SMIC Data Sheet

Version: 0.2.a Release Date: June 30, 2018

Semiconductor Manufacturing International Corporation

SMIC Data Sheet

Notice

DISCLAIMER

SMIC hereby provides the quality information to you but makes no claims, promises or guarantees about the accuracy, completeness, or adequacy of the information herein. The information contained herein is provided on an "AS IS" basis without any warranty, and SMIC assumes no obligation to provide support of any kind or otherwise maintain the information. SMIC disclaims any representation that the information does not infringe any intellectual property rights or proprietary rights of any third Parties. SMIC makes no other warranty, whether express, implied or statutory as to any matter whatsoever, including but not limited to the accuracy or sufficiency of any information or the merchantability and fitness for a particular purpose. Neither SMIC nor any of its representatives shall be liable for any cause of action incurred to connect to this service.

STATEMENT OF USE AND CONFIDENTIALITY

The following/attached material contains confidential and proprietary information of SMIC. This material is based upon information which SMIC considers reliable, but SMIC neither represents nor warrants that such information is accurate or complete, and it must not be relied upon as such. This information was prepared for informational purposes and is for the use by SMIC's customer only. SMIC reserves the right to make changes in the information at any time without notice. No part of this information may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of SMIC. Any unauthorized use or disclosure of this material is strictly prohibited and may be unlawful. By accepting this material, the receiving party shall be deemed to have acknowledged, accepted, and agreed to be bound by the foregoing limitations and restrictions. Thank you.

Semiconductor Manufacturing International Corporation
No. 18 Zhangjiang Road
Pudong New Area
Shanghai 201203
The People's Republic of China
URL: www.smics.com

Chapter 1 Installing and Using the Compiler

1.1 Introduction

This user guide provides the information of SMIC 0.18um Dual-Port SRAM compiler about how to generate embedded SRAM macros and corresponding Design Kits.

1.2 Memory Compiler Features

- High speed and high density
- Optimized power distribution scheme (Over the cell power routing)
- Bits write option
- Low active power and low standby leakage power
- Timing and power models for advanced design tools

1.3 Compiler Installation

This section contains System Environment Requirements and Installation Instructions for the SMIC 0.18um Dual-Port SRAM Compiler.

1.3.1 System Environment Requirements

Requires Java Version 1.6.0_12（JDK 1.6.0_12) . If Java Version is not 1.6.0_12, Compiler may crash by chance. If it happened, please restart compiler.
To verify your Java version using the following command:
% java –version

1.3.2 Installation Instructions

- The SMIC 0.18um Dual-Port Memory Compiler installation package: S018DP.tar.gz, includes the following files:

1. S018DP.jar # main memory Generator application
2. S018DP.csh # setup file for C shell
3. S018DP_ug.pdf #user guide
4. S018DP.notes #release note

5. Library #library for GDS Smart Option

- Installation Procedure:

1. Copy the installation package into a stand-alone directory.

2. Uncompress and untar the application package:

% gunzip < S018DP .tar.gz | tar xvf -

or

% gtar xzvf S018DP .tar.gz

3. Verify that the files in the install directory are executable.

% chmod -R 755 /your_directory/

1.4 Launching the Compiler from GUI

Invoke the SMIC Memory Generator GUI in /your directory/ using the following command with no arguments:

% S018DP.csh

1.5 Compiler Window

For Example, SMIC 0.18um Dual-Port Memory Compiler GUI is shown in Figure 1-1

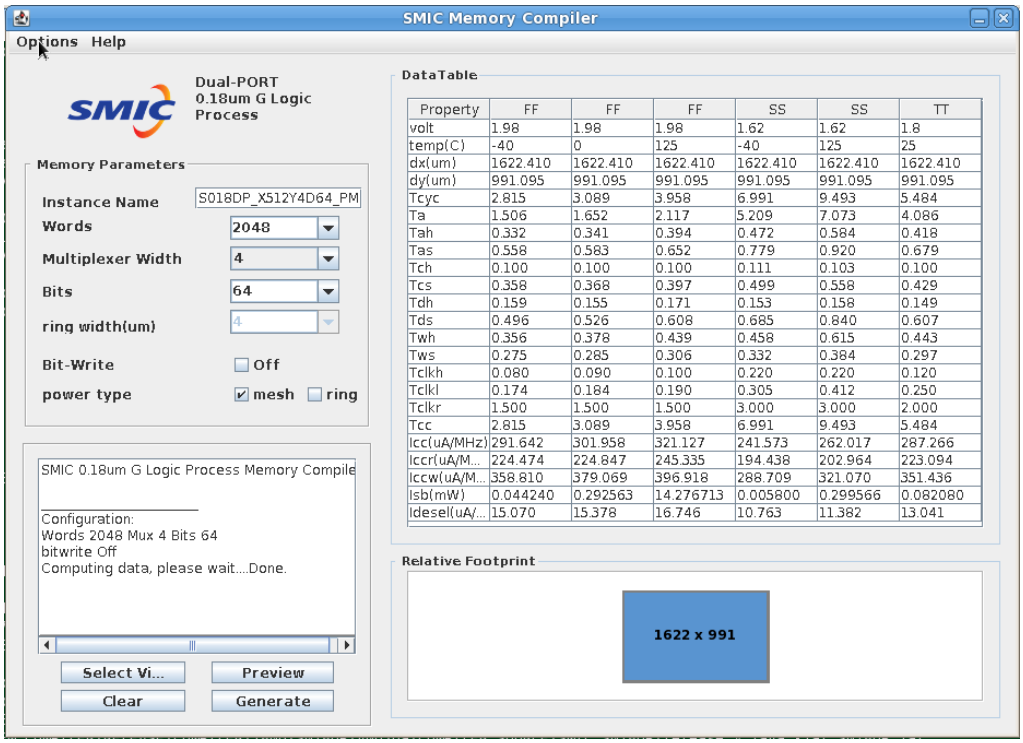

Figure 1-1 SMIC Memory Compiler GUI

1.5.1 Memory Parameters Panel

The Memory Parameters panel contains input field and check boxes. You can change values of *Word, Multiplexer* and *Bits* by selecting the value you want from the pull-down submenu corresponding to each parameter or typing the value in the input field. Also you could select *off* or *on* for *Bit-Write* and *mesh* or *ring* for power structure in check boxes. If you select *ring*, the Power ring width pull-down submenu will turn optional, and the power rings must be properly sized considering the chip-level power distribution methodology, the number, width and placement of supply wire connections to the power rings, and current consumption.

When you finish filling the parameter values, you could view the update *DataTable* panel, *Message* panel and *Relative Footprint* panel by click *Preview* button.

For Example, you could generate a view for a specific instance with 8192 words, 64 bits and 16 Multiplexer Width. Select 8192 from *Words* pull-down submenu, 16 from *Multiplexer Width* pull-down submenu and 64 from *Bits* pull-down submenu. Click *Preview* button. Then *DataTable* panel, *Message* panel and *Relative Footprint* panel will be updated.

1.5.2 Output View Panel

You could generate multiple views at a time. To generate a single view or multiple views, select the view you want from the *Output View* checkboxes. Click the *Generate* button, then *save as…* submenu will pop up, enter path or folder name, click OK. The view will be generated and be saved at the path you entered just now.

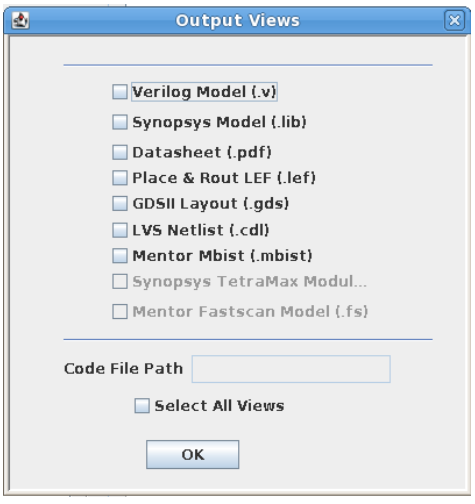

Figure 1-2 SMIC Memory Compiler output view

1.5.3 Relative Footprint Panel

The *Relative Footprint* panel shows the aspect ratio of the corresponding specific instance. When you change the *Memory parameters* and click *Preview* button, the *Relative Footprint* will be updated.

1.5.4 Data Table Panel

The *Data Table* shows corners, layout size, timing, current, and power information for the corresponding specific instance.

When you change the *Memory parameters* and click *Preview* button, the *Data Table* will be updated.

1.5.5 Message Panel

The *Message* panel shows messages when you generate an instance. The message includes the configuration information of the instance be generating, the associated view type information be generating and successful generation information.

The *Message* panel also report problem when you enter the values out of the pre-determined range. For Example, when enter 322 in the *bits* input field, it will report the following message (see Figure 1-3).

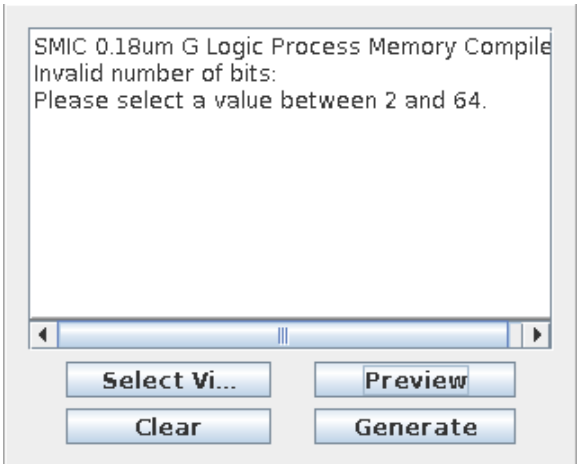

Figure 1-3 Message Panel

1.5.6 File Menu

To exit the Compiler GUI, select *File* menu, then select *Exit* submenu.

