SYNTHESIZABLE SYNCHRONOUS-DRAM CONTROLLER INTELLECTUAL PROPERTY

1.0 SYSTEM FEATURES

- Interfaces readily, without further modifications, to 1M x 16 SDRAMs such as:

Samsung	KM416S1120D
NEC	uPD4516161AG5
OKI	MSM56V1616

  and other compatibles.
- Flexible refresh-cycle generation, including “burst” refresh and normal refresh, and everything in between.
- Allows the SDRAM’s internal mode register to be programmed easily by the host.
- “smart_mode” provides 1 level deep write queue for “dump-and-run” writes.
- Internal synchronizers allows the host (micro) and the SDRAM to operate at differing clock speeds.
- Built-in comprehensive synthesizable SDRAM tester.

2.0 GENERAL OVERVIEW

The synthesizable *Synchronous DRAM controller IP* is a complete design solution which allows virtually any type of microprocessor, microcontroller and DSP to interface to large capacity SDRAMs effortlessly.

All of the low-level SDRAM functions such as address demultiplexing, refresh generation and busy generation are handled by the IP transparently to the host. The non-trivial initialization sequence required by most SDRAMs is also performed transparently to the host upon powerup or system reset.

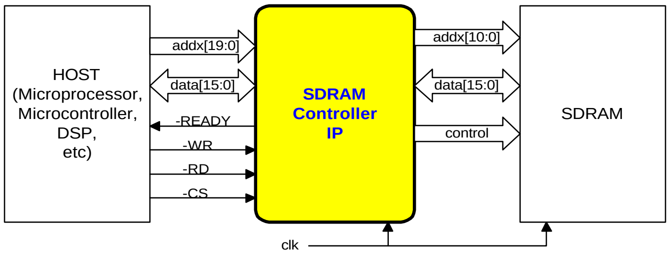


Figure 1 Typical interface block diagram

As with most DRAM controllers, this SDRAM controller provides to the host an SRAM-like bus interface. Figure 1 illustrates the typical connection block diagram. By embedding the SDRAM controller IP along with processor cores, a highly integrated System-on-Chip (SoC) designs are possible.

To write, the host drives the bus with a typical SRAM-like write operation. The SDRAM controller then generates the appropriate signals to transfer the host’s data to the specified address of the SDRAM. A selectable “dump-n-run” feature allows the host to continue with other operations immediately following the write, and not wait for the SDRAM controller to relinquish the bus.

Similarly, to read, the host drives the bus with a typical read operation. The SDRAM controller keeps the host in busy state until data is retrieved from the SDRAM and made available to the host.

If a write or read request is made by the host while the SDRAM is being accessed, the host is placed in busy state until the pending access is completed. Similarly, if a request is made while a refresh operation is in progress, the host is busied until its completion.

The host’s write and read bus operation is completely asynchronous, much like the access to a SRAM. The SDRAM controller generates a busy output which the host can use as handshake control. This avoids the need of fixed wait states.

Essentially, a write to the SDRAM controller represents a write to the SDRAM, while the read from the SDRAM controller represents a read from the SDRAM.

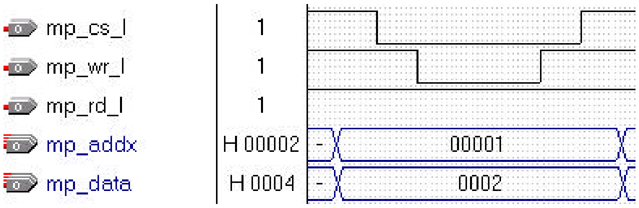


Figure 2 To write to the SDRAM, the host issues a write bus cycle to the SDRAM controller which resembles that of a typical SRAM.

