Supported Products

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

GW1NZ series FPGA products: GW1NZ-1.

1.3 Related Documents

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

1. GW1NZ series of FPGA Products Data Sheet
2. Gowin YunYuan Software User Guide

1.4 Abbreviations and Terminology

Table 1-1 shows the abbreviations and terminology that is used in this guide.

Table 1-1 Abbreviations and Terminologies

Abbreviations and Terminology	Full Name
FPGA	Field Programmable Gate Array
SPMI	System Power Management Interface
BOM	Bus Owner Master
MID	Master Identifier
CPC	Clock per Cycle
BOD	Bus Ower Device
BRD	Bus Receiver Device
TBO	Transfer Bus Ownership
RCS	Request Capable Slave
SUID	Slave Unic Identifier

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

2SPMI Parameters and Interfaces

2.1 Master

2.1.1 Parameter Configuration

Users must set the parameters listed in the table below when instantiating a SPMI as a Master.

Table 2-1 Definition of Master Parameter

Parameter	Unit	Description	Range	Default Value
CLK_FREQ	MHz	System clock frequency	1-500	50
SCLK_PERIOD	ns	The period length of the generated SCLK	40-31000	100
ABTR_DELAY	ns	The delay of BOM in response to Arbitration	1-30	
REQ_PIPE	clock cycle	The delay cycles of the sample clock after the req signal	0-7	1
MID		zMaster MID	0-3	0

2.1.2 Interface

The definition and function of the Master interfaces is as listed in the table below.

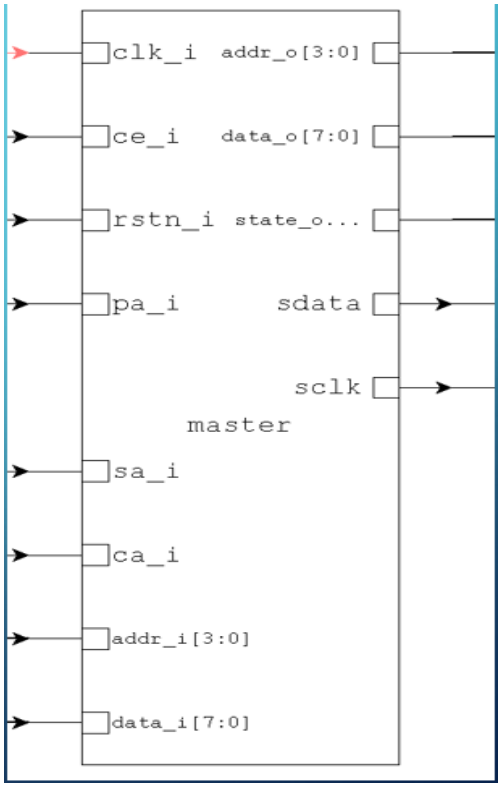
Table 2-2 The Definition of Master Interface

Name	I/O/Width	Function	Remarks
clk_i	input 1	System clock signal	
ce_i	input 1	Clock enable signal	
rstn_i	input 1	Reset signal	
ca_i	input 1	Attempt to connect	
pa_i	input 1	Initiate pa calls	

Name	I/O/Width	Function	Remarks
sa_i	input 1	Initiate sa calls	
addr_i	input [3:0]	Address input	
data_i	input [7:0]	Data input	
addr_o	output [3:0]	Address Output	
data_o	output [7:0]	Data output	
state_o	output [15:0]	State output	
sdata	inout 1	SPMI serial data line	
sclk	inout 1	SPMI serial clock line	

The figure below shows the Master interfaces.

Figure 2-1Master Interfaces



2.2 Slave

2.2.1 Parameter Configuration

Users must set the parameters listed in the table below when instantiating a SPMI as a Slave.

