SAED Memory Compiler

User Guide

Version 3.0, July 2019

Content

1. Introduction

This document describes the SAED Memory Compiler and provides procedures for accessing the basic functionality.

SAED Memory Compiler is software for automatic generation of static RAM circuits (SRAMs) based on the parameters supplied by the user. It has the ability to generate a range of types of SRAMs, including a broad spectrum of output data needed for further integration of the memory into design flow.

SAED Memory Compiler is designed to be free from intellectual property restrictions and is anticipated for the use in educational purposes aimed at training highly qualified specialists in the area of microelectronics in:

- SYNOPSYS Customer Education Services
- SYNOPSYS Global Technical Services
- Universities included in SYNOPSYS University Program

For the use of some parts of SAED Memory Compiler it is assumed that European or North American bundle of SYNOPSYS EDA tools is available to trainees.

1.1. Usage Overview

Usage starts by filling the appropriate commands in configuration file which then is provided to compiler. By supplying corresponding values it is possible to change memory type, word count and width, list of generated output files, their generation options, etc. All of these parameters must be specified in a text file containing SAED Memory Compiler commands for console type of application. Such file will be called a *configuration file* from this point on.
Current version of SAED Memory Compiler has the following capabilities and limitations:

- Memories are generated for SAED 14nm, 32/28nm and 90nm technologies
- 4 types of memories are supported
  - Dual port SRAMs
  - Single port SRAMs
  - Low power dual port SRAMs
  - Low power single port SRAMs
- Number of words for memory is currently limited to be one of the following values:
  - 14nm technology:
    - 4,8,16, 32, 64, 128, 256,512,1024
  - 32/28nm and 90nm technologies:
    - 16, 32, 64, 128, 256 for dual port memories
    - 16, 32, 64, 128, for dual port memories
- Word length is limited to appropriately 1024 bits per word and 512 bits per word

Next section describes installation, usage and features of compiler.

Also a complete usage example is given in the appendix. This can be used as a starting point.

2. Installation and Setup

2.1. System Requirements

SAED Memory Compiler requires the following environment and tools on the hosting system:

- Environment
  - GNU/Linux operating system (tested on CentOS Linux 7.6)
  - Perl 5.8.8 or above
  - Bash
- Synopsys EDA tools[1]
  - Custom Compile, version 2016.06-3 or above
  - [2]Library Compiler, version 2016.12-SP3 or above
  - [3]IC Validator, version 2016.06-1
  - [4]StarRC, version 2016.06-SP3-1 or above

Notes:

1. Required to load the tools then us SAED_MC, otherwise SAED_MC will not work and will not show any Errors.
2. Required only with for do_lib option selected.
3. Required only with for do_drc or do_lvs options selected. It is highly recommended to use exact version of IC Validator as version changes can cause undesired errors related to input runset syntax incompatibility with both older and newer versions.
4. Required only with for do_cx or do_rcx options selected.

2.2. Installation and Setup

This chapter describes how SAED Memory Compiler must be installed and setup for further use.

The distribution of SAED Memory Compiler is an archived file of *.tar.gz format with the following naming convention:

```
saed_mc_<version number>.tar.gz
```

Installation process requires simply unpacking the distribution in the final destination where it is supposed to be installed. For unpacking the following Linux shell commands can be used:

```
% tar -zxvf saed_mc_<version number>.tar.gz
```

After unpacking the installation should be set up for the environment. This is done by modifying file "`setup.sh`" in the root directory of the. In this file the value of variable MC_HOME need to be modified to point to current installation directory. For example:

```
export  MC_HOME =/remote/home/user/saed_mc/
```

The contents of the file should be sourced in `bash` before running Memory Compiler.

```
% source setup.sh
```

For convenience the content of this file could be added to user bash profile.

3. Running Memory Compiler

SAED Memory Compiler is an console application which can be invoked using `saed_mc` command after performing installation steps described in Chapter 2. The compilation of required memory is performed based on configuration file provided as input to the compiler. The detailed description of commands available in configuration file can be found in Chapter 4 where an example of a configuration file is given which can be found also in the `demo/` directory of the installation.

The configuration file is mandatory and has no default file location. So in case of any failure with configuration file, the compiler will halt its execution until all noticed issues are solved.

3.1. Running compilation

Suppose the configuration file is named *SRAM4x16.config* and is placed in *demo/* directory. Then the execution of compilation will look like this:

```
% saed_mc demo/SRAM4x16.config
```

A complete usage example is given in the appendix.

3.2. Command line options

SAED Memory Compiler also provides command line options which are listed below:

Option	Definition
`--help`	Prints help message and exits
`--version`	Prints program versions and exits
`--quiet`	Print only errors and important messages
`--work <work_path>`	Selects work directory

3.3. Troubleshooting

Depending on user input and system setup several errors can be reported by compiler which should be fixed to continue compilation.

3.3.1. Configuration file

During the parsing of configuration file several errors may occur. These errors can occur because:

- The special command is misspelled. In this case a hint should be printed about the desired command name and possible values of that command
- The command value is not supported, misspelled or missing. In this case a hint should be printed out about the desired command value.

- The user have no write permission to the working directory file

3.3.2. Environment setup

SAED Memory Compiler uses different Synopsys EDA tools which should be available in PATH at the time of compilation execution. The list of tools and their version is provided in System requirements section. Missing tools will be reported and compilation will stop.

3.4. SAED Memory Compiler Output

After the successful compilation, the compiler will put the output files requested in the configuration file in the work directory specified. The number and types of files created depend on the configuration file supplied. The full list of output files/directories and their dependencies on the configuration file are presented in Table 3.1.

Table 3.1 Table of compiler deliverables

Folder Name	Content	Description
mc_work	<instance_name>.sp	SPICE netlist
	<instance_name>.gds	GDSII file
	<instance_name>.v	Verilog model
	<instance_name>.vhd	VHDL model
	<instance_name>.lef	LEF view of cell
	<instance_name>.lib <instance_name>.db	Logic Libraries
	cx/rcx<instance_name>.spf	C/RC Extraction results
	layout_results/temp	OpenAccess library containing resulted layout (can be opened using Synopsys Custom Designer)
	<instance_name>.DRC	DRC report
	<instance_name>.LVS	LVS report

4. Command Reference

Commands described in this section are used to configure the actions performed by the compiler and the number/type and other attributes of output deliverables. They must be specified in a text file called a *configuration file* which is passed to SAED Memory Compiler console application.

An example of configuration file is presented below:

```
#This is a line comment
word_count=16                 # Memory word count
word_bits=4                   # Memory word width in bits
mem_type=dual                 # Memory type, one of the supported types
do_lvs=0                      # Run LVS, Boolean value, 0 or 1
do_spice=1                    # Generate SPICE netlist, 0 or 1
do_layout=0                   # Generate layout, 0 or 1
do_gds=0                      # Generate GDSII, 0 or 1
do_drc=0                      # Run DRC, 0 or 1
do_logic=0                    # Generate VHDL and Verilog files, 0 or 1
set_ver_tool=icv              # Select ICV for verification
```

4.1. Configuration File Commands

SAED Memory Compiler configuration file commands and their descriptions are listed below. There are the following commands available.

mem_type
word_count
word_bits
instance_name
do_spice
do_layout[1]
do_gds
do_lvs
do_drc
do_cx
do_rcx
do_logic
do_lef
do_lib_nldm
do_mw
do_lib_ccs[1]
work_dir

Notes:

1. do_layout and do_lib_ccs features don’t available for 14nm SAED_MC.
2. do_mw feature available for anly 14nm SAED_MC

Next sections present requirements for the listed commands.

4.1.1. mem_type

This command is required and has no default value.

SYNTAX

mem_type = single | dual | singlelp | duallp

COMMAND INFORMATION

Value Type	Default Value	Requirement
String	N/A	One of the possible values*

The values must be one of the memory types supported by compiler.

DESCRIPTION

Generated SRAM's type.

EXAMPLE

Argument	Description
dual	Dual Port SRAM
single	Single Port SRAM

4.1.2. word_count

This command is required and has no default value.

SYNTAX

word_count = 4 | 8* | 16 | 32 | 64 | 128 | 256* | 512* | 1024**

COMMAND INFORMATION

Value Type	Default Value	Requirement
integer	N/A	One of the allowable values * sizes are currently only available for 14nm memories

DESCRIPTION

The numbers words in SRAM.

EXAMPLE

Argument	Description
128	SRAM will have 128 words

4.1.3. word_bits

This command is required and has no default value.

SYNTAX

words_bits = integer

COMMAND INFORMATION

Value Type	Default Value	Requirement
integer	N/A	Integer value in the range 4..512

DESCRIPTION

It shows numbers of bits per word. Its minimal value is 4, maximal value is 512. The word bits don’t have max limitation for 14nm SAED_MC.

EXAMPLE

Argument	Description
128	-

4.1.4. instance_name

SYNTAX

instance_name = (string_identifier)

COMMAND INFORMATION

Value Type	Default Value	Requirement
String	SRAM<word_count>x<word_bits><mem_type>	Alphanumeric identifier with no spaces allowed

DESCRIPTION

Generated SRAM's name.

EXAMPLE

Argument	Description
SRAM16x4dual	Dual Port SRAM
SRAM64x8single	Single Port SRAM

4.1.5. do_spice

SYNTAX

do_spice = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	1	0 or 1

DESCRIPTION

Enables SPICE netlist generation. This parameter takes two possible values.