1.6 Views and Output Files

Table 1-1 lists the kinds of views you could generate and the corresponding output files.

Table 1-1 Views and Output Files

View	Output Files
LEF footprint	<instance name>.lef
Synopsys model	<instance name>.lib
PDF datasheet	<instance name>.pdf
Verilog model	<instance name>.v
GDSII	<instance name>.gds
LVS Netlist	<instance name>.cdl
MBIST model	<instance name>.mbist

1.7 Generating Views

You can generate single or multiple views from GUI and command line.

1.7.1 Generating Views from GUI

You can generate multiple views from GUI. To generate a new instance, change the *Memory parameters*, click *Preview* button. Select a view from *Output View* checkbox, click *Generate* button. The view will be generated.

1.7.2 Generating Views from Command Line

Generate single or multiple output views from the command line by specifying additional arguments. Enter “java –jar S018DP.jar -help" for usage options:

```
% java –jar S018DP.jar [options...]
```

-help : display usage options
-bitwrite : toggle bitwrite
-bits VAL : number of bits
-cdl : generate cdl
-gds : generate GDS
-instname VAL : define instance name
-lef : generate .lef footprint
-lib : generate Synopsys .lib model
-mux VAL : column multiplexer value
-mbist : generate mentor mbist model
-pdf : generate pdf datasheet
-powerring : generate gds with power ring
-powermesh : generate gds with power mesh
-ringwidth VAL : define the width of power ring
-savepath VAL : define saved file path
-v : generate verilog model
-words VAL : number of words

-words, -mux, -bits, -powerring or -powermesh and at least one output view must be specified. When choosing -powerring, the default ringwidth is 4 without specifying -ringwidth. The range of ring width is from 4 to 50.

Ex. % java –jar S018DP.jar -words 8192 -mux 16 -bits 32 –gds -v -lef -lib –pdf -powerring –ringwidth 5

Chapter 2 Synchronous SRAM Compiler

2.1 Synchronous Dual-Port SRAM Structure and Timing Specifications

2.1.1 Dual-Port SRAM Description

2.1.1.1 Basic Functionality

The synchronous dual-port SRAM has A, B ports to access the same memory location. Both ports can be independently read or written from the memory array.
The SRAM access is synchronous and is triggered by the rising edge of the clock. The write enable (WENA, WENB) and bit-write enable (BWENA[n-1:0], BWENB[n-1:0]), chip enable (CENA, CENB), address (AA[i-1:0], AB[i-1:0]), and data in (DA[n-1:0], DB[n-1:0]) signals are latched on the rising edge of the clock.

- If bit-write is off:

When CENA is low and WENA is high, the memory will be in read operation. Data is read from the location specified by the address AA[i-1:0], and is output to QA[n-1:0].

When CENB is low and WENB is high, the memory will be in read operation. Data is read from the location specified by the address AB[i-1:0], and is output to QB[n-1:0].

When CENA and WENA are both low, the memory will be in write operation. The word on the data port DA[n-1:0] will be written into the location specified by the address AA[i-1:0] and the data will appear on the output port QA[n-1:0].

When CENB and WENB are both low, the memory will be in write operation. The word on the data port DB[n-1:0] will be written into the location specified by the address AB[i-1:0] and the data will appear on the output port QB[n-1:0].

When CENA and CENB are high, the memory is in standby mode. Meanwhile, the data stored in memory is retained but cannot be read or written.

- If bit-write is on:

When CENA is low and WENA is high, the memory will be in read operation. Data is read from the location specified by the address AA[i-1:0], and is output on the output port QA[n-1:0].

When CENB is low and WENB is high, the memory will be in read operation. Data is read from the location specified by the address AB[i-1:0], and is output on the output port QB[n-1:0].

When CENA is low, WENA is low and BWENA[j] is high, the memory will be in write block operation. DA[j] cannot be written in, QA[j] is the data stored in selected bit cell.

When CENB is low, WENB is low and BWENB[j] is high, the memory will be in write block operation. DB[j] cannot be written in, QB[j] is the data stored in selected bit cell.

When CENA is low, WENA is low and BWENA[j] is low, the memory will be in write operation, the corresponding data on the data port DA[j] will be written into the location specified by the address AA[i-1:0] and the data will appear on the corresponding output port QA[j].

When CENB is low, WENB is low and BWENB[j] is low, the memory will be in write operation, the corresponding data on the data port DB[j] will be written into the location specified by the address AB[i-1:0] and the data will appear on the corresponding output port QB[j].