Table 2-3 Definition of Slave Parameters

Parameter	Unit	Description	Range	Default Value
CLK_FREQ	MHz	System clock frequency	1-500	50
REQ_PIPE	clock cycle	The delay cycles of the sample clock after the req signal	0-7	1
SUID		Slave ID	0-15	0

2.2.2 Interface

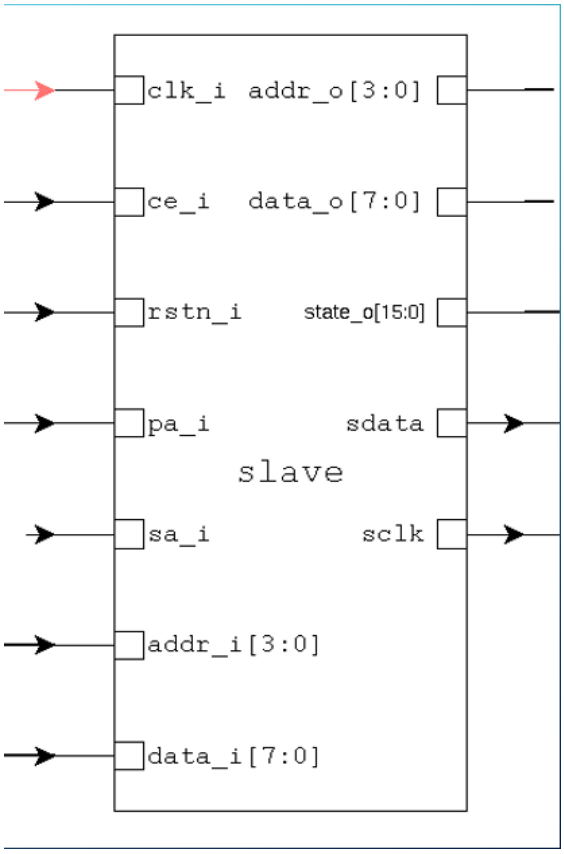
The definition and function of the Slave interfaces is as listed in the table below.

Table 2-4 The Definition of Slave Interface

Name	I/O/Width	Function	Remarks
clk_i	input 1	System clock signal	
ce_i	input 1	Clock enable signal	
rstn_i	input 1	Reset signal	
pa_i	input 1	Initiate pa calls	
sa_i	input 1	Initiate sa calls	
addr_i	input [3:0]	Address input	
data_i	input [7:0]	Data input	
addr_o	output [3:0]	Address Output	
data_o	output [7:0]	Data output	
state_o	output [15:0]	State output	
sdata	inout 1	SPMI serial data line	
sclk	inout 1	SPMI serial clock line	

The figure below shows the Slave interfaces.

Figure 2-2Slave Interfaces



2.3 Parameter Constraint

The parameter constraint is as below:

$$CPC = \left[\frac{CLK\_FREQ \times SCLK\_PERIOD}{2000}\right] - 1$$ Definition

[x] takes integers that are not greater than x

The constraints are as below:

$CPC \geq 3, \quad REQ\_PIPE \leq CPC - 1$

If CPC=3, the system has reached the edge of stability. If the system has good timing closure at the moment, it still can work normally; if not, the system may be failure.

2.4 state_o Description

The state_o description is as shown in the table below.

Table 2-5 state_o Description

bit	Definition	Description	Function	Master	Slave
state[4:0]	state code	stae code	stae code	√	√
state[5]	meSyn	me synchronization	SPMI connected (High)/ SPMI disconnected (Low)	√	-
state[6]	meFree	me free	SPMI free (High)/	√	√

			SPMI busy (Low)		
state[7]	meBom	me Bus Owner Master	Owner Master provides SCLK clock	√	-
state[8]	meBod	me Bus Owner Device	Owner Device has Bus control, being caller	√	√
state[9]	meBrd	me Bus Receiver Device	Receiver Device, being callee	√	√
state[10]	reqAddr	Request Address	Request Address	√	√
state[11]	reqData	Request Data	Request Data	√	√
state[12]	validAddr	valid address	valid address	√	√
state[13]	validData	valid data	valid data	√	√
state[14]	Reserved				
state[15]	Reserved				

