



**DO NOT SHARE  
SLIDES AND CLASS MATERIALS  
ON ONLINE SITES**

Course Hero

# Memory Compiler

Xiaofu Pei

23<sup>rd</sup> Oct, 2019

# Memory Compiler

---

- Use of Standard Library 130nm to 250nm SRAM generators to create instances with a variety of parameters and views
- One can generate single and dual-port, high speed and high density SRAM.
- Located at:  
***/courses/ee6321/share/mem\_comp***

# SRAM Generator Features

---

- Optimized for High Speed/Density or Low-Power or High Density
- Aspect Ratio Control for Efficient Floor Planning
- Memory Operation and Retention at Low Frequency
- Low Active Power and Leakage-Only Standby Power
- Timing and Power Models for Industry-Leading Design Tools
- Configurable Word-Write Mask Option

# Generating SRAM example (1)

---

- Create “mem\_comp” folder in your directory
- Copy the contents from  
/courses/ee6321/share/mem\_comp  
to your local folder
- Use  
**For one-port memory**  
“/courses/ee6321/share/mem\_comp/rf1shd/bin/rf1shd”  
**For two-port memory**  
“/courses/ee6321/share/mem\_comp/ra2shd\_v20/bin/ra2shd\_v20”

# SRAM Standard Support

---

- PostScript Datasheet
- ASCII Datable
- Verilog Model ( .v)
- VHDL Model
- Synopsys Model ( .lib)
- PrimeTime Model ( .data .mod)
- TLF Model
- VCLEF Footprint
- GDSII Layout
- LVS Netlist

# Generating SRAM example (2)

! There is a limited range for each parameters

**Number of Words**

- [16, 256]

**Number of Bits**

- [4, 128]

Choose output file listed in slide 4

ARM@CADPC20

File Utilities Help

Artisan<sup>®</sup>  
ARM Physical IP

High Speed/Density Single Port Register File Generator  
for CMRF8SF-LPVT Process

## Memory Parameters

**GENERIC PARAMETERS**

Instance Name: memory

Number of Words: 128

Number of Bits: 16

Frequency <MHz>: 100

Ring Width <um>: 2.0

Multiplexer Width: ☐ 1 ☒ 2 ☐ 4

Word-Write Mask: ☐ on ☒ off

Top Metal Layer: ☒ m6-m8\_m4


Power Type: ☒ rings

Horizontal Ring Layer: ☒ m1 ☐ m2 ☐ m3

Vertical Ring Layer: ☐ m1 ☒ m2 ☐ m3

Default Update

**RELATIVE FOOTPRINT**



**ASCII DATATABLE**

name	ff_1p32v...	ff_1p65v...	tt_1p2v_25c	ss_1p08v...
geomx	74.240	74.240	74.240	74.240
geomy	190.750	190.750	190.750	190.750
ring_size	6.240	6.240	6.240	6.240
volt	1.320	1.650	1.200	1.080
temp	-55.000	125.000	25.000	-55.000
tcyc	1.307	1.227	3.236	8.040
ta	1.102	0.931	2.636	6.892
tas	0.298	0.226	0.764	2.261
tah	0.000	0.003	0.013	0.034
tcs	0.215	0.195	0.645	1.875
tch	0.000	0.000	0.000	0.000
tws	0.241	0.202	0.540	1.540
twh	0.000	0.000	0.000	0.000
tds	0.130	0.109	0.391	1.312
tdh	0.089	0.084	0.000	0.000
tckh	0.175	0.165	0.355	0.712
tkcl	0.204	0.175	0.585	1.654
tckr	1.500	1.500	1.500	1.500
icap_a	0.012	0.012	0.013	0.013
icap_d	0.003	0.003	0.003	0.003
icap_clk	0.054	0.055	0.057	0.058
icap_cen	0.006	0.006	0.008	0.008
icap_wen	0.017	0.018	0.018	0.018
load_q	0.638	0.562	1.343	3.331
icc	9.78E-1	1.283	9.52E-1	8.45E-1
icc_r	9.44E-1	1.228	9.21E-1	8.17E-1

PostScript Datasheet

Default Generate

ASCII Datable updated  
ASCII Datable updated  
ASCII Datable updated  
Initialized Generic parameters to defaults  
ASCII Datable updated  
ASCII Datable updated  
Parameters have not changed since last ASCII Datable update

# Generating SRAM example (3)

ARM@CADPC20

File Utilities Help

High Speed/Density Single Port Register File Generator  
IBM CMRF85F-LPVT Process