EXAMPLE

Argument	Description
1	Generate spice netlist
0	Do not generate spice netlist

4.1.6. do_cx

SYNTAX

do_cx = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables generation of C extracted netlist. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate CX extracted netlist
0	Do not generate CX extracted netlist

4.1.7. do_rcx

SYNTAX

do_rcx = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables generation of RC extracted netlist. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate RCX extracted netlist
0	Do not generate RCX extracted netlist

4.1.8. do_layout

SYNTAX

do_layout = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	1	0 or 1

DESCRIPTION

Enables layout files generation. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate
0	Do not generate

4.1.9. do_gds

SYNTAX

do_gds = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	1	0 or 1

DESCRIPTION

Enables GDS files generation. This parameter takes two possible values 1 or 0 which stand for yes/no. This option should be enabled for do_drc/do_lvs options.

EXAMPLE

Argument	Description
1	Generate
0	Do not generate

4.1.10. do_lvs

SYNTAX

do_lvs = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	1	0 or 1

DESCRIPTION

Enables LVS verification of layout. This parameter takes two possible values 1 or 0 which stand for yes/no. This option requires do_gds option to be set to 1.

EXAMPLE

Argument	Description
1	Do LVS
0	Do not LVS

4.1.11. do_drc

SYNTAX

do_drc = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables DRC verification of layout. This parameter takes two possible values 1 or 0 which stand for yes/no. This option requires do_gds option to be set to 1.

EXAMPLE

Argument	Description
1	Do DRC
0	Do not DRC

4.1.12. do_logic

SYNTAX

do_logic = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables Verilog/VHDL files generation. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate
0	Do not generate

4.1.13. do_lef

SYNTAX

do_lef= 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables LEF file generation. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate
0	Do not generate

4.1.14. do_lib_nldm

SYNTAX

do_lib_nldm = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables .lib/.db file generation. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate .lib/.db files
0	Do not generate .lib/.db files

4.1.15. do_mw

SYNTAX

do_lib_nldm = 1 | 0

COMMAND INFORMATION

Value Type	Default Value	Requirement
Boolean	0	0 or 1

DESCRIPTION

Enables MWdb generation. This parameter takes two possible values 1 or 0 which stand for yes/no.

EXAMPLE

Argument	Description
1	Generate milkyway database
0	Do not generate milkyway database

4.1.16. set_ver_tool

SYNTAX

set_ver_tool = icv | hercules

COMMAND INFORMATION

Value Type	Default Value	Requirement
String	icv	*icv* or *hercules*

DESCRIPTION

Choose physical verification tool. This parameter takes two possible values icv for IC Validator and hercules for Hercules.

EXAMPLE

Argument	Description
icv	Use IC Validator
hercules	Use Hercules

4.1.17. work_dir

SYNTAX

work_dir = string

COMMAND INFORMATION

Value Type	Default Value	Requirement
String	./mc_work	Valid directory name

DESCRIPTION

Output directory of all results.

EXAMPLE

Argument	Description
/mc_work	Output results directory

5. Supported Memory Types

This chapter contains the description of all types of memories supported by the compiler.

This chapter presents the following sections:

- Dual Port SRAM
- Single Port SRAM
- Low Power Dual Port SRAM
- Low Power Single Port SRAM

SAED Memory Compiler supports variety of static memory types, the full list of which is presented in the table below:

Table 5.1. SAED Memory Compiler supported memory types and notations

Configuration name (mem_type)	Memory Type	Symbol
dual	Single Port SRAM	SRAMnxm_1rw
single	Dual Port SRAM	SRAMnxm
duallp*	Single Port Low Power SRAM	SRAMLPnxm_1rw
singlelp*	Dual Port Low Power SRAM	SRAMLPnxm

** Not fully implemented in current version*

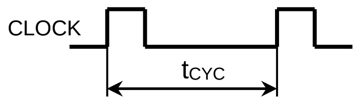
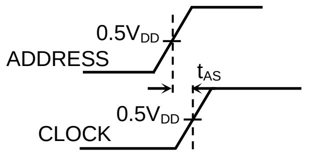
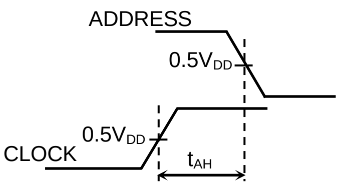
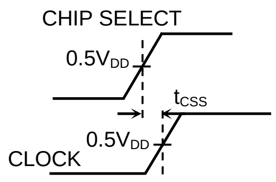
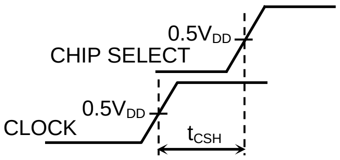
The used symbols of SRAMnxm states are shown in Table 5.2 The synchronous dual-port SRAMnxm have two ports (Primary and Dual) for the same memory location. Both ports can be independently accessed for read and write operations. The basic pins of dual-port SRAMnxm are shown in Figure 5.1 and their descriptions are shown in Table 5.3

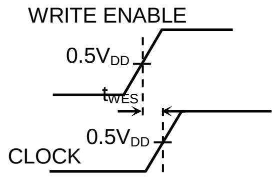
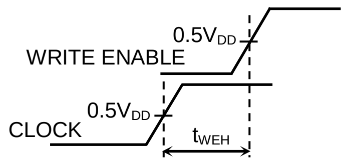
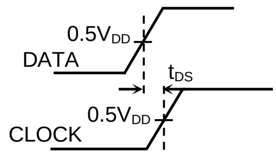
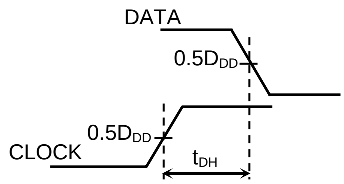
Table 5.2. Symbols of SRAMnxm states

Symbol	State
L (“0”)	LOW Logic Level
H (“1”)	HIGH Logic Level
Z	High-impedance State
LH (“0”→”1”)	LOW to HIGH Transition
HL (“1”→”0”)	HIGH to LOW Transition
X	Either HIGH or LOW Logic Level

Timing parameters and their definitions of SRAM memories are shown in Table 5.3

Table 5.3.. Timing Parameters of SRAM memories

No	Parameter	Unit	Symbol	Figure	Definition
Timing parameters					
1	Cycle time	ns	t_{CYC}		The amount of time between two sequential active edges of clock signal
2	Access time	ns	t_A	None	The amount of time between applying Write/Read Enable signal and obtaining Access to Data in Memory
3	Address setup	ns	t_{AS}		The minimum amount of time in which the address to a SRAMnxm must be stable before the active edge of the clock occurs
4	Address hold	ns	t_{AH}		The minimum amount of time in which the address to a SRAMnxm must remain stable after the active edge of the clock has occurred
5	Chip select setup	ns	t_{CSS}		The minimum amount of time in which the Chip select signal to a SRAMnxm must be stable before the active edge of the clock occurs
6	Chip select hold	ns	t_{CSH}		The minimum amount of time in which the Chip select signal to a SRAMnxm must remain stable after the active edge of the clock has occurred

No	Parameter	Unit	Symbol	Figure	Definition
7	Write enable setup	ns	t_{WES}		The minimum amount of time in which the Write enable signal to a SRAMnxm must be stable before the active edge of the clock occurs
8	Write enable hold	ns	t_{WEH}		The minimum amount of time in which the Write enable signal to a SRAMnxm must remain stable after the active edge of the clock has occurred
9	Data setup	ns	t_{DS}		The minimum amount of time in which the input data to a SRAMnxm must be stable before the active edge of the clock occurs
10	Data hold	ns	t_{DH}		The minimum amount of time in which the input data to a SRAMnxm must remain stable after the active edge of the clock has occurred
11	Output Z state entry time	ns	t_{OZ}	None	The amount of time that takes the outputs to change to Z state after output enable signal is applied
12	Output Z state exit time	ns	t_{ZO}	None	The amount of time that takes the outputs to exit from Z state after output enable signal is applied

5.1. Dual port SRAM

5.1.1. Basil pins

The general block-diagram of dual-port SRAMnxm is shown in Figure 5.1.
Dual port SRAMnxm basic operations are shown in Table 5.4.
Dual port SRAMnxm access is synchronous and triggered by the rising edge of the clock signals (CE1, CE2). Read/Write addresses (A1, A2), Input data (I1, I2), Write enable

signals (WEB1, WEB2), and Chip select signals (CSB1, CSB2) are latched by the rising edge of the clocks (CE1, CE2).

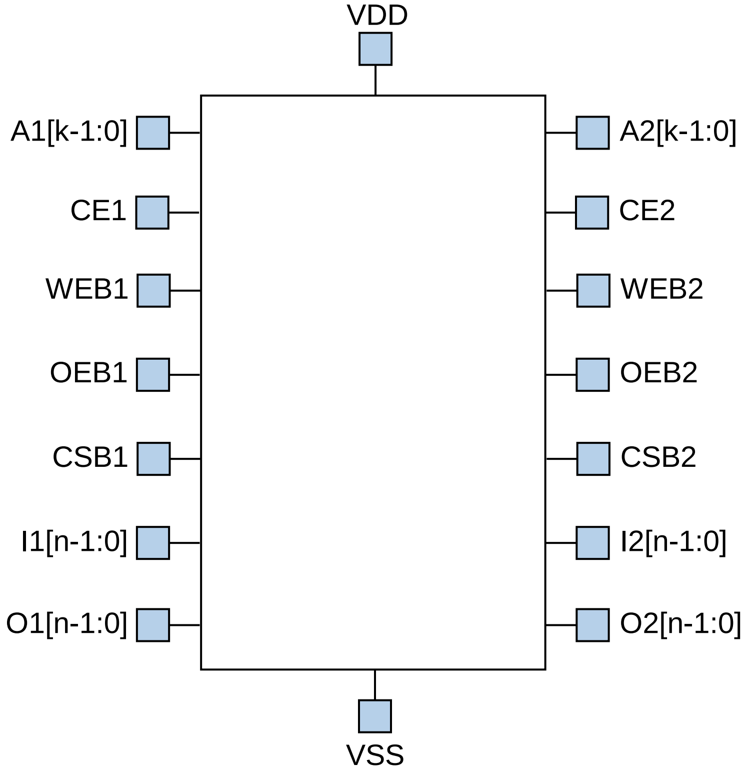


Figure 5.1. Dual port SRAMnxm Basic Pins

Table 5.4. Dual port SRAMnxm Pin Definition

Pin Symbol	Width (bits)	Type	Name and Function
A1	k	Input	Primary Read/Write Address
CE1	1	Input	Primary Positive-Edge Clock
WEB1	1	Input	Primary Write Enable, Active Low
OEB1	1	Input	Primary Output Enable, Active Low
CSB1	1	Input	Primary Chip Select, Active Low
I1	n	Input	Primary Input data bus
O1	n	Output	Primary Output data bus
A2	k	Input	Dual Read/Write Address
CE2	1	Input	Dual Positive-Edge Clock
WEB2	1	Input	Dual Write Enable, Active Low
OEB2	1	Input	Dual Output Enable, Active Low
CSB2	1	Input	Dual Chip Select, Active Low
I2	n	Input	Dual Input data bus
O2	n	Output	Dual Output data bus
VDD	Power supply		
VSS	Power ground		