2.5 State Code Definition

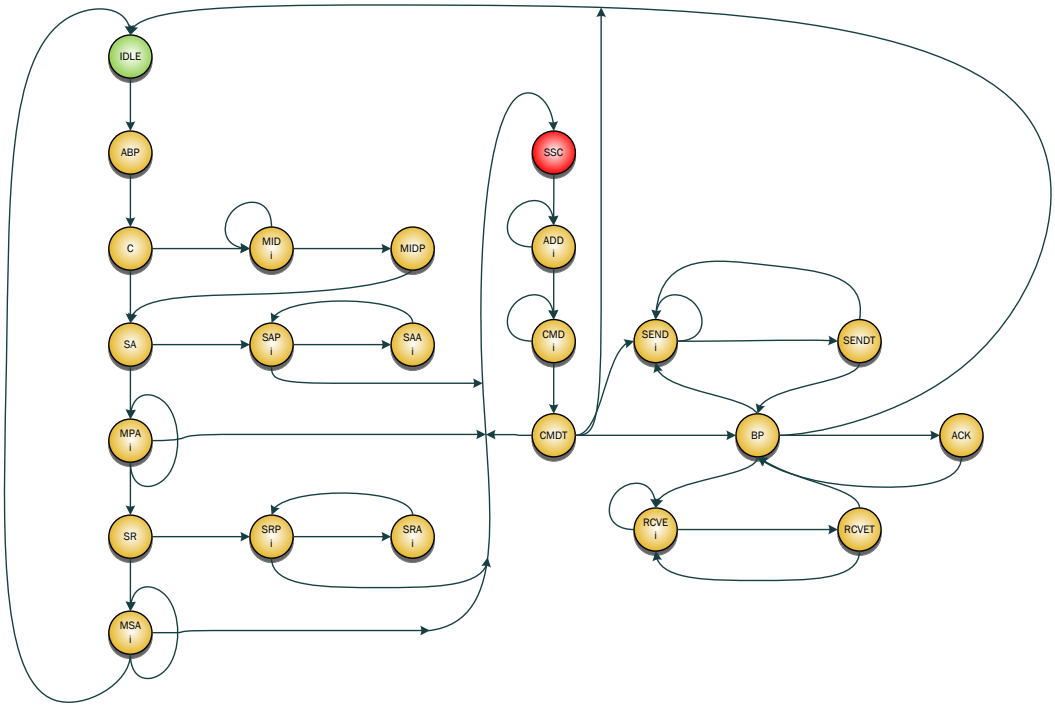
The state code definition is as shown in the table below.

Table 2-6 State Code Definition

State Code	Definition	Description	Users Concerned
5‘h00	IDLE	IDLE State	√
5 ‘h01	ABP	Bus Arbitration State	
5‘h02	C	C-bit state	
5‘h03	SA	SA-bit state	
5‘h04	MPA	Master Priority Arbitration state	
5‘h05	SR	SR-bit state	
5‘h06	MSA	Master Secondary Arbitration state	
5‘h07	MID	Send BOM ID to the connecting Master	
5‘h08	MIDP	Send BOM ID to the corresponding bus park of the connecting Master	
5‘h09	SAP	The corresponding bus park state of A-bit	
5‘h0a	SAA	Send Slave address SA[i]	
5‘h0b	SRP	The corresponding bus park state of SR-bit	
5‘h0c	SRA	Send Slave address SR[i]	
5‘h0d	SSC	SSC state	
5‘h0e	ADD	4-bit address in the command sequence	√
5‘h0f	CMD	8-bit command in the command sequence	√
5‘h10	CMDT	Parity state in the command sequence	√
5‘h11	SEND	Data transmission state	√
5‘h12	SENDT	The parity check in data transmission state	
5‘h13	BP	bus pack	
5‘h14	RCVE	Data receive state	√
5‘h15	RCVET	The parity check in data receive state	
5‘h16	ACK	ACK response state	√
5‘h17-1f		Reserved	

The state-transition is as shown in the figure below.

Figure 2-3 State Transition



3 SPMI Operation

3.1 System Architecture

One SPMI Bus has at most 4 SPMI Masters. These Masters can connect at most 16 Slaves. Master ID can be set using the "MID" parameter. Slave ID can be set using the "SUID" parameter.

