

Introduction to Macros and

SRAM
NCTU-EE ICLab
Fall-2023

Outline

- ✓ **Section 1 – Macro**
 - ✓ **(Intellectual property, IP)**
- ✓ **Section 2 – Hard IP: Memory**
 - ✓ **Behavior**
 - ✓ **Usage**



Outline

- ✓ **Section 1 – Macro**
 - ✓ **(Intellectual property, IP)**
- ✓ **Section 2 – Hard IP: Memory**
 - ✓ **Behavior**
 - ✓ **Usage**



Introduction to Intellectual Property

✓ Intellectual Property (IP) core

- **What:** IP is a design of a logic function that specifies how the elements are interconnected
// e.g. square root
- **Why:** A designer can develop more quickly by applying IPs
- **How:** IPs may be licensed to another party
- **Soft macro(IP):** Synthesizable RTL
 - Portable and Editable
 - Unpredictable in terms of performance, timing, area, or power
 - IP protection risks
- **Firm macro(IP):** Netlist format
 - Performance optimization under a specific fabrication technology
 - Need not synthesizing (sometimes it's time wasting)
- **Hard macro(IP):** Hardware (LEF, GDS2 file format)
 - Specifies the physical pathways and wiring (proved under specific tech.)
 - Moving, rotating, flipping freedom but can't touch the interior (APR)



- ✓ **Imagine that your design is for cellphone screen processing**
 - Assume the resolution is 1920×1080 , 24 bits per pixel
 - 50M registers!!
- ✓ **Cellphone becomes large and power-consuming!**

size



power

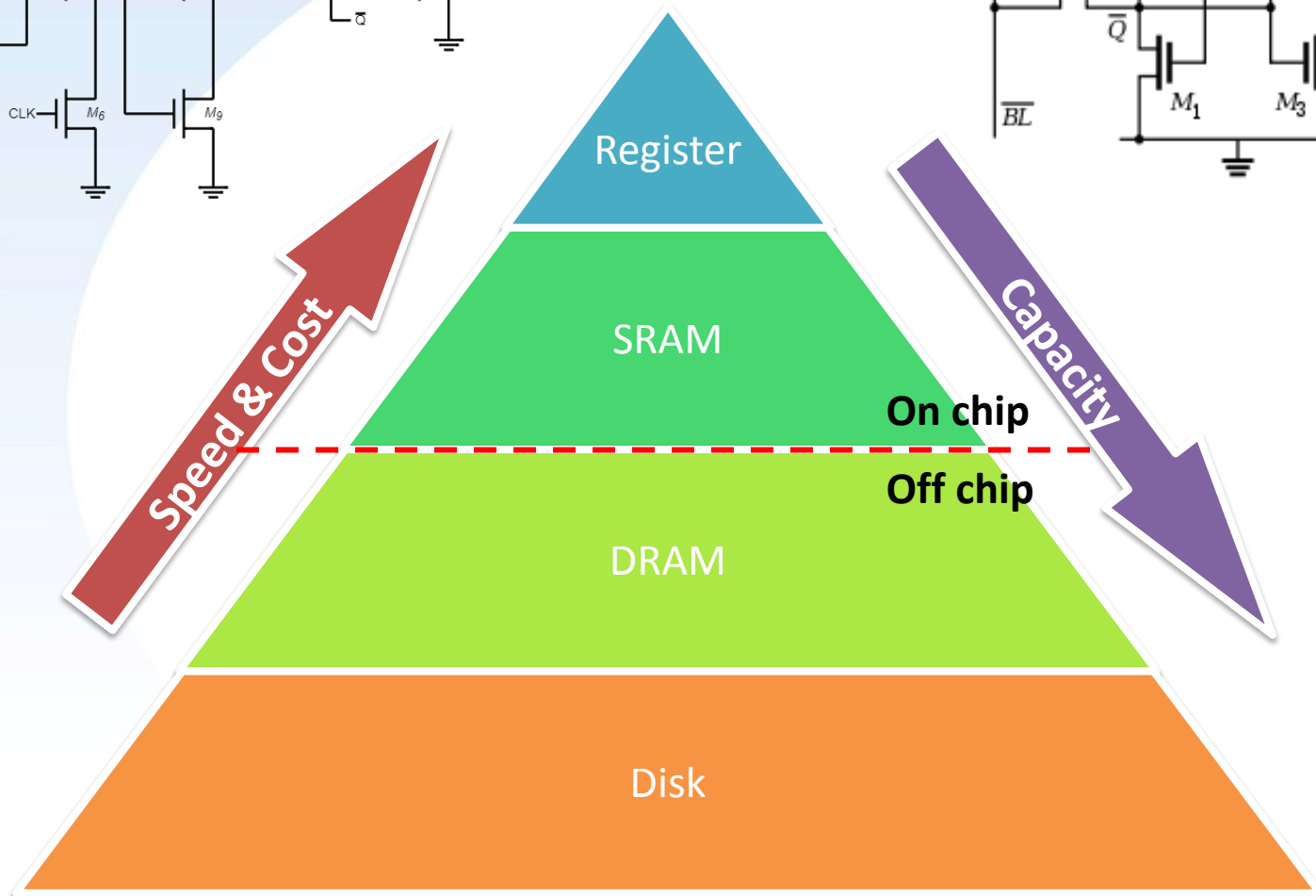
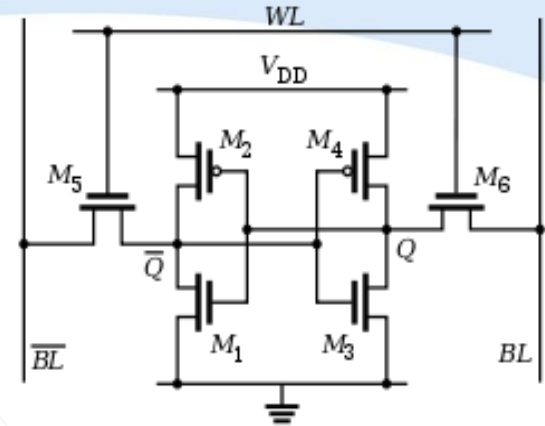
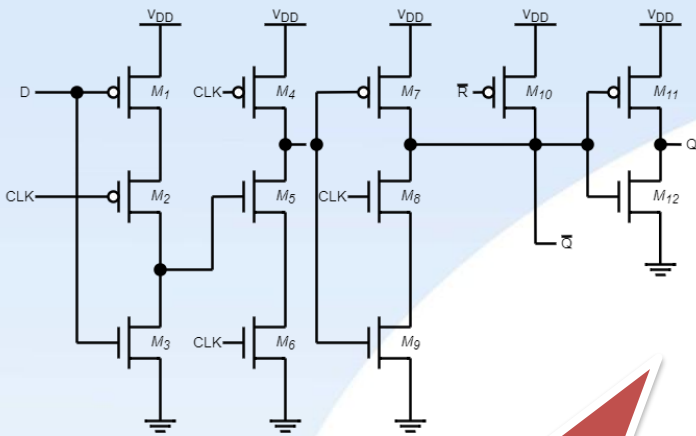