5.1.2. Description

The value of Chip Select signal is low (CS1/CS2=0) for read/write operation. The SRAMnxm enter read mode when CS1/CS2=0 and WEB1/WEB2=1. During read operations, data read from the memory location D(A1[k-1:0])/D(A2[k-1:0]) specified on the address bus I1[n-1:0]/I2[n-1:0] and appear on the data output bus O1[n-1:0]/O2[n-1:0].
Dual port SRAMnxm enter write mode when CSB1/CSB2=0 and WEB1/WEB2=0. During write mode, data on the data input bus I1[n-1:0]/I2[n-1:0] is writing into the memory location D(A1[k-1:0])/D(A2[k-1:0]) specified on the address bus I1[n-1:0]/I2[n-2:0].

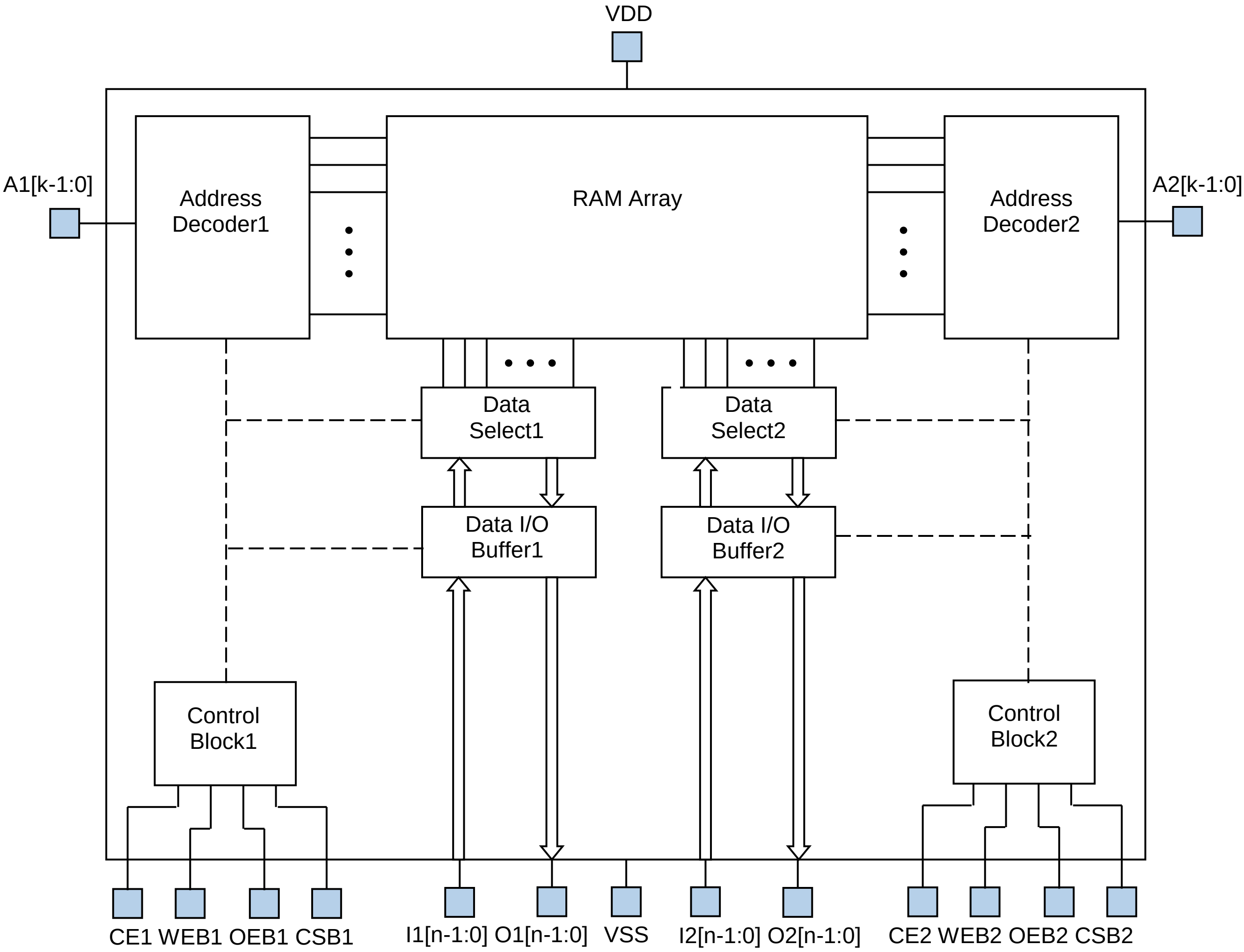


Figure 5.2. Dual port SRAMnxm block diagram

If OEB1/OEB2=1, data on the output bus O1[n-1:0]/O2[n-1:0] placed in Z state. At that time read/write operation continue. When OEB1/OEB2=0, the data appear on the output bus O1[n-a:0]/O2[n-1:0].
Power dissipation is minimized using static circuit implementations. A standby mode is provided to further reduce power dissipation during periods of non-operation (CCB1/CSB2=1). While in standby mode, address and data inputs are disabled; data stored in the memory D(A1[k-1:0])/D(A2[k-1:0]) is retained, but the memory cannot be accessed for reads or writes.
Address contention will occur when both ports simultaneously access the same address. In this case, both ports will read the same data.

Table 5.5. Dual port SRAMnxm Basic Operations

Pins						Data in Memory	Access to Memory	Operation
A1[k-1:0]	WEB1	OEB1	CSB1	I1[n-1:0]	O1[n-1:0] (t+1)	D(A1[k-1:0]) (t+1)		
X	X	0	1	Disabled	O1[n-1:0] (t)	D(A1[k-1:0]) (t)	No	Standby
		1			Z			
X	0	0	0	Enabled	I1[n-1:0]	I1[n-1:0]	Yes	Write
		1			Z			
X	1	0	0	X	D(A1[k-1:0]) (t)	D(A1[k-1:0]) (t)	No	Read
		1			Z			
A2[k-1:0]	WEB2	OEB2	CSB2	I2[n-1:0]	O2[n-1:0] (t+1)	D(A2[k-1:0]) (t+1)		
X	X	0	1	Disabled	O2[n-1:0] (t)	D(A2[k-1:0]) (t)	No	Standby
		1			Z			
X	0	0	0	Enabled	I2[n-1:0]	I2[n-1:0]	Yes	Write
		1			Z			
X	1	0	0	X	D(A2[k-1:0]) (t)	D(A2[k-1:0]) (t)	No	Read
		1			Z			

Note: O1[n-1:0] (t) is the value of Primary Port Output bus in the previous moment of time, and O1[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A1[k-1:0]) (t) is the data in the RAM location specified on the address bus A1[k-1:0] in the previous moment of time, and D(A1[k-1:0]) (t+1) in the next moment of time.
O2[n-1:0] (t) is the value of Dual Port Output bus in the previous moment of time, and O2[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A2[k-1:0]) (t) is the data in the RAM location specified on the address bus A2[k-1:0] in the previous moment of time, and D(A2[k-1:0]) (t+1) in the next moment of time.

5.1.3. Timing Waveforms

SRAMnxm will function according to the block-diagrams shown in Figures 5.3. – 5.6.

