SIEMENS

Memory Products

Modular embedded DRAM

DRM256 Version 1.1

Preliminary Datasheet 06.97

DRM256 Revision History 06.97
Previous Releases: versions 1.0 and 0.9 of this document. Only editorial changes have been done in the current release of this document, no changes to the silicon occured

Edition 06.97

This edition was realized using the software system FrameMaker®.

**Published by Siemens AG,
Bereich Halbleiter, Marketing-
Kommunikation, Balanstraße 73,
81541 München**

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Modular embedded DRAM is the core of a new service provided by the SIEMENS Memory Products group. Custom logic can be combined with the latest SIEMENS dynamic memory technology providing application specific embedded DRAM solutions manufactured by SIEMENS. Modular embedded DRAM is based on 256 Kbit DRAM building blocks. A library of core modules is provided which allows the fast creation of dynamic memory optimized for specific applications. The memory size can be customized up to 16 MBit with increment steps of 256 Kbit.

The synchronous interface of Modular embedded DRAM operates at high speed using fast page mode access. Optional multibank operation allows high performance applications. Multiple block configurations enable large data busses with up to 512 bit bus width. The resulting maximum bus throughput rate is 42.4 Gbit/s at a clock frequency of 83 MHz. Programmable pipeline stages of depth one, two or three stages allow to optimize data transmission by trading clock frequency for latency.

Application examples include signal processing or graphics devices which benefit from the large bandwidth of the wide on-chip interface. Dedicated memory architectures with customized memory modules are supported.

This preliminary data sheet describes features and application of the 256 Kbit DRAM memory block DRM256 for custom memory modules between 256 Kbit and 16 Mbit.

The 256 Kbit DRAM Core is available today in the latest SIEMENS 0.35µ 16-MBit trench technology, enhanced with up to four metal layers. It works from a 3.3 Volt supply.

1.1 Features

- 256 Kbit DRAM building block for memories of up to 16Mbit in customized bank configurations
- Wide configurable interface, up to 512 bit
- Optional BIST with on-chip test controller
- LATENCY 1, 2 and 3 modes
- 256 refresh cycles in 64ms refresh interval
- 1024 local sense amps per block
- 8 redundant rows per block
- 1-T trench capacitor DRAM cells
- Chip area < 8.8 mm² for 2 Mbit with customized interface
- Less than 1.5mW/MHz power consumption per active building block
- 0.35µ CMOS process, up to 4 metal layers
- 3.3V operating voltage

1.2 Block Diagram

Modular embedded DRAM includes three main parts: The actual DRAM, an optional test controller, and a power supply unit. The power supply unit includes a stabilized 2.5 Volt supply for the memory core and an n-well voltage generator for the memory cell transistors. The test controller allows different test options from direct access to built-in self test. See section 5 for a detailed description of the test controller.

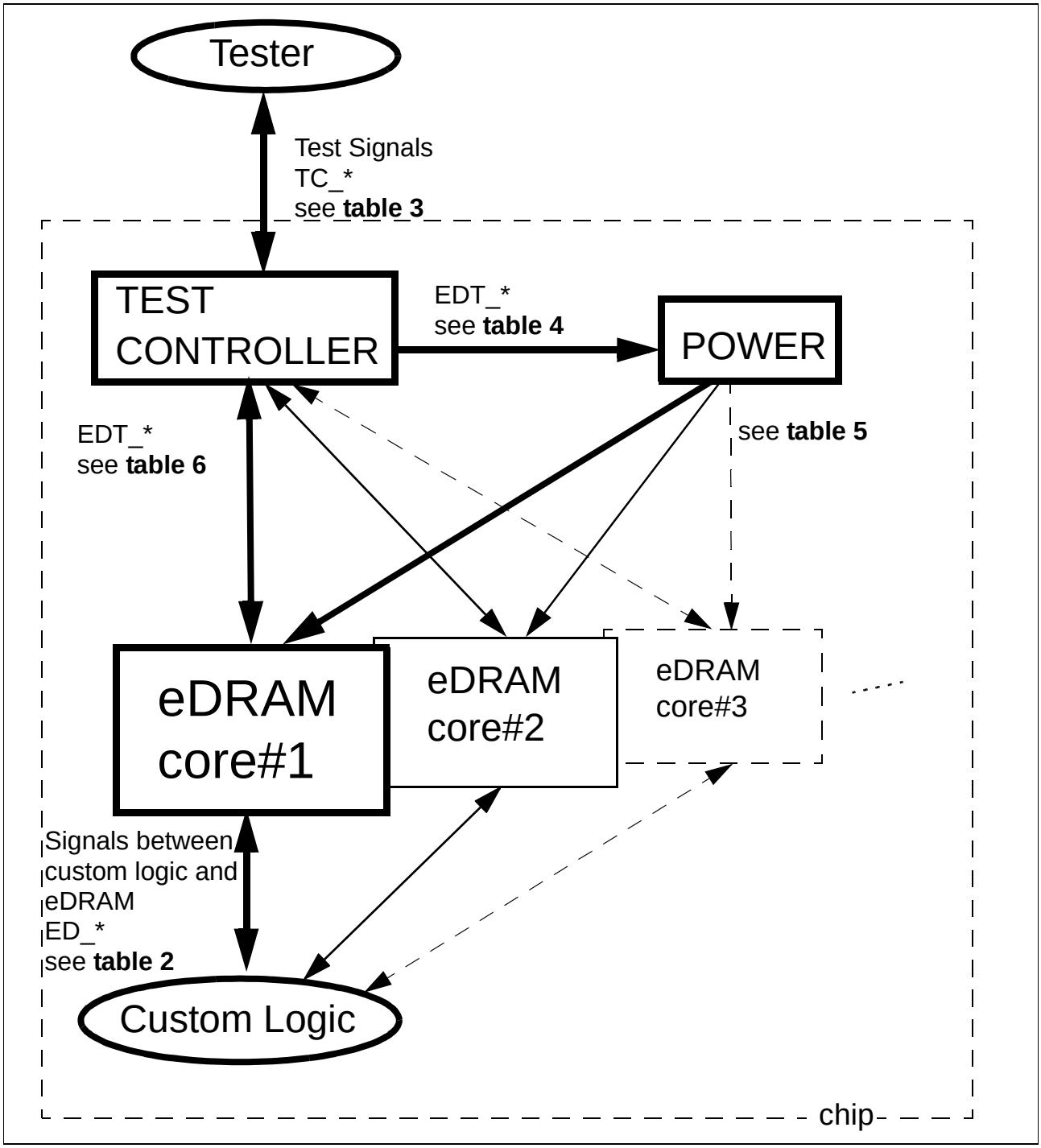


Figure 1
Block Diagram

1.3 Signal Definitions and Functions

The following tables group the signals according to their functions. They include signal name, bus width, direction and a brief description of the function. As described on figure 1, signals are grouped into five different clusters as follows:

- “ED_“ signals are connections between the embedded DRAM core and the custom logic part or the power module, see table 2 in section 1.3.1 and table 5 in section 1.3.3, respectively.
- “TC_” signals are connections between the Test Controller and the tester, see table 3 in section 1.3.2.
- “EDT_” signals are connections between Test Controller and either the DRAM core or the power unit, see table 6 in section 1.3.3 and table 4 in section 1.3.3, respectively.

1.3.1 Signals from Custom Logic to DRAM

Table 1
Power Supply Signals

VDD	3.3V +/-10% supply voltage
GND	0V

Table 2
Input Signals from Custom Logic, Output Signals to Custom Logic

Signal Name	Width	Direction at DRAM	Description
ED_CLK	1	I	**CLOCK:** Input to the internal clock buffer of the memory module. CLK can be kept at low for minimum power dissipation.
ED_CKE	1	I	**CLOCK ENABLE:** 1: The internal clock of the memory module is running and the module is ready for operation. 0: The internal clock is stopped and the memory module is in standby. Note that there is no automatic refresh.
ED_RESETN	1	I	**RESET:** Asynchronous input, active low. The memory module is reset while set to „0“. Should be applied together with reset of the entire chip.
ED_CMD[3:0]	4	I	**COMMAND:** Controls the operation of the memory module. See tables 10 to 14 for a description of the operating modes.The value has to be valid at the rising edge of CLK.
ED_BROW[]	b	I	**BANK ROW:** Selects one bank for a row access. The width of the bus depends on the application configuration of the memory module. The value has to be valid at the rising edge of CLK. Connect to GND, if only one bank is implemented.

ED_AROW[7:0]	8	I	**ADDRESS ROW:** Selects one of the memory modules for a page access. 8 row addresses are available. The value has to be valid at the rising edge of CLK.
ED_BCOL[]	b	I	**BANK COLUMN:** Selects one bank for a column access. The width of the bus depends on the application configuration of the memory module. BROW has the same width as BCOL. The value has to be valid at the rising edge of CLK. Connect to GND, if only one bank is implemented.
ED_ACOL[]	c	I	**ADDRESS COLUMN:** Selects one word of the open page. The width of the bus depends on the application configuration of the memory module, see section 3. The value has to be valid at the rising edge of CLK.
ED_LAT[2:0]	3	I	**LATENCY:** Activates the registers in the data path. Three stages are implemented. Asynchronous signals. Data on this bus must be valid for a complete access. Changes are only allowed while all memory banks of a memory module are in the precharge state.
ED_DI[]	w	I	**DATA INPUT:** Input data bus.The width is application dependent and is the same as DO[].
ED_BWE[]	w/8	I	**Byte Write Enable:** Enables a write operation for 8 bit of the data word if set to 1. Width is equal to wordwidth divided by 8. Set to 1 if not used.
ED_OEN	1	I	**OUTPUT ENABLE:** 0: The tri-state buffers of DO[] are active for data output. 1: The tri-state buffers of DO[] are switched off. Asynchronous signal, active low. Connect to Ground if not used.
ED_GROUP	1	I	**GROUP:** This signal is used during the ACTIVATE and PRECHARGE commands. It selects a group of memory blocks.
ED_DO[]	w	O	**DATA OUTPUT:** This bus carries the output data. The width is application dependent and is the same as DI[]. /OE controls the state of the buffers.

Definitions:

w is the word width of the data bus and is currently 16, 32, 64 or 128.

b for b_addrwidth, used to select a bank. See section 3.4.2 for more details.

c for col_addrwidth, used to select a column. See section 3.4.2 for more details.

1.3.2 Signals to Tester Interface

These signals must be connected to external pads during test. A NWA pad for wafer test and analysis is required. Do not connect this pad to anything else.

Table 3
Signals to the Test Controller

Signal Name	Width	Direction at TC	Description
TC_DI[15:0]	16	I	Test Controller Data In
TC_DO[15:0]	16	O	Test Controller Data Out
TC_DDIR	1	I	**I/O Data Pad Direction Select:** Select direction of data at pad. TC_DI and TC_DO are connected to one bidirectional pad. 1:Input 0:Outpout
TC_ADDR [7:0]	8	I	Test Controller Address Bus
TC_CMD [3:0]	4	I	Test Controller Command Bus
TC_FAIL	1	O	Output Fail Indication 1:fail 0:no error
TC_CLK	1	I	Controller Clock[1)]
TC_RESN	1	I	Test Controller Reset (active low)
TC_BURNIN	1	I	set Burn in mode (active high)
TC_MARCH	1	I	Set March Test Mode (active high)
TC_TEST	1	I	Activate Test Controller (active high)

1) The signal TC_CLK is also used for the DRAM core.

1.3.3 Internal Signals

Internal signals are signals that connect the different blocks but are neither connected to the custom logic nor used for test access. These signals have to be connected during the routing of the embedded DRAM product. Internal signal names are identical at source and destination except where indicated differently.

Table 4
Signals from Test Controller to Voltage Regulator & Margin Generator:

Signal Name	Width	Direction at TC	Description
EDT_REG_ON	1	O	Voltage regulator on (active high)
EDT_MARGIN[4:0]	5	O	Set Margin Voltage Steps (active high)
EDT_MARG_OFF	1	O	Disable Margin Generator variable margin (active high)

Table 5
Signals from Power Module to DRAM cores

Signal Name	Width	Direction at DRAM	Description
ED_V25	1	I	2.5V for the DRAM core. This is a pad for wafer test only.
ED_NWA	1	I	n-Well Voltage. This is a pad for wafer test only.
EDT_MARX	1	I	Voltage for margin test

The n-well voltage has to be varied in different DRAM patterns. A NWA **pad** for wafer test and analysis is required, as is the 2.5 Volt source pad ED_V25 (this does need not to be an external pin). This is an analog signal and cannot be generated from test controller logic. A NWA pad for wafer test and analysis is required. Do not connect this pad to anything else.

In the following table, most signals are similar to those defined in table 1.The only difference is that they are now coming from the Test Controller and no longer from the custom logic. To select the origin of these signals (TC or custom logic) 2 signals are used: EDT_TEST_MUX and EDT_CLK_MUX.

Table 6
Signals from Test Controller to DRAM Cores:

Signal Name	Width	Direction at TC	Description
EDT_BROW []	b	O	Bank Row and Col are equal and come from Load Bank [3:0]
EDT_BCOL []	b	O	Bank Row and Col are equal and come from Load Bank [3:0]
EDT_AROW[7:0]	8	O	Test Address Row
EDT_ACOL[]	c	O	Test Address Column
EDT_RESETN	1	O	Test Reset for DRAM core
EDT_CMD[3:0]	4	O	DRAM Command Bus
EDT_DI[3:0]	4	O	DRAM Data Input
EDT_DO[15:0]	16	I	DRAM Data Output
EDT_LAT[2:0]	3	O	latency 1, 2 or 3
EDT_GROUP	1	O	Group: Selection of a group of memory blocks
EDT_CKE[]	m	O	Test Clock Enable;
EDT_OEN[]	m	O	Output Enable
EDT_TEST_MUX[]	m	O	**Test Select (Multiplexer)** 0: Normal mode 1: Test mode
EDT_CLK_MUX[]	m	O	**Clock Select (Multiplexer)** 0: Normal mode, ED_CLK is used 1: Test mode, TC_CLK is used
Test Modi			
EDT_REDTST	1	O	Redundancy Test On
EDT_BOFF	1	O	Bit Redundancy Off
EDT_ROFF	1	O	Row Redundancy Off
EDT_MWLS	1	O	MWL Select test mode enabled
EDT_BOOST_OFF	1	O	Boost Off (cf. Margin control)
EDT_WBI	1	O	Wafer Burn In test mode en. In this mode EDT_MWLS must be set active
EDT_COMP	1	O	Compare Output Bits
EDT_SELW[1:0]	2	O	Output Multiplexer

Definition:

m is the number of memory module connected to one test controller.

2 Device Integration

The integration of Modular embedded DRAM into an application specific circuit is part of the service provided by **SIEMENS**. Given the specific requirements of the DRAM core and the application specific logic, **SIEMENS** provides design integration and manufacturing of the device.The following main steps are applied to create an embedded DRAM product:

- Siemens provides all necessary information for the design of the custom logic part, i.g. design rules, timing parameters etc.
- The DRAM module has to be specified as described in section 3.5. A parameterizable VHDL model is available for device level simulations (see section 3.4). Siemens generates the design and provides all views. An abstract view is provided to be placed in the layout of the product in advance.
- A test concept has to be selected. The test controller is provided in form of a VHDL netlist. It can be synthesized together with the custom logic part. The test concept is described in detail in section 5.
- Test software for the DRAM part can be provided by Siemens, depending on the tester equipment and the selected test concept.
- Mask making and manufacturing are performed by Siemens.
- Testing of the DRAM is done by Siemens, while the custom logic part may be tested by the customer himself. Intellectual property residing in the custom logic remains completely with the customer.

This document describes architecture, test concept, and electrical parameters of Modular embedded DRAM. Note that this document focusses on Modular embedded DRAM, while standard DRAM cores with customized interface are available as well.

The design flow may start at different entry points, depending on the extent of the cooperation. See figure 2 for an overview of the design flow.