When CENA and CENB are high, the memory is in standby mode. Meanwhile, the data stored in memory is retained but cannot be read or written.

For instance, a memory has 16 bits. When CENA is low meanwhile WENA is low, the bit-write enable BWENA [7:0] is low and BWENA [15:8] is high, the memory will be in write operation. The word on DA [7:0] will be written into the location specified by the address AA [i-1:0] and the data will appear on the output port QA [7:0]. DA [15:8] could not be written into the memory and could not be output on the output port. The output port QA [15:8] will output the data stored in addressed location.

For read and write detailed description and clock conflict issue, please see Table 2-1.

Table 2-1 Read and write behavior when access the same address

Read/Write truth table							
Function	**CLKA/B**	**CENA/B**	**WENA/B**	**BWENA/B**	**DA/B**	**QA/B**	
Deselect	X	H	X	X	X	Q-1	
Read	L→H	L	H	X	X	Q	
Write	L→H	L	L	L	Data-in	Data-in	
Write Block	L→H	L	L	H	Data-in	Q-1	
（A address=B address） & （CLKA=CLKB）							
Port A	**Port B**	**DA**	**DB**	**CENA**	**CENB**	**QA**	**QB**
Read	Read	DA	DB	L	L	Q	Q
Write	Read	DA	DB	L	L	Write and read operation at the same address is forbidden in one cycle.	
Read	Write	DA	DB	L	L		
Write	Write	DA	DB	L	L		
（A address≠B address） || （CLKA≠CLKB）							
Port A	**Port B**	**DA**	**DB**	**WENA**	**WENB**	**QA**	**QB**
Read	Read	DA	DB	H	H	QA	QB
Write	Read	DA	DB	L	H	DA	QB
Read	Write	DA	DB	H	L	QA	DB
Write	Write	DA	DB	L	L	DA	DB

2.1.2 Dual-Port SRAM Pins

Figure 2-2 shows the basic Dual-Port SRAM Pins.

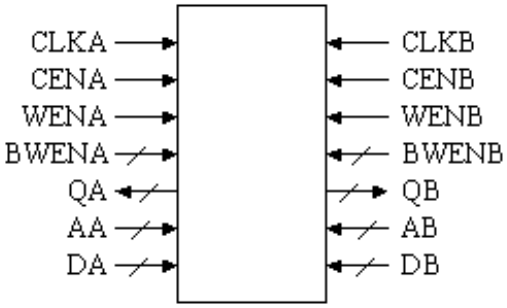

Figure 2-1 Dual-Port SRAM Basic Pins

Table2-2 shows the basic Dual-Port SRAM Pins Definition.

Table2-2 the basic Pins Definition

PIN	DIRECTION	DEFINITION
AA[i-1:0]	Input	A Port Address Inputs
DA[n-1:0]	Input	A Port Data Inputs
AB[i-1:0]	Input	B Port Address Inputs
DB[n-1:0]	Input	B Port Data Inputs
WENA	Input	A Port Write Enable
BWENA[n-1:0]	Input	A Port Bit-Write Enable
WENB	Input	B Port Write Enable
BWENB[n-1:0]	Input	B Port Bit-Write Enable
CENA	Input	A Port Enable
CENB	Input	B Port Enable
CLKA	Input	A Port Clock Input
CLKB	Input	B Port Clock Input
QA[n-1:0]	Output	A Port Data Outputs
QB[n-1:0]	Output	B Port Data Outputs

2.1.3 Dual-Port SRAM Logic Table

This section shows logic tables for Dual-Port SRAM basic functions.

- 0.18um Dual-Port SRAM basic functions are described in Table 2-3 and Table 2-4

Table 2-3 0.18um Dual-Port SRAM basic functions (Bit-Write is off)

CEN	WEN(A/B)	Operation	Output	Function
Low	High	Read	Stored data	Data is read from the location specified by the address A(A/B)[i-1:0] and is output on the output port Q(A/B)[n-1:0]
Low	Low	Write	Data In	The word on the data port D(A/B)[n-1:0] is written into the memory location specified by the address A(A/B)[i-1:0] and is output on the output port Q(A/B)[n-1:0]
High	X	Standby	Last Data	All input pins are inactive. The data stored in memory is retained but cannot be read or written
High	X	Deselect	Last Data	Half of D(A/B) [n-1:0] and A(A/B) [i-1:0] pins are active. The data stored in memory is retained but cannot be read or written

Table 2-4 0.18um Dual-Port SRAM basic functions (Bit-Write is on)