GENERIC PARAMETERS

Instance Name: memory

Number of Words: 128

Number of Bits: 16

Frequency <MHz>: 100

Ring Width <um>: 2

Multiplexer Width: ☐ 1 ☒ 2

Word-Write Mask: ☐ on ☒ off

Top Metal Layer: ☒ m6 ☐ m5

Power Type: ☒ rings ☐ m1

Horizontal Ring Layer: ☒ m1 ☐ m2

Vertical Ring Layer: ☐ m1 ☐ m2

RELATIVE FOOTPRINT

Generate Views@CADPC20

- ☐ PostScript Datasheet
- ☐ ASCII Datable
- ☒ Verilog Model
- ☐ VHDL Model
- ☒ Synopsys Model
- ☐ PrimeTime Model
- ☐ TLF Model
- ☐ VCLEF Footprint
- ☐ GDSII Layout
- ☐ LVS Netlist

All None Invert

Default Generate Close

Views

PostScript Datasheet

Default Generate

1p32v...	ff_1p65v...	tt_1p2v_25c...	ss_1p08v...
240	74.240	74.240	74.240
0.750	190.750	190.750	190.750
40	6.240	6.240	6.240
20	1.650	1.200	1.080
5.000	125.000	25.000	-55.000
07	1.227	3.236	8.040
02	0.931	2.636	6.892
98	0.226	0.764	2.261
00	0.003	0.013	0.034
15	0.195	0.645	1.875
00	0.000	0.000	0.000
41	0.202	0.540	1.540
00	0.000	0.000	0.000
30	0.109	0.391	1.312
89	0.084	0.000	0.000
75	0.165	0.355	0.712
0204	0.175	0.585	1.654
1500	1.500	1.500	1.500
0.012	0.012	0.013	0.013
0.003	0.003	0.003	0.003
0.054	0.055	0.057	0.058
0.006	0.006	0.008	0.008
0.017	0.018	0.018	0.018
0.638	0.562	1.343	3.331
9.78E-1	1.283	9.52E-1	8.45E-1
0.44E-1	1.228	0.21E-1	8.17E-1

ASCII Datable updated  
Parameters have not changed since last ASCII Datable update  
command: /user5/fall15/sjk2206/6321/mem\_comp/mem/rf1shd/bin/rf1shd verilog -instname memory -words 128 -bits 16 -frequency 100 -ring\_w  
Verilog Model generator succeeded, created:  
memory.v  
command: /user5/fall15/sjk2206/6321/mem\_comp/mem/rf1shd/bin/rf1shd synopsys -instname memory -words 128 -bits 16 -frequency 100 -ring  
Synopsys Model generator succeeded, created:  
memory\_ff\_1p32v\_m55c\_syn.lib



# Outputs You Need

---

Instantiation name

```
ACI.log
memory_ff_1p32v_m55c.data
memory_ff_1p32v_m55c_syn.lib
memory_ff_1p65v_125c.data
memory_ff_1p65v_125c_syn.lib
memory_mod
memory_ss_1p08v_m55c.data
memory_ss_1p08v_m55c_syn.lib
memory_tt_1p2v_25c.data
memory_tt_1p2v_25c_syn.lib
memory.v
```

Please use 1.2V 25°C as your library file for assignment 4, you can try different libraries later if you want

.v can be used in Modelsim simulation

# Sram Ports

Figure 3-1. Single-Port SRAM Basic Pins

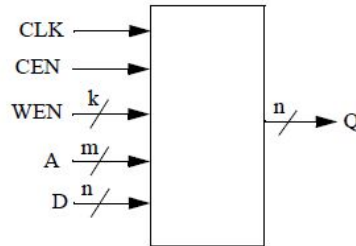


Table 3-2 provides the single-port SRAM generator pin descriptions.

Table 3-2. Pin Descriptions for Single-Port SRAM Generators

Name	Type	Description
<b>Basic Pins</b>		
A[m-1:0]	Input	Addresses (A[0] = LSB)
D[n-1:0]	Input	Data inputs (D[0] = LSB)
CEN	Input	Chip Enable active low
WEN [*]	Input	Write Enable, active low. *If word-write mask is enabled, this becomes a bus.
CLK	Input	Clock
Q[n-1:0]	Output	Data outputs (Q[0] = LSB)
OEN	Output	Tristate Output Enable <sup>1</sup>

<sup>1</sup> The OEN pin applies to some 180nm generators.