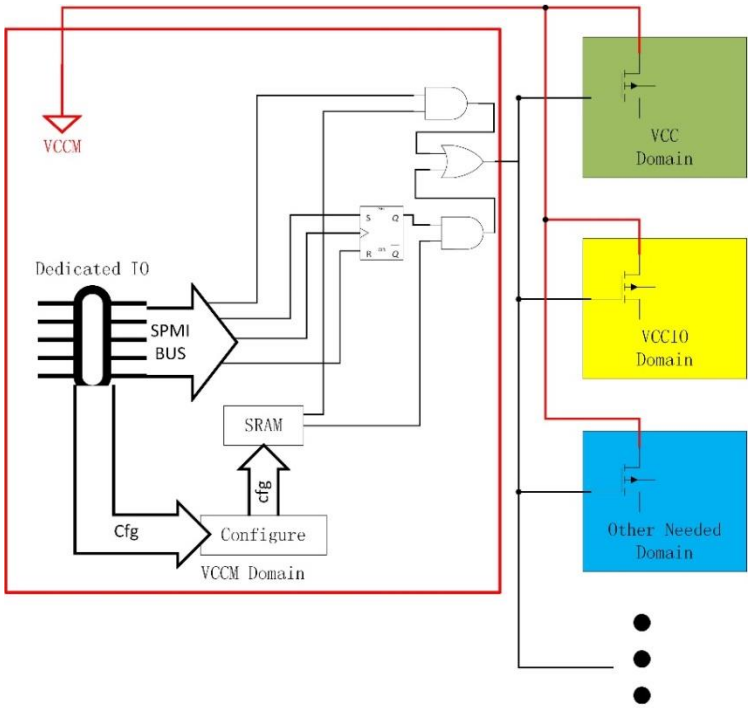
BOM: Bus Owner Master, providing SCLK signal for Bus. The system selects the BOM automatically.

BOD: Bus Owner Device, acting as the caller device in communication.

BRD: Bus Receiver Device, acting as the callee device in communication.

The FPGA power management circuit is as shown in the figure below.

Figure 3-1 Power Management Circuit



3.2 Master Connection

After power-on, the SPMI Master is in isolation state and begins to listen on the Bus immediately.

The SPMI Master in isolation state tries to connect to the Bus when it is idle and ca is pulled up for one cycle. If it connects successfully, the SPMI Master will be in connected state.

If the connected Master disconnects from the Bus via TBO command, it will be in isolation state again and keep listening on the Bus. Connection actions still need to be performed if the Master wants to connect to the Bus again.

3.3 Communication

3.3.1 Mode

Master and RCS (Request Capable Slave) can all be the caller or the callee.

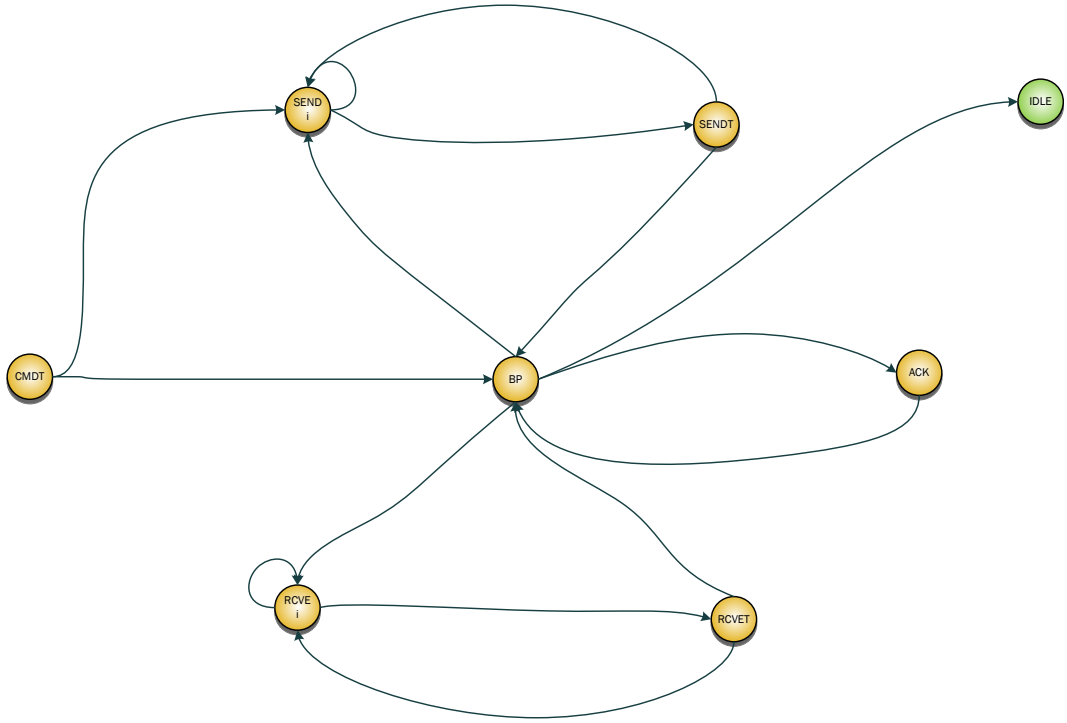
Devices on the system that do not communicate are all in listening state, including the Masters in isolation state.