CEN	WEN (A/B)	BWEN (A/B)[j]	Operation	Output	Function
Low	High	X	Read	Stored data	Data is read from the location specified by the address A[i-1:0] and is output on the output port Q[n-1:0]
Low	Low	High	Write	Stored data	Data is read from the location specified by the address A[i-1:0] and is output on the output port Q[n-1:0]
Low	Low	Low	Write	Data In	The corresponding data on the data port D(A/B)[j] will be written into the memory location specified by the address A(A/B)[i-1:0] and the data will appear on the corresponding output port Q(A/B)[j]
High	X	X	Standby	Last Data	All input pins are inactive. The data stored in memory is retained but cannot be read or written
High	X	X	Deselect	Last Data	Half of D(A/B) [n-1:0] and A(A/B) [i-1:0] pins are active. The data stored in memory is retained but cannot be read or written

2.1.4 0.18um Dual-Port SRAM Parameters

The 0.18um Dual-Port SRAM standard input and block parameters are listed in Table 2-5. You could refer to Compiler GUI for pre-determined input ranges. If you enter an invalid value and *Preview* the window, a problem message and the pre-determined range of the value will be displayed in the *Message* panel.

Table 2-5 0.18um Dual-Port SRAM Parameters

Parameters	Ranges
Words Number	Ymux = 4 32~2048, increment = 8*Ymux
	Ymux = 8 64~4096, increment = 8*Ymux
	Ymux = 16 128~8192, increment = 8*Ymux
Bits Number	Ymux = 4 2~64, increment = 1
	Ymux = 8 2~32, increment = 1
	Ymux = 16 2~16, increment = 1
MUX	4, 8,16; default 4
Bit-Write	on, off; default off
Top chip metal layers support	Metal4 for power mesh and Metal3 for power ring to top metal layer supported by design process
Power type	metal4 for power mesh and metal2/metal3 for power ring power strap

2.1.5 0.18um Dual-Port SRAM Block Diagram

The instance block diagram of 0.18um Dual-Port SRAM is shown in Figure 2-2

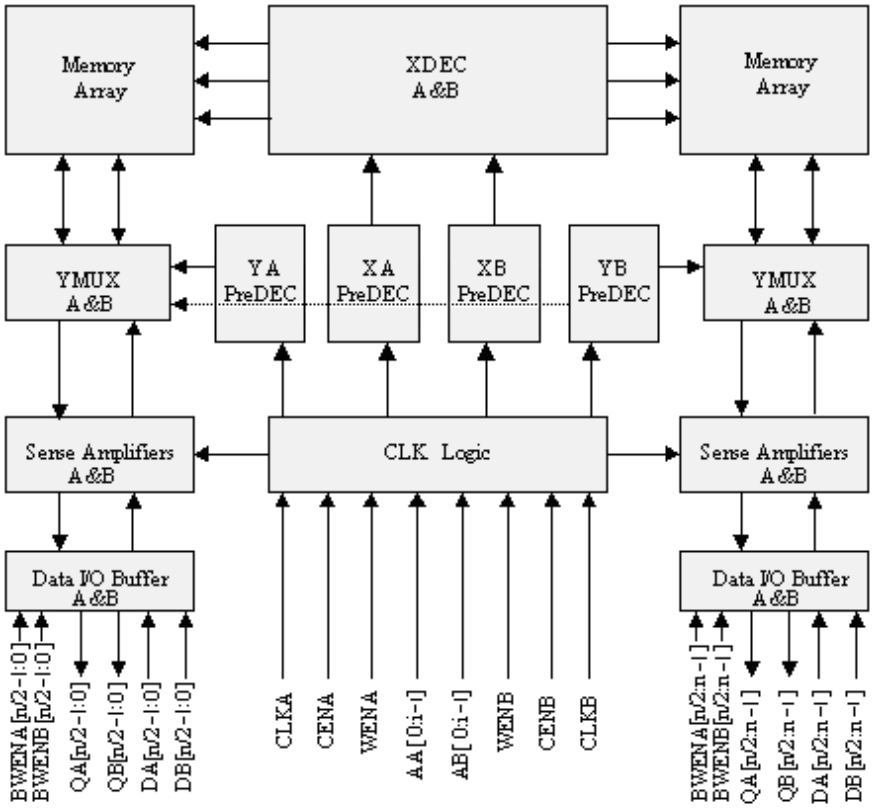

Figure 2-2 0.18um Dual-Port SRAM Block Diagram

2.1.6 0.18um Dual-Port SRAM Core Address Maps

This section shows the core address maps. An example of the core address maps is shown in Figure 2-3