# Detailed function description

---

**Table 3-3. Single-Port Register File Basic Functions**

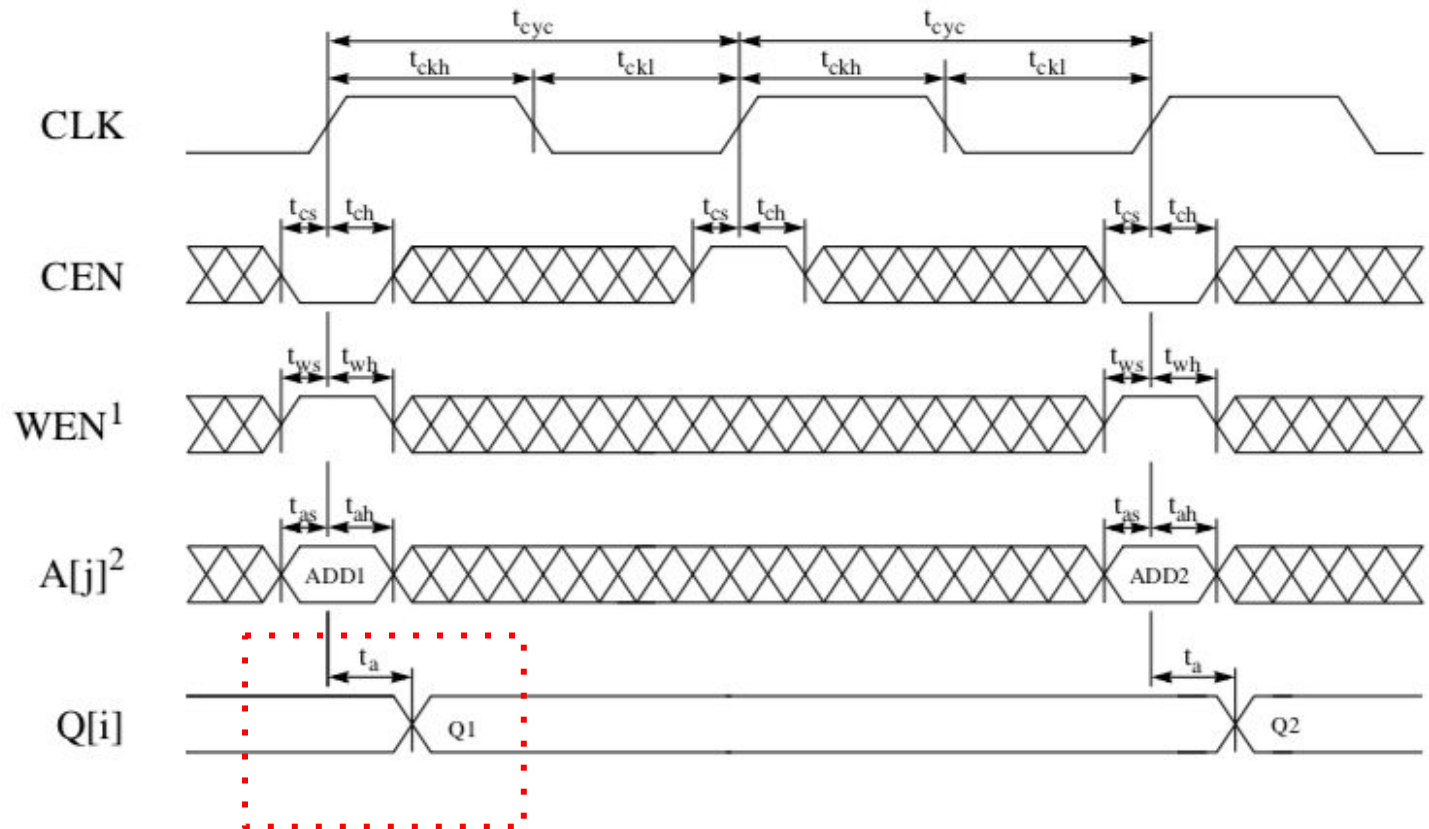
CEN	WEN[]	Data Output	Mode	Function
H	X	Last Data	Standby	Address inputs are disabled; data stored in the memory is retained, but memory cannot be accessed for new reads or writes. Data outputs remain stable.
L	L	Data In	Write	<b>Word-write:</b> Data on the data input bus, D[n-1:0], is written to the memory location specified by the address bus, A[m-1:0]; and is driven through to the data output bus Q[n-1:0] <b>Bit-write:</b> The corresponding data partition is selected by the write enables, WEN[k-1:0]; and that data is written to the memory location specified by the address bus, A[m-1:0], and is driven through to the data output bus Q[n-1:0]. Data on the data input bus, D[n-1:0], is written to the memory location specified by the address bus, A[m-1:0]; and is driven through to the data output bus Q[n-1:0]
L	H	SRAM data	Read	Data on the data output bus Q[n-1:0] is read from the memory location specified on the address bus A[m-1:0].