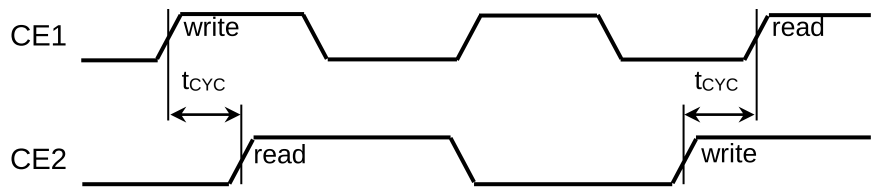


Figure 5.3. Dual port SRAMnxm Write-Read Clock Timing Waveforms

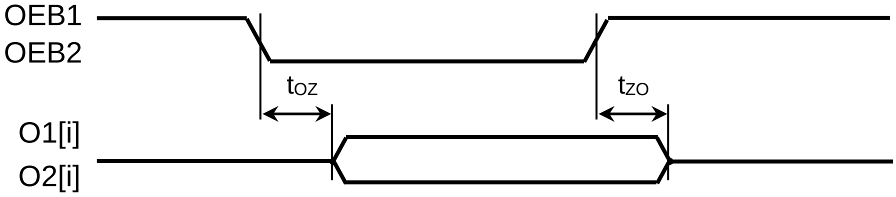


Figure 5.4. Dual port SRAMnxm Output-Enable Timing Waveforms

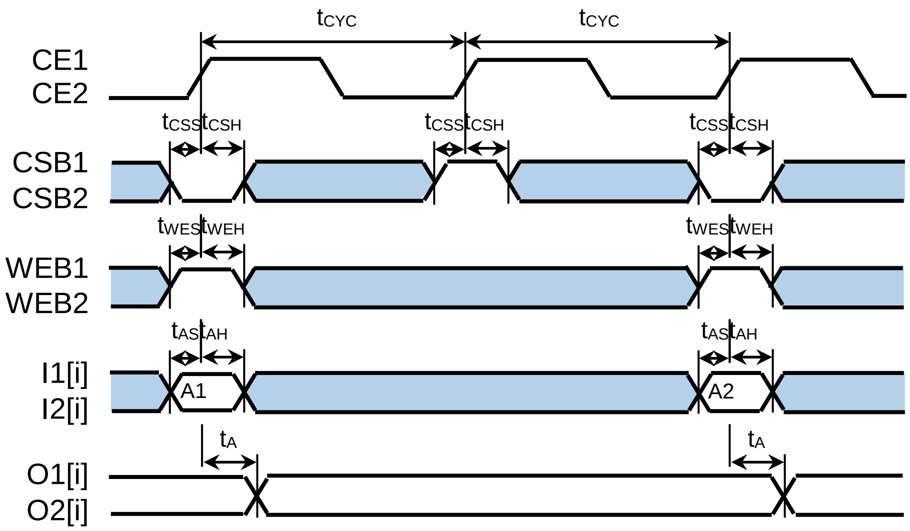


Figure 5.5. Dual port SRAMnxm Read-Cycle Timing Waveforms

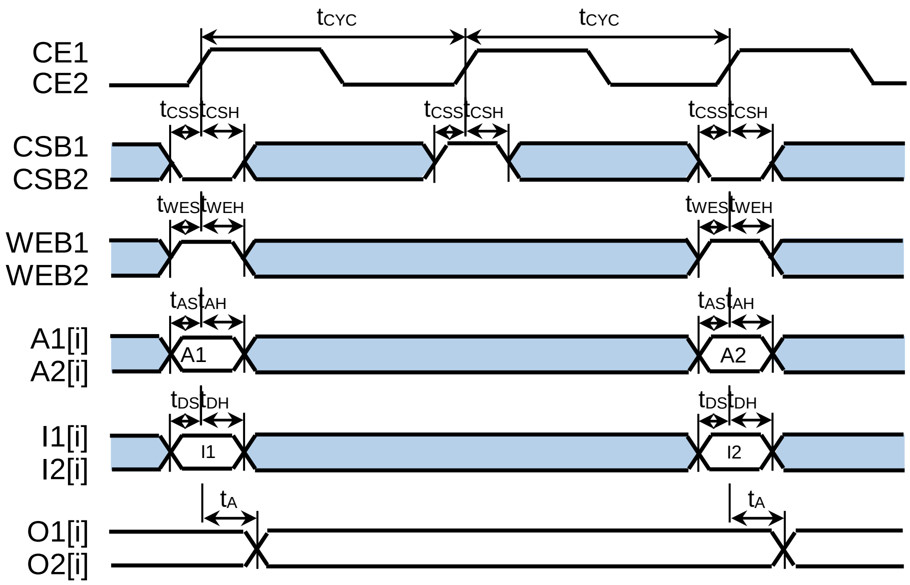


Figure 5.6. Dual port SRAMnxm Write-Cycle Timing Waveforms

5.2. Single port SRAM

5.2.1. Basic pins

Basic pins of single port SRAMnxm_1rw are shown in Figure 5.7, and their descriptions are in Table 5.6.

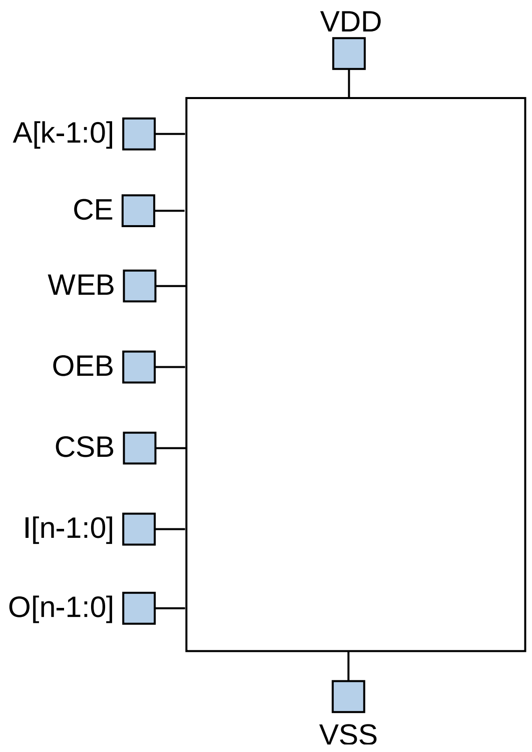


Figure 5.7. Single port SRAMnxm_1rw Basic Pins

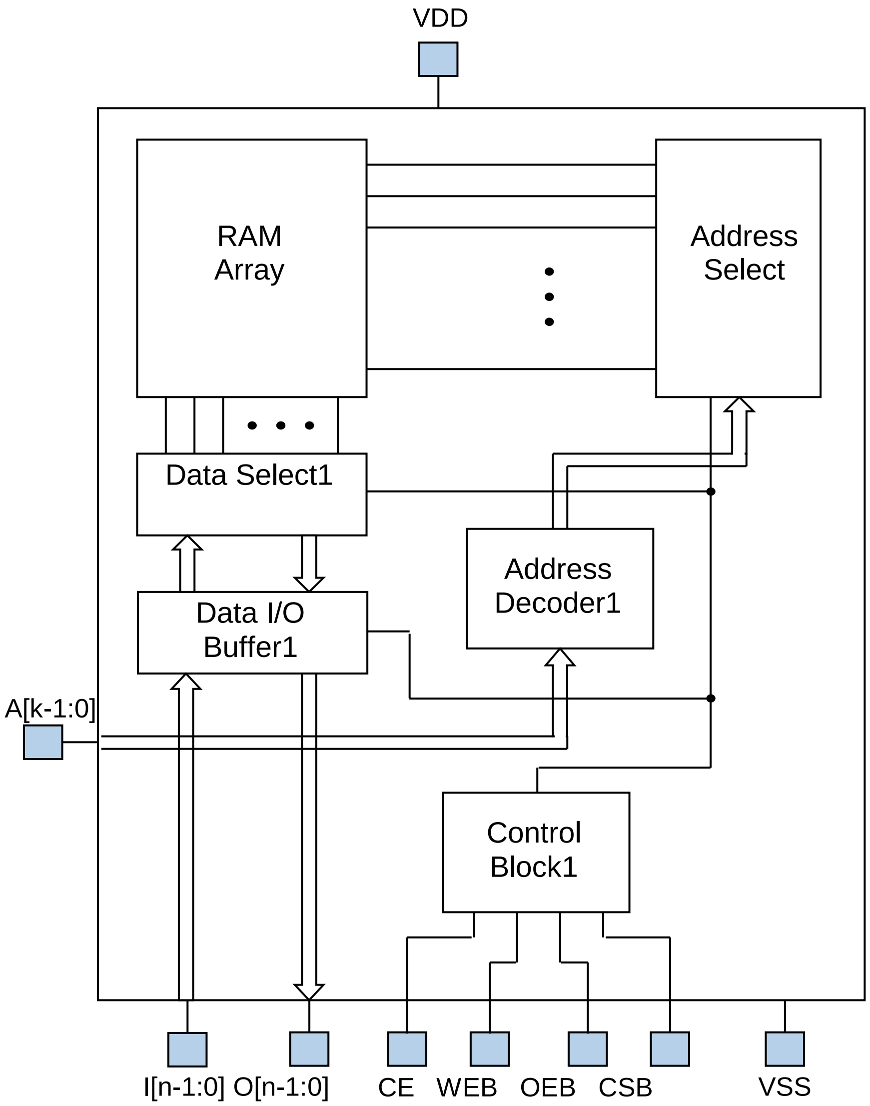
Table 5.6. Single port SRAMnxm_1rw Pin Definition

Pin Symbol	Width(bits)	Type	Name and Function
A	k	Input	Primary Read/Write Address
CE	1	Input	Primary Positive-Edge Clock
WEB	1	Input	Primary Write Enable, Active Low
OEB	1	Input	Primary Output Enable, Active Low
CSB	1	Input	Primary Chip Select, Active Low
I	n	Input	Primary Input data bus
O	n	Output	Primary Output data bus
VDD	Power supply		
VSS	Power ground		

5.2.2. Description

The general block-diagram of single port SRAMnxm_1rw is shown in Figure 5.8. and its basic operations are shown in Table 5.7.

Single port SRAMnxm_1rw access is synchronous and triggered by the rising edge of the clock signals (CE1). Read/Write addresses (A1), Input data (I1), Write enable signals



(WEB1), and Chip select signals (CSB1) are latched by the rising edge of the clocks (CE).

Figure 5.8. Single port SRAMnxm_1rw block diagram

The value of Chip Select signal is low (CS=0) for read/write operation. The single port SRAMnxm_1rw enter read mode when CS=0 and WEB=1. During read operations, data read from the memory location D(A[k-1:0]) specified on the address bus I[n-1:0] and appear on the data output bus O[n-1:0].
Single port SRAMnxm_1rw enter write mode when CSB=0 and WEB=0. During write mode, data on the data input bus I[n-1:0] is writing into the memory location D(A[k-1:0]) specified on the address bus I[n-1:0].
If OEB=1, data on the output bus O[n-1:0] placed in Z state. At that time read/write operation continue. When OEB=0, the data appear on the output bus O[n-a:0].
Power dissipation is minimized using static circuit implementations. A standby mode is provided to further reduce power dissipation during periods of non-operation (CCB=1). While in standby mode, address and data inputs are disabled; data stored in the memory D(A[k-1:0]) is retained, but the memory cannot be accessed for reads or writes.