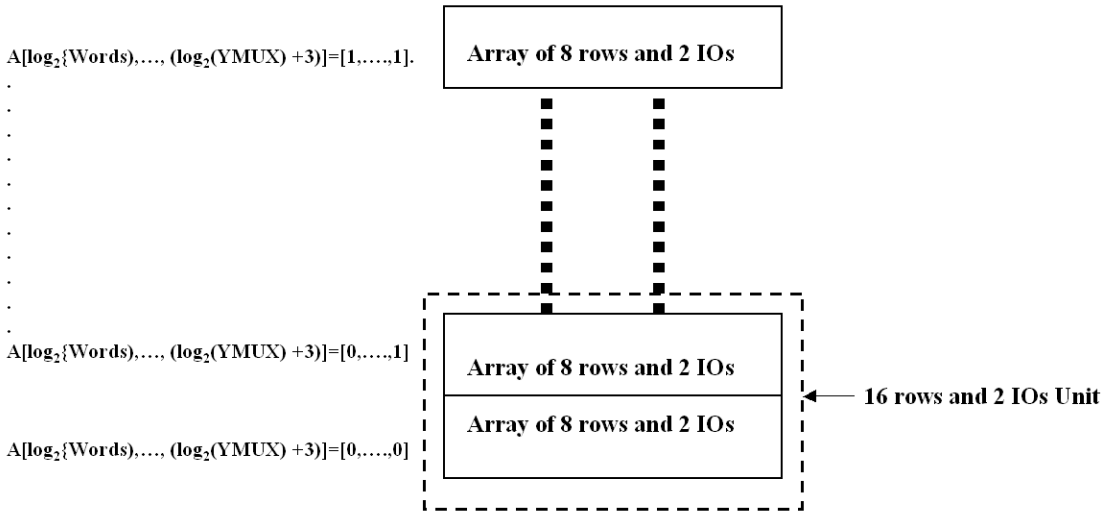

Figure 2-3 0.18um Dual-Port SRAM Core Address Maps

2.1.7 0.18um Dual-Port SRAM Timing Description

This section describes the timing diagrams, timing parameters and power parameters of 0.18um Dual-Port SRAM.

2.1.7.1 0.18um Dual-Port SRAM Timing Diagrams

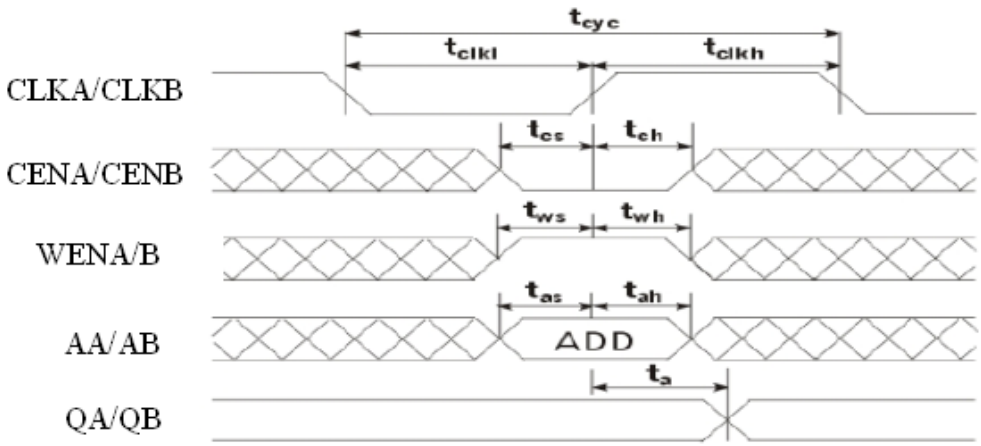

Figure 2-4 0.18um Dual-Port SRAM Read-Cycle Timing

[WEN is a single pin. BWEN is a bus pins. (bit-write is on)]

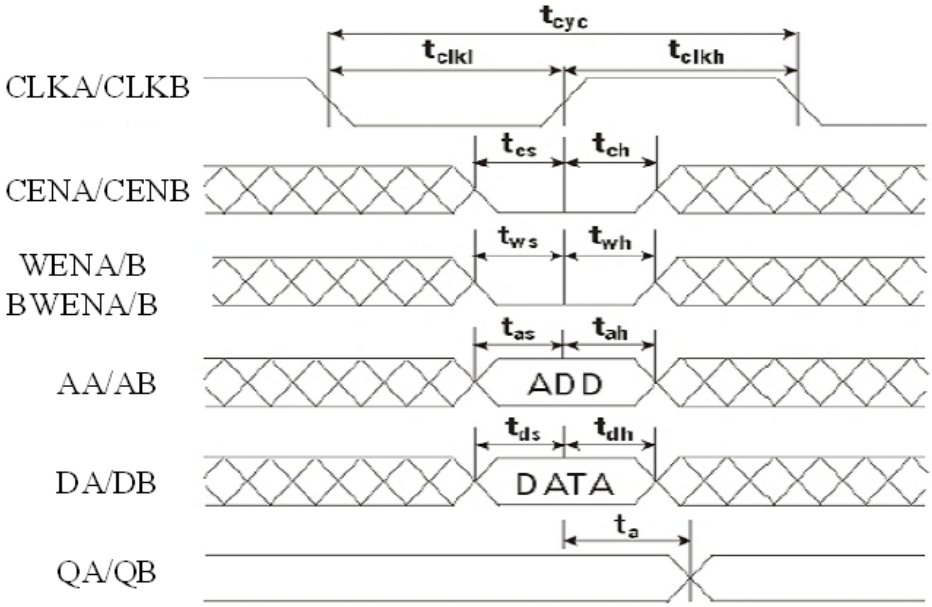

Figure 2-5 0.18um Dual-Port SRAM Write-Cycle Timing

[WEN is a single pin. BWEN is a bus pins. (Bit-write is on)]