Outline

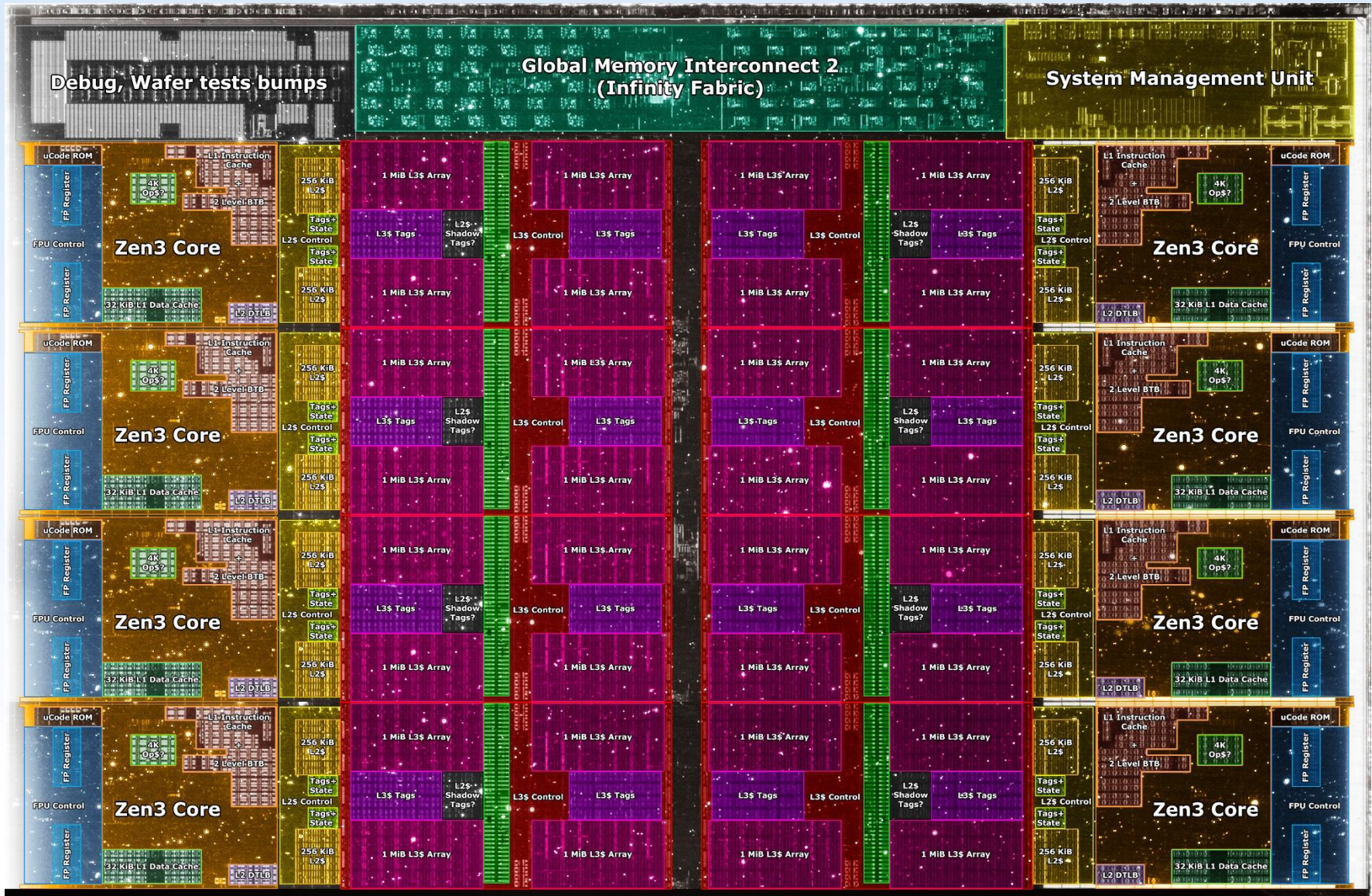
- ✓ **Section 1 – Macro**
 - ✓ **(Intellectual property, IP)**
- ✓ **Section 2 – Hard IP: Memory**
 - ✓ **Behavior**
 - ✓ **Usage**