The caller operation flow is: Initiate a calling request, Bus arbitration, set calling address, set communication command, transmit communication data.

The callee response flow is: Listen on address (If required), listen on commands, and transmit communication data.

The figure below shows the communication state transition.

Figure 3-2 Communication State Transition



3.3.2 Introduction to the Caller

When the device is idle, initiate a request via pa or sa. The Master can operate only when it is connected.

After the Bus arbitration, the device can be the caller if it can take control of the Bus.

The callee address is set in ADD state.

The communication command is set in CMD state.

Data is sent in SEND state.

Data is received in RCVE state.

meBod will be automatically cleared by hardware after communication.

3.3.3 Introduction to the Callee

The address is monitored in ADD state.

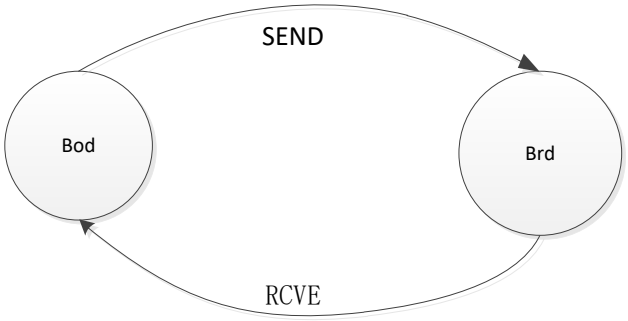
The command is monitored in CMD state.

After the device is confirmed as the callee:

Data is received in SEND state.

Data is sent in RCVE state.

Figure 3-3"Callee" State Transition



3.4 Introduction to Communication Priority

pa calling from the Slave > pa calling from the Master > sa calling from the Slave > sa calling from the Master

The Slave priority is determined by SUID. Larger address has higher priority.

The Master priority is determined by round rabin algorithm automatically to ensure that the Masters have equal polling opportunities.

3.5 Timing

3.5.1 Set/Check the Callee Address

Setting:

In ADD state, the system will sample addr_i after reqAddr is valid and REQ_PIPE cycles delayed.

Check:

In ADD state, the system will output address on addr_o whjen addrValid is valid.

3.5.2 Set/Check Communication Commands

Setting:

In CMD state, the system will sample data_i after dataReq is valid and REQ_PIPEcycles delayed.

Check:

In CMD state, the system will output address on addr_o whjen addrValid is valid.

3.5.3 Read/Write Communication Data

Read data:

- In SEND state, the callee reads data_o when dataValid.
- In RCVE state, the caller reads data_o when dataValid.

Write data:

- In SEND state, the caller writes data to data_i after dataReq is valid and REQ_PIPEcycles delayed.
- In RCVE state, the callee writes data to data_i after dataReq is valid and REQ_PIPEcycles delayed.

4Commands Supported

4.1 Commands List

The commands list that SPMI supported is as shown in the table below.

Table 4-1Commands List

Command Frame Payload	Description
0x00 to 0x0F	Extended Register Write
0x10	Reset
0x11	Sleep
0x12	Shutdown
0x13	Wakeup
0x14	Authenticate
0x15	Master Read
0x16	Master Write
0x1A	Transfer Bus Ownership
0x1B	Device Descriptor Block Master Read
0x1C	Device Descriptor Block Slave Read
0x20 to 0x2F	Extended Register Read
0x30 to 0x37	Extended Register Write Long
0x38 to 0x3F	Extended Register Read Long
0x40 to 0x5F	Register Write
0x60 to 0x7F	Register Read
0x80 to 0xFF	Register 0 Write

The data frames of TBO commands will be generated by their Masters automatically.

4.2 Examples

Master0 and slave0 (SUID=0): Extended Register Write

Function:

Master0（meBod）:

Master sets pa_i to initiate Master Priority Arbitration to enter MPA state ->SSC state when meFree and meSyn are high.