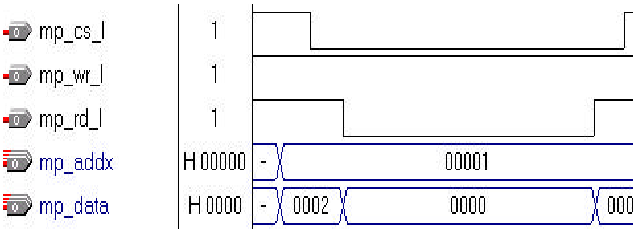


Figure 3 To read from the SDRAM, the host issues a read bus cycle to the SDRAM controller which resembles that of a typical SRAM read

2.1 IP Signal Conventions

The signaling conventions used in the core are detailed below:

- Any active low signal is appended with “**`_l`**”. For example, the active low system reset is denoted as `sys_rst_`**`l`**
- Any signals connecting to the host is preceded with “**`mp_`**”. For example, the data bus connecting the controller to the host is denoted as **`mp_`**`data[15:0].`
- Any signals connecting to the SDRAM is preceded with “**`sd_`**”. For example, the data bus connecting the controller to the SDRAM is denoted as **`sd_`**`data[15:0].`

2.2 IP Pinout

The pinout for the default IP (target the typical 1Mx16 SDRAM) consists of 2 system signals, 42 host side signals and 36 SDRAM side signals. The tables below describes the functions of the pins.

System Level Signals	Directi on	Description
sys_rst_l	input	System level active low reset. Brings all state machines into the intial state. The reset is asynchronous.
Sys_clk	input	This is the main clock used to clock the state machine. An optional clock divider can be used internally.

Host Side Signals	Directi on	Description

mp_data[15:0]	bidi	Bidirectional data bus connected to the host (micro, DSP, etc).
mp_addx[19:0]	input	Linear address bus connected to the host
mp_cs_l	input	Active low chip-select input from the host. This signal must go low on all reads or writes meant to the SDRAM. This is an asynchronous signal.
mp_rd_l	input	Active low signal which the host uses to indicate to the controller that the current bus transaction is a read from the SDRAM. This is an asynchronous signal.
mp_wr_l	input	Active low signal which the host uses to indicates to the controller that the current bus trasaction is a write to the SDRAM
sdram_busy_l	output	The SDRAM controller sets this output to low to indicate that there is a pending transaction (read,write, refresh) with the SDRAM and thus the host must wait. Additionally, this signals is used as handshake control betweent the controller and the host.
sdram_mode_set	input	A write to the controller with this signal low will update the SDRAM's mode register.
smart_h	input	When high, enables 1 deep write queue during writes allowing dump-n-run writes.

SDRAM Side Signals	Direction	Description
sd_data[15:0]	bidi	This is the bidirectional data bus conneting the controller and the SDRAM.
sd_addx[10:0]	output	This is the multiplexed address bus connected to the SDRAM.
sd_clke	output	This is the active high clock enable signal.
sd_wr_l	output	This is the active low write signal.
sd_cs_l	output	This is the active low SDRAM chip select signal.
sd_ras_l	output	This is the active low row address select signal.
sd_cas_l	output	This is the active low colum address select signal.
sd_udqm	output	This is the active high upper byte mask signal. When high, no data can be read to written to the SDRAM upper byte.
sd_ldqm	output	This is the active high lower byte mask signal. When high, no data can be read to written to the SDRAM lower byte.
sd_ba	output	This is the bank select (lower 512K or upper 512K) signal. It is also known as the 11th address bit (A11).
sd_clk	output	This is the clock driving the SDRAM.

2.3 SDRAM Controller IP Files

The SDRAM controller IP is composed of the following files:

```
SDRAM.V
HOSTCONT.V
SDRAMCNT.V
MICRO.V
SIM_MP.V
INC.H
TST_INC.H
```

SDRAM.V is the top hierarchy module. It serves as a wrapper and instantiates the lower sub modules, HOSTCONT.V , SDRAMCNT.V and MICRO.V. Additionally, as an option, it contains a system clock divider.

SDRAMCNT.V is the main state machine which generates all of the control signals to the SDRAM. It also generates signals to control the HOSTCONT.V module. This module is responsible for essentially governing the SDRAM.