Memory Hierarchy



AMD Ryzen ZEN3

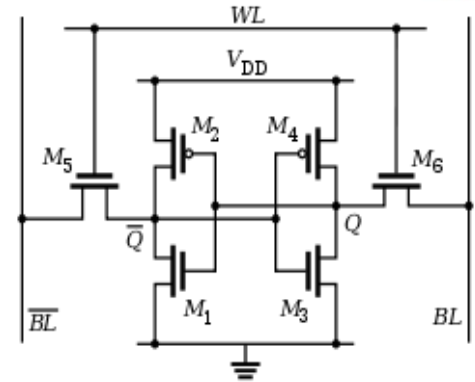


Memory

✓ SRAM

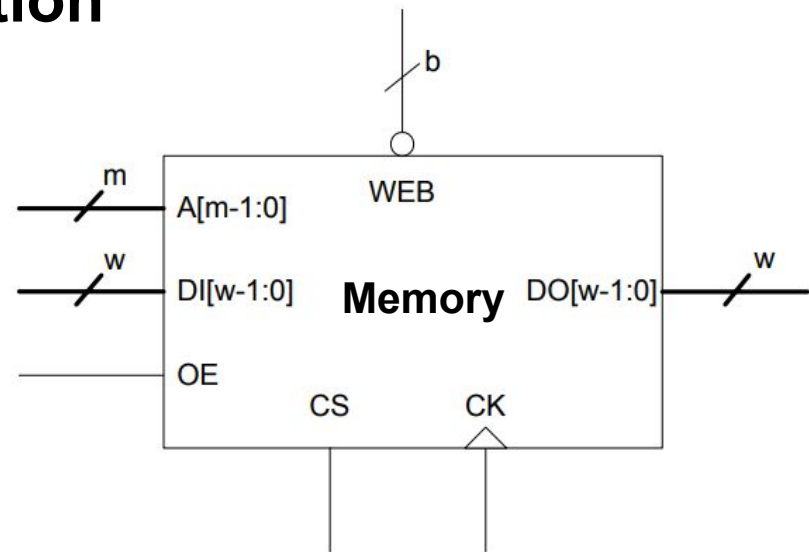
- Read and Write Data only
- Memory has less area than register
- Memory is slower than register
- Only one address can be accessed in the same time (single port SRAM vs. dual port)

6T SRAM



✓ Single port SRAM I/O Description

Pin	Description
A[9:0]	Address(A[0]=LSB)
DI[7:0]	Data input(DI[0]=LSB)
CK	Clock input
CS	Chip Enable
OE	Write Enable
WEB	Output Enable
DO[7:0]	Data Output(Q[0]=LSB)