The callee address (4‘h0) is set in ADD state.
The communication command (8’h00) is set in CMD state.
The register address is sent in SEND state.
Data is sent in SEND state.
Slave0 (meBrd):
Listen to the ADD address (4‘h0) in ADD state
Listen to the communication command (8’h00) in CMD state
The register address is received in SEND state.
The written data is received in SEND state.

m0_ERWL_s0:

a). Function:

This example is about the Extended Register Write Long (ERWL) command. That is Master initiates a ERWL command to Slave.

b). Master and Slave configuration:

cmd = 8'h36
m0：mid=0
s0：SUID=10

c). Steps:

Press m0 key1 and key2 in turn.

m0_m1_ca:

a). Function:

This example demonstrates how a Master connect to the Bus. That is Master 0 and Master 1 request to connect to the Bus simultaneously.

b). Master configuration:

m0：mid=3
m1：mid=1

c). Steps:

j9_38 of m0 and m1are connected via a cable. Download m0 bit files first, download m1 bit files, and then press m0 key1 and key2.

m0_m1_pa:

a). Function:

This example is about Master arbitration. That is Master 0 and Master 1 initiate pa (Priority Arbitration) simultaneously.

b). Master configuration:

m0: mid=0 ，cmd=8’h10
m1: mid=1 ，cmd=8’h11

c). Steps:

j9_38 and j9_37 of m0 and m1are connected via cables. Download m0 bit files first, download m1 bit files, and then press m0 key1 and key2.

m0_pa_s0_sa

a). Function:

This example is about Master and Slave arbitrations. That is one Master initiates a pa (Priority Arbitration) request and one Slave initiates a sa (SR-bit Slave Arbitration) request simultaneously.

b). Master and Slave configuration:

m0: mid=1 ，cmd=8’h10
s0：SUID=7，cmd=8’h16

c). Steps:

j9_37 of m0 and s0 are connected via a cable. Download m0 bit files

first, download s0 bit files, and then press m0 key1 and key2.

s0_write_m0

a). Function:

This example is about MW commands. That is Slave performs write operations on Master.

b). Master and Slave configuration:

m0：mid=0

s0：SUID=8，cmd=8'h16

c). Steps:

j9_38 of m0 and s0 are connected via a cable. Download s0 bit files first, download m0 bit files, and then press s0 key1 and key2.

5 Instantiation

5.1 Master Instantiation

Master Instantiation is as described below:

```
master iMaster (
    .clk_i      (sysClk),
    .ce_i       (ce_r),
    .rstn_i     (rstn),
    .pa_i       (pa),
    .sa_i       (sa),
    .ca_i       (ca),
    .addr_i     (addrI),
    .data_i     (dataI),
    .addr_o     (addrO),
    .data_o     (dataO),
    .state_o    (state_o),
    .sdata      (j10_1),
    .sclk       (j10_5)
);

defparam iMaster.MID            =1;
defparam iMaster.CLK_FREQ       =CLK_FREQ;
defparam iMaster.SCLK_PERIOD    =1000;
defparam iMaster.REQ_PIPE       =1;
```

5.2 Slave Instantiation

Slave Instantiation is as described below:

```
slave iSlave1 (
    .clk_i      (sysClk),
    .ce_i       (ce_r),
    .rstn_i     (rstn),
    .pa_i       (pa_s),
    .sa_i       (sa),
    .addr_i     (addrI_s),
```