All the details of the sram will be in **ra2shd\_v20.pdf (Artisan User Manual)**

**Design & memory-compiler.pdf** is slides for memory compiler from ARES Lab

# Sram Read Timing Diagram

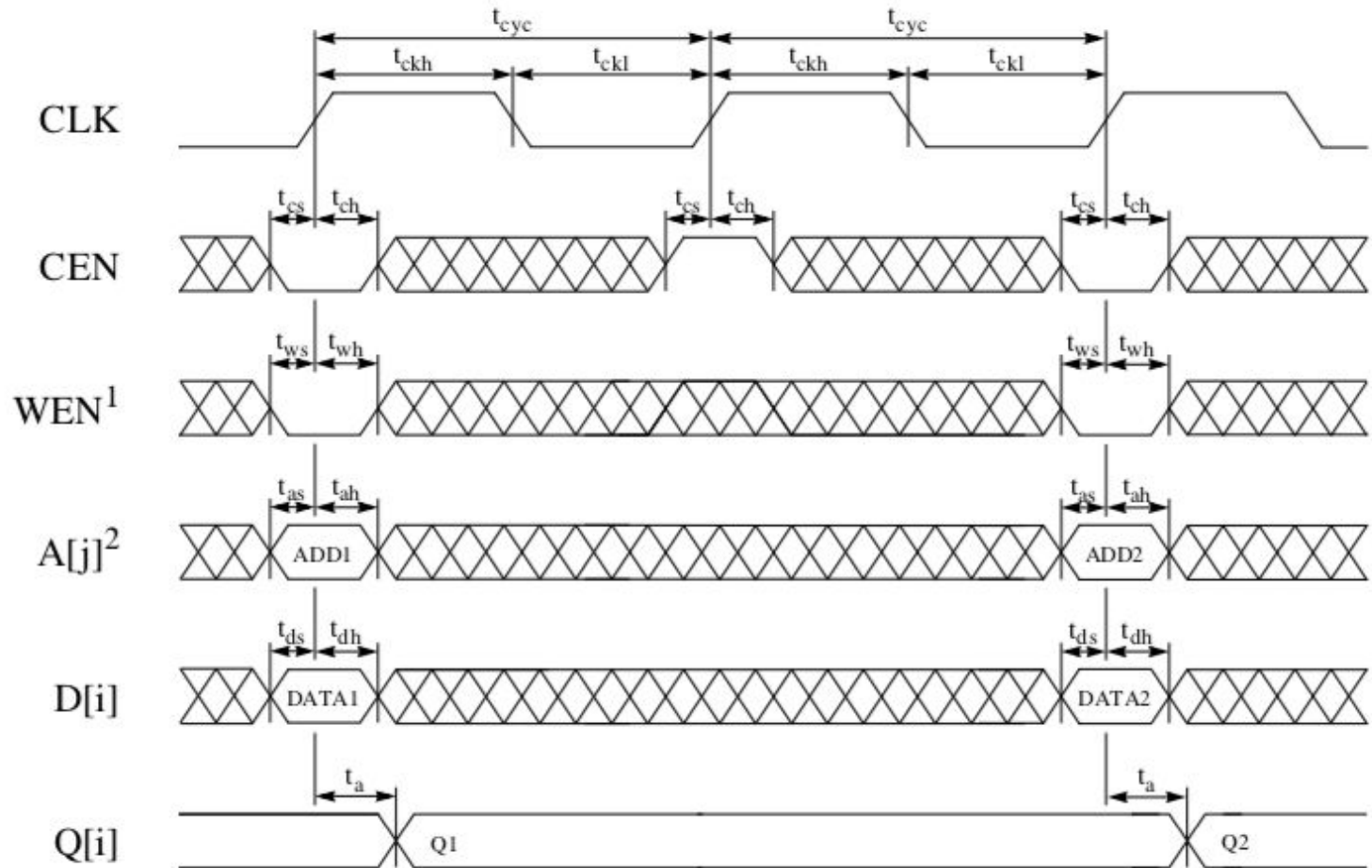
Figure 3-7. Single-Port Register File (rf1 sh, rf1 shd) Read-Cycle Timing