2.1.7.2 0.18um Dual-Port SRAM Timing Parameters

The timing parameters that show up in GUI *Data Table* are listed in Table 2-6.

Table 2-6 0.18um Dual-Port SRAM Timing Parameters

Parameter (ns)	**Description**
Tcyc	Cycle Time
Ta	Access Time
Tas	Address Setup
Tah	Address Hold
Tds	Data Setup
Tdh	Data Hold
Tcs	Chip Enable Setup
Tch	Chip Enable Hold
Tws	Write Enable Setup
Twh	Write Enable Hold
Tckh	Clock High
Tckl	Clock Low
Tckr	Clock Rise Skew
Tcc	Clock Collision

2.1.7.3 0.18um Dual-Port SRAM Power Parameters

The timing parameters that show up in GUI *Data Table* are listed in Table 2-7.

Table 2-7 0.18um Dual-Port SRAM Power Parameters

Parameter	Description
icc	AC Current (uA/MHz)
iccr	Read AC Current (uA/MHz)
iccw	Write AC Current (uA/MHz)
isb	Standby Power (mW)
Idesel	Deselect power (uA /MHz)

2.2 0.18um Dual-Port SRAM Power Structure

2.2.1 Current Calculations

2.2.1.1 Average current

The average current, I_avg, which is reported in the datasheet and GUI timing table can be calculated by following formula:
I_avg = I_avg_read*M_read_percentage + I_avg_write*N_write_percentage + (C_pin*V*F)*S_data_switch_percentage
Where,

- I_avg_read= average read current (mA)
- I_avg_write= average write current (mA)
- M_read_percentage= percentage of read operation (%)
- N_write_percentage= percentage of write operation (%)
- S_data_switch_percentage= percentage of input and output pins switch (%)
- C_pin = average capacitance of output port (pF)
- V= power supply voltage (V)
- F= operation frequency (Hz)

The I_avg in our datasheet is calculated by the assumption of 100% address switch, M=N=S=50%.
User can recalculate the I_avg base on the special value of M, N and S in your design.

2.2.1.2 Peak current

The peak current, Ipeak, is calculated by the SPICE simulator during the Read/Write

operations. The Ipeak is the simulated value and the duration is short.

2.2.1.3 Standby current

The standby current, I_standby, is measured from following operation condition:
CEN is disabled; all of the input pins are stable.

2.2.1.4 Deselect current

The deselect current, Idesel, is measured from following operation condition:
CEN is disabled; half of the input pins are active.

2.2.2 Power Distribution Methodology

User's full chip level power mesh which connects to SRAM macro must meet the spec of EM(electro-migration) rules and the limitation of average and peak IR drop. The voltage supplied to the SRAM macro must be the same value as the characterized voltage to make sure the SRAM speed accuracy.

User can calculate the minimum power bus width connect to SRAM as following formulas:
(This formula does not include the other elements which could consume power through the same power wire. User needs to take it into account for minimum power bus calculating)

- W = max (WEM, WIR_PEAK, WIR_AVERAGE)
- W_EM = I_avg/D (um)
- W_IR_PEAK = I_peak*(R_square* L_effective_power)/⊿V_IR_PEAK (um)
- W_IR_AVERAGE = I_avg*(R_square* L_effective_power)/⊿ V_IR_AVERAGE (um)

Where,

- W = Final minimum power bus width
- W_EM = Power bus width based on EM rules limitation (um)
- W_IR_AVERAGE= Power bus width based on average current IR drop voltage limitation (um)
- W_IR_PEAK = Power bus width based on peak current IR drop voltage limitation (um)
- I_avg = Average current of SRAM macro (mA)
- I_peak = Peak current of SRAM macro (mA)
- D = The value of metal current density rules (uA/um)
- R_square = Square resistance of metal (Ohm/square)

- L_effective_power = The effective power bus connection length from IO power PAD to the SRAM macro (um)
- ⊿V_IR_PEAK = The spec of peak current IR drop of the chip level power mesh (mV)
- ⊿V_IR_AVERAGE = The spec of average current IR drop of the chip level power mesh (mV)

2.2.3 Noise Limits

The characterized clock noise limit, vn_ck, is the maximum CLK voltage allowable for the indicated pulse width without causing a spurious memory cycle or other memory failure. The standard pulse width, pwn_ck, used in characterizing this limit is 10ns.