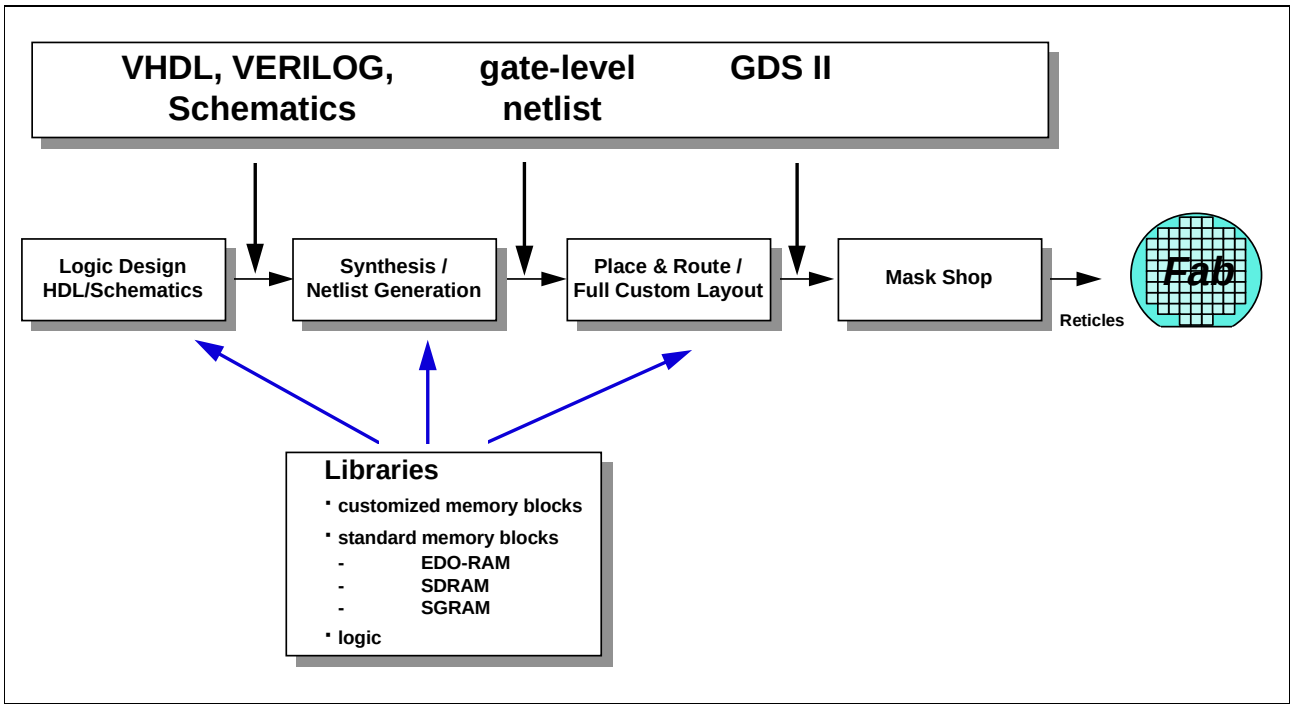


Figure 2
Design Flow

Manufacturing may include packaging, or end at wafer level. See figure 3 for an overview of the backend flow. Depending on the customer's needs, tested wafers, mounted devices or tested devices may be provided.

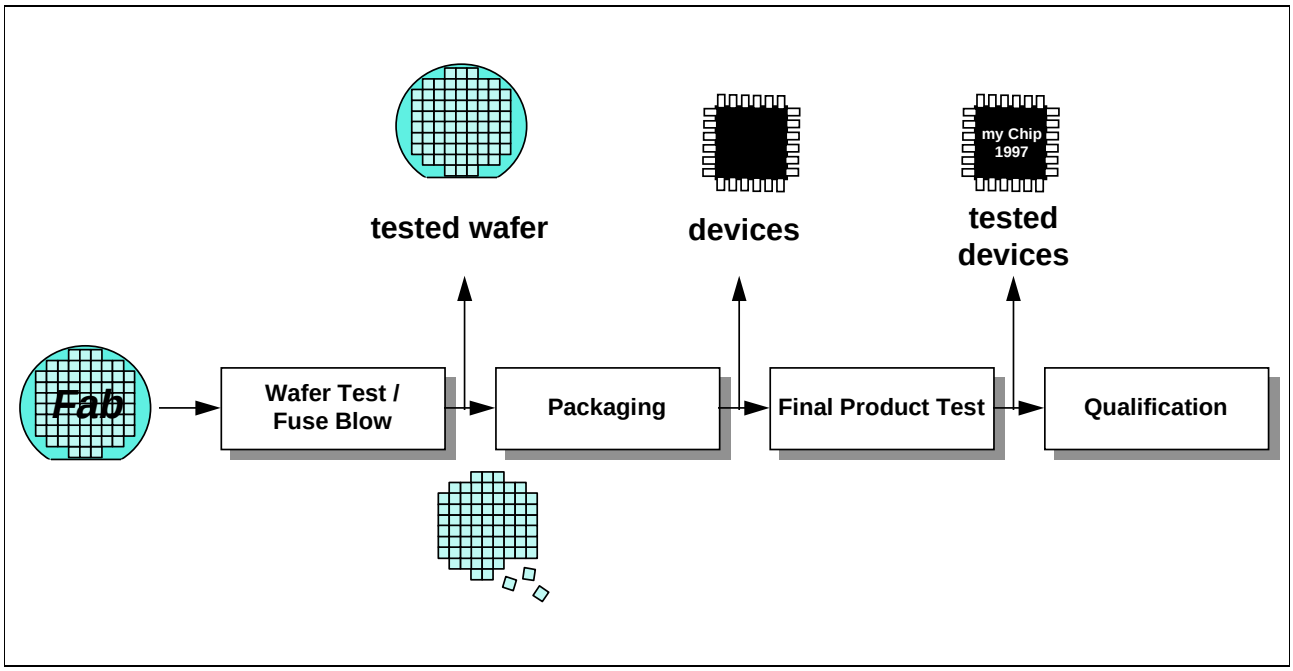


Figure 3
Backend Manufacturing

3 Technical Description

Siemens provides a library of DRAM modules for the fast creation of customized dynamic memory. One memory block contains 256 Kbit, organized in 1024 columns and 256 rows. This way 1024 sense amplifiers build one of 256 available pages. The data is accessible via a 64 bit wide bus. Adding blocks vertically, horizontally or both ways affects bus width and the amount of interface logic needed. The page length can be set separately by providing an appropriate decoder. This decoder is part of the DRAM core and is provided by Siemens according to the requirements of the design. Thus, the architecture can be optimized for the given application.

3.1 Building Blocks

Currently the library contains the following building blocks:

- memory block DRM256_0
- bus interface BIF32R_0
- bus bridge BB8_0
- control bridge CB_0

Memory Block DRM256_0

The dynamic memory module DRM256_0 contains:

- 262144 bits of user memory
- 1024 local sense amplifiers
- 64 global bit lines
- 8 redundant rows
- 4 redundantglobal bit-lines

The layout is depicted in figure 4.

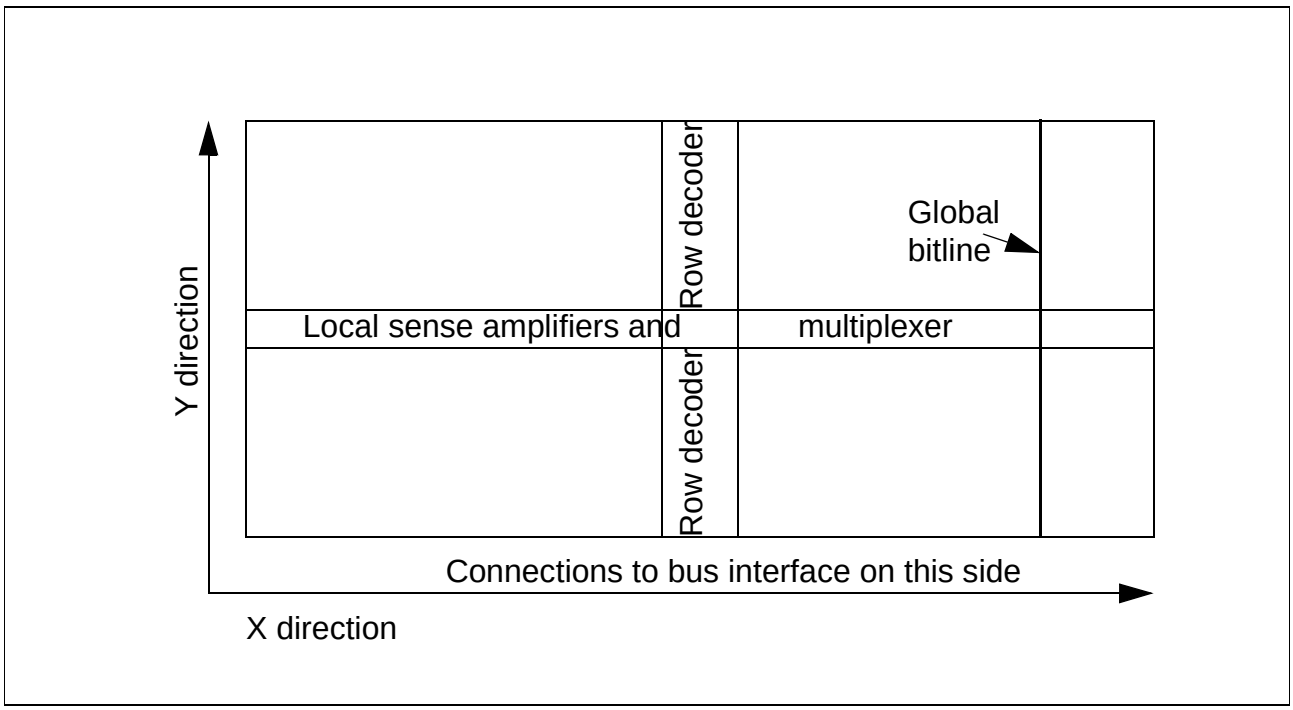


Figure 4
Layout principle of a Memory Block

Memory blocks can be placed besides each other on two sides. One memory block is accessed via a 64 bit wide bus, also referred to as the global bit lines. These 64 lines are multiplexed to the 1024 local bit lines. Consider two memory blocks placed aside each other in x - direction as defined in figure 4. This doubles the number of global bit lines and the number of sense amplifiers. A 512 Kbit memory bank with a 128 bit wide interface results. If two memory blocks are abutted in y-direction, the number of global bit lines remains 64. Thus, the same amount of memory is obtained with a smaller interface, less interface logic and hence less power consumption. A more detailed description of the architectural possibilities is given in section 3.2.

Bus Interface BIF32R_0

The bus interface is placed at one end of a line of memory blocks, see figure 5. It performs the multiplexing of the 64 global bit lines down to 16, 32, or again 64 bit and thus connects one line of memory blocks to the memory bus. It also converts the voltage levels from the 64 global bit lines into pure logical levels. It contains drivers and buffers to the memory bus and to the memory blocks as well as the bit redundancy multiplexers. It is available in different configurations.

Bus Bridge BB8_0

The bus bridge is the connection between the memory bus and the custom logic. The bus bridge provides a well defined electrical interface to allow easy integration of memory modules and custom logic. The number of bus bridges connected to the memory bus determines the width of the actual user interface between DRAM and the custom logic part.

Special configurations for embedded memory are possible. In case the memory bus is not needed for a certain configuration, a dedicated block can be built, which contains the function of the bus-interface and the bus-bridge in one block.

Control Bridge CB_0

The control bridge contains all buffers and drivers for the address busses and control signals for the memory module and bus interfaces. It is necessary once per memory module. This block is only used once in one memory module.

Memory Bus

The memory bus connects the data bus of several lines of memory blocks to a memory segment. This bus is generated specifically for the required configuration. It also connects the Control Bridge to the bus interfaces and the memory blocks.

NWA Pump NWAP_0

This is the voltage regulator which generates the voltage for the N-well. It is necessary once on a chip.

Voltage Regulator VR05_0

The voltage of the memory array is reduced to 2.5 V. The surrounding logic operates at 3.3 V. All level shifters are included in the bus and control bridges. The reduced voltage is provided by this separate regulator. It is placed outside the memory module.

3.2 Architecture of a Memory Module

The following definitions are used throughout this document:

Memory stripe: all memory blocks with common global bit lines together with the bus interface (see figure 5).

Memory segment: all memory stripes that share same lines of the external data bus (see figure 6).

Memory module: all memory segments sharing one control bridge.

It is possible that one stripe is a segment and that this segment is the whole module. Figures 5 to 7 show different examples for a memory module.