Table 5.7. Single port SRAMnxm_1rw Basic Operations

Pins						Data in Memory	Access to Memory	Operation
A[k-1:0]	WEB	OEB	CSB	I[n-1:0]	O[n-1:0] (t+1)	D(A[k-1:0]) (t+1)		
X	X	0	1	Disabled	O[n-1:0] (t)	D(A[k-1:0]) (t)	No	Standby
		1			Z			
X	0	0	0	Enabled	I[n-1:0]	I[n-1:0]	Yes	Write
		1			Z			
X	1	0	0	X	D(A[k-1:0]) (t)	D(A[k-1:0]) (t)	No	Read
		1			Z			

Note: O[n-1:0] (t) is the value of Primary Port Output bus in the previous moment of time, and O[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A[k-1:0]) (t) is the data in the RAM location specified on the address bus A[k-1:0] in the previous moment of time, and D(A[k-1:0]) (t+1) in the next moment of time.
O[n-1:0] (t) is the value of Dual Port Output bus in the previous moment of time, and O[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A[k-1:0]) (t) is the data in the RAM location specified on the address bus A[k-1:0] in the previous moment of time, and D(A[k-1:0]) (t+1) in the next moment of time.

Address contention will occur when both ports simultaneously access the same address. In this case, both ports will read the same data.
The list of expressions to be used in this section and their meanings is presented in Table 5.7.

5.2.3. Timing Waveforms

Single port SRAMnxm_1rw functions according to the block-diagrams shown in Figures 5.9. – 5.11.

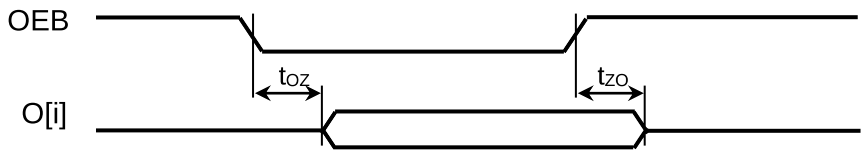


Figure 5.9. Single port SRAMnxm_1rw Output-Enable Timing Waveforms

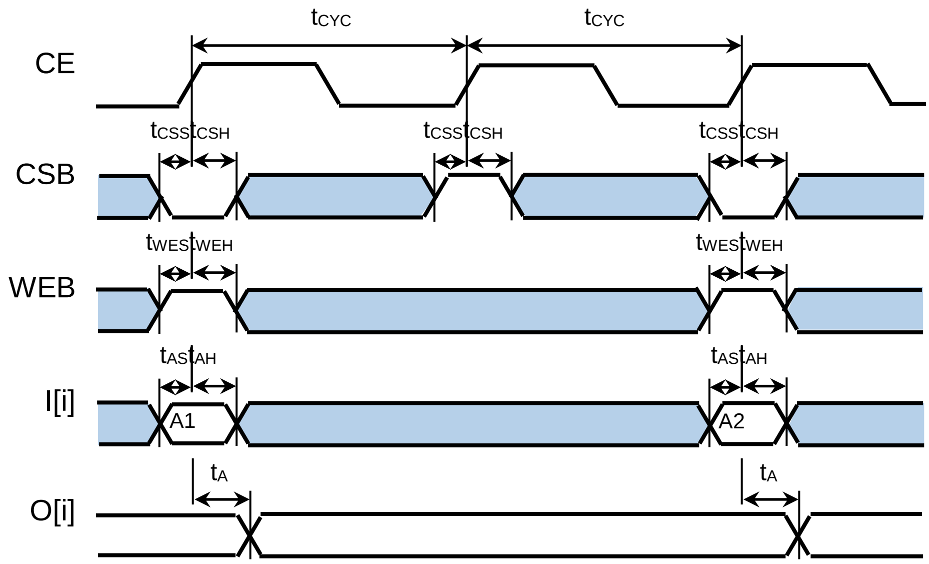


Figure 5.10. Single port SRAMnxm_1rw Read-Cycle Timing Waveforms

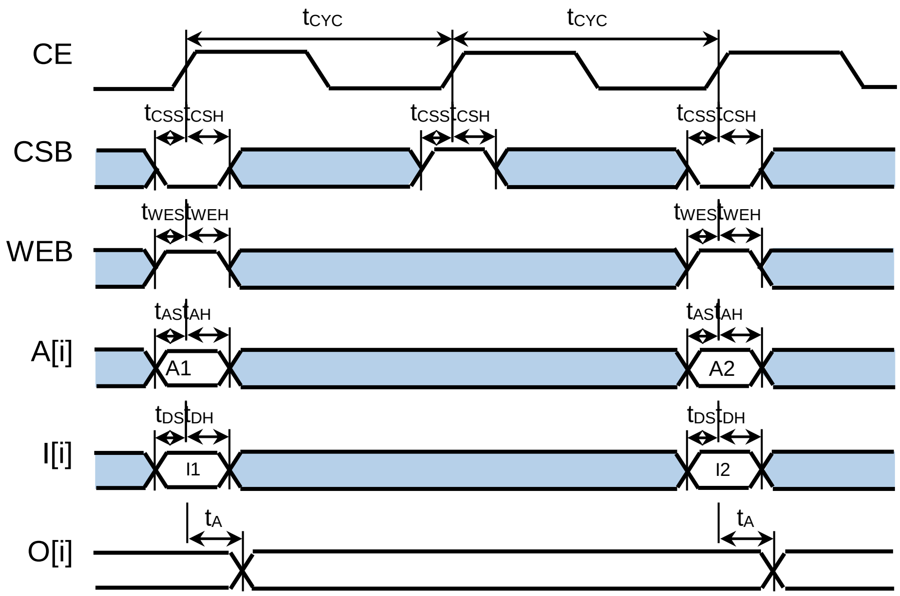


Figure 5.11. Single port SRAMnxm_1rw Write-Cycle Timing Waveforms

5.3. Dual port Low power SRAM

5.3.1. Basic Pins

The Basic Pins of dual port low power SRAMLPnxm are shown in Figure 5.12. and its descriptions are shown in Table 5.8.

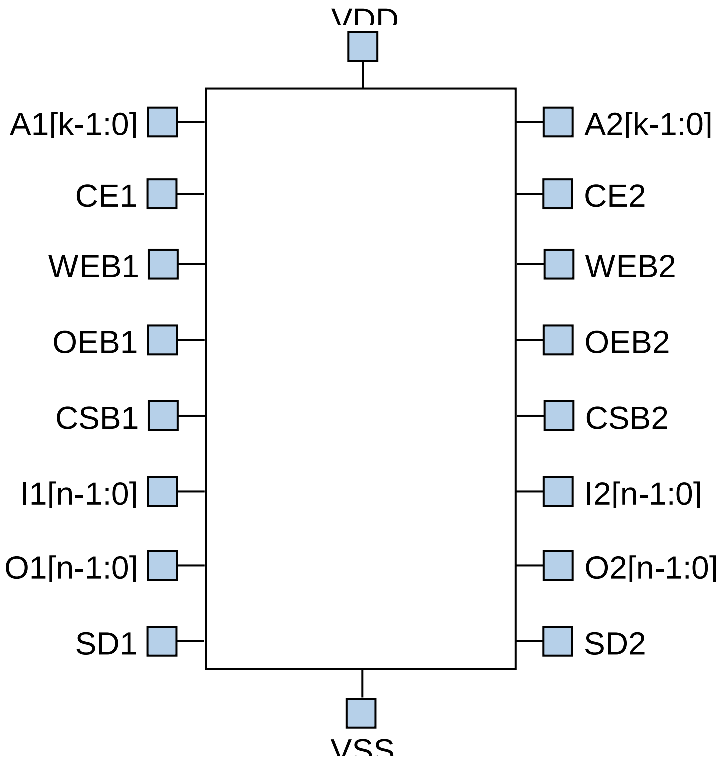


Figure 5.12. Dual port low power SRAMLPnxm Basic Pins

Table 5.8. Dual port low power SRAMLPnxm Pin Definition

Pin Symbol	Width (bits)	Type	Name and Function
A1	K	Input	Primary Read/Write Address
CE1	1	Input	Primary Positive-Edge Clock
WEB1	1	Input	Primary Write Enable, Active Low
OEB1	1	Input	Primary Output Enable, Active Low
CSB1	1	Input	Primary Chip Select, Active Low
SD1	1	Input	Primary Supply down, Active High
I1	N	Input	Primary Input data bus
O1	N	Output	Primary Output data bus
A2	k	Input	Dual Read/Write Address
CE2	1	Input	Dual Positive-Edge Clock
WEB2	1	Input	Dual Write Enable, Active Low
OEB2	1	Input	Dual Output Enable, Active Low
CSB2	1	Input	Dual Chip Select, Active Low
SD2	1	Input	Dual Supply down, Active High
I2	n	Input	Dual Input data bus
O2	n	Output	Dual Output data bus
VDD	Power supply		
VSS	Power ground		

5.3.2. Description

The general block-diagram of SRAMnxm is shown in Figure 5.13.