The power and ground noise limits vn_pwr and vn_gnd respectively are the maximum supply or ground voltage transition allowable without causing a memory failure. Power and ground noise limits are assured at 10% of the characterized voltage.

2.3 0.18um Dual-Port SRAM Physical Characteristics

This section indicates physical characteristics of 0.18um Dual-Port SRAM/ Register-File.

2.3.1 TOP Metal Layer

The Compiler supports different top metal layer designs. If users choose power ring structure, they have the flexibility to choose METAL4, METAL5, METAL6, METAL7, METAL8 or METAL9 as top metal base on user's design spec. If users choose power mesh structure, they have the flexibility to choose METAL5, METAL6, METAL7, METAL8 or METAL9 as top metal base on user's design spec.

2.3.2 Pin Connections

All of the signal pins of the SRAM macro are located on the bottom of the block. And the signal pins can be accessed by the router from METAL1 to METAL3 routing layers.

2.3.3 Characterization Environments

The generator is characterized at FF, TT and SS corners. SMIC recommends user to perform setup, hold and critical path timing analysis at all applicable corners.

Chapter 3 Compiler Views

3.1 Overview

This chapter lists the EDA tools used for verification and the tools supported by the Compiler.

3.2 Tool Verification

This section introduces the views generated by the Compiler and the tools used for verification.

Table3-1 Views and corresponding verification tools

Views	Tool
Verilog(.v)	VCS (Synopsys)
Liberty model(.lib)	Design Compiler (Synopsys)
LEF file(.lef)	SOC Encounter (Cadence)
GDSII and Netlist	Calibre (Mentor)
MBIST file(.mbist)	Mbistarchitect(Mentor)

3.3 Using the Views

3.3.1 Using Verilog Model

When finish generating the Verilog model, you could simulate the file using VCS.

Simulation:

```
%   vcs   <TestBench>.v
%   simv > verilog.log
```

The simulation output will be put into a file verilog.log.

Also you could check syntax of the Verilog model using VCS.
Check syntax:

```
%   vcs <instance_name>.v
```

% simv

<instance_name>.v is the verilog model file.

3.3.2 Using LEF file

LEF file provides an abstract view for the memory. It contains the information required by Cadence placement and routing tools. This abstract view defines the size of the instance, pin names, pin locations and layers information.

LEF view only contains MACRO definition. You should have a technology file which contains layer names, layer properties and via definitions. The technology file should be loaded into the router before reading the LEF file.

You could also load the LEF file into Milkyway.
Start Milkyway. You could type in *read_lef* in the command window or select menu *Cell Library -> Lef In…,* a submenu will popup, fill in the .lef file name, then click OK to create FRAM view and CEL view.

3.3.3 Using Liberty model

The Compiler could generate TT/FF/SS Synopsys .lib files. The SS corner model is generated with maximum delays, and the FF corner model is generated with minimum delays.

The Liberty model contains delay model, environment conditions and lookup table. Delay model provides information for lookup table delay calculation. Environment conditions provide operating conditions, default attributes and k-factors for circuit timing evaluation. Lookup table describes pin capacitance, timing and power information.

You could generate SDF from .lib using the following steps:
Start Synopsys Design Compiler:

% dc_shell

You will in the Synopsys Design Compiler command window, type in the following commands:

% read_lib <instance_name>.lib
% write_lib <userdefine>
% link_library=<userdefine>.db

```
% target_library=<userdefine>.db
% read –f verilog <instance_name>.v
% write_timing –context verilog –f sdf –v2.1 –0 <out_file_name>
```

Where <instance_name>.lib is the Liberty file; <userdefine> is the name of your library; <instance_name>.v is the verilog file; <out_file_name> is the name of out put file.

3.3.4 Using GDSII

The GDSII file contains layouts of the instances. It is GDSII format and contains the mask data. It could also be used to do LVS check.

You could also load the GDSII file into Milkyway.
Start Milkyway. You should already read the .lef file into the Milkyway before this. You could type in *auStreamIn* in the command window or select menu *Cell Library -> Stream in…* to read in the GDSII file. This will overwrite the CEL view with the GDSII layout.

3.3.5 Using Netlist

After finish generating the Netlist, you could use it for LVS check or function simulation.
Usually, you may use Calibre(Mentor Graphics) to do LVS check.
The netlist could be added to the chip level netlist, then you could do chip level LVS check.

3.3.6 Using MBIST

After finish generating the MBIST model, you could use it as a reference model for generating memory BIST logic with Mbistarchitect(Mentor Graphics).