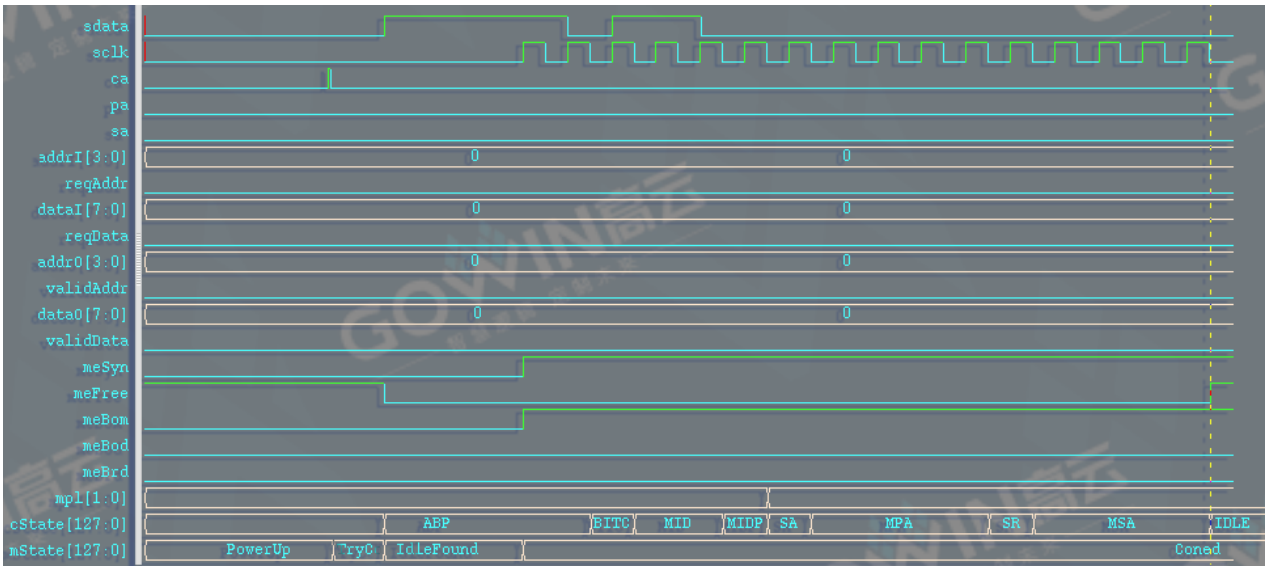
```
    .data_i     (dataI_s),
    .addr_o     (addrO_s),
    .data_o     (dataO_s),
    .state_o    (state_o_s),
    .sdata      (j10_1),
    .sclk       (j10_5)
);

defparam iSlave1.SUID           =12;
defparam iSlave1.CLK_FREQ       =CLK_FREQ;
defparam iSlave1.REQ_PIPE       =1;
```

6 Timing Examples

Note that to view all the waveform, the timeline may be cropped and unbalanced scaled.

6.1 Connection

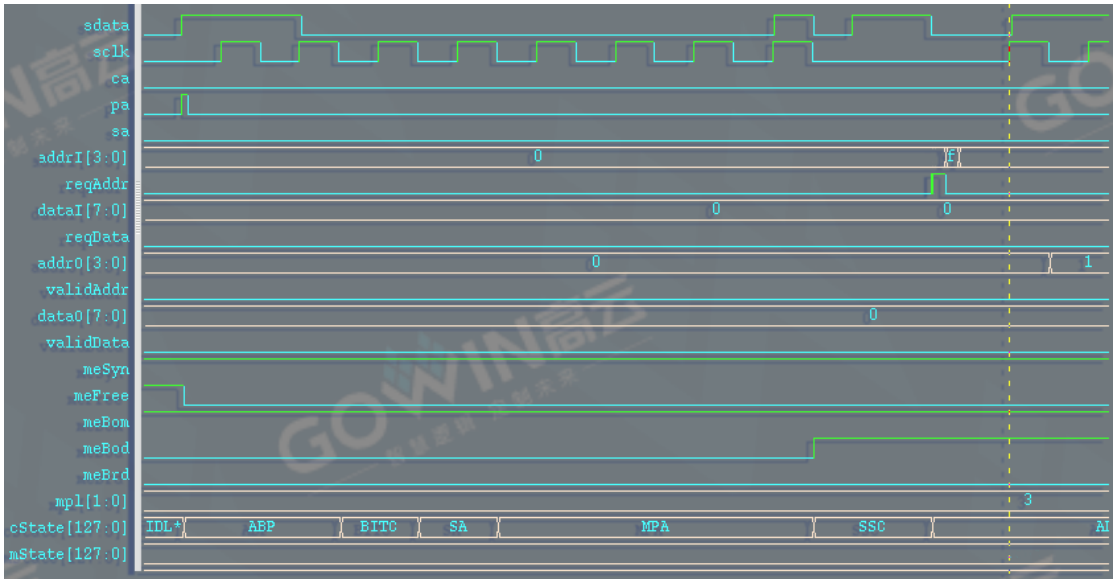
Figure 6-1 Connection Timing



As shown in the figure above, after Master is powered on, meFree is detected as High, and meSyn is low, ca will be pulled up for one cycle to connect to the Bus.