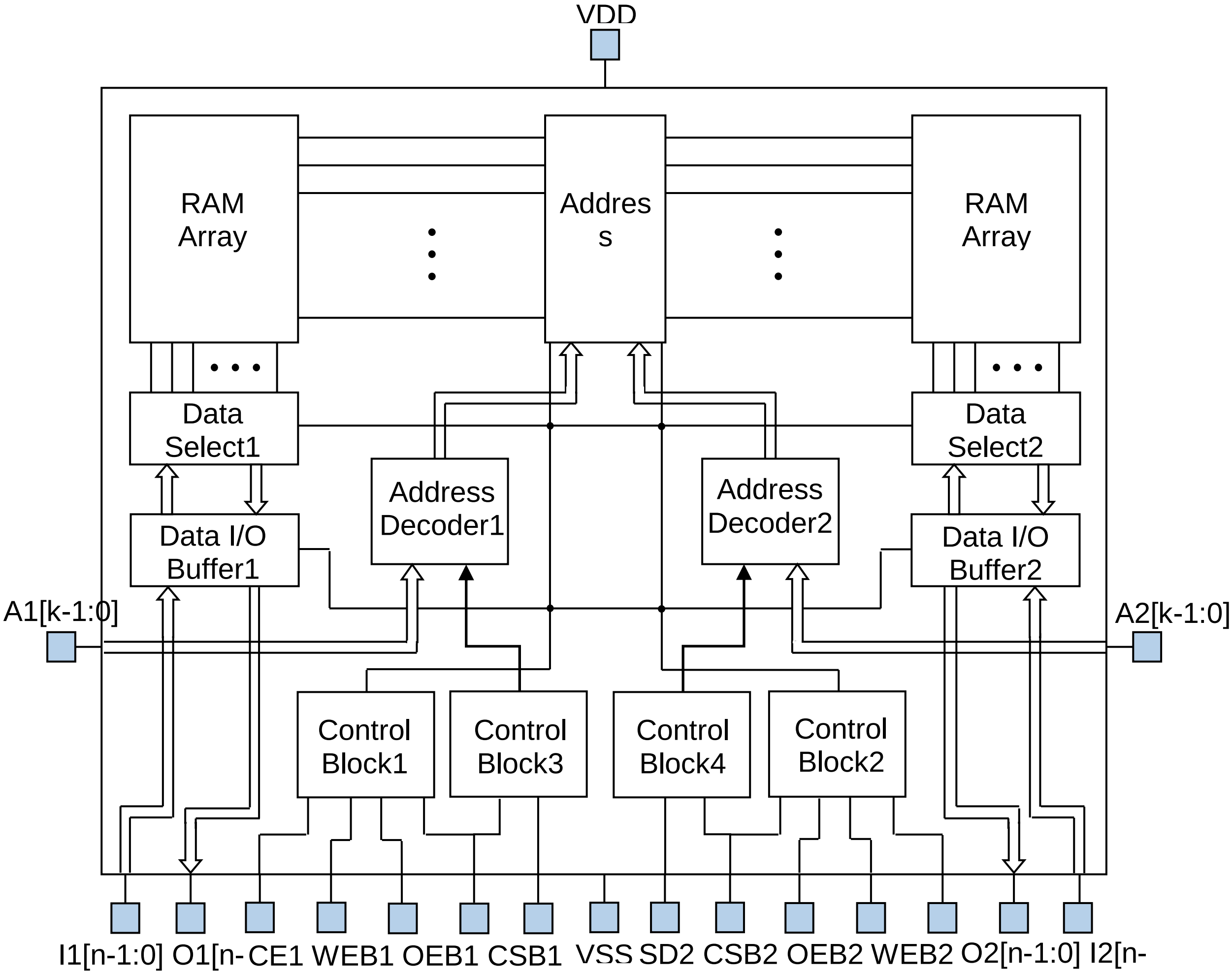


Figure 5.13. Dual port low power SRAMLPnxm block diagram

Dual port low power SRAMLPnxm Basic Operations is shown in Table 5.9.
Dual port low power SRAMLPnxm access is synchronous and triggered by the rising edge of the clock signals (CE1, CE2). Read/Write addresses (A1, A2), Input data (I1, I2), Write enable (WEB1, WEB2), Supply down signals (SD1, SD2), and Chip select signals (CSB1, CSB2) are latched by the rising edge of the clocks (CE1, CE2).

Table 5.9. Dual port low power SRAMLPnxm Basic Operations

	Pins						Data in Memory	Access to Memory	Operation
A1[k-1:0]	WEB1	OEB1	CSB1	SD1	I1[n-1:0]	O1[n-1:0] (t+1)	D(A1[k-1:0]) (t+1)		
X	X	0	1	x	Disabled	O1[n-1:0] (t)	D(A1[k-1:0]) (t)	No	Standby
		1				Z			
X	0	0	0	1	Enabled	I1[n-1:0]	I1[n-1:0]	Yes	Low Power Write
		1				Z			
X	0	0	0	0	Enabled	I1[n-1:0]	I1[n-1:0]	Yes	Normal Write
		1				Z			
X	1	0	0	0	X	D(A1[k-1:0]) (t)	D(A1[k-1:0]) (t)	No	Read
		1				Z			
A2[k-1:0]	WEB2	OEB2	CSB2	SD2	I2[n-1:0]	O2[n-1:0] (t+1)	D(A2[k-1:0]) (t+1)		
X	X	0	1	x	Disabled	O2[n-1:0] (t)	D(A2[k-1:0]) (t)	No	Standby
		1				Z			
X	0	0	0	1	Enabled	I2[n-1:0]	I2[n-1:0]	Yes	Low Power Write
		1				Z			
X	0	0	0	0	Enabled	I2[n-1:0]	I2[n-1:0]	Yes	Normal Write
		1				Z			
X	1	0	0	0	X	D(A2[k-1:0]) (t)	D(A2[k-1:0]) (t)	No	Read
		1				Z			

Note: O1[n-1:0] (t) is the value of Primary Port Output bus in the previous moment of time, and O1[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A1[k-1:0]) (t) is the data in the RAM location specified on the address bus A1[k-1:0] in the previous moment of time, and D(A1[k-1:0]) (t+1) in the next moment of time.
O2[n-1:0] (t) is the value of Dual Port Output bus in the previous moment of time, and O2[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A2[k-1:0]) (t) is the data in the RAM location specified on the address bus A2[k-1:0] in the previous moment of time, and D(A2[k-1:0]) (t+1) in the next moment of time.

5.3.3. Operation modes

Read Mode: The value of Chip Select signal is low (CS1/CS2=0) for read operation. The SRAMnxm enter read mode when CS1/CS2=0 and WEB1/WEB2=1. In this mode the value of SD1/SD2 signal is low. Low SD1/SD2 signal keeps the supply voltage of selected row to high. During read operations, data read from the memory location D(A1[k-1:0])/D(A2[k-1:0]) specified on the address bus I1[n-1:0]/I2[n-1:0] and appear on the data output bus O1[n-1:0]/O2[n-1:0].

Normal Write Mode: The value of Chip Select signal is low (CS1/CS2=0) for write operation. Dual port low power SRAMLPnxm enter write mode when CSB1/CSB2=0 and WEB1/WEB2=0. In write mode the value of SD1/SD2 signal is low. Low SD1/SD2 signal keeps the supply voltage of selected row to high. During write mode, data on the data input bus I1[n-1:0]/I2[n-1:0] is writing into the memory location D(A1[k-1:0])/D(A2[k-1:0]) specified on the address bus I1[n-1:0]/I2[n-2:0].

If OEB1/OEB2=1, data on the output bus O1[n-1:0]/O2[n-1:0] placed in Z state. At that time read/write operation continue. When OEB1/OEB2=0, the data appear on the output bus O1[n-a:0]/O2[n-1:0].

<u>Low Power Write Mode:</u> Low power write mode differs from normal write mode with the value of SD1/SD2 signal. In this mode the value of SD1/SD2 signal is high in order to reduce the supply voltage of selected row and to minimize power dissipation.

<u>Standby Mode:</u> Power dissipation can also be minimized by using static circuit implementations. A standby mode is provided to further reduce power dissipation during periods of non-operation (CSB1/CSB2=1). While in standby mode, address and data inputs are disabled; data stored in the memory D(A1[k-1:0])/D(A2[k-1:0]) is retained, but the memory cannot be accessed for reads or writes.

Address contention will occur when both ports simultaneously access the same address. In this case, both ports will read the same data.

5.3.4. Timing Waveforms

SRAMnxm will function according to the block-diagrams shown in Figures 5.14-5.18.