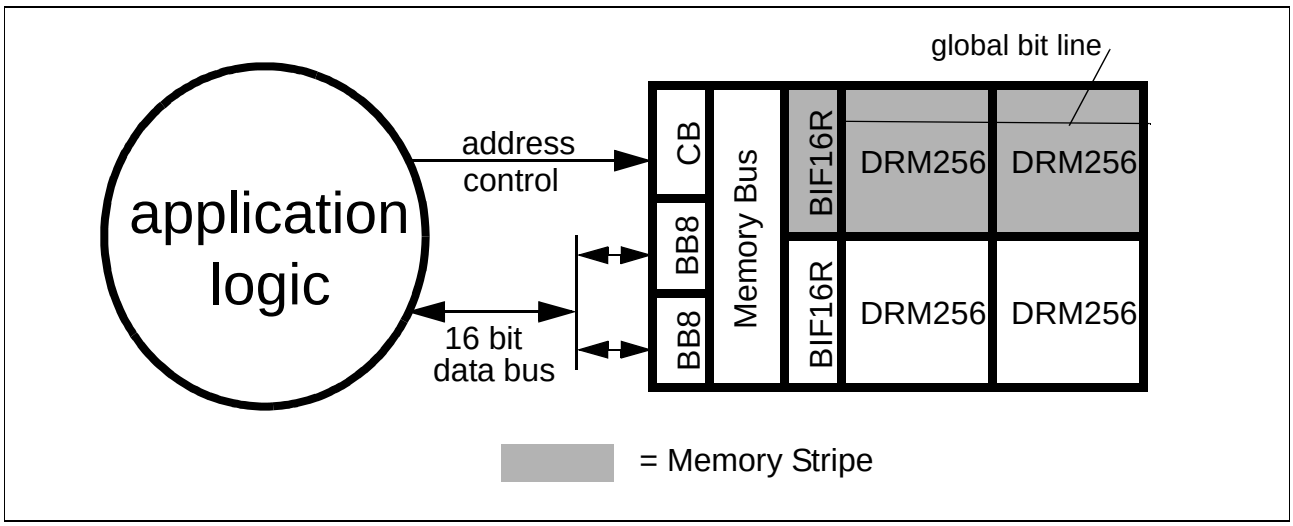


Figure 5
Memory Stripe in a Memory Module with 1 Mbit and 16 Bit Data Bus

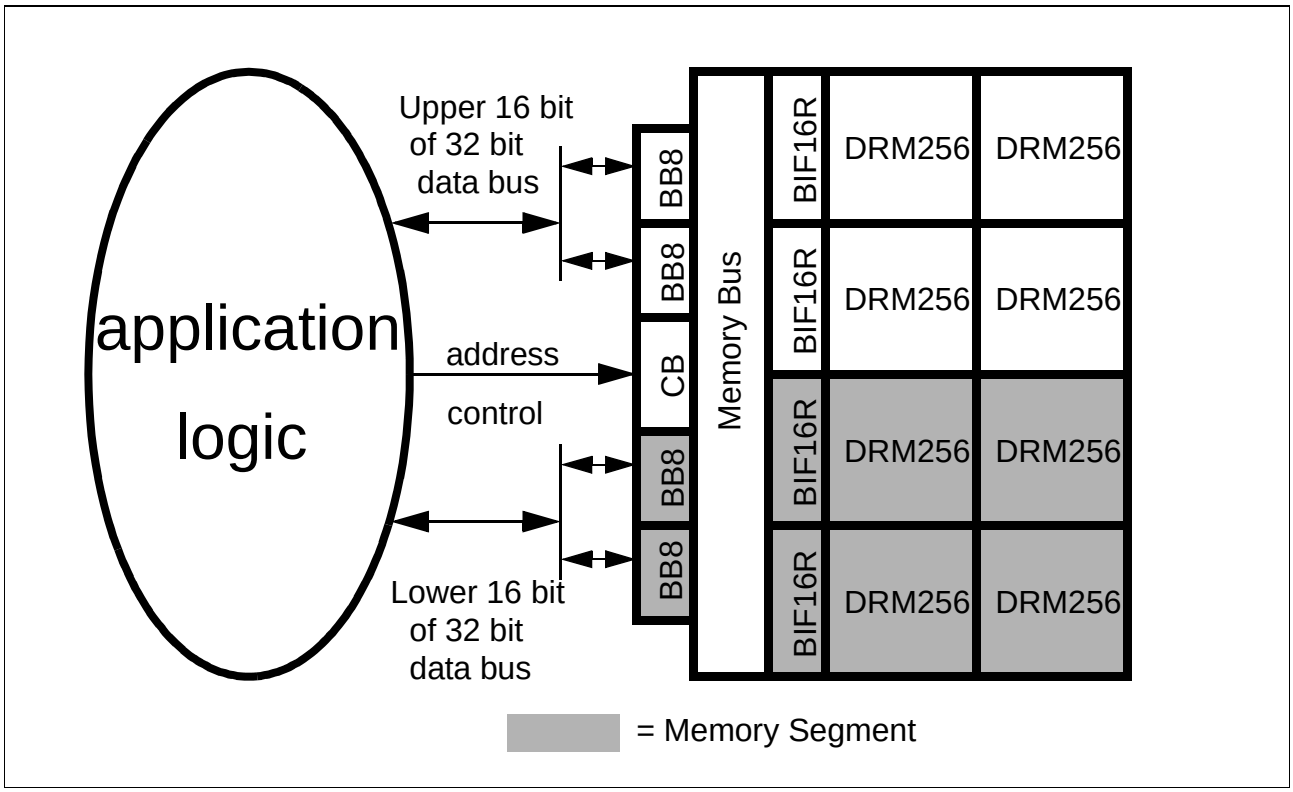


Figure 6
Example for a Memory Module with 2 Mbit and 32 Bit Data Bus

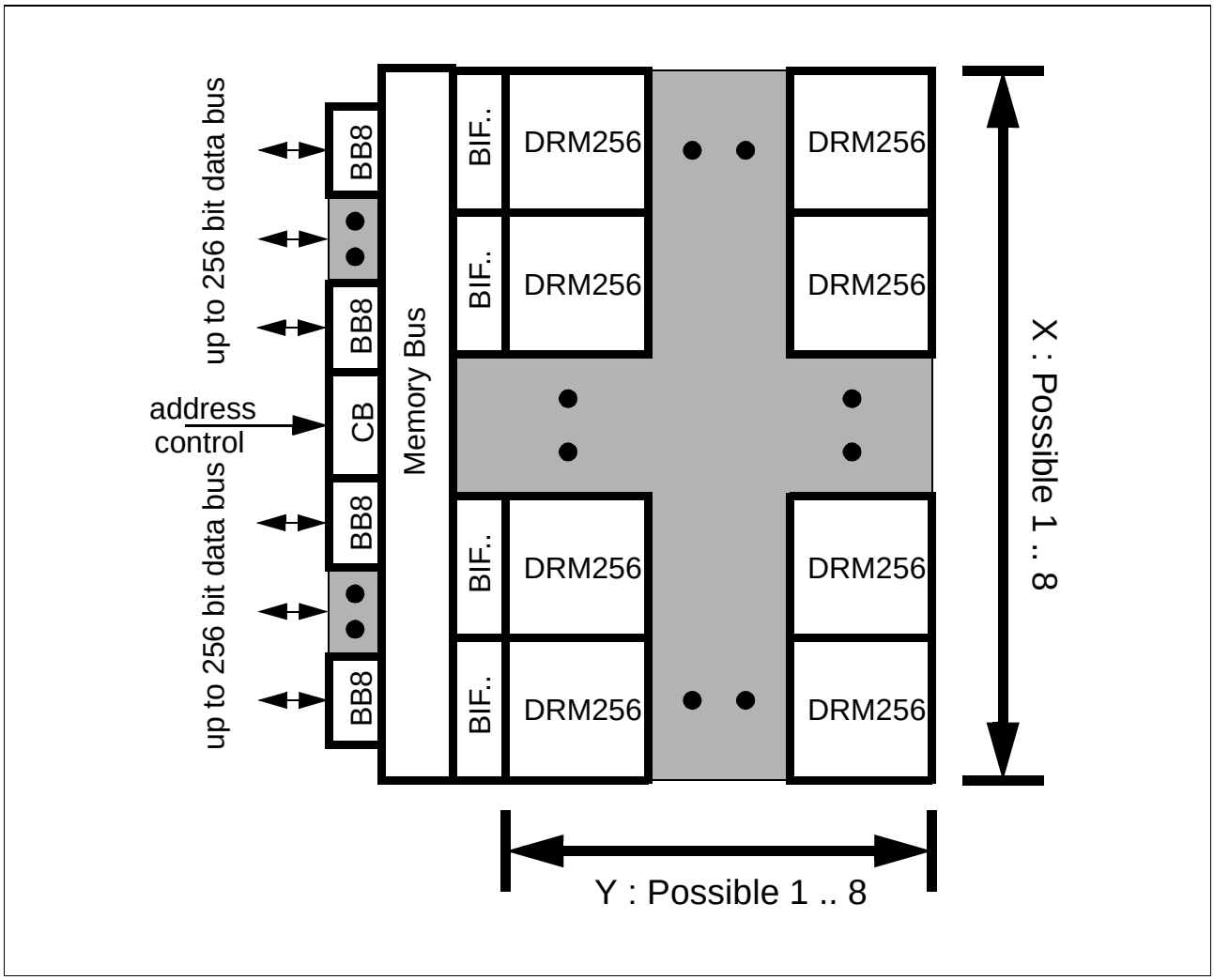


Figure 7
Maximum Configuration of a Memory Module

The number of blocks in y direction for one memory module is limited to 8. Otherwise the length of the global bit line exceeds the driver capability of the bus interface and a serious speed degradation occurs. The number of blocks in x direction is limited to 8 because of the implemented bus width for selection. This also limits the total data bus width to 8 * 64 = 512 bits. In total one memory module can have up to 64 memory blocks or 16 MBit of memory.

3.3 Concept of Memory Banks

The logical separation of the memory module is the memory bank. One, two, four or eight memory blocks form one memory bank. The advantage of this organization is the increased page length. One memory block has 1024 local sense amplifiers and therefore offers 1024 bits for „fast page mode“ style access. If *n* blocks form one bank the page length of this bank is *n* * 1024 bits. The purpose of using *multiple* banks is to parallelly keep different data segments in different banks. This allows faster page mode accesses to all data segments with hidden precharge as the probability for a pagehit is increased.

The maximum number of banks of a memory module is equal to the number of memory blocks. However, the effort to keep track of all open banks rises significantly with the number of banks. Thus,

the maximum number of banks supported by Modular embedded DRAM is 16. The maximum size of one bank is 2 MBit. This corresponds to a maximum page length of 8 Kbit.

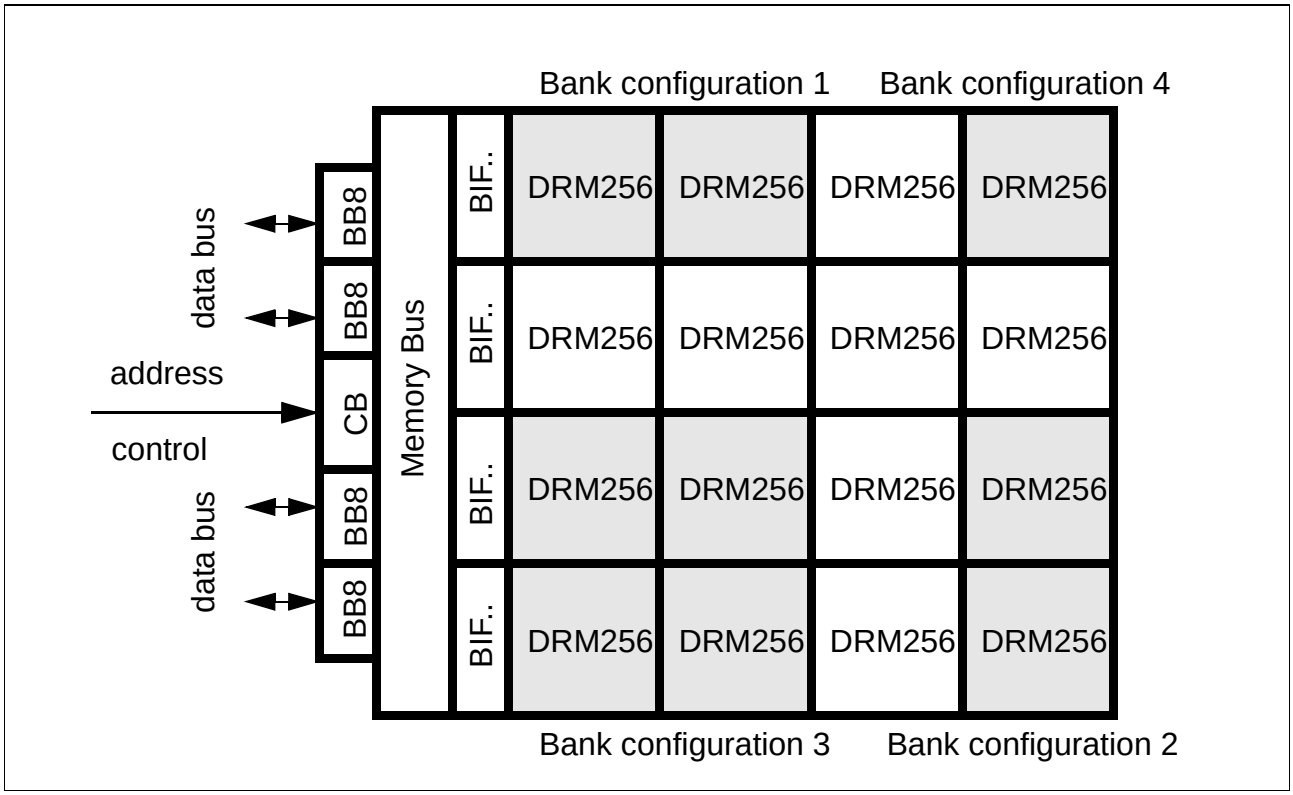


Figure 8
Possible Bank Configurations

The memory block contains a bank decoder part, which is configured by Siemens to the request of the actual design. Several different bank configurations are possible. Figure 8 shows examples, they are described below. Each configuration has its own area of application. However, a memory module may only contain identical banks, as opposed to the module sketched in figure 8.

Bank Configuration 1

One memory stripe is connected to all data bits of the data bus. This configuration provides a 2K page size with a data bus of up to 64 bits, depending on the choice of bus interface and bus bridge. It provides optimized power consumption, because only one bus interface is needed for 512 Kbit of memory.

Bank Configuration 2

Two memory blocks are connected in parallel to the memory bus. Again, there are 512 Kbit with a 2K page size, but the bus interface width is doubled as compared to configuration 1. This configuration is useful, if a wide data bus is needed and also a large number of banks is required.

Bank Configuration 3

Four memory blocks are combined to one bank with a bus interface of 128 bit. This configuration gives a pagelength of 4096 bit together with a wide data bus.

Bank Configuration 4

A single memory block could also be a memory bank. This allows a large number of memory banks per memory module. As the number of sense amplifiers to be charged at a time is minimized, minimum power consumption results. Note that the maximum supported number of banks is 16.

3.4 VHDL Model

Modular embedded DRAM comes with a VHDL model for early simulation. The VHDL model is foreseen for the logic- and gate-level simulation with leapfrog. Upcoming additional packages will provide all timing values for the gate level simulation for the three modes “slow” (worst), “typical” and “fast” (best). Currently the VHDL model uses the timing values as given in table 20.

All timing values and configuration parameters are checked within the DRAM model. If any configuration parameter is incorrect or a timing violation occurs, a warning or an error in the log window and log file of leapfrog is issued. Therefore it is recommended always to check these warning and error messages.

3.4.1 Implementation

To implement the DRAM model the following files are necessary:

dram_behav_comp_pack.vhd	*(package & package body)*
conversions.vhd	*(package & package body)*
dram_mem_packp.vhd	*(package)*
dram_mem_packb.vhd	*(package body)*
top_of_dram_conf_ssc.vhd	*(configuration)*
top_of_dram_sse.vhd	*(entity)*
top_of_dram_rtl_ssa.vhd	*(architecture)*
dram_behav_conf_ssc.vhd	*(configuration)*
dram_behav_sse.vhd	*(entity)*
dram_behav_rtl_ssa.vhd	*(architecture)*

Additionally the DRAM model uses the packages *std_logic_1164* and *std_logic_arith* from the IEEE library and the package *bv_arithmetic* from the Synposys library.

To implement the DRAM model instantiate it in your *architecture* as usual and add the following declarations:

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_arith.ALL;
USE WORK.dram_behav_comp_pack.ALL;

CONSTANT severity_lvl       : severity_level:= WARNING;
CONSTANT wordwidth_c        : Integer :   = 64;          --example value!
CONSTANT x_c                : Integer :   = 4;           --example value!
CONSTANT y_c                : Integer :   = 4;           --example value!
CONSTANT banks              : Integer :   = 4;           --example value!
CONSTANT blocks_per_bank    : Integer :   = 4;           --example value!
CONSTANT b_addrwidth        : Integer :   = 2;           --example value!
CONSTANT col_addrwidth      : Integer :   = 6;           --example value!

COMPONENT top_of_dram
GENERIC(
                    severity_lvl      : severity_level;
```

```
                    wordwidth       : Integer;
                    x               : Integer;
                    y               : Integer;
                    banks           : Integer;
                    blocks_per_bank : Integer;
                    b_addrwidth     : Integer;
                    col_addrwidth   : Integer);

PORT(
                    edt_test_mux    : IN   std_ulogic;
                    edt_clk_mux     : IN   std_ulogic;
                    ed_resetn       : IN   std_ulogic;
                    edt_resetn      : IN   std_ulogic;
                    edt_cke         : IN   std_ulogic;
                    ed_cke          : IN   std_ulogic;
                    ed_cmd          : IN   std_ulogic_vector(3 DOWNTO 0);
                    edt_cmd         : IN   std_ulogic_vector(3 DOWNTO 0);
                    ed_brow         : IN   std_ulogic_vector(b_addrwidth-1 DOWNTO 0);
                    edt_brow        : IN   std_ulogic_vector(b_addrwidth-1 DOWNTO 0);
                    ed_arow         : IN   std_ulogic_vector(7 DOWNTO 0);
                    edt_arow        : IN   std_ulogic_vector(7 DOWNTO 0);
                    ed_bcol         : IN   std_ulogic_vector(b_addrwidth-1 DOWNTO 0);
                    edt_bcol        : IN   std_ulogic_vector(b_addrwidth-1 DOWNTO 0);
                    ed_acol         : IN   std_ulogic_vector(col_addrwidth-1 DOWNTO 0);
                    edt_acol        : IN   std_ulogic_vector(col_addrwidth-1 DOWNTO 0);
                    ed_lat          : IN   std_ulogic_vector(2 DOWNTO 0);
                    edt_lat         : IN   std_ulogic_vector(2 DOWNTO 0);
                    ed_di           : IN   std_ulogic_vector(wordwidth-1 DOWNTO 0);
                    edt_di          : IN   std_ulogic_vector(wordwidth-1 DOWNTO 0);
                    ed_do           : OUT  std_ulogic_vector(wordwidth-1 DOWNTO 0);
                    edt_do          : OUT  std_ulogic_vector(wordwidth-1 DOWNTO 0);
                    ed_bwe          : IN   std_ulogic_vector((wordwidth/8)-1 DOWNTO 0);
                    edt_bwe         : IN   std_ulogic_vector((wordwidth/8)-1 DOWNTO 0);
                    tc_clk          : IN   std_ulogic;
                    ed_clk          : IN   std_ulogic;
                    ed_oen          : IN   std_ulogic;
                    edt_mwls        : IN   std_ulogic;  --multi word line select, dummy pin
                    edt_marx        : IN   std_ulogic;  --margin voltage, dummy pin
                    edt_boost_off   : IN   std_ulogic;  --margin on/off, dummy
                    pin (old name: margin)
                    edt_wbi         : IN   std_ulogic;  --
                    edt_redtst      : IN   std_ulogic;  --redundancy test, dummy pin
                    edt_roff        : IN   std_ulogic;  --row redundancy off, dummy pin
                    edt_boff        : IN   std_ulogic); --bit redundancy off, dummy pin
END COMPONENT;

BEGIN
top_of_dram1:top_of_dram
   GENERIC MAP(
                    severity_lvl    => severity_lvl,
```

```
                wordwidth       => wordwidth,
                x               => x,
                y               => y,
                banks           => banks,
                blocks_per_bank  => blocks_per_bank,
                b_addrwidth     => b_addrwidth,
                col_addrwidth    => col_addrwidth)
PORT MAP(
                edt_test_mux       => edt_test_mux,
                edt_clk_mux        => edt_clk_mux,
                ed_resetn          => ed_resetn,
                edt_cke            => edt_cke,
                ed_cke             => ed_cke,
                ed_cmd             => ed_cmd,
                edt_cmd            => edt_cmd,
                ed_brow            => ed_brow,
                edt_brow           => edt_brow,
                ed_arow            => ed_arow,
                edt_arow           => edt_arow,
                ed_bcol            => ed_bcol,
                edt_bcol           => edt_bcol,
                ed_acol            => ed_acol,
                edt_acol           => edt_acol,
                ed_lat             => ed_lat,
                edt_lat            => edt_lat,
                ed_di              => ed_di,
                edt_di             => edt_di,
                ed_do              => ed_do,
                edt_do             => edt_do,
                ed_bwe             => ed_bwe,
                edt_bwe            => edt_bwe,
                edt_resetn         => edt_resetn,
                tc_clk             => tc_clk,
                ed_clk             => ed_clk,
                ed_oen             => ed_oen,
                edt_mwls           => edt_mwls,
                edt_marx           => edt_marx,
                edt_boost_off      => edt_boost_off,
                edt_wbi            => edt_wbi,
                edt_redtst         => edt_redtst,
                edt_roff           => edt_roff,
                edt_boff           => edt_boff);
```

For all generic parameters which end with “_c” enter your own integer constant. Replace your own std_ulogic_vector signals with all interface signals which end with “_i”. The bus widths of each signal are described in the entity top_of_dram_sse.vhd.

In your *configuration* insert the following lines