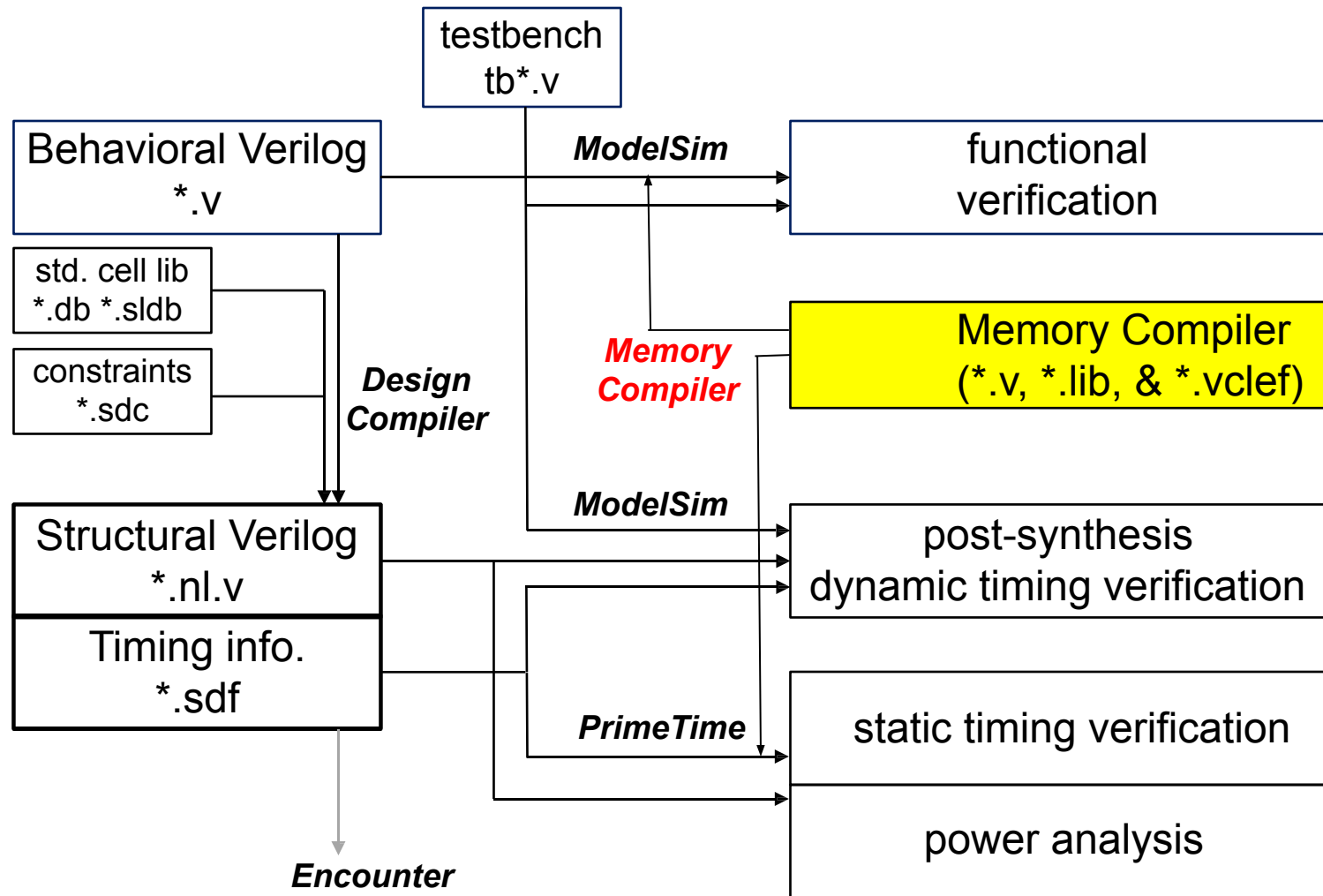
Data (Q) will be available after the rising edge of the clock (1 clock cycle delay)

# Sram Write Timing Diagram

Figure 3-8. Single-Port Register File (rf1sh, rf1shd) Write-Cycle Timing



# Summary



Automatic Placement and Routing

# Suggestions on Lab Assignment 4

---

1. Make sure you are familiar with previous labs
2. Take a look at the Memory Generator **User Manual** to learn the architecture & timing specification of SRAM.
3. First simulate the sram block you generated from memory compiler and then try to build the larger block. (***RTL on-chip memory model guide.pdf*** as reference).
4. When running DC, you do **not** need to use read\_verilog to include the [mem\_cell\_name].v generated by memory compiler and you do not need to use command to **black-box your sram**. You only need to include the correct db file. (.db file is generated by compiling .lib file through DC, codes are in **/courses/ee6321/share/4823-fall2020/lib2db/**).
5. You can use the same db file as above in your PT
6. Start early. This one will take some time.