HOSTCNT.V contains the logic in the datapath connecting the host bus and the SDRAM bus. Most of the datapath logic receives the control signals from SDRAMCNT.V

MICRO.V is the optionally synthesizable tester. It essentially acts as a host to the SDRAM controller core. Based on the selected test, it will instruct the SDRAM controller to do a series of repeating write/read tests to the SDRAM.

SIM_MP.V is a submodule to MICRO.V and acts as a write-bus or read-bus cycle generator. The type of bus cycle it will emulate is that of an Intel processor. That is, of an asynchronous in nature.

INC.H contains a set of globals which determine the interval of refresh and the number of refreshes per interval. The user specifies the SDRAM clock speed in this file. The synthesizable test core also gets to be enabled here. Finally, the state machine's in SDRAMCNT.V is defined here.

TST_INC.H defines the type of test to perform when the test core is enabled.

2.4 Built-In Synthesizable SDRAM Tester

The SDRAM IP contains a set of synthesizable test cores which can aid the testing and development of embedded SDRAM controllers greatly. The test core is useful primarily on FPGA/CPLD platforms.

The test cores emulates a typical microprocesors' write and read bus cycles. When enabled, the tester becomes a host to the SDRAM controller. The tester/controller pair can then be used to test the performance and feature of a particular SDRAM chip quickly. This can be of particular use when the simulation model of the SDRAM is unavailable or if the simulation is not possible.

The test core is enabled at compile time through the use of `define statement. The header file tst_inc.h contains a set of define statements which when uncommented enables the test core. At the time of the writing of this documentation, three test cores are available:

(I) do_single_burst_write_read_test

This test core writes a specified amount of words to the SDRAM controller. It then delays for a specified amount of clocks, and proceeeds to read from the just written addresses of the SDRAM. This delay and read cycles indefinetely.

(II) do_burst_write_read_test

This test core writes a specified amount of words to the SDRAM controller. It then delays for a specified amount of clocks, and then proceeds to read from the just written addresses of the SDRAM. The write-delay-read cycles indefinetely.

(III) do_read_write_test

This test core performs a single write, then delays for a specified clock ticks, then performs a single read from the just written address. The write-delay-read cycles indefinitely.

3.0 OVERVIEW OF BASIC SDRAM OPERATION

This section is not intended to discuss all of the detailed operation of a SDRAM, but does cover the essentials of a typical read, write and refresh operations. The reader should consult the specification manual for the particular memory for further details.

The majority of SDRAM employs a simple set of signal combinations called *commands* to carry out the basic IO. Most of