```
FOR ALL: top_of_dram
```

```
    USE CONFIGURATION work.top_of_dram_conf;
END FOR;
```

The following section describes how to determine the integer constants for the generic map.

3.4.2 Configuration of the DRAM

Configuration of the DRAM is to chose the required number of memory blocks in „x“ and in „y“ direction together with the correct interface and bank setting. The VHDL model has to reflect this choice by having set the corresponding parameters correctly.

If the memory size in bits is M then the product x*y is:

$$x \times y = \frac{M}{262144}$$

„x” stores the number of 256 Kbit blocks in x-direction, „y” stores the number of 256 Kbit blocks in y-direction. Please refer to figure 4 on page 16 for the illustration of „x“ and „y“ direction. One 256 Kbit block contains 262144 bits.

“Wordwidth” is the bus width of the data input signal *di* and the data output signal *do*. Further the constant “banks” stores the number of banks. With the number of banks the value for “blocks_per_bank” can be calculated:

$$\text{blocks\_per\_bank} = \frac{M}{\text{banks} \times 262144}$$

The bus width of the bank address signals “brow” and “bcol” is:

$$\text{b\_addrwidth} = \log_2(\text{banks})$$

Bcol and brow have the same width, given by $\log_2$ of the number of implemented banks. However, if only one bank is implemented, the signals ED_BCOL and ED_BROW still exist, resulting in b_addrwidth = 1. They both then have to be tied to GND for proper operation.

The bus width of the column address signal “acol” is

$$\text{col\_addrwidth} = \log_2\left(\frac{M}{\text{wordwidth} \times \text{banks} \times 256}\right)$$

Table 7 below gives the possible ranges for all configuration parameters.

Table 7
Configuration Ranges

Parameter	Value Range
wordwidth	16, 32, 64, 128
x	1 .. 8
y	1 .. 8
banks	1 .. 16
blocks_per_bank	1, 2, 4, 8
b_addrwidth	1 .. 6
col_addrwidth	1 .. 12
M	256k .. 16Mbit

3.4.3 Simulation

As the selected latency mode influences the maximum clock frequency, it is necessary to have the correct timing parameters set for simulation. The minimum clock period is a function of the latency mode and the number „y“ of 256 Kbit blocks in y-direction as given in table 8. The parameter „n“ in this table also depends on „y“ and has to be set accordingly.

Table 8
Minimum clock period

Latency Mode	minimum clock period	Note
1	19 ns + *n* *1 ns	if y = 1, 2, 3, or 4 *n* = 0 else *n* = y - 4
2	15 ns + *n* *1 ns	
3	12 ns + *n* *1 ns	

3.5 Specification of a Memory Module

This section summarizes the parameters that are necessary to fully specify an embedded DRAM module.The following table gives parameter, possible range and a short description.

Table 9
Memory Module Specification

Parameter	Range	Description
memory size	256Kbit .. 16 Mbit	These two parameters are redundant as one block has 256 Kbit = 262 144 Bit
no. of 256 Kbit blocks	1 .. 64	
no. of blocks in x - direction	1 .. 8	increases possible bus width (refer to figure 7 on page 20)
no. of blocks in y - direction	1 .. 8	increases length of bit lines (see section 3) with constant interface width

Parameter	Range	Description
Width of the data bus	16 .. 512 bit	
size of one bank in x- direction		all banks have to be equal; no. of blocks per bank must be 1, 2, 4, or 8
size of one bank in y- direction		

4 Operation of the Memory Module

Operation of the memory module is controlled via the command bus, the latency setting, the bank address, column address and row address. Data is read and written via the data bus.

4.1 Data Bus

The width of the data bus depends on the actual implementation. It may be configured in a range from 16 bit to 512 bit. DI[] and DO[] can be tied together in order to get a bidirectional bus. In that case the OEN signal must be used for arbitration. If data is to be read from DO[], OEN must be set to „0“, enabling the tri-state buffers. If data is to be written, the tri-state buffers are disabled with OEN = „1“. The layout of the memory module is prepared for such applications, i.e. the DI[] and DO[] signals are paired for each bit.

4.2 Command Bus

The command bus CMD[3..0] is divided in two parts in order to allow operations in different banks with „hidden precharge“. Tables 10 and 11 give the possible row and column commands as controlled by CMD[3:0].

Table 10
ROW Commands

CMD3	CMD2	State	Description
1	1	NOP	No operation
1	0	-	Reserved for future use
0	1	ACTIVATE	One row of a bank is opened for access
0	0	PRECHARGE	The open row is copied back to the memory

Table 11
COLUMN Commands

CMD1	CMD0	State	Description
1	1	NOP	No operation
1	0	-	Reserved for future use
0	1	READ	Read data from the open row
0	0	WRITE	Write data to the open row

The state diagram given in figure 9 shows the possible command flow for one memory bank of a memory module

.

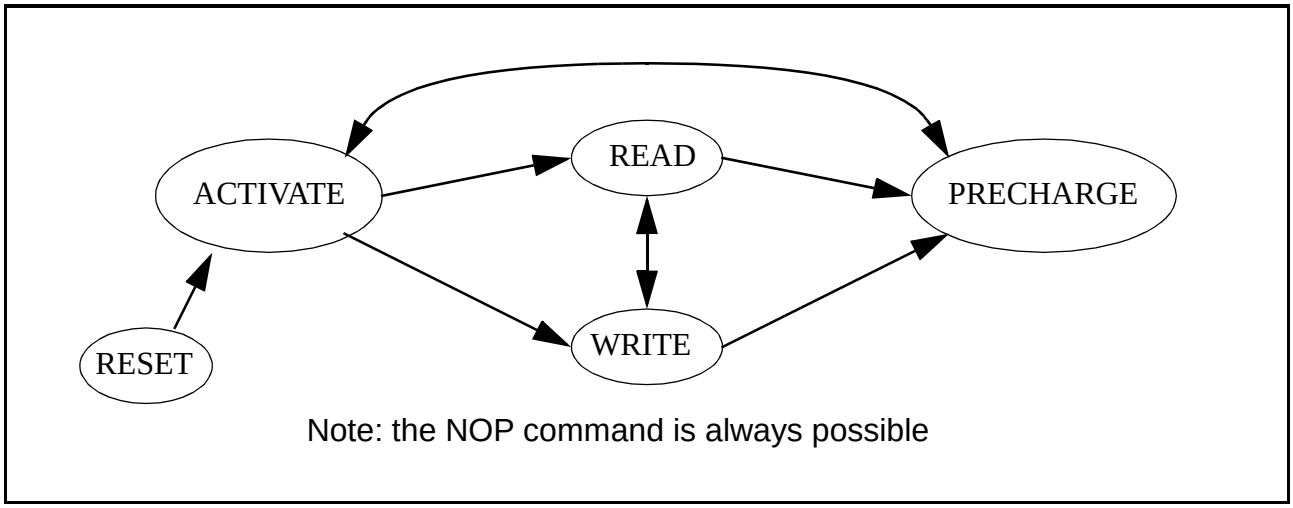


Figure 9
Command Sequences for one Memory Bank

The command bus allows two operations in parallel. ACTIVATE/PRECHARGE of one bank can be combined with READ/WRITE of another bank. This way, multibank operation is provided giving accelerated data transmission with hidden precharge. However, some restrictions apply. The following three tables give the possible combinations.