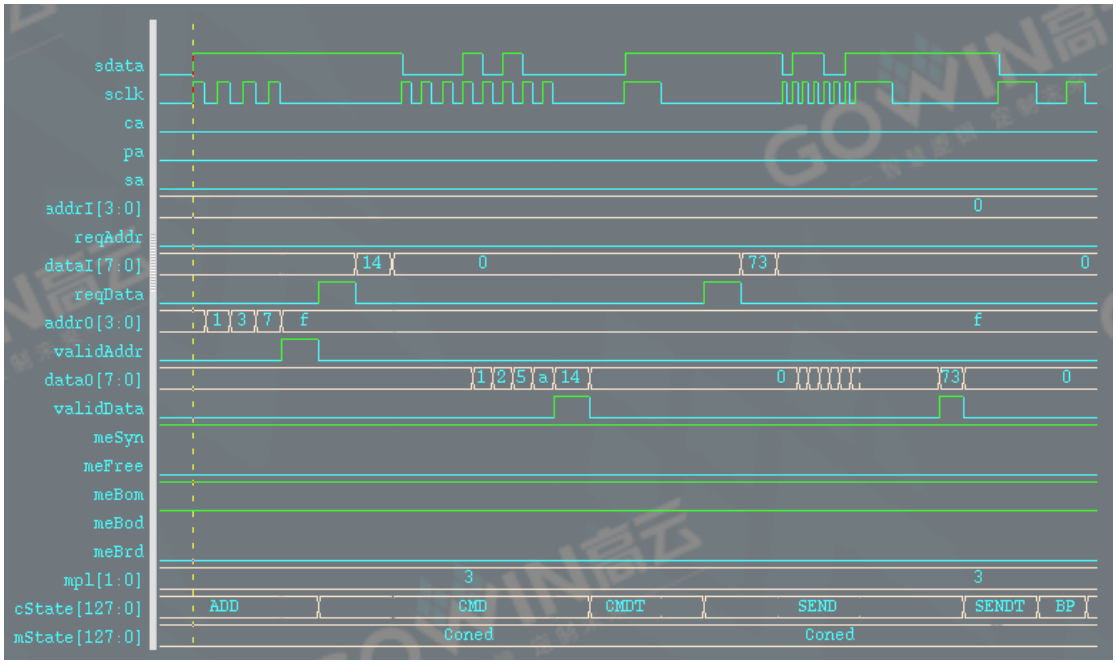
6.2 Master Initiates Pa Request, SUID=15, CMD=ATT

Figure 6-2 Initiates Master Primary Communication



As shown in the figure above, meFree is detected as High and meSyn is low. ca will be pulled up for one cycle to initiate Master Primary communication. If meBod is detected as high, the request is successful. In ADD state, set the communication address 0xF on the addrI interface after reqAddr is high and REQ_PIPE=1 cycle delayed.

Figure 6-3 State Transition in Authenticate Command



In CMD state, set the communication command 0x14 on the dataI interface after reqData is high and REQ_PIPE=1cycle delayed. This is the "Authenticate" command. According to SPMI protocol, Master in this command transmits one byte to Slave, and then Slave transmits one byte to Master. Repeat 4 times. In the first SEND state, set Master transmitting

the first byte to Slave on dataI after reqData is high and REQ_PIPE=1 cycle delayed; In the n SEND state, set Master transmitting the n byte to Slave on dataI after reqData is high and REQ_PIPE=1 cycle delayed, n=1-4. Similarly, in the first RCVE state, the first data that Slave sends back is read on dataO when validData is high; in the n RCVE state, the n data that Slave sends back is read on dataO when validData is high, n=1-4.

It's similar for the Slave communication. The Bus communication starts when meFree is detected as low. In ADD state, read the output address of addrO when validAdd is high, and this step can be omitted. In CMD state, read the output command of dataO when validData is high. After this, if meBrd is detected as high, it denotes that the communication request is sent to the Slave. Data can be prepared according to the commands at this time.

Assume that ATT is the command described above. Similarly, in the first SEND state, the first data that Master sends is read on dataO when validData is high; in the n SEND state, the n data that Master sends is read on dataO when validData is high, n=1-4. In the first RCVE state, set the first data that Slave sends to Master on dataI after reqData is high and REQ_PIPE=1 cycle delayed; In the n RCVE state, set the n data that Slave sends to Master on dataI after reqData is high and REQ_PIPE=1 cycle delayed, n=1-4.


GOWIN
PROGRAMMING FOR FUTURE