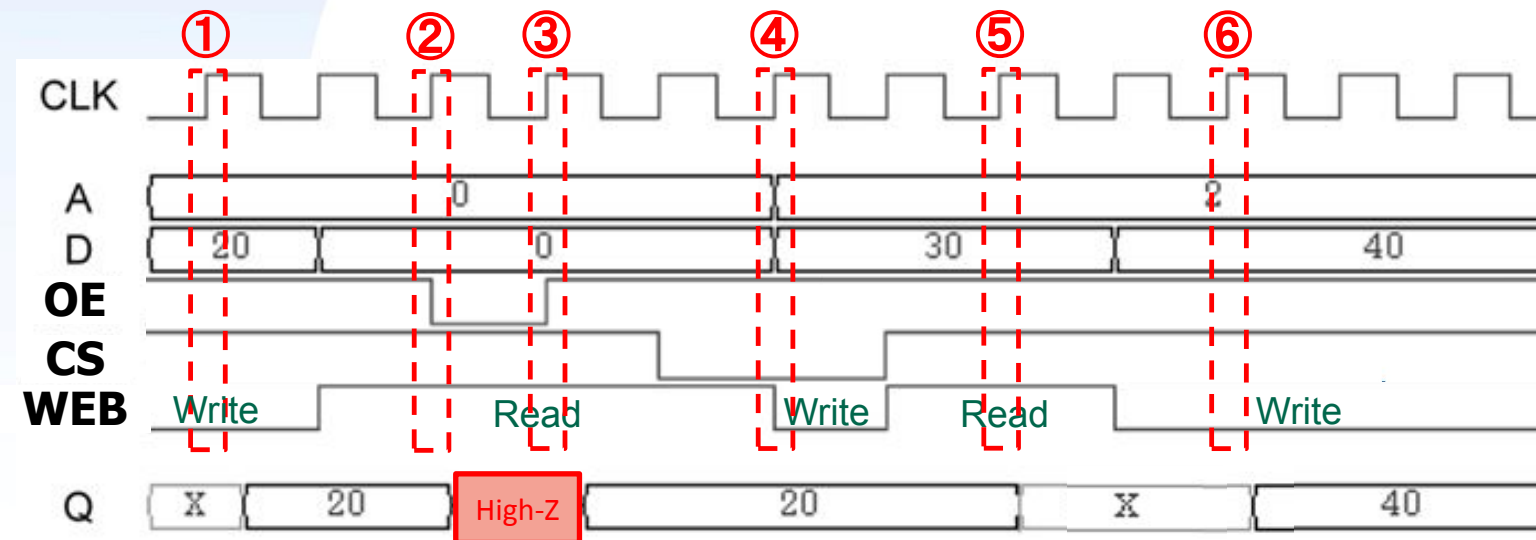


SRAM Logic Table

- OE is a tri-state buffer
- Considering CLK skew, Enable Chip at least one cycle before use

SRAM Logic Table

CS	WEB	OE	Data Out	Mode	Function
X	X	L	Z	High-Z	The data output bus Q[n-1:0] is placed in a high impedance state. Other memory operations are unaffected.
L	X	H	Last Data	Standby	Address inputs are disabled; data stored in the memory is retained, but the memory cannot be accessed for new reads or writes. Data outputs remain stable.
H	H	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].
H	L	H	Data In	Write	Data on the data input bus D[n-1:0] is written to the memory location specified on the address bus A[m-1:0], and driven through to the data output bus Q[n-1:0].