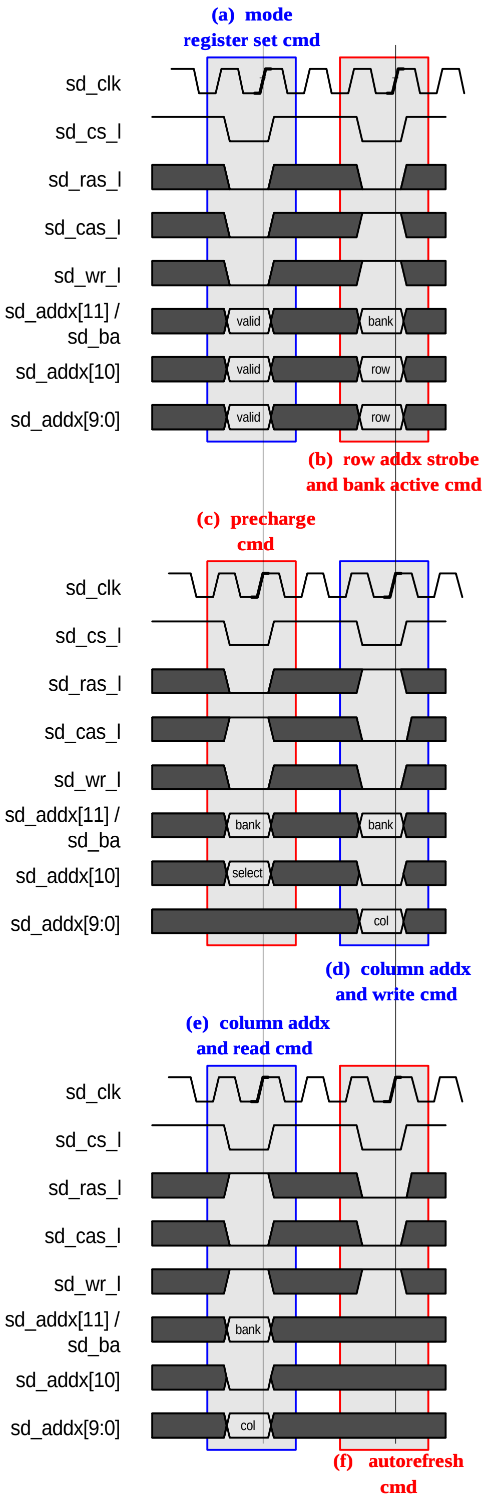


Figure 4 (a) - (f) This figure illustrates the basic commands most SDRAMs recognize. The proper combination of these command comprises the read, write and refresh operations.

these commands are one clock cycle in duration, and are clocked by the SDRAM on the rising edge. A proper setup and hold times must be observed. A sequence of these commands comprises the primitive operation of read, write and refresh. Figure 4 illustrates a typical set of command used by most SDRAMs.

The 1M x 16 SDRAM (IP default) is organized as 512K x 16 x 2 banks. The bank selection is done by a pin called BA (bank address) or sometimes also called A11. This IP refers to as BA, and from this point forward, A11 and BA will be used interchangeably.

The address organization of these SDRAMs are 11 rows by 8 columns. During the row address strobing, A[10:0] provides the row to select and during the column strobing A[7:0] provides the column to select (A[10:8] should be set to logic low).

- *Mode register set command*

This command is used to program the SDRAM's mode register. The mode register controls the operation of the SDRAM, including the CAS latency, burst type, burst length, test mode and other vendor specific options. Most SDRAMs do not initialize the mode register upon power up, thus it is critical this register be initialized prior to the normal use. The SDRAM controller initializes the mode register following every system reset with a default value. During the mode register programming, the SDRAM receives the data from A[10:0] and BA, rather than from the data bus.

- *Row address strobe and bank active command*

This command is used to select the bank and the row where the data access is to take place. The BA input selects the bank, while A[10:0] provides the row address.

- *Precharge*

The precharge command is used to begin the precharge operation to the selected bank(s). When A[10] is high, both banks are activated, while when A[10] is low, the bank selected by BA is activated.

- *Column address and write command*

This command is used to select the column of the selected bank where the write is to take place. It is important that the bank selected at the row address strobe be selected again. The address bus A[7:0] selects the column. A[10:8] should be logic low.

- *Column address and read command*

This command is used to select the column of the selected bank where the read is to take place. The address bus A[7:0] selects the column. A[10:8] should be logic low.