Table 12
Combinations of ROW and COLUMN Commands for LATENCY 1 and 2

CMD[3:2]	CMD[1:0]	ROW BANK	COLUMN BANK	Status
ACTIVATE	READ/WRITE	X	X	not allowed
ACTIVATE	READ/WRITE	X	Y	not allowed
PRECHARGE	READ/WRITE	X	X	not allowed
PRECHARGE	READ/WRITE	X	Y	allowed

Table12 reads as follows: An ACTIVATE command on a row in bank X together with a READ/WRITE on a column in the same bank X is not allowed (see first line). A PRECHARGE command on a row in bank X at the same time as a READ/WRITE on a column in another bank Y is possible (see last line). The above table is valid independently from the clock cycle time.

If LATENCY 3 is applied, care must be taken to the combinations during succeeding clock periods. The table below gives possible combinations for two clock periods n and n+1, respectively. See section 4.4 for a description of latency modes.

Table 13
Combinations of ROW and COLUMN Commands for LATENCY 3, part 1

CLK	CMD[3:2]	CMD[1:0]	ROW BANK	COLUMN BANK	Status
n	don't care	READ/WRITE	-	X	
n+1	ACTIVATE	don't care	X	-	not allowed
n	don't care	READ/WRITE	-	Y	
n+1	ACTIVATE	don't care	X	-	not allowed
n	don't care	READ/WRITE	-	X	
n+1	PRECHARGE	don't care	X	-	not allowed
n	don't care	READ/WRITE	-	Y	
n+1	PRECHARGE	don't care	X	-	allowed

Table 13 is valid independently from the clock cycle time.

The following table gives possible combinations for LATENCY 3 if row and column commands are given on the same clock edge.

Table 14
Combinations of Row and Column Commands for LATENCY 3, part 2

CMD[3:2]	CMD[1:0]	ROW BANK	COLUMN BANK	Status
ACTIVATE	READ/WRITE	X	X	$t_{CL3} >= t_{ACD}$
ACTIVATE	READ/WRITE	X	Y	$t_{CL3} >= t_{ACD}$
PRECHARGE	READ/WRITE	X	X	not allowed
PRECHARGE	READ/WRITE	X	Y	allowed

4.3 Address Bus

The address bus is divided into four parts: column address ACOL, row address AROW, and bank addresses BCOL and BROW.

As one block has 256 rows, the AROW bus is always 8 bit wide. The width of the ACOL bus depends on the number of columns in one bank, the page length and the wordwidth (see section 3.4.2).

BCOL and BROW have the same width, given by $\log_2$ of the number of implemented banks. Note that two banks require a single signal to select the banks. However, if only one bank is implemented, the signals BCOL[] and BROW [] still exist. They then have to be tied to GND for proper operation.

4.4 Setting Latency Modes

A programmable pipe line of three registers in the interface logic allows to trade clock frequency for latency. The clock frequency for LATENCY 3 is higher than for LATENCY 1. „LATENCY 1“ means that the data is available on DO[] one clock cycle after a given command. „LATENCY 2“ means the data is available two clock periods after the command, etc. Please refer to sections 6.4 to 6.6 for detailed timing diagrams. The setting of possible latency configurations for the data path with LAT[2:0] are given in table 15.

Table 15
Setting Latency Modes

LAT2	LAT1	LAT0	State	Description
0	0	0		reserved for future use
0	0	1	LATENCY 1	The register for the address is active
0	1	0		reserved for future use
0	1	1		
1	0	0		
1	0	1	LATENCY 2	LATENCY 1 + output register
1	1	0		reserved for future use
1	1	1	LATENCY 3	LATENCY 2 + decoded address register

4.5 Byte Write Enable

Setting a bit on BWE to „1“enables writing of the corresponding byte only instead of the complete data word. This allows to write bytes of 8 bits each separately on a given location into an addressed word. The following figure gives an example for a memory module with a 32 bit interface using the byte write capability. As the 32 bit word contains four 8-bit bytes, BWE is a 4-bit wide bus selecting the corresponding bytes of the data word. If all four bits of BWE are set to „1“during a WRITE; all 32 data bits are written into the memory block. Setting only the LSB of BWE to „1“only writes the data on DI[7..0] into the memory block without affecting the other bits.

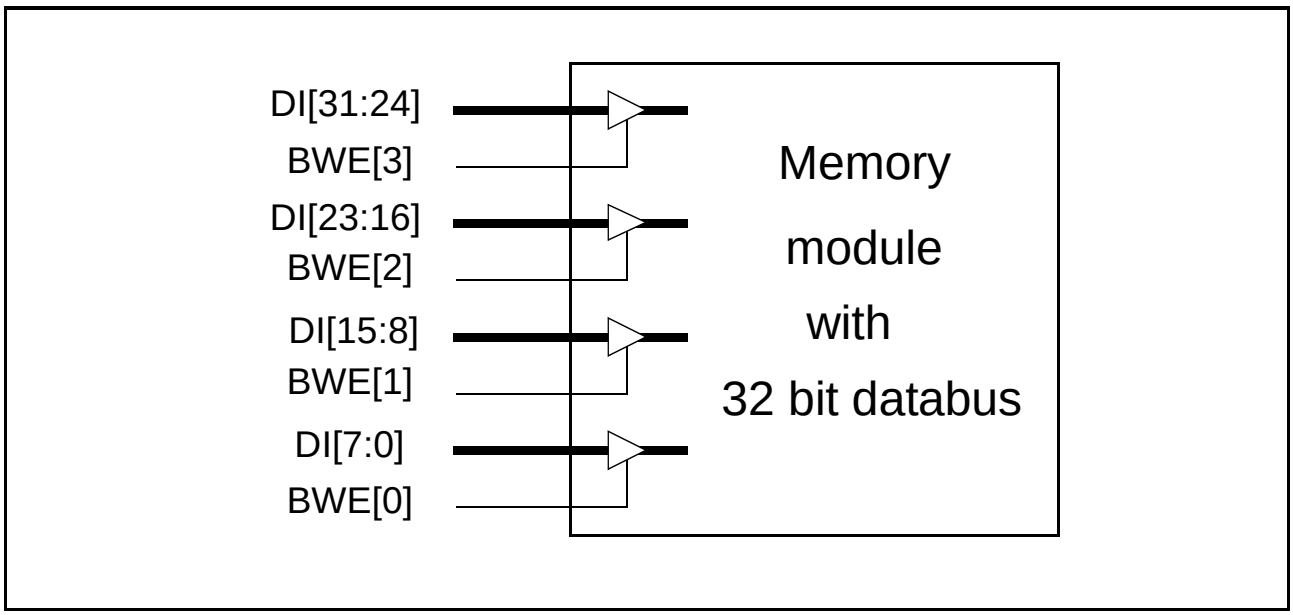


Figure 10
Use of the BYTE WRITE ENABLE Signal

5 Test Concept

Embedded DRAM requires a dedicated solution for testing, derived from common DRAM test methods and taking into account the interaction with logic such as CPU cores, SRAM, ROM, etc. Two approaches are supported to access and communicate with the embedded DRAM core:

- direct access but with the help of multiplexer logic and registers to reduce the number of pins which must be accessed simultaneously
- built-in logic to partially support algorithmic pattern generation and expected-value comparison.

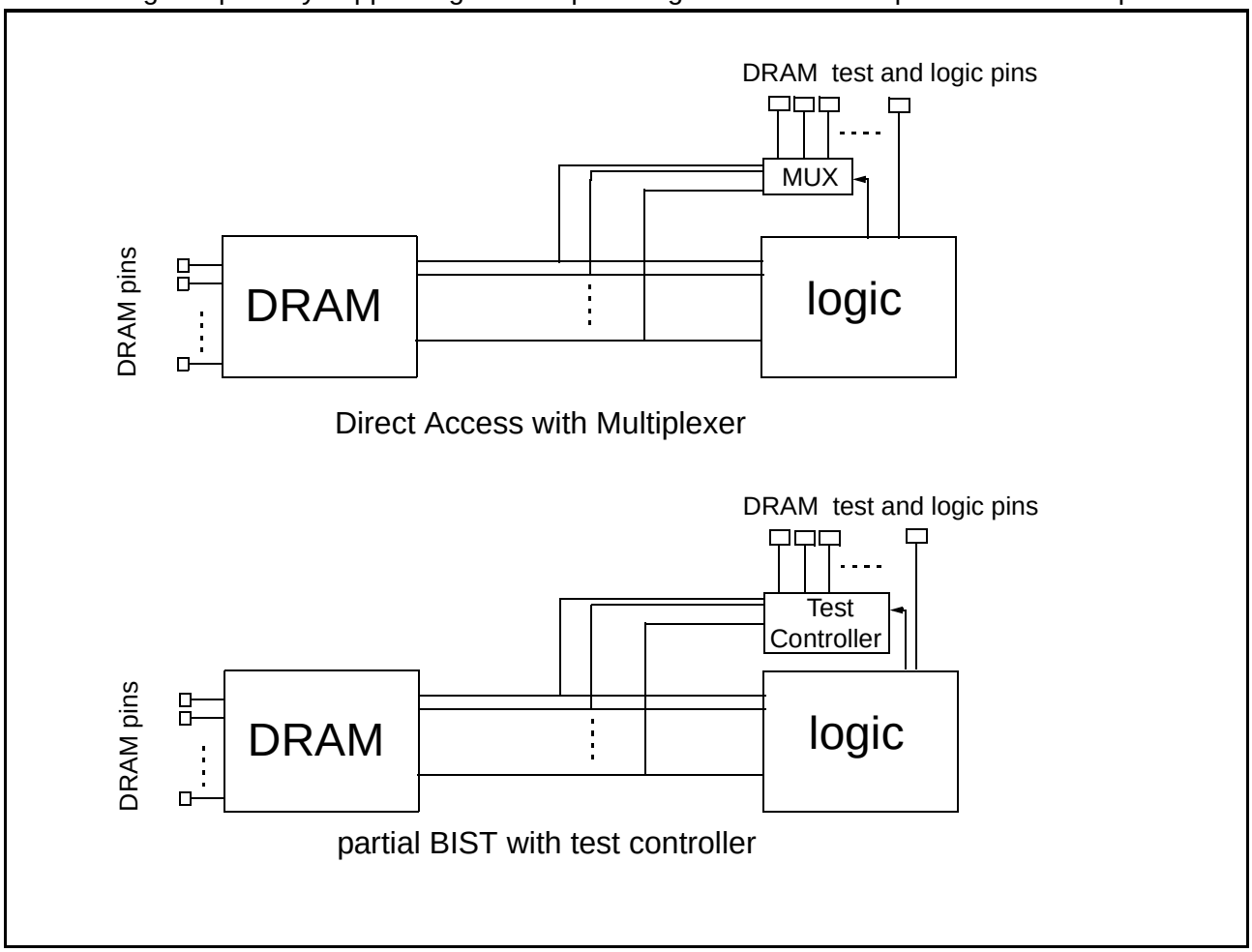


Figure 11
DRAM Test Methods

See figure 11 for an illustration of different test methods. These methods are in order of increased silicon area and decreased test effort. Direct access would require additional pins or multiplexer logic and registers. The Test Concept allows the use of standard memory test software, for example the ‘ig900’ suite from Teradyne. Thus, existing memory test software may be used for embedded DRAM products. However, ATE’s (Automated Test Equipment, or “tester”) with memory test option “MTO” capability are required.