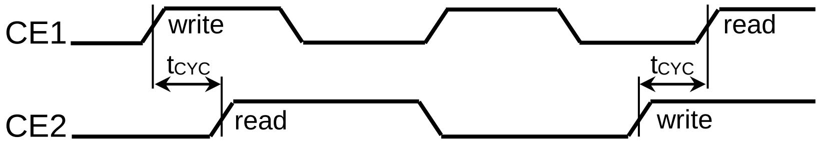


Figure 5.14.Dual port low power SRAMLPnxm Write-Read Clock Timing Waveforms

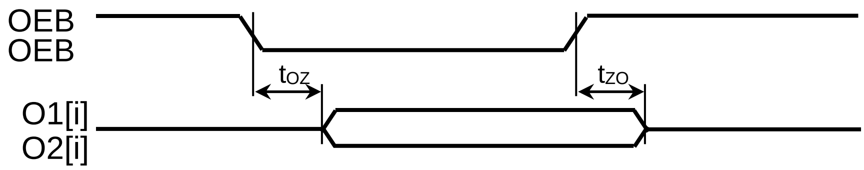


Figure 5.15. Dual port low power SRAMLPnxm Output-Enable Timing Waveforms

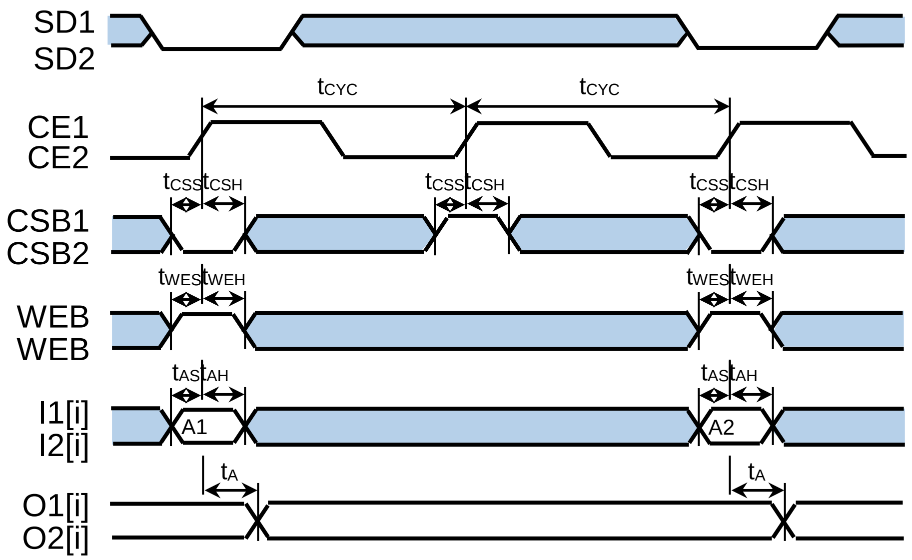


Figure 5.16. Dual port low power SRAMLPnxm Read-Cycle Timing Waveforms

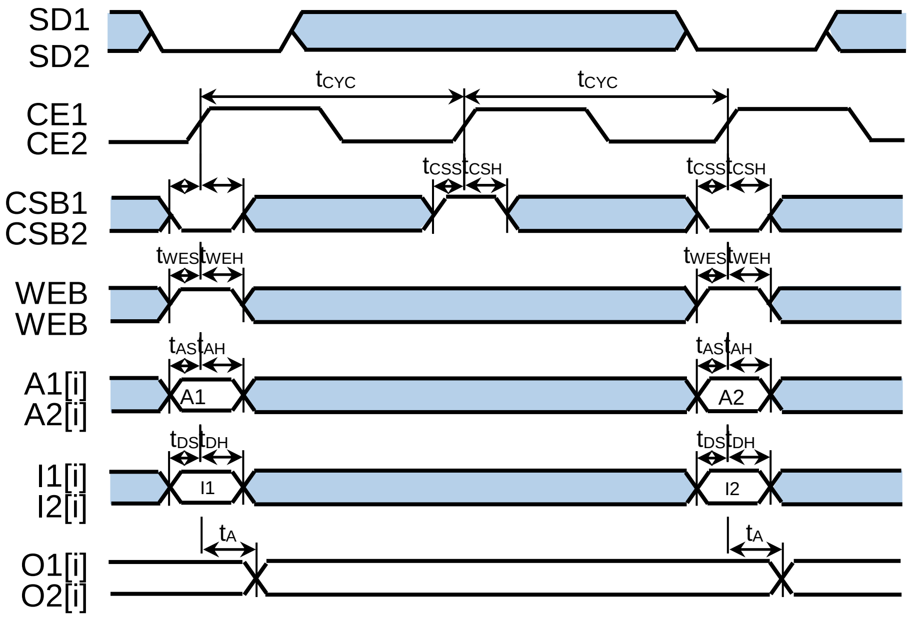


Figure 5.17. Dual port low power SRAMLPnxm Normal Write-Cycle Timing Waveforms

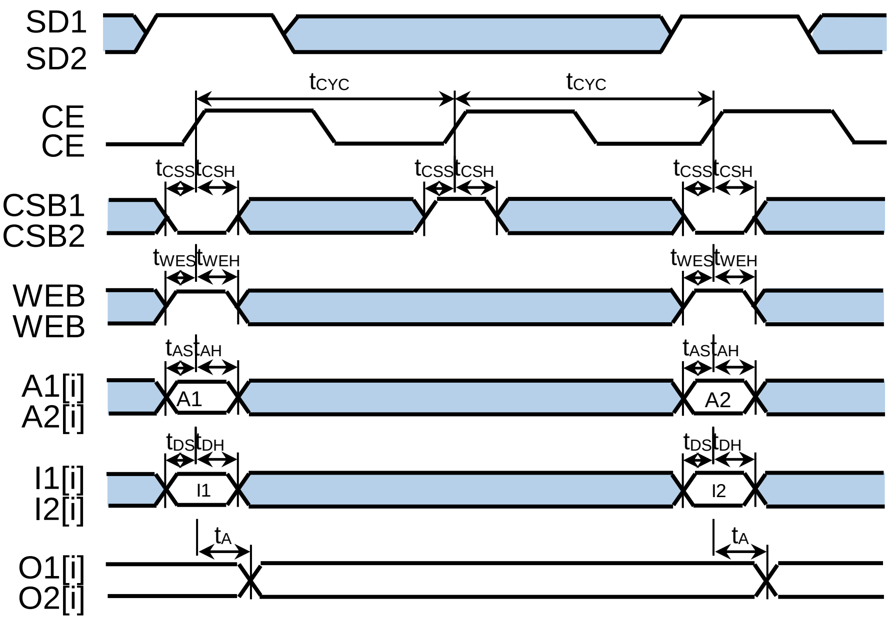


Figure 5.18. Dual port low power SRAMLPnxm Low Power Write-Cycle Timing Waveforms

5.4. Single port low power SRAMs

5.4.1. Basic Pins

The Basic Pins of single port low power SRAMLPnxm_1rw are shown in Figure 5.19. and its descriptions are shown in Table 5.10.

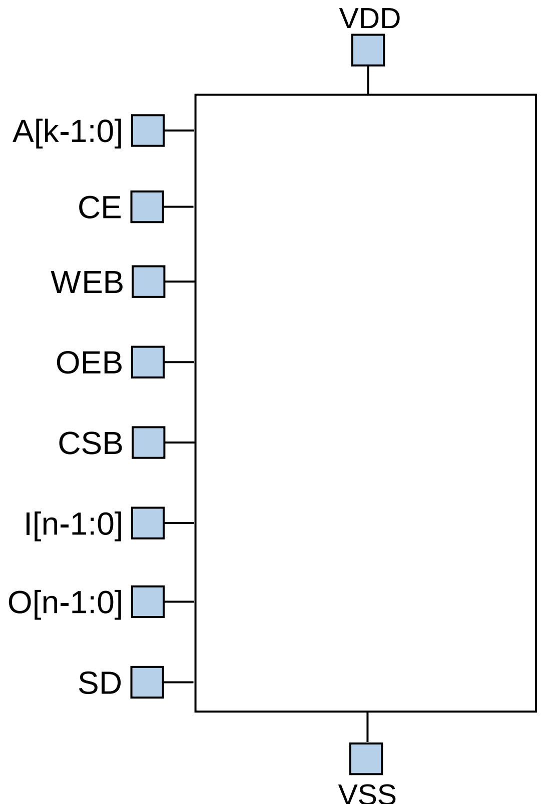


Figure 5.19. Single port low power SRAMLPnxm_1rw Basic Pins

Table 5.10. Single port low power SRAMLPnxm_1rw Pin Definition

Pin Symbol	Width (bits)	Type	Name and Function
A	k	Input	Primary Read/Write Address
CE	1	Input	Primary Positive-Edge Clock
WEB	1	Input	Primary Write Enable, Active Low
OEB	1	Input	Primary Output Enable, Active Low
CSB	1	Input	Primary Chip Select, Active Low
SD	1	Input	Primary Supply down, Active High
I	n	Input	Primary Input data bus
O	n	Output	Primary Output data bus
VDD	Power supply		
VSS	Power ground		

5.4.2. Description

The general block-diagram of single port low power SRAMLPnxm_1rw is shown in Figure 5.20.

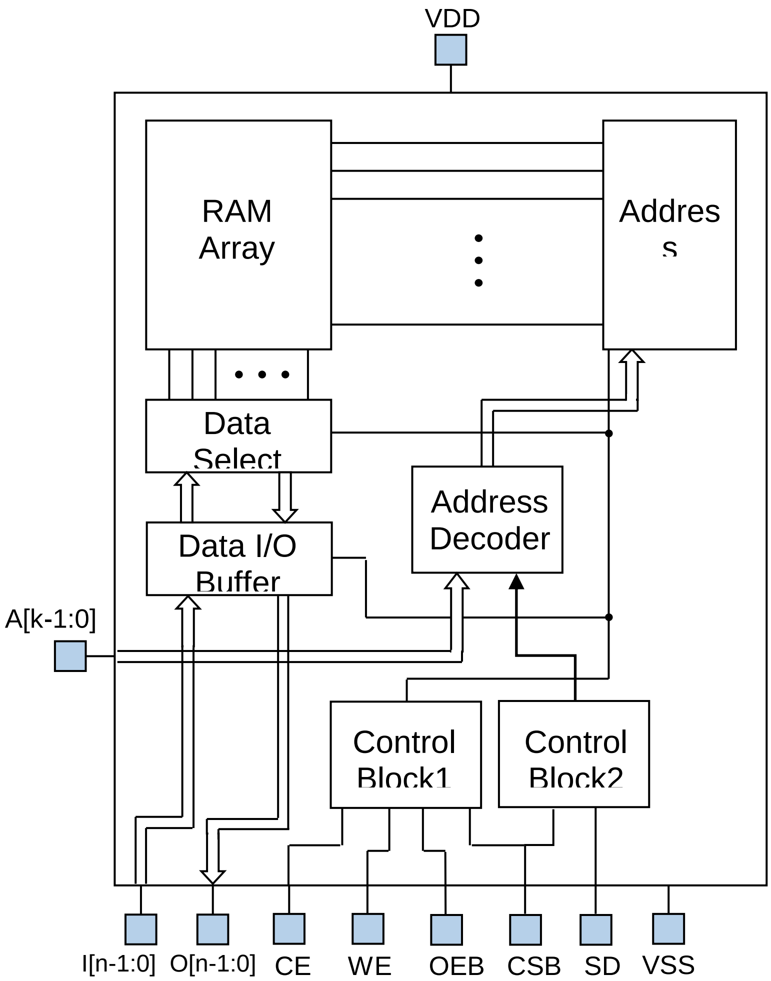


Figure 5.20. Single port low power SRAMLPnxm_1rw block diagram

Single port low power SRAMLPnxm_1rw Basic Operations is shown in Table 2.12.
Single port low power SRAMLPnxm_1rw access is synchronous and triggered by the rising edge of the clock signals (CE). Read/Write addresses (A), Input data (I), Write enable (WEB), Supply down (SD) signals, and Chip select signal (CSB) are latched by the rising edge of the clocks (CE).