- *Auto refresh command*

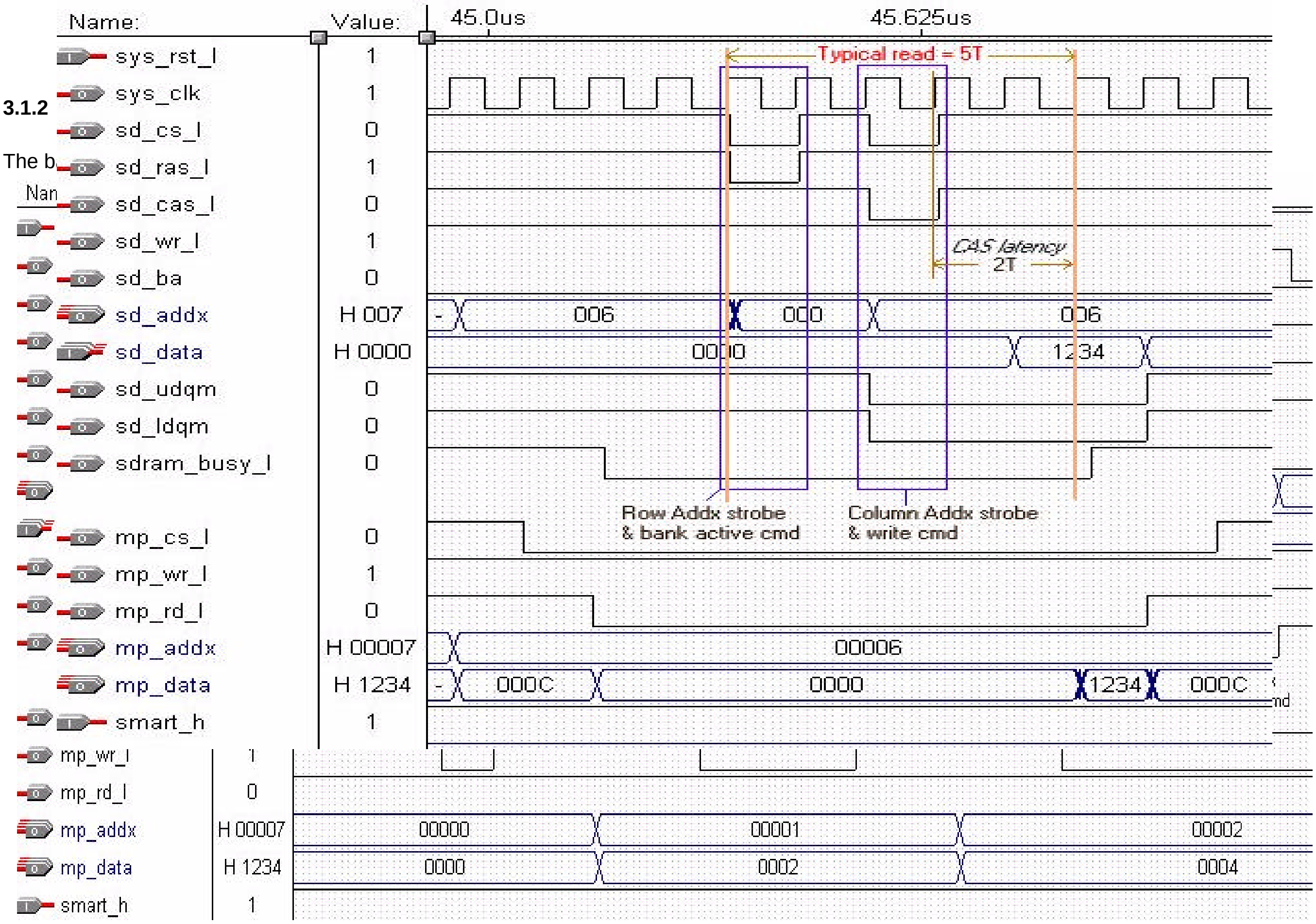
This command is used to start the internal refresh cycle. An internal row counter is adjusted to point to the next row.

3.1 Basic Operations

Since in order to take the full advantage of the capabilities of a particular SDRAM, the application has to be known, the SDRAM controller IP only support the basic types of READ, WRITE and refresh operations.

3.1.1 Basic READ Operation

The basic read operation consists of reading one 16 bit word from the SDRAM.



3.1.3 Refresh Operation

The refresh operation commands the SDRAM to perform its internal auto-refresh cycle.

3.2 SDRAM Mode Register and Initialization Sequence

Upon powerup, the SDRAM’s mode register must be initialized for proper operation. This is due to the fact that most SDRAMs do not initialize the mode register following a reset. Before the register can be initialized, however, the SDRAM must be subjected to an intialization sequence.

4.0 FUNCTIONAL DESCRIPTION OF SDRAM CONTROLLER

As previously mentioned, the SDRAM controller sits between the host and manages the data flow to and from the host/SDRAM. From the host’s point of view the controller/SDRAM pair represents a large capacity SRAM. All of the primitive tasks of refresh, arbitration, address demultiplexing are transparently handled by the controller. The only data flow control (hanshaking) signal used is the sdram_busy_l signal.