If the embedded DRAM product has DRAM Algorithmic Pattern Generation logic on-chip, any standard ATE can be used for testing. During wafer test, memory test ATE is required for redundancy and repair. For package test, where no wafer fail map is generated, internally generated patterns can be applied and the results analyzed as pass/fail. Such a test can be performed by any standard logic tester.

The embedded DRAM test concept is based on an on-chip test controller. It provides the necessary multiplexers to allow direct access as well as optional additional logic to partially support algorithmic pattern generation and expected-value comparison. This provides the following advantages:

- efficient production test to standard DRAM quality,
- information recovery for repair & yield statistics,
- transparent access for fault analysis

The test concept is derived from Siemens' 16 Mbit technology and offers a quality level comparable to that achieved for standard DRAM components in this technology. All of the tests used for 16 Mbit EDO and SDRAM are available.

Tester software is available for the **Teradyne J-971** tester with **Memory Test Option** (MTO). Test of packaged parts can be performed using any standard logic tester, provided that the test software in TOPS format can convert the test patterns to the target ATE.

The test software together with the test controller provide industrial level quality, being:

- product defect quality: 20 - 100 dpm
- average failure rate: for > 3000 h 10 fit

If lower quality standards are sufficient, the test can be simplified, reducing test time.

5.1 The Test Controller

The test controller provides a generic hardware interface between ATE and DRAM core. As it presents the same hardware to the ATE regardless of the chip in which it is embedded, it allows the re-use of existing test software for DRAMs. Only minor modifications are required, e.g. for start-up sequences.

It is delivered in the form of a synthesizeable VHDL description which is configured for the particular design. This VHDL code is integrated in the customer VHDL code. The logic then is synthesized using the customer's library. The logic requires approximately (3+*i*/5)k gates, *i* being the actual DRAM size in MBit. This offers a "shrink-wrapped" solution, which requires no additional logic development and limited tester software development on the part of the customer.

The test controller has a 16-bit control/address interface and an 8-bit, 16-bit or 32-bit data interface as options. The width of the data bus can be selected depending on the application.

It provides two operating modes, *direct access* and *pattern generation (partial BIST)*. The *direct access* mode reduces the number of external pins needed for direct DRAM access. It corresponds to the top diagram in figure 11. It allows the use of test programs with the same functionality as standard tester software for DRAMs. This includes gathering Bit Fail Maps for repair (fusing) and yield analysis. The *pattern generation* mode uses built-in logic to support algorithmic pattern generation and expected-value comparison using an external ATE.

A typical application of the test concept with test controller has three phases during product ramp-up and production:

- *First silicon, engineering samples*: test with test controller in *direct access mode*,
- *Wafer test:* test with test controller in *direct access mode* and *pattern generation mode*
- *Package test:* test with test controller in *pattern generation mode* (partial BIST).

The test controller is activated by the custom logic with a „1"at the TEST signal. This signal can be provided with a dedicated pin or by register programming in the custom logic. In the latter case, the implementation has to provide a start-up sequence with the correct default values.

5.1.1 Tester Interface

The test controller in a 16-bit data configuration requires 32 external pins for ATE access, and three test signals (local signals) for activation and test mode selection. As an option, 32 bit data is provided resulting in a 48 bit interface. If these local signals are not provided by the user, register bits are used for activation of the test interface logic, and the user has to provide a start-up sequence for putting the chip in embedded DRAM test mode. Note that the speed and drive capability of pins that are used for tester access may affect the maximum test program speed.

Table 16
Signals to the Tester:

Signal Name	Width	Type	Channel	Description
TC_DATA [15/31:0]	16/32	I/O	VMO	Data In/Out
TC_ADDR [7:0]	8	O	VMO	Address Bus
TC_CMD [3:0]	4	O	STD	Test Controller Command Bus
TC_FAIL	1	I/O	STD	Output Fail
TC_CLK	1	O	STD	Controller Clock
TC_RESN	1	O	STD	Reset
TC_DDIR	1	O	STD	I/O Data Pad Direction Select

Table 17
Signals from Custom Logic:

Signal Name	Type	Description
TC_TEST	I	Set Test Controller and DRAM test mode
TC_BURNIN	I	Set general burn-in mode (e.g. regulator off)
TC_MARCH	I	Starts March6 Test if set to „1“. Connect to GND if not used.