Table 5.11. Single port low power SRAMLPnxm_1rw Basic Operations

	Pins						Data in Memory	Access to Memory	Operation
A[k-1:0]	WEB	OEB	CSB	SD	I[n-1:0]	O[n-1:0] (t+1)	D(A[k-1:0]) (t+1)		
X	X	0	1	x	Disabled	O[n-1:0] (t)	D(A[k-1:0]) (t)	No	Standby
		1				Z			
X	0	0	0	1	Enabled	I[n-1:0]	I[n-1:0]	Yes	Low Power Write
		1				Z			
X	0	0	0	0	Enabled	I[n-1:0]	I[n-1:0]	Yes	Normal Write
		1				Z			
X	1	0	0	0	X	D(A[k-1:0]) (t)	D(A[k-1:0]) (t)	No	Read
		1				Z			

Note: O[n-1:0] (t) is the value of Primary Port Output bus in the previous moment of time, and O[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A[k-1:0]) (t) is the data in the RAM location specified on the address bus A[k-1:0] in the previous moment of time, and D(A[k-1:0]) (t+1) in the next moment of time.
O[n-1:0] (t) is the value of Dual Port Output bus in the previous moment of time, and O[n-1:0] (t+1) is the value of the same bus in the next moment of time.
D(A[k-1:0]) (t) is the data in the RAM location specified on the address bus A[k-1:0] in the previous moment of time, and D(A[k-1:0]) (t+1) in the next moment of time.

5.4.3. Operation modes

Read Mode: The value of Chip Select signal is low (CS=0) for read operation. The SRAMnxm enter read mode when CS=0 and WEB=1. In this mode the value of SD signal is low. Low SD signal keeps the supply voltage of selected row to high. During read operations, data read from the memory location D(A[k-1:0]) specified on the address bus I[n-1:0] and appear on the data output bus O[n-1:0]].

Normal Write Mode: The value of Chip Select signal is low (CS=0) for write operation. Single port low power SRAMLPnxm enter write mode when CSB=0 and WEB=0. In write mode the value of SD signal is low. Low SD signal keeps the supply voltage of selected row to high. During write mode, data on the data input bus I[n-1:0] is writing into the memory location D(A[k-1:0]) specified on the address bus I[n-1:0].
If OEB=1, data on the output bus O[n-1:0] placed in Z state. At that time read/write operation continue. When OEB=0, the data appear on the output bus O[n-a:0].

Low Power Write Mode: Low power write mode differs from normal write mode with the value of SD signal. In this mode the value of SD signal is high in order to reduce the supply voltage of selected row and to minimize power dissipation.

Standby Mode: Power dissipation can also be minimized by using static circuit implementations. A standby mode is provided to further reduce power dissipation during periods of non-operation (CCB=1). While in standby mode, address and data inputs are disabled; data stored in the memory D(A1[k-1:0]) is retained, but the memory cannot be accessed for reads or writes.

5.4.4. Timing Waveforms

Single port low power SRAMLPnxm_1rw functions according to the block-diagrams shown in Figures 5.21 – 5.24.

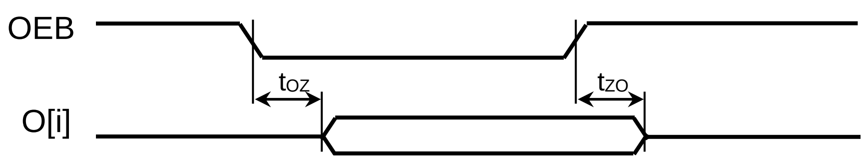


Figure 5.21. Single port low power SRAMLPnxm_1rw Output-Enable Timing Waveforms

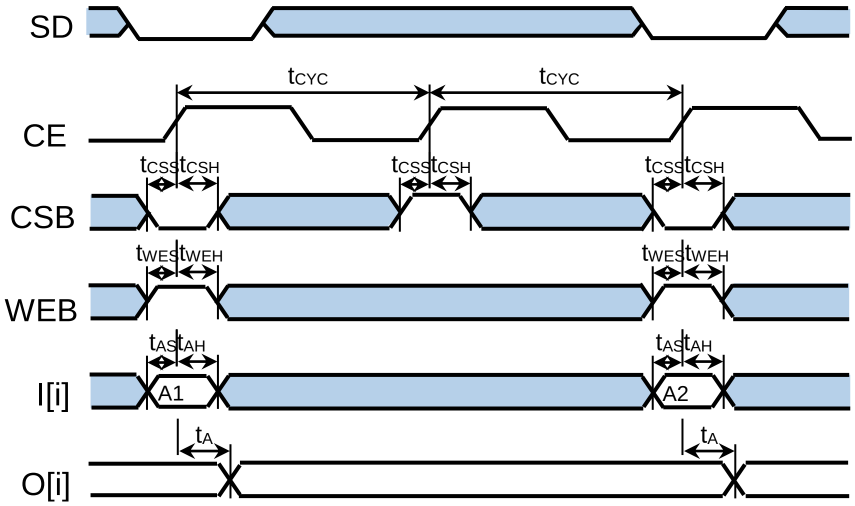


Figure 5.22. Single port low power SRAMLPnxm_1rw Read-Cycle Timing Waveforms

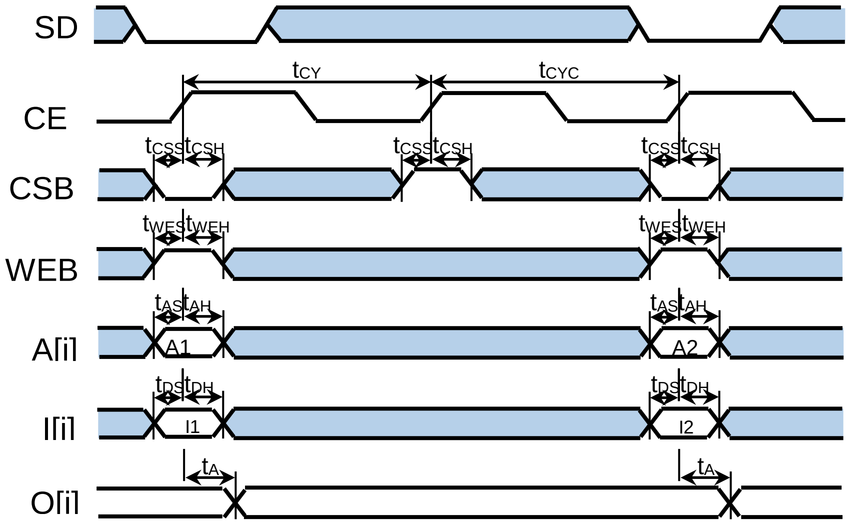


Figure 5.23. Single port low power SRAMLPnxm_1rw Normal Write-Cycle Timing Waveforms

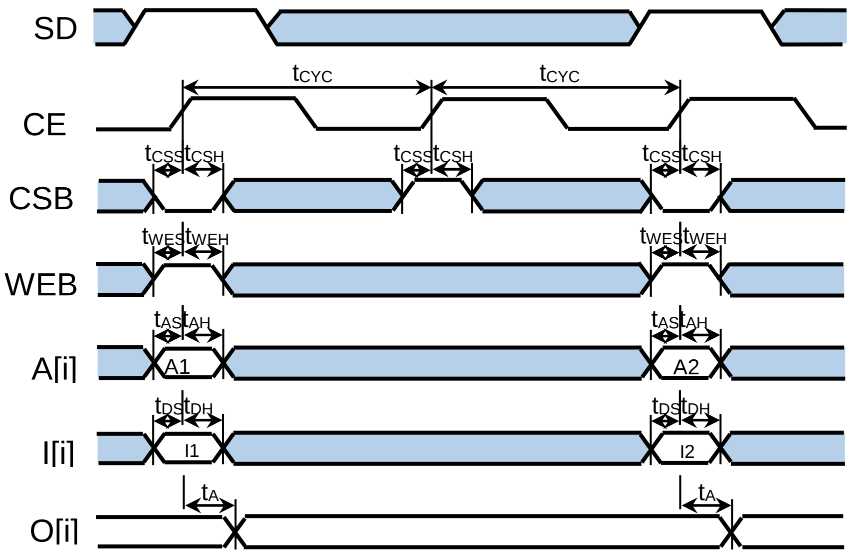


Figure 5.24. Single port low power SRAMLPnxm_1rw Low Power Write-Cycle Timing Waveforms

6. Appendix A: Usage example

The appendix below describes step by step example f SAED memory compiler usage. Before running the steps please make sure that the installation was successfully implemented and system meets system requirements.

1. Create any directory to work in.

```
% mkdir work
% cd work
```

2. Copy sample configuration file from memory compiler demo/ directory to current directory.

```
% cp /remote/install/saed_mc/demo/SRAM14nm.config.
```

3. Run `saed_mc`

```
% saed_mc SRAM.config
```

7. Revision history

Table 7.1. Revision history

Revision	Date	Change