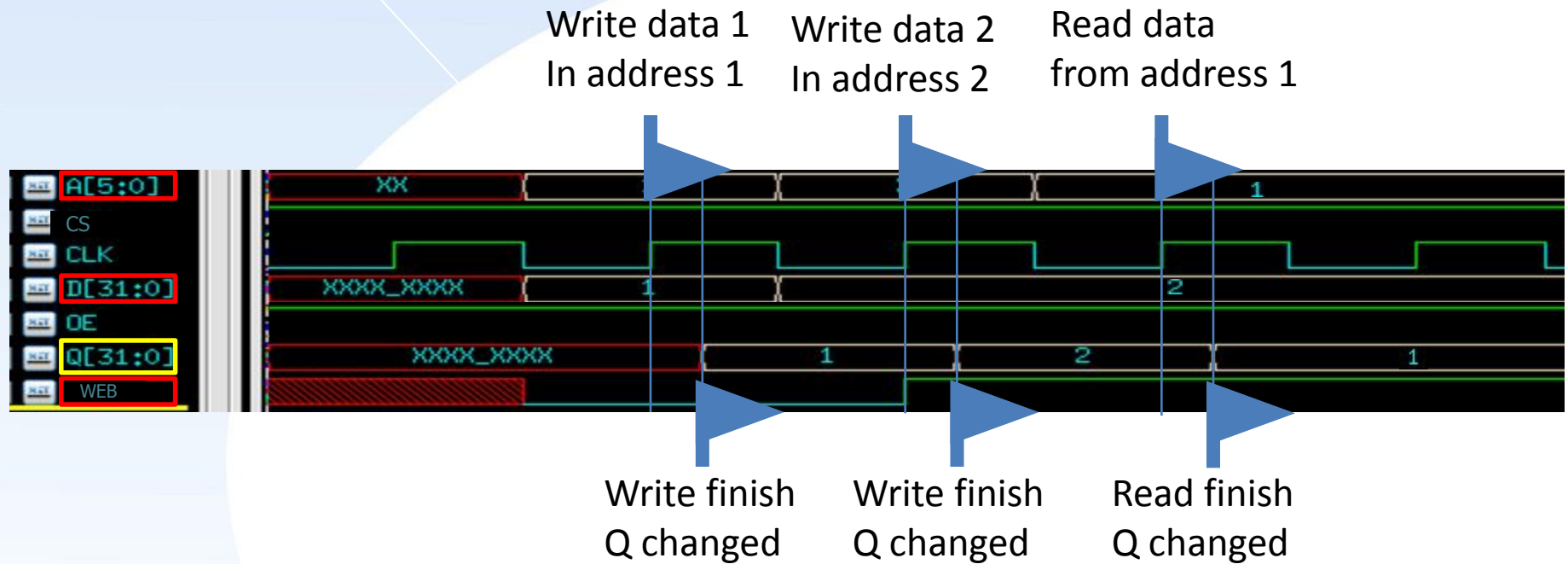


SRAM

Address	Data
0	20
1	X
2	40
⋮	⋮



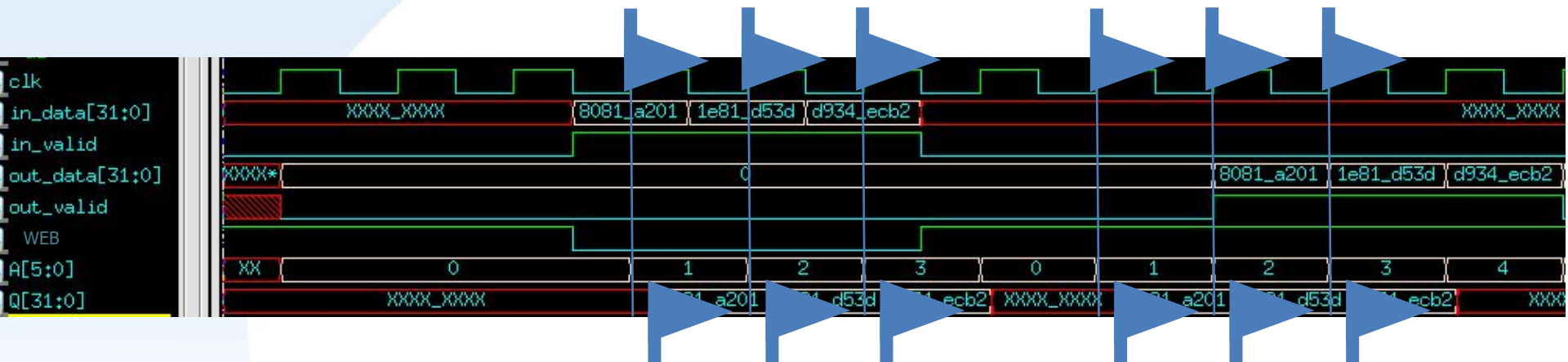
Signal example



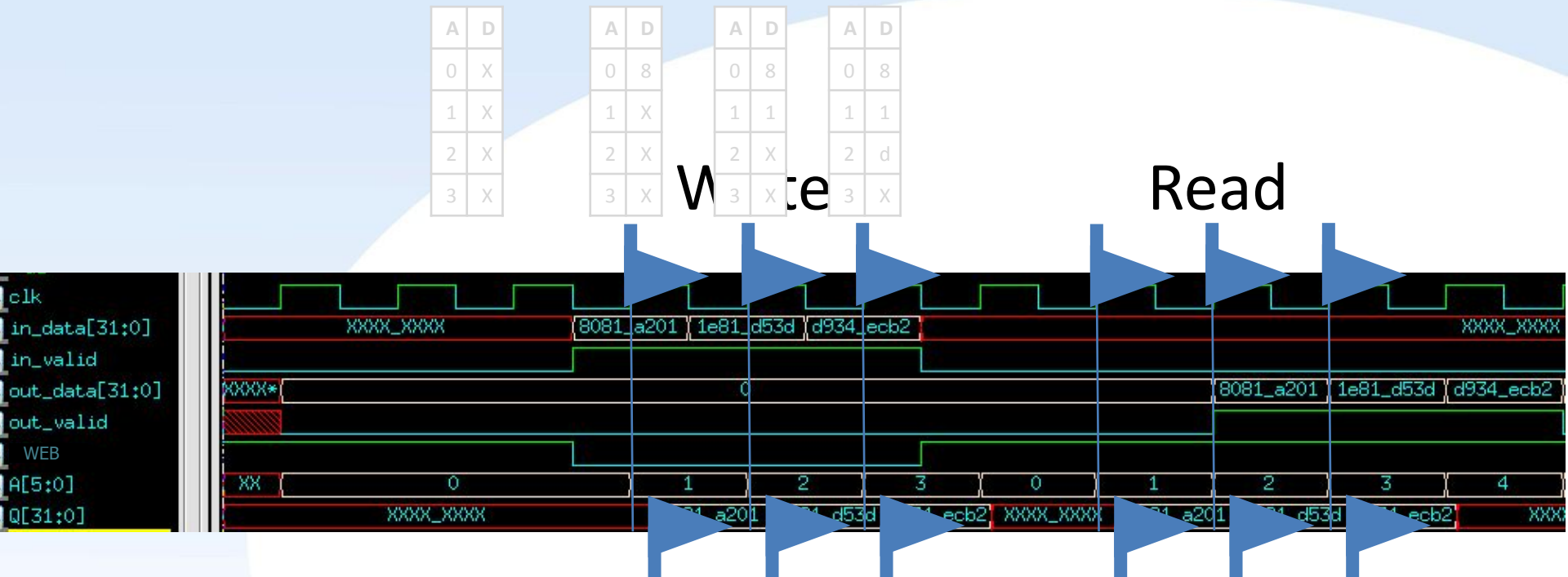
WEB:

- 0 -> write
- 1 -> read

Appendix-Write and read in order

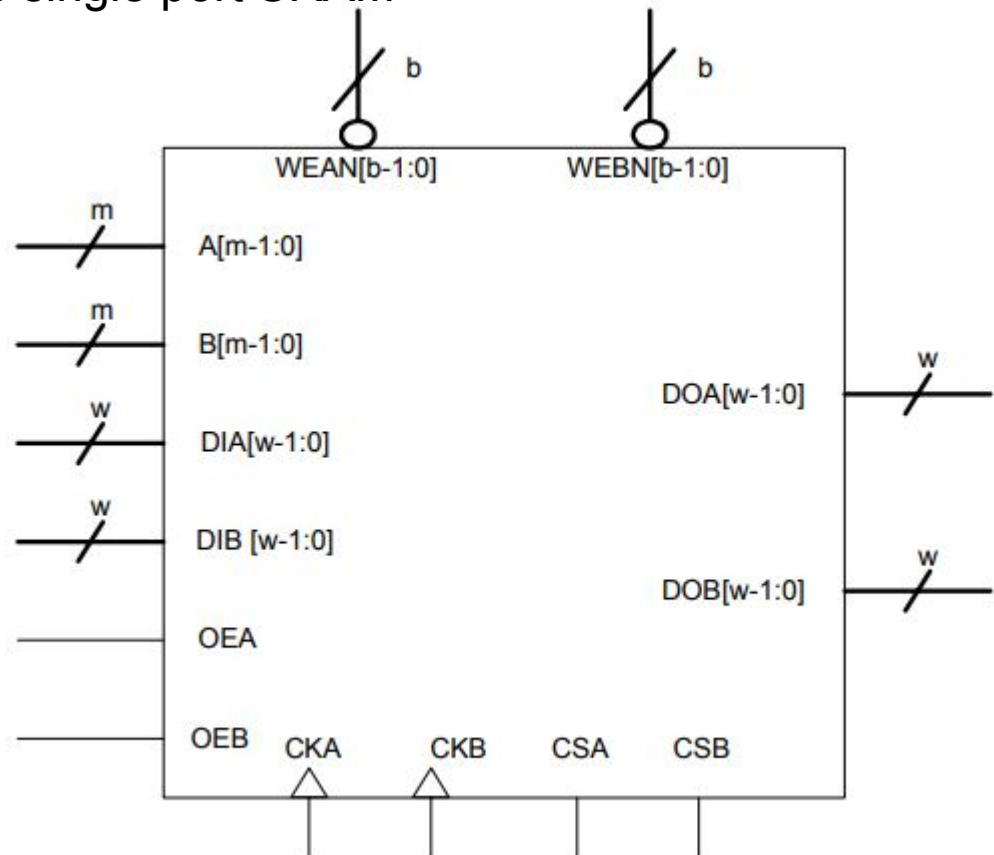


Appendix-Write and read in order



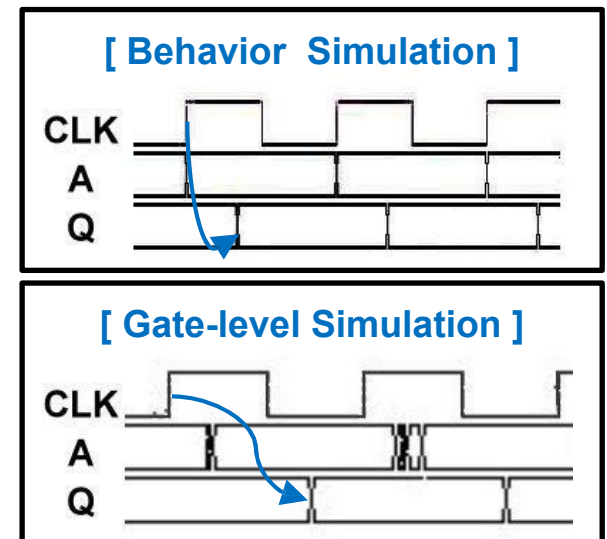
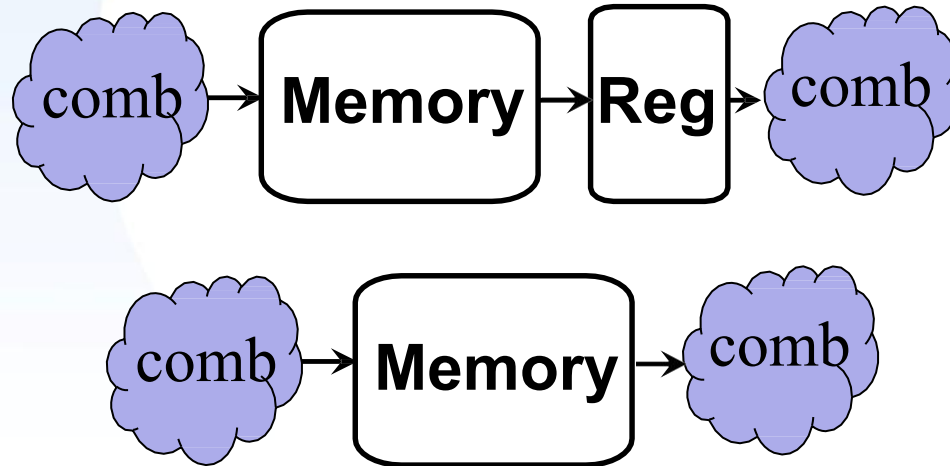
Dual port SRAM

- Same behavior as the single port SRAM



Design Tips

- ✓ **To avoid critical path causing timing violation**
 - Add registers after the hard macro
 - Use enable signal to control output register to avoid reading unknown value
- ✓ **If a memory macro is used in your design, the timescale should be set according to the timescale specified by memory file**