The n-well voltage has to be varied in different DRAM patterns. A NWA **pad** for wafer test and analysis is required, as is a 25-volt source pad (this does need not to be an external pin). This is an analog signal and cannot be generated from test controller logic.

The DRAM has to have separate VDD and GND pins and pads for Wafer Burn In and Package Burn-In.

5.1.2 Multiplexed Data Access

Two different buses TC_DI and TC_DO have to be supplied to the Test Controller, while only one bus TC_DATA connects the tester to the device. A multiplexer is provided in the pads to select the direction as given by the signal TC_DDIR. See figure 12 for an illustration.

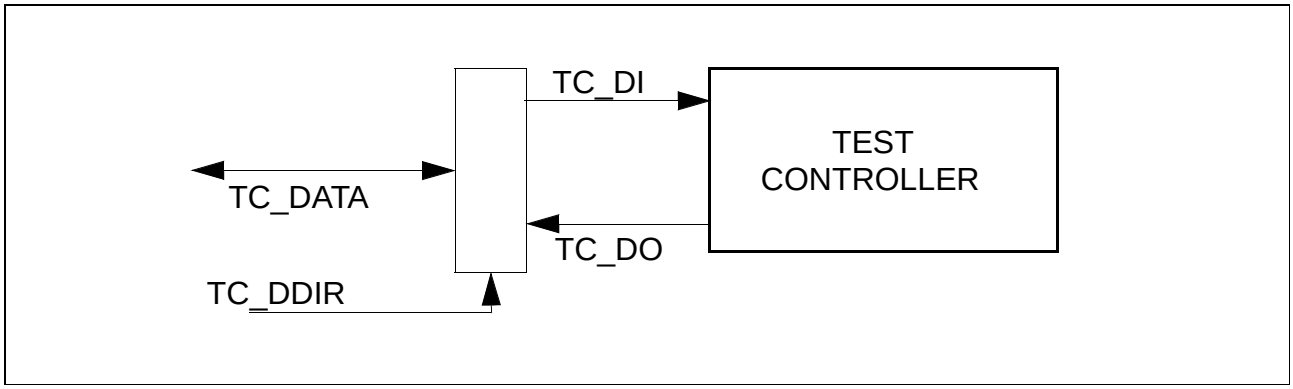


Figure 12
Bidirectional Pad for Tester Access

5.1.3 Built-in March 6 Test Sequence

The test controller can automatically execute a March-6 Test sequence. The test controller enters the March-6 Test sequence if, when the signals ‘tc_test’ and ‘tc_resn’ become ‘1’, the command ‘set_pg_run’ is applied to the command bus ’tc-cmd’. The March sequence continues until: ‘tc_test’ is no longer ‘1’, ‘tc_resn’ is no longer ‘1’, or the command ‘set_pg_run’ is no longer applied.

During the March-6 test, each cell of each module is addressed in sequence; every cell of the DRAM (except redundant cells) is tested for hard failures. The signal ’tc_burn_in’ affects the March test as follows: when ’tc_burn_in’ is ‘1’, the regulator is turned off, and the ‘tc_fail’ signal is latched if it goes high; otherwise, the regulator is on and ‘tc_fail’ is a combinatorial output. Once the ‘tc_fail’ signal has been latched, it can be cleared either by toggling ‘tc_burn_in’ or with a Reset.

Note that the DRAM is addressed block by block, without background refresh. In the case of a burn-in or a test clock which is of a much lower frequency, the maximum refresh interval must be respected. A typical minimum frequency is 4 Mhz, for 3 Modules of the same size (or 2 Modules of sizes differing by 1 block).

The steps of the March-6 test are:

1. Write (Physical) Zero, single cycle, ripple word, all addresses
2. Read Zero & write One, single cycle, ripple word, all addresses
3. Read One & write Zero, single cycle, ripple word, all addresses
4. Read Zero, single cycle, ripple word, all addresses

During the March-6 tests, the inputs’tc_addr’, ‘tc_data’, and ‘tc_ddir’ are ignored.

5.1.4 Special Test Modes

The test controller supports special test modes of the DRAM core as described below.

Margin On/Off, Margin, Boost Off

When the Margin control is enabled, the margin can be set over a range specified by 5 bits of Margin. In normal operation the Boost is turned off whenever a specific Margin level is set, however the Boost can also be turned off separately.

Multi-Word-Line Select

When the “mwls” signal is active, the DRAM interface logic activates groups of Word Lines simultaneously. The Word Lines {0, 4, 8,...} form a group, as do {1, 5, 9,...}, {2, 6, 10,...} and {3, 7, 11,...}. An entire group is activated whenever any of the Word Lines in the given group is addressed.

Regulator On/Off

The built-in voltage regulator is enabled or disabled using this control. If the regulator is turned off, the external supply voltage is applied directly to the DRAM core.

Wafer Burn-in

When the “wbi” signal is active and the “mwls” is active, the DRAM interface logic activates all Word Lines simultaneously whenever an Activate command is given. *The wafer burn-in control only works in conjunction with multi-word-line select!*

Redundancy Controls

The redundant Bit and Word lines can be accessed using a combination of the three controls “redtest”, “boff” and “roff”. These controls affect, respectively, General Redundancy, Row (Word-line) Redundancy and Bit (Column) Redundancy.

n-Well

The n-Well voltage is not accessible through the test controller. It can be set directly from the ATE if an “nwa” pad is made available. The 25-volt supply voltage can be controlled directly from the ATE if a ’’V-25’’ pad is made available.

5.1.5 Test Controller Test

The test controller logic itself also is tested using functional patterns. The pattern generator is loaded using scan chains; if scan test is to be included, these scan chains can also serve that purpose.

6 Operating Conditions

Table 18
Operating Conditions

Symbol	Parameter	min	max	Unit
V_{DD}	Supply voltage	3.0	3.6	V
T_{OP}	Operating temperature	-25 [1)]	70	°C

1) -25 °C is the technological limit. For consumer applications it is recommended to use a higher limit of 0 °C.

6.1 DC Parameters

Table 19
Electrical Parameters

Symbol	Parameter	min	max	Unit
I_{DD}	Supply current		t.b.d.	mA
C_I	Input capacitance of CLK, TCLK		200	fF
C_I	Input capacitance of all other inputs		15	fF
C_O	Output capacitance of DO, TDO		t.b.d.	fF

6.2 AC Parameters

The following table contains all timing parameters of the timing diagrams in the next chapters. The given values are pre-silicon and based on simulations with a high production yield as main design target.

Table 20
Timing Parameters

Symbol	Parameter	min	max	Unit	Notes
t_{CLL}	Clock low time	4		ns	
t_{CLH}	Clock high time	4		ns	
t_{CLR}	Clock rise time		1	ns	
t_{CLF}	Clock fall time		1	ns	
t_{SU}	Setup time	2		ns	