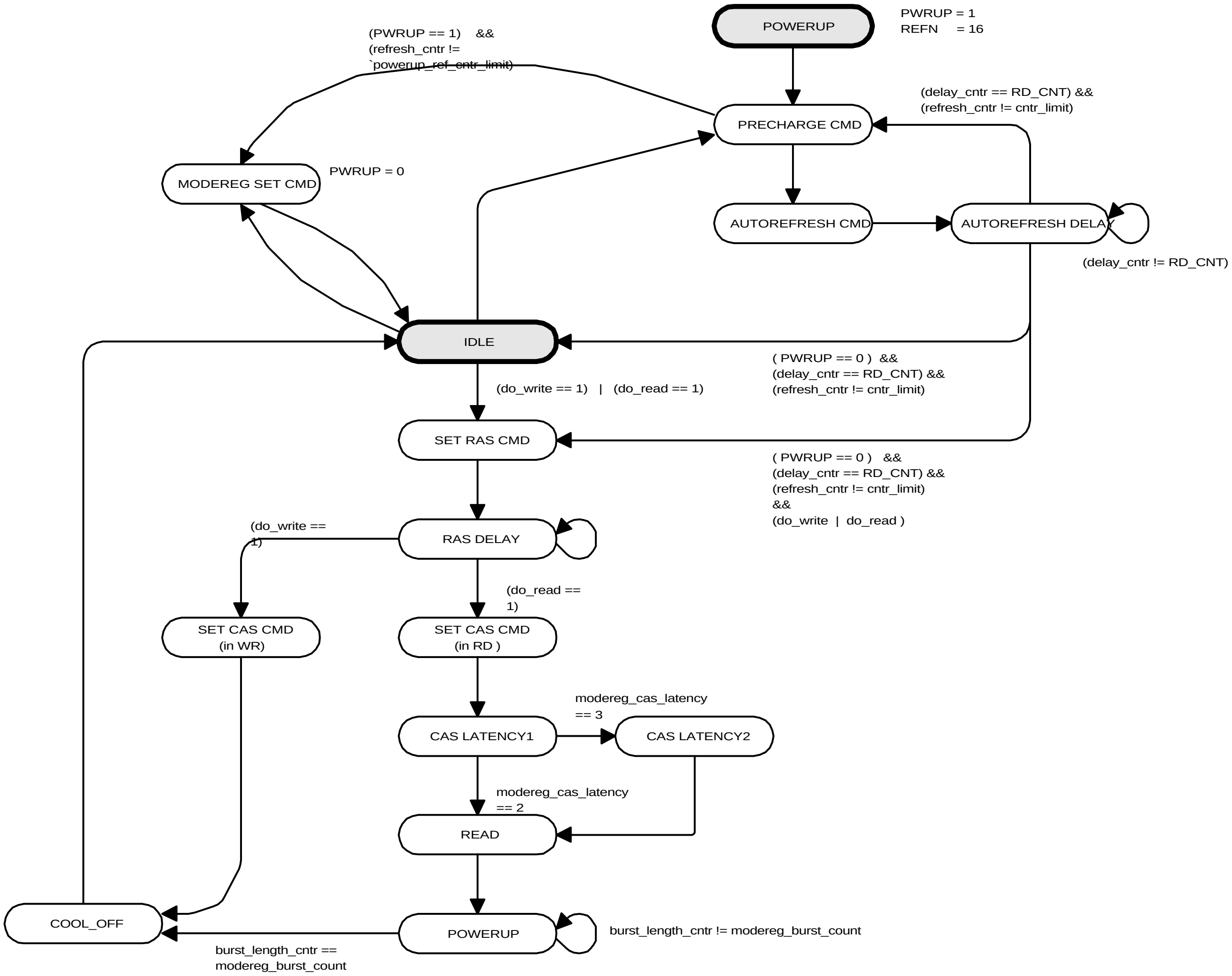


Figure 5 State Flow of the state machine in SDRAMCNT.v