- ✓ **Be aware of features and characteristics of hard macro before you use it in your design**



Memory generation example

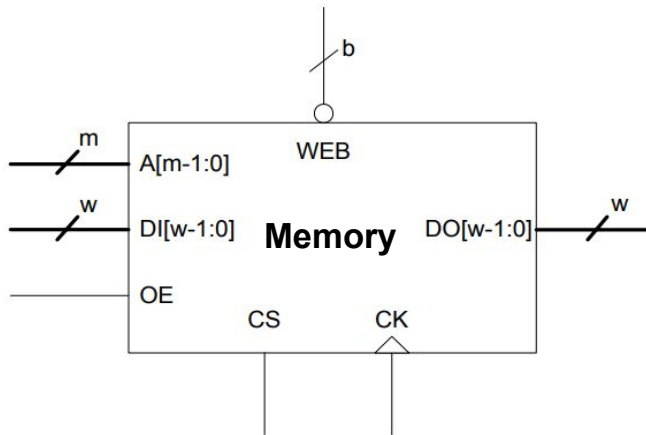
Example :

Number of Words : 600

Number of Bits : 8

8 bits	Entry 0
8 bits	Entry 1
8 bits	Entry 2
8 bits	Entry 4
⋮	⋮
8 bits	Entry 599

1. How many bits of data pins (D and Q) are needed?
2. How many bits of address are needed?



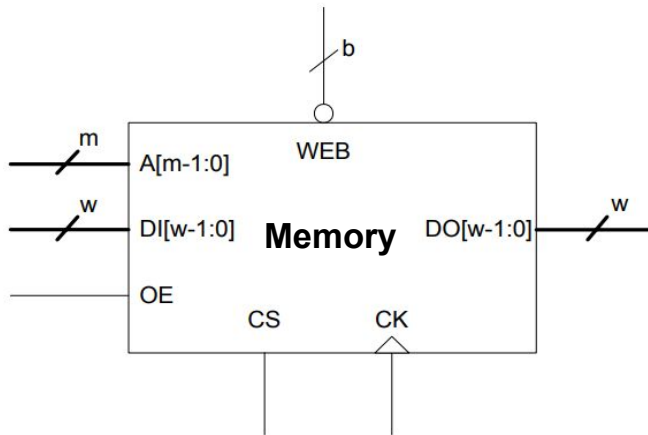
Memory generation example

Example :

Number of Words : 600

Number of Bits : 8

1. How many bits of data pins (D and Q) are needed?
2. How many bits of address are needed?



● Answer:

- **DI [7:0]**
- **QO [7:0]**
- **A [9:0]**



Memory Compiler

NCTU-EE ICLab
Fall-2022

 **OASIS
LAB** Lecturer: Jia Fu, Tsao

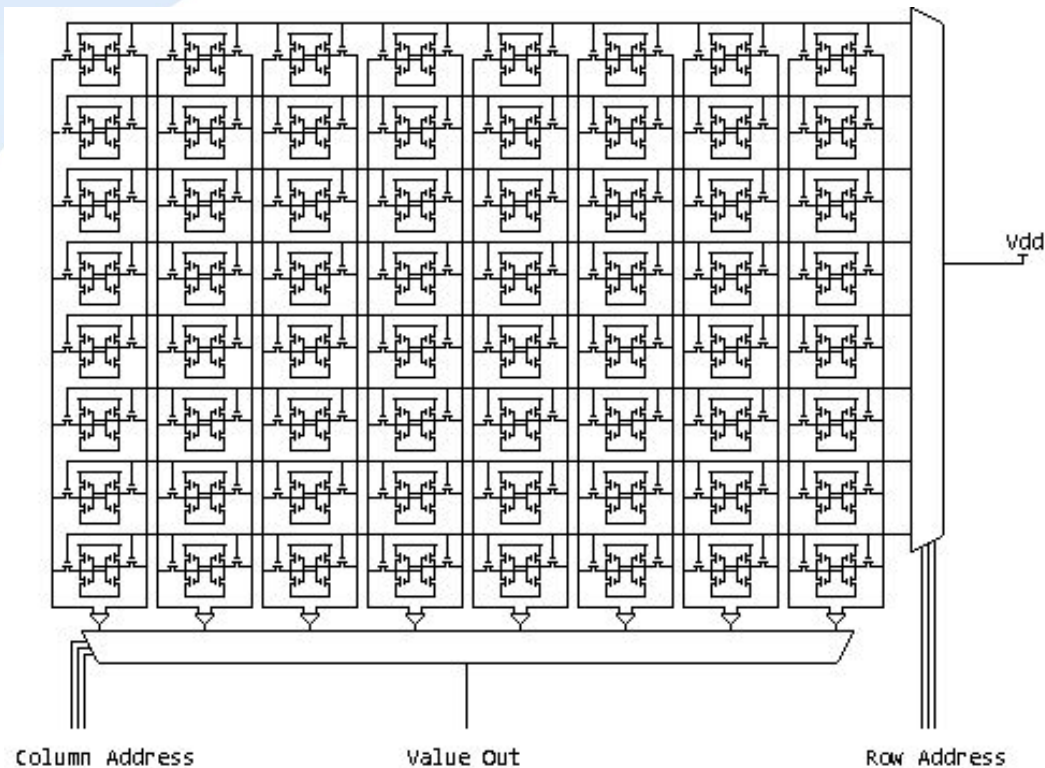
Outline

✓ **Section 1 – GUI**



Memory Compiler

Memory Architectures



Memory Compiler Parameter

- **Instance Name** : memory name
- **Number of Words** : number of entry for the designed memory
- **Number of Bits** : number of bits for every entry
- **Multiplexer Width** : 1-to-1, 2-to-1, 4-to-1, 8-to-1, 16-to-1 multiplexer



Memory Compiler GUI steps

- ✓ Step 1. >> cd Lab05/Exercise/Memory/ftclib_200901.2.1/
- ✓ Step 2. Open memaker.env file
- ✓ Step 3. Replace the path after ' setenv FTC ' with your own absolute path

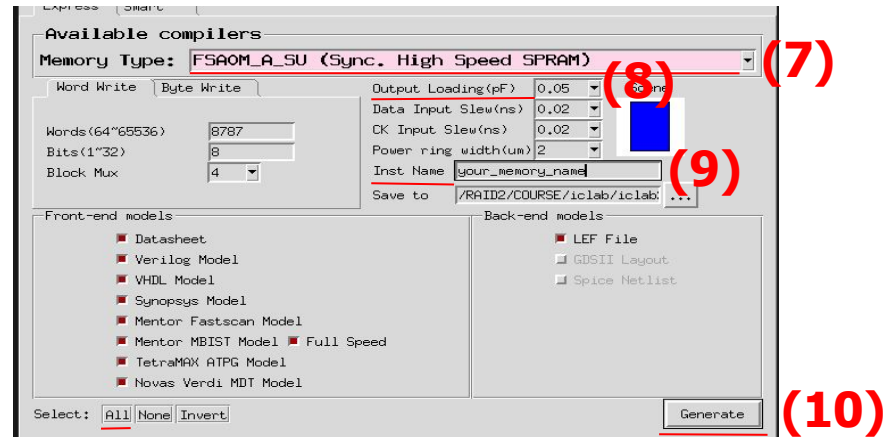
```
##### Edit this #####  
setenv FTC /user/DSD/dk/DesignKit/UMC018_Faraday/UMC018_Faraday_v1.0/CBDK018_UMC_Faraday_v1.1/CIC/Memory/ftclib_200901.2.1
```

- ✓ Step 4. >> source memaker.env
- ✓ Step 5. >> cd EXE/