t_{HO}	Hold time	0		ns	

Symbol	Parameter	min	max	Unit	Notes
t_{SUL}	Setup time for LATENCY	2		clocks	
t_{HOL}	Hold time for LATENCY	2		clocks	
t_{ZHL}	Delay time, buffer tri-state to buffer active		3	ns	
t_{HLZ}	Delay time, buffer active to buffer tri-state		3	ns	
t_{RLO}	Reset low time with inactive memory	200		ns	
t_{RVV}	Delay time from RESET inactive to first ACTIVATE	200		ns	
t_{APD}	Delay time between ACTIVATE and PRECHARGE	30		ns	
$t_{PAD\ (1)}$	Delay time between PRECHARGE and ACTIVATE of same row	30	see t_{PAD} (2)	ns	1)
$t_{PAD\ (2)}$	Refresh period for one memory row	see t_{PAD} (1)	64	ms	2)
t_{ACD}	Delay time between ACTIVATE and COLUMN operation to the same bank	25		ns	
t_{CL1}	Clock period for LATENCY 1	19+n		ns	3)
t_{AA1}	Address access time for LATENCY 1	44+n		ns	3)
t_{CO1}	Clock to valid output delay at LATENCY 1		19+n	ns	3), 4)
t_{DH1}	Clock to output hold at LATENCY 1	2		ns	
t_{MO1}	Modify time: t_{CL1} - t_{CO1} - t_{SU}				
t_{CL2}	Clock period for LATENCY 2	15+n		ns	3)
t_{AA2}	Address access time for LATENCY 2	42+n		ns	3)
t_{CO2}	Clock to valid output delay for LATENCY 2 and LATENCY 3		4	ns	4)
t_{DH2}	Clock to output hold at LATENCY 2	2		ns	
t_{MO2}	Modify time: t_{CL2} - t_{CO2} - t_{SU}				
t_{CL3}	Clock period for LATENCY 3	12+n		ns	3)
t_{AA3}	Address access time for LATENCY 3	42+n		ns	3)

Symbol	Parameter	min	max	Unit	Notes
t_{CP3}	Delay between COLUMN operation and PRECHARGE to the same bank, LATENCY 3	2 * t_{CL3}			5)
t_{MO3}	Modify time: t_{CL3} - t_{CO2} - t_{SU}				

1) The same parameter is relevant for the refresh period.

2) During the refresh period all 256 rows of one memory block must be activated and precharged.

3) If the number of memory blocks in a memory stripe is less or equal than 4 then n = 0. For all other values *n = (number of blocks in stripe) - 4*.

4) Output load 1 pF

5) Due to LATENCY 3 a rising clock edge is required during t_{CP3}.

6.3 General Timing Diagrams

The following timing diagrams are only valid during normal operation of the memory module, e.g. at TEST = 0. The aspects of operating the memory module in test mode are covered in section 4.

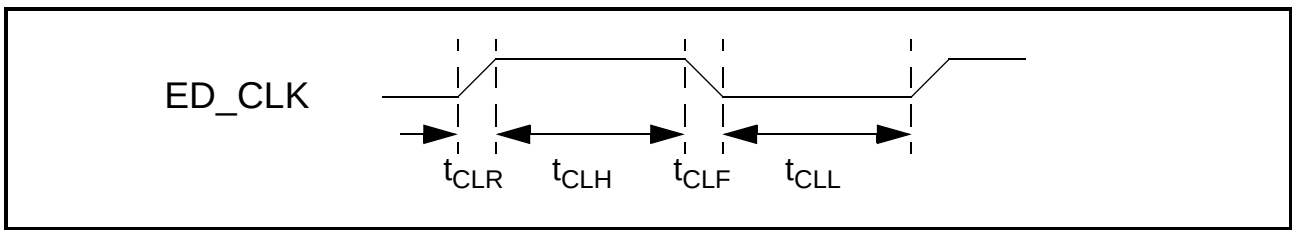


Figure 13
Clock Parameters

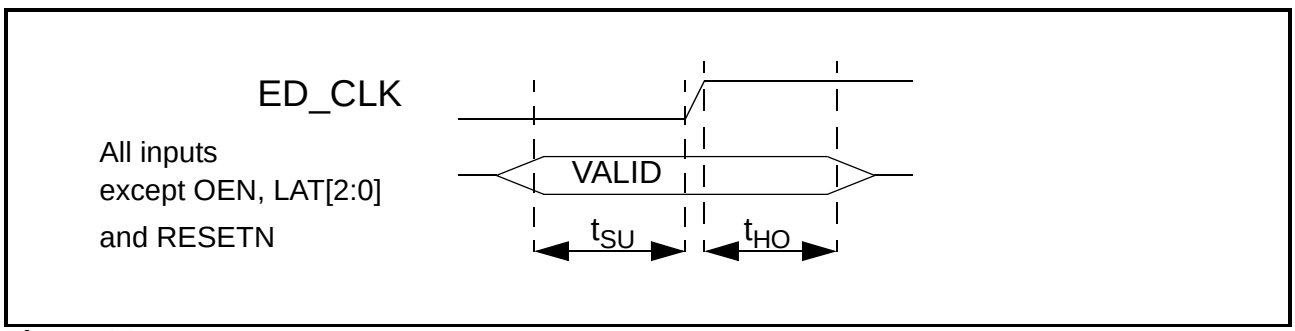


Figure 14
Setup and Hold Timing

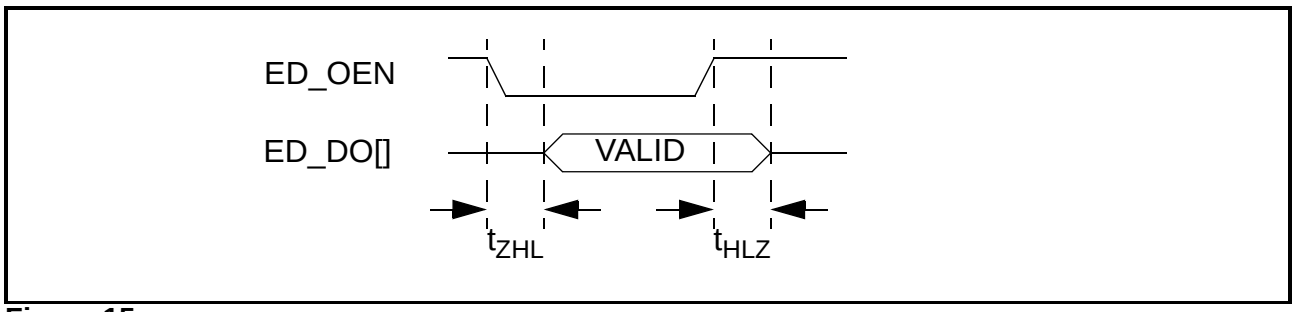


Figure 15
OEN Timing

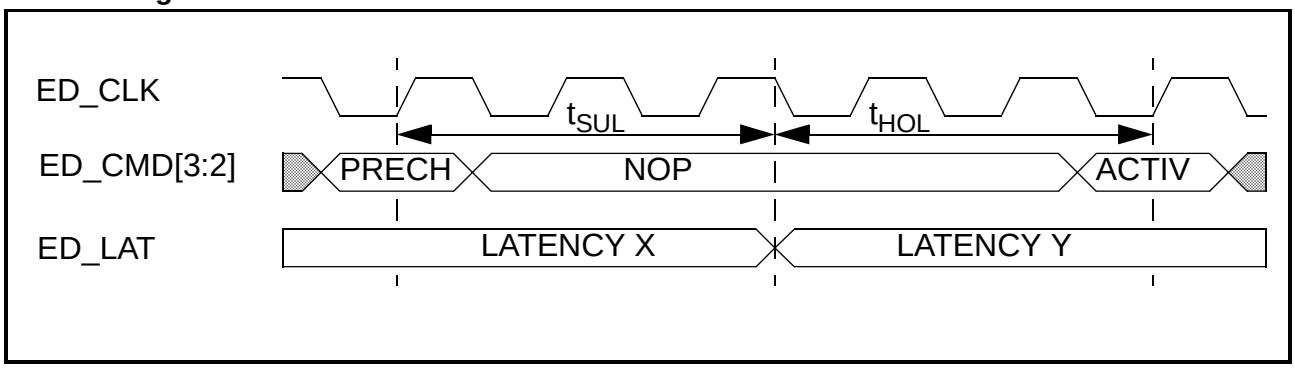


Figure 16
Setup and Hold Timing for LATENCY

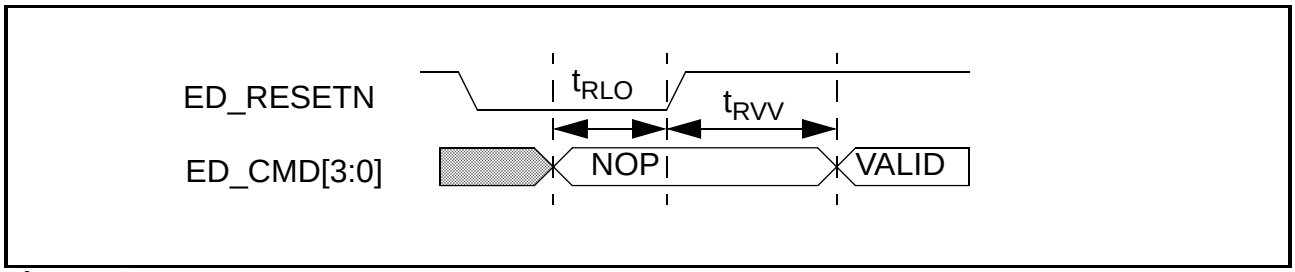


Figure 17
RESETN Timing

To enhance the appearance of the following timing diagrams, a convention for the row and column command NOP is introduced, see figure 18.

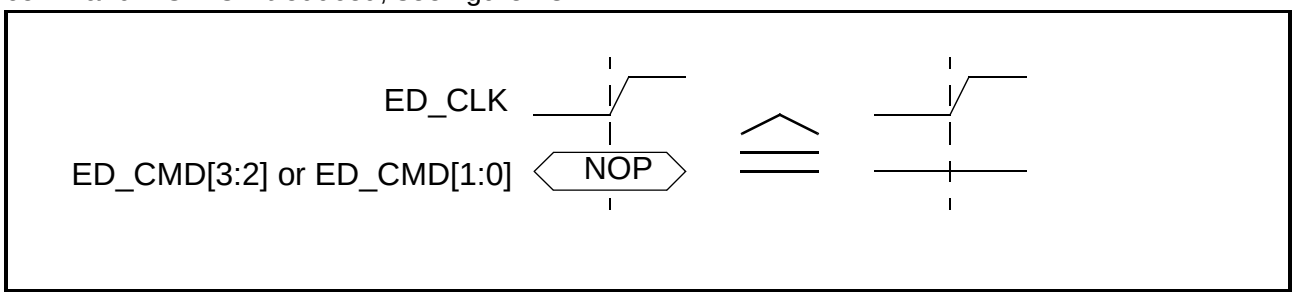


Figure 18
Convention for Presentation

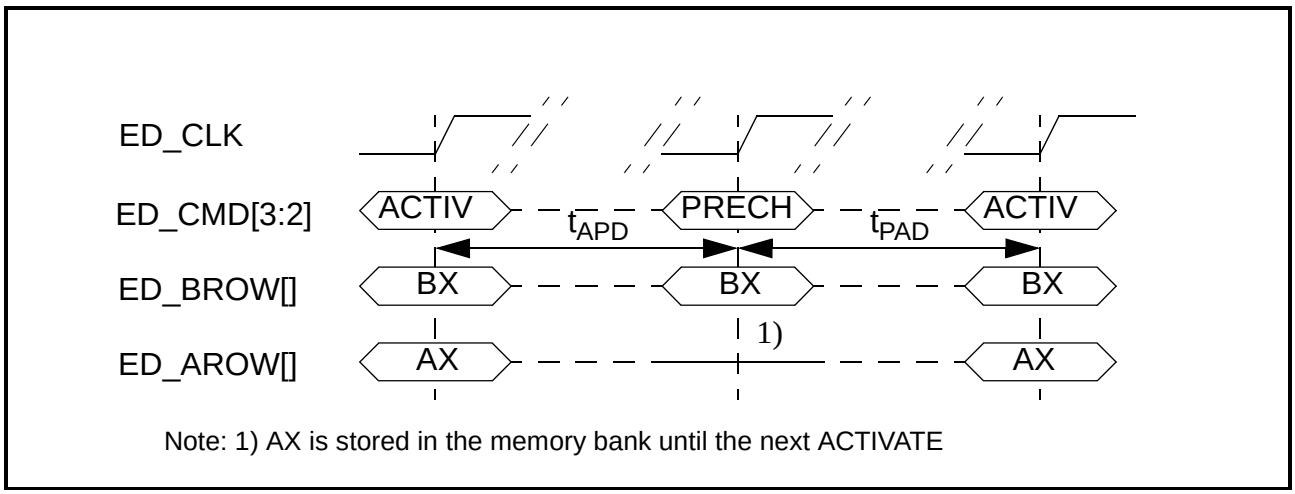


Figure 19
Delay between ROW Commands to the same Bank

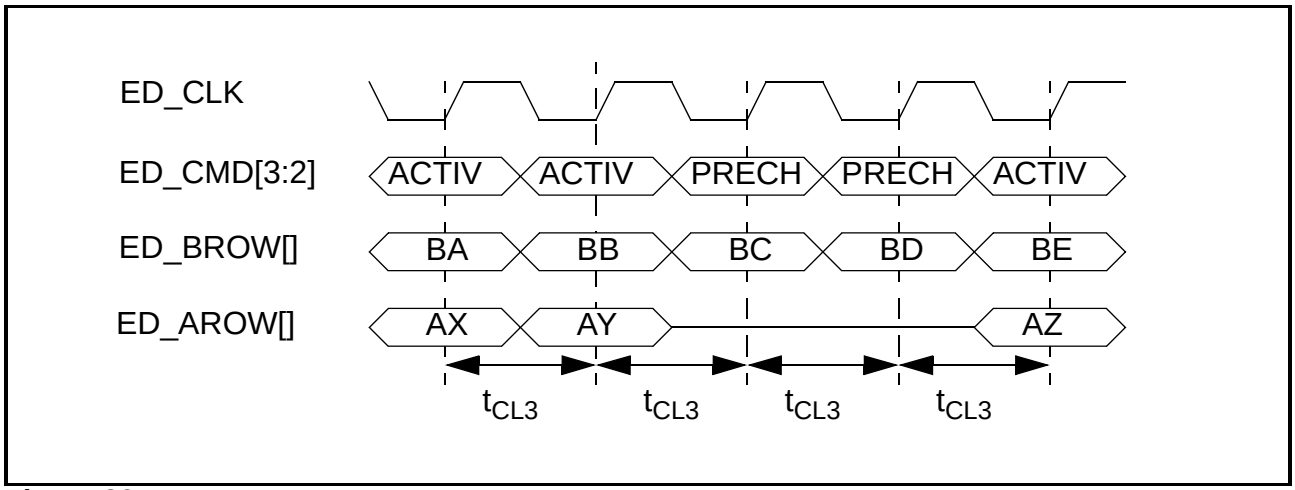


Figure 20
Delay between ROW Commands to different Banks

6.4 Timing Diagrams for LATENCY 1

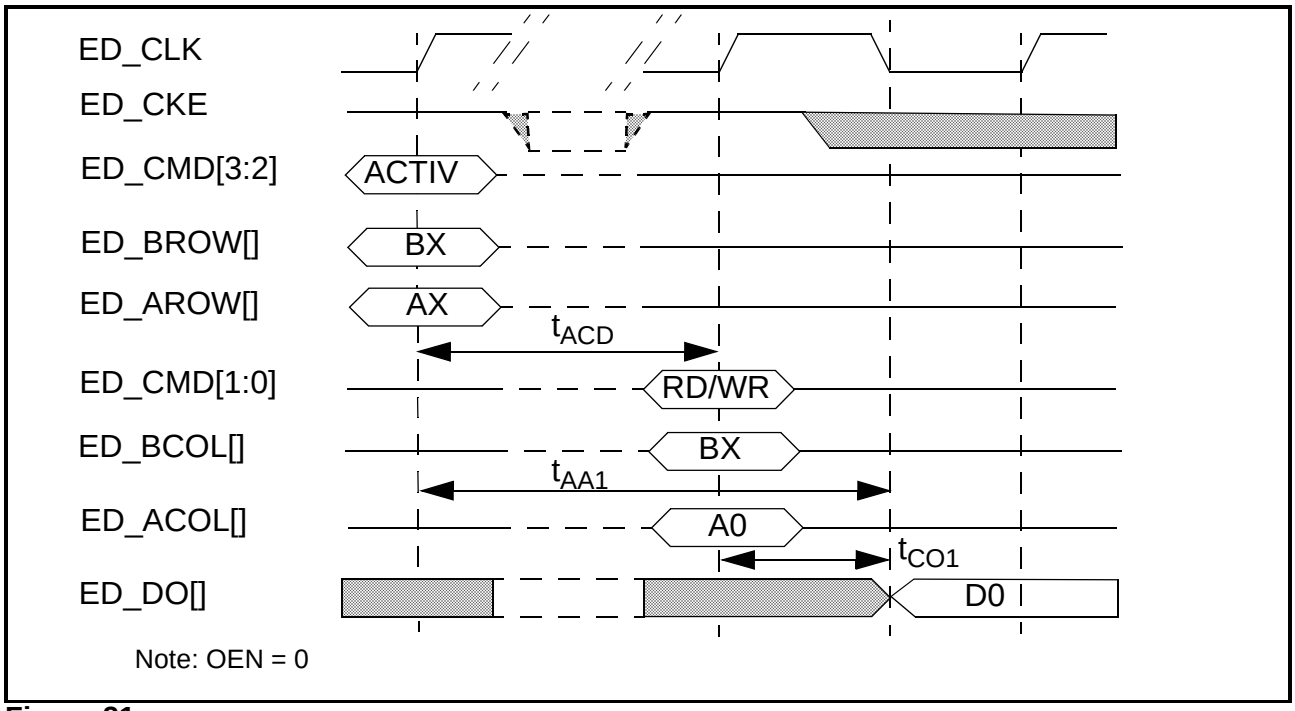


Figure 21
Delay between ACTIVATE and READ/WRITE with LATENCY 1

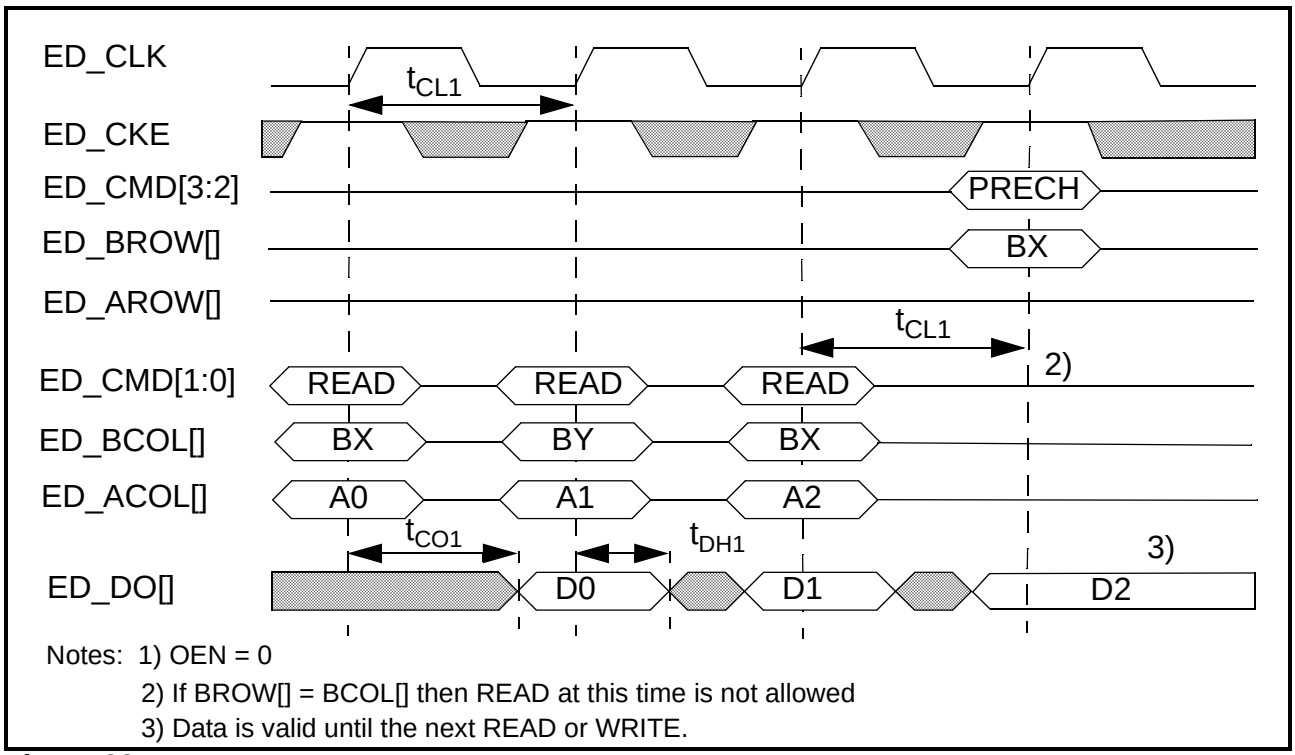


Figure 22
Read Timing, LATENCY 1

.

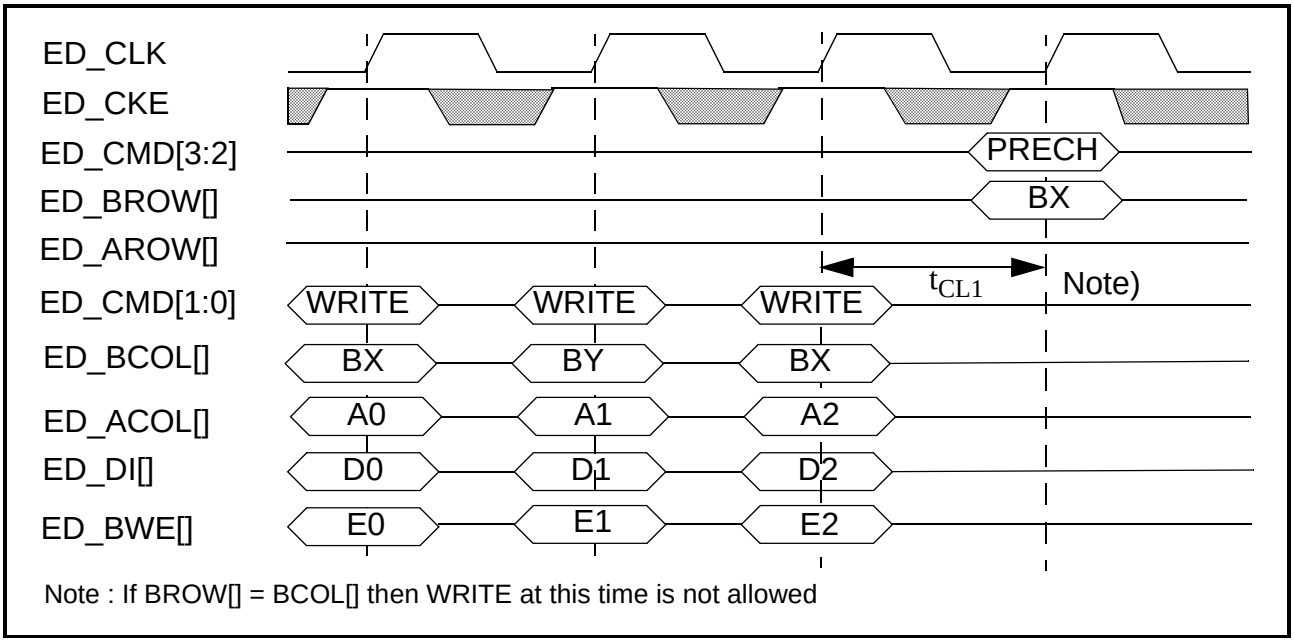


Figure 23
Write Timing, LATENCY 1

.

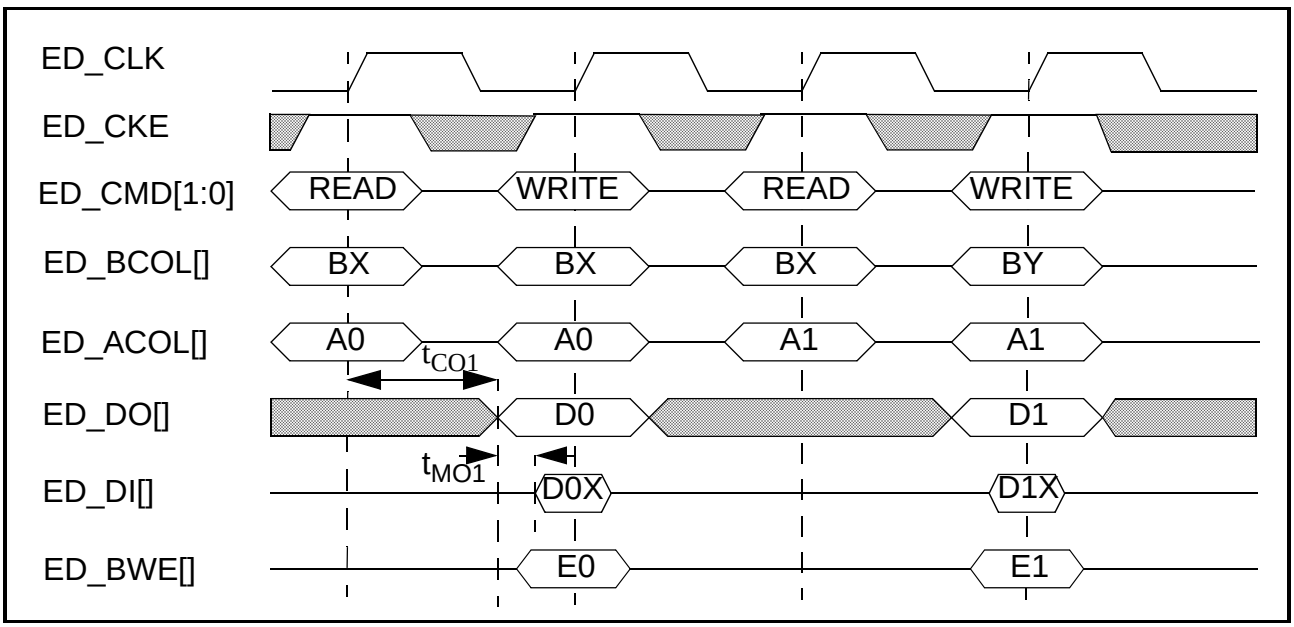


Figure 24
Read-Modify-Write Timing, LATENCY 1

6.5 Timing Diagrams for LATENCY 2

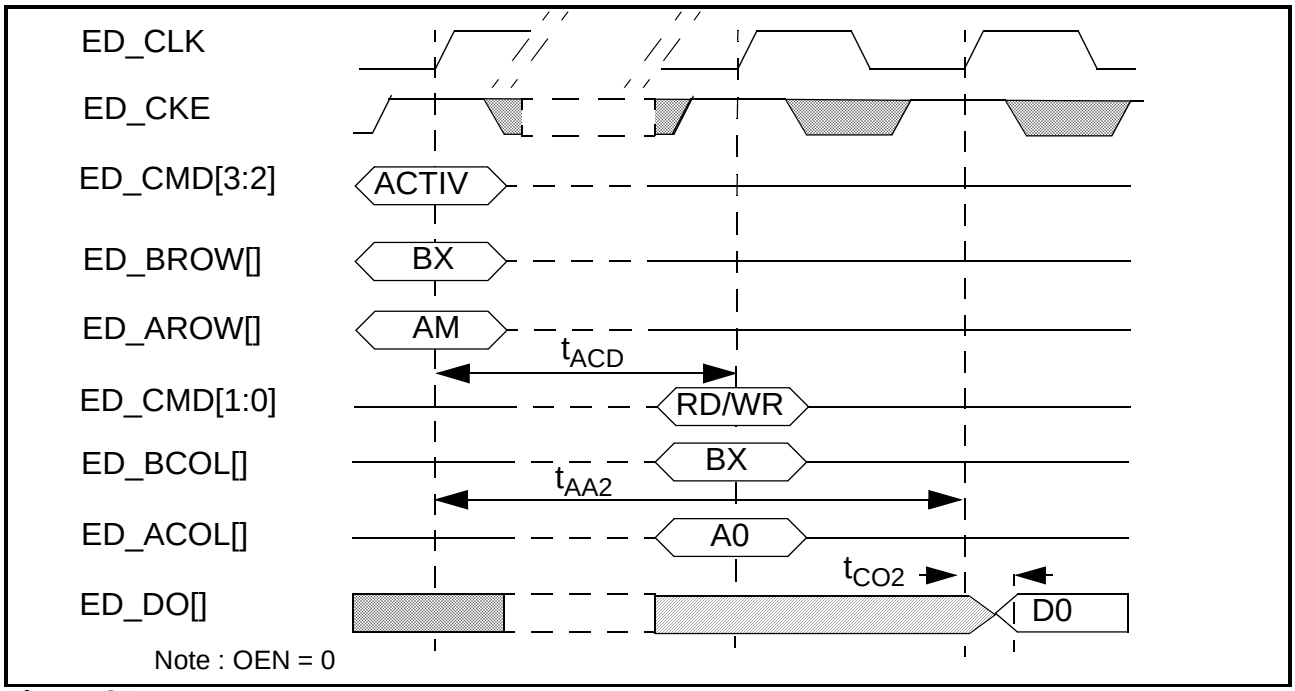


Figure 25
Delay between ACTIVATE and READ/WRITE with LATENCY 2

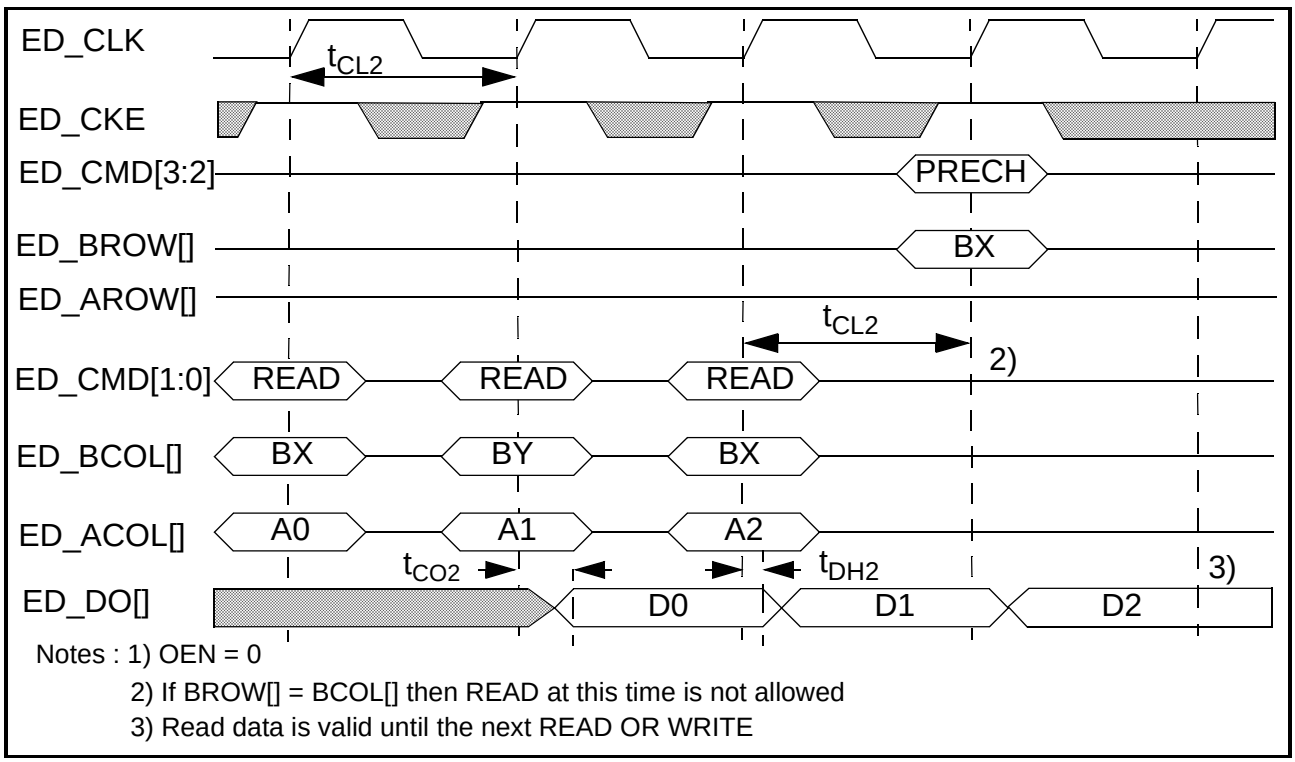


Figure 26
Read Timing, LATENCY 2

.

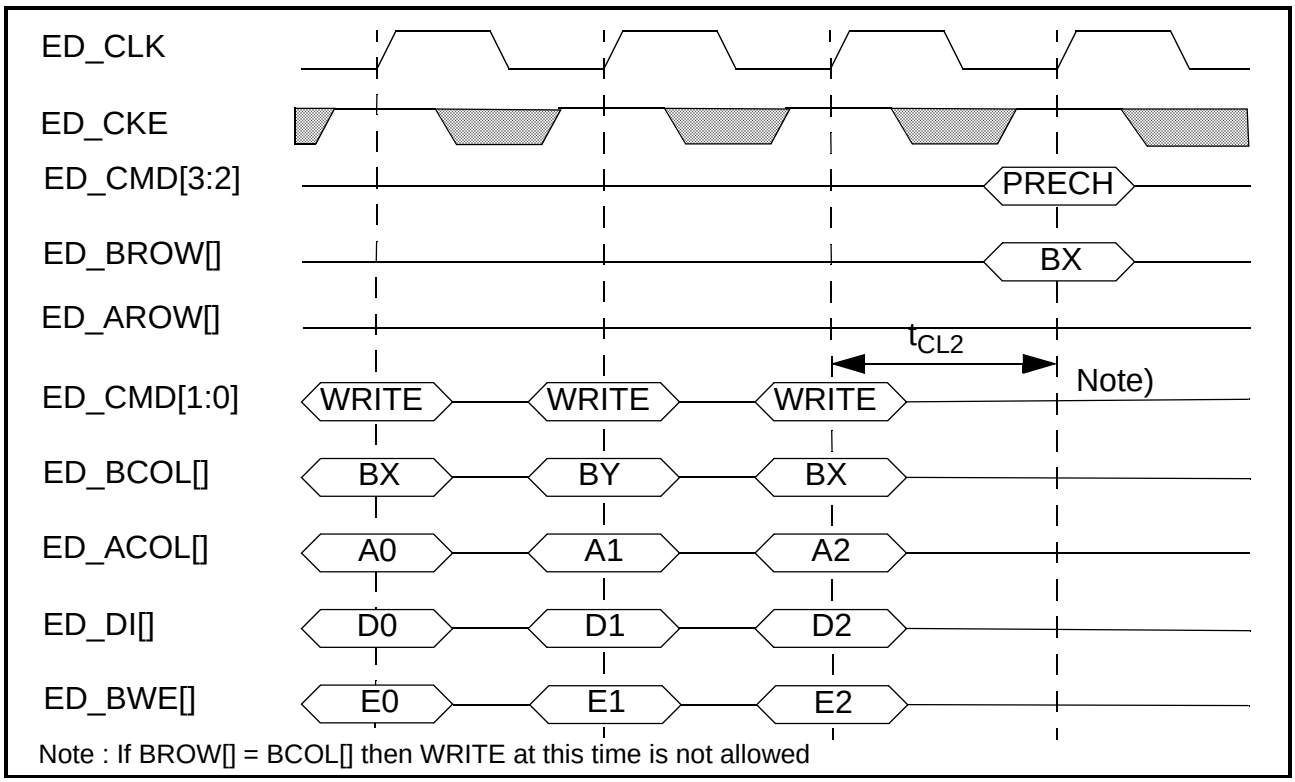


Figure 27
Write Timing, LATENCY 2

.

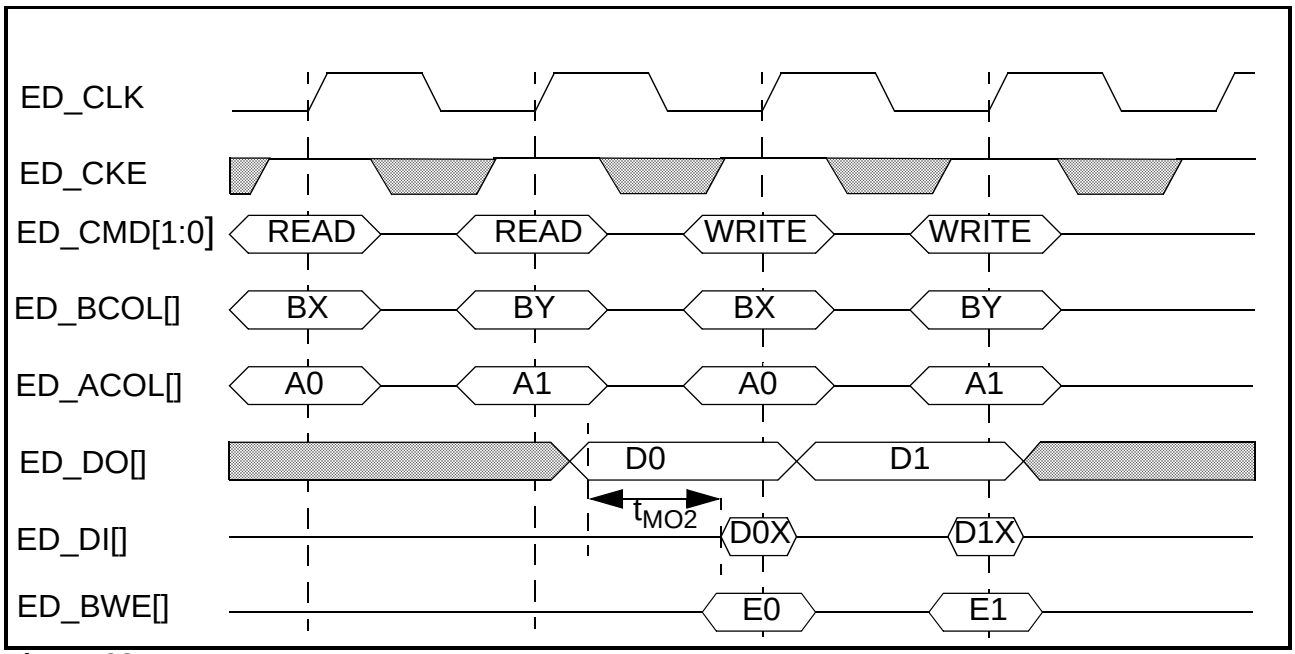
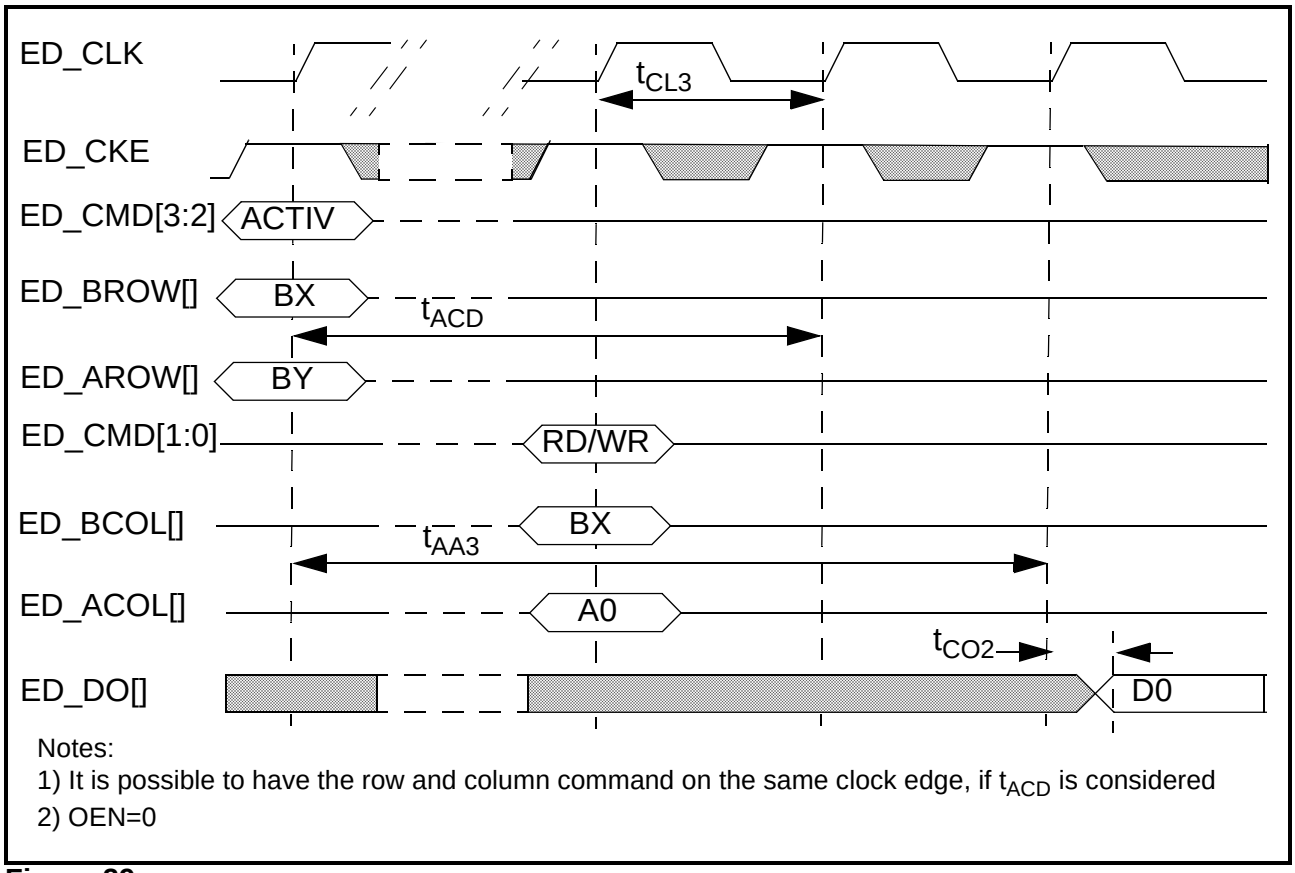


Figure 28
Read-Modify-Write Timing, LATENCY 2

6.6 Timing Diagrams for LATENCY 3



Notes:
1) It is possible to have the row and column command on the same clock edge, if t_{ACD} is considered
2) OEN=0

Figure 29
Delay between ACTIVATE and READ/WRITE with LATENCY 3

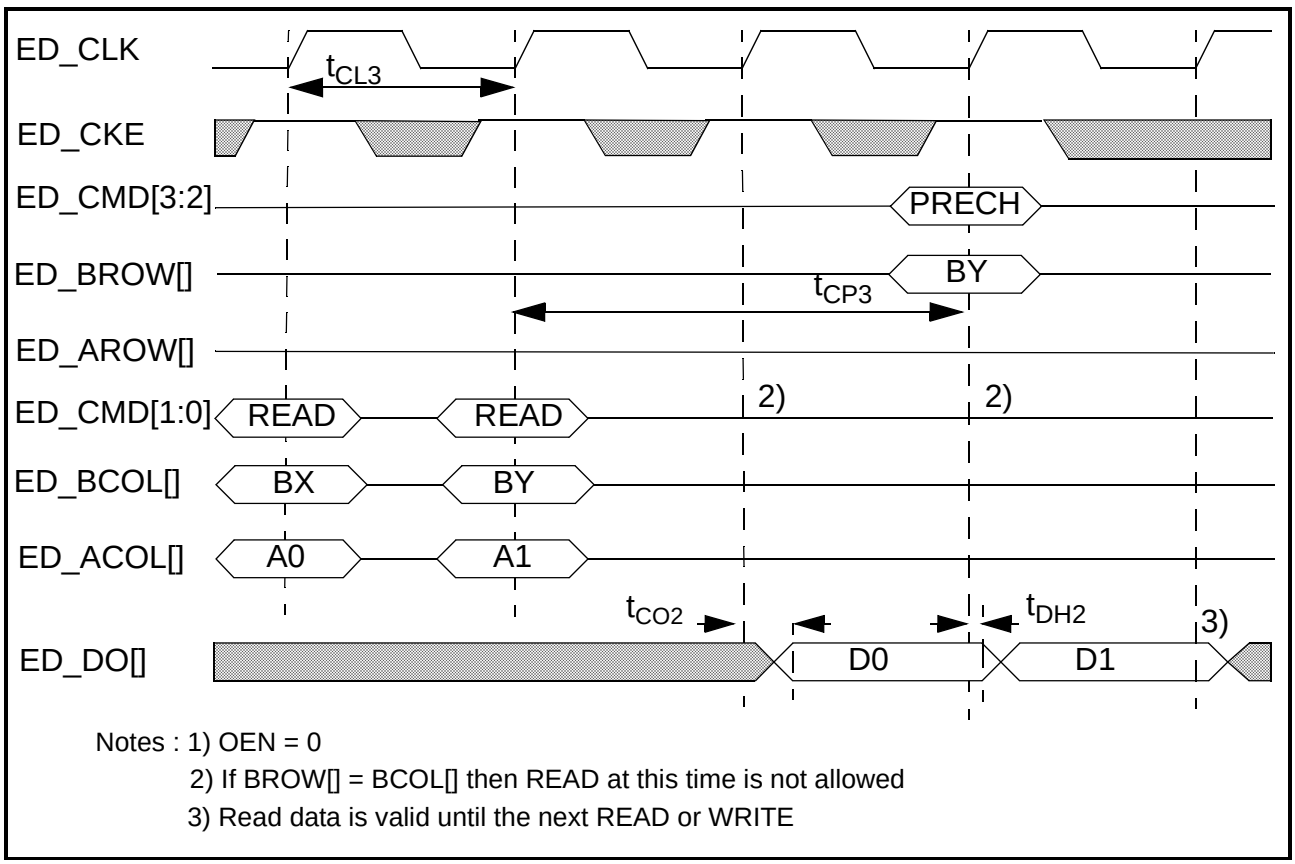


Figure 30
Read Timing, LATENCY 3

.

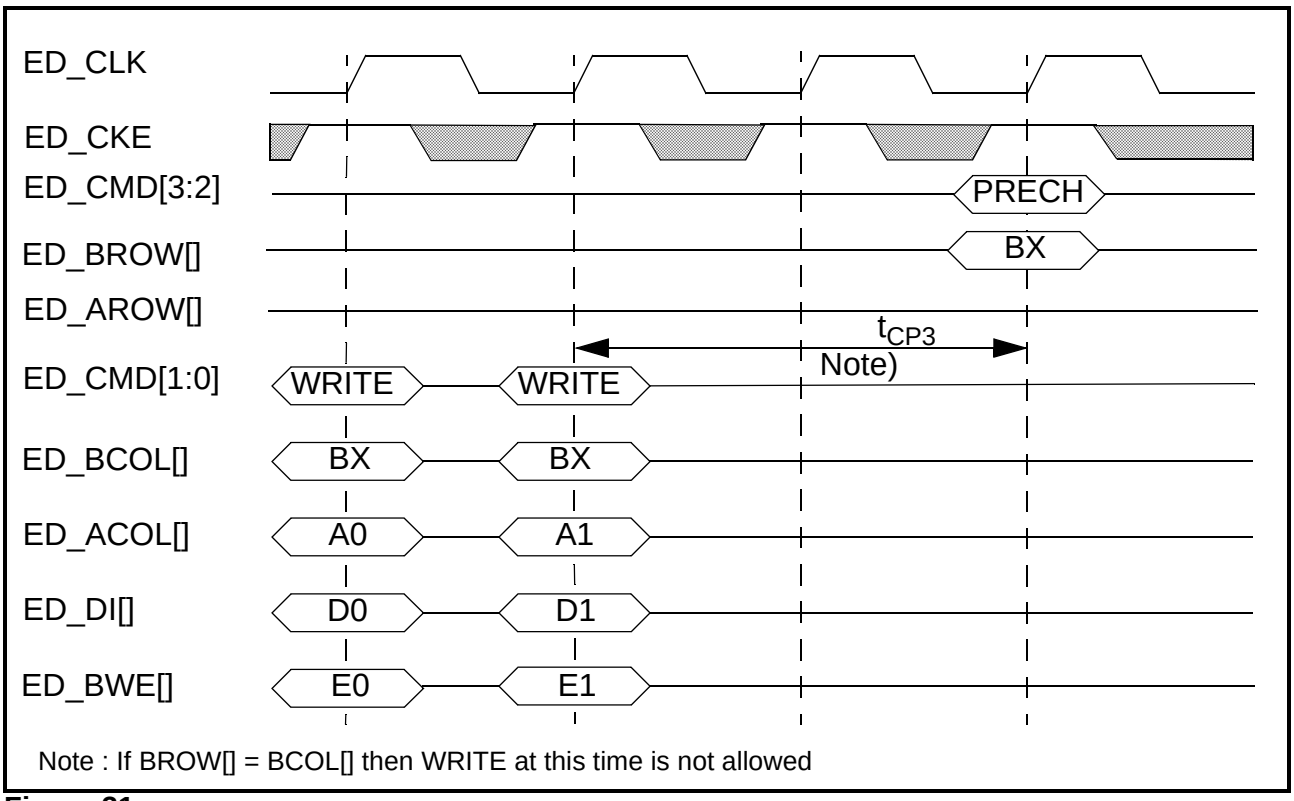


Figure 31
Write Timing, LATENCY 3

.

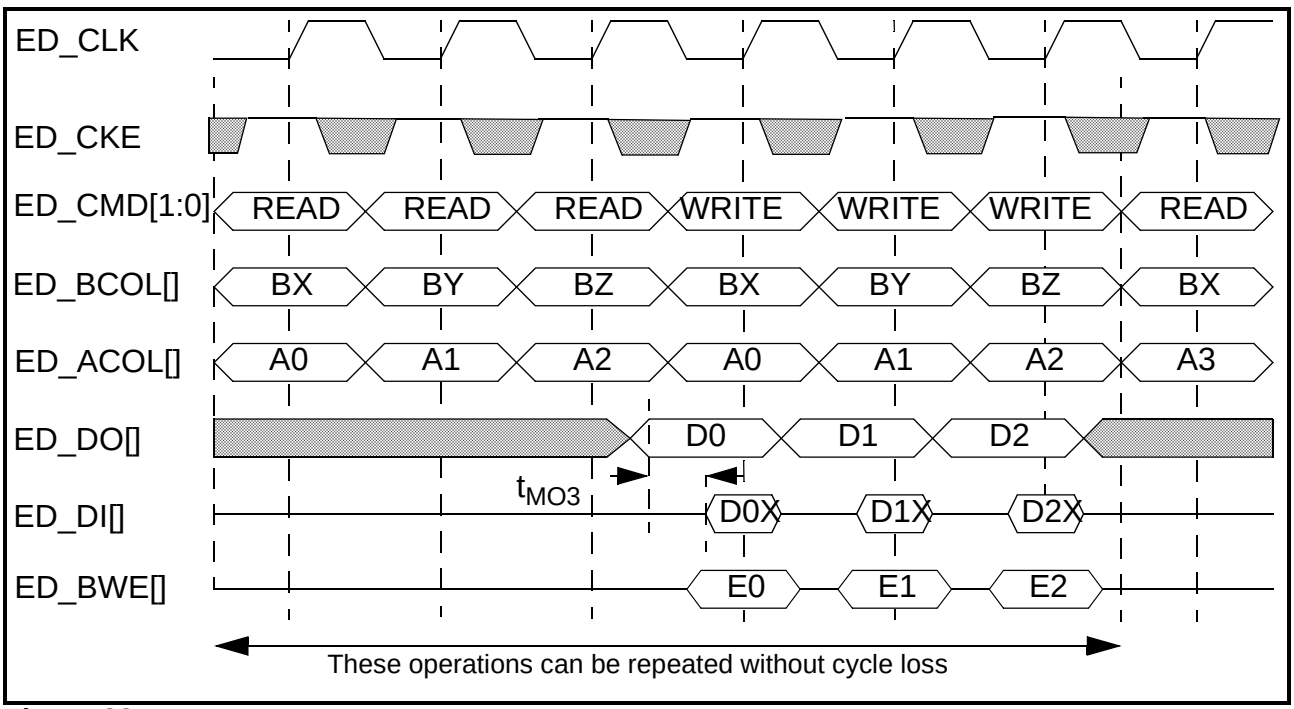


Figure 32
Read-Modify-Write Timing, LATENCY 3

7 Maximum Ratings

Stresses above those listed in the table below may cause permanent damage to the device. Exposure to conditions beyond those indicated below may affect device reliability.

This is a stress rating only and functional operation of the device under these conditions or at any other condition beyond those indicated in the operational conditions of this specification is not implied. It is not implied, that more than one of these conditions can be applied simultaneously.

Table 21
Maximum Ratings

Parameter	Limit Values		Unit
	min.	max.	
positive Supply Voltage	3.0	4.6	V
Voltage applied at any input	tbd	tbd	V

8 Environmental Requirements

8.1 Storage and Transportation

The rated (limited capability) storage and transportation temperature range prior to printed board assembly shall be - 50 to +150°C (without supply voltage)

8.2 Operating Ambient

The operating ambient temperature shall be within 0 °C to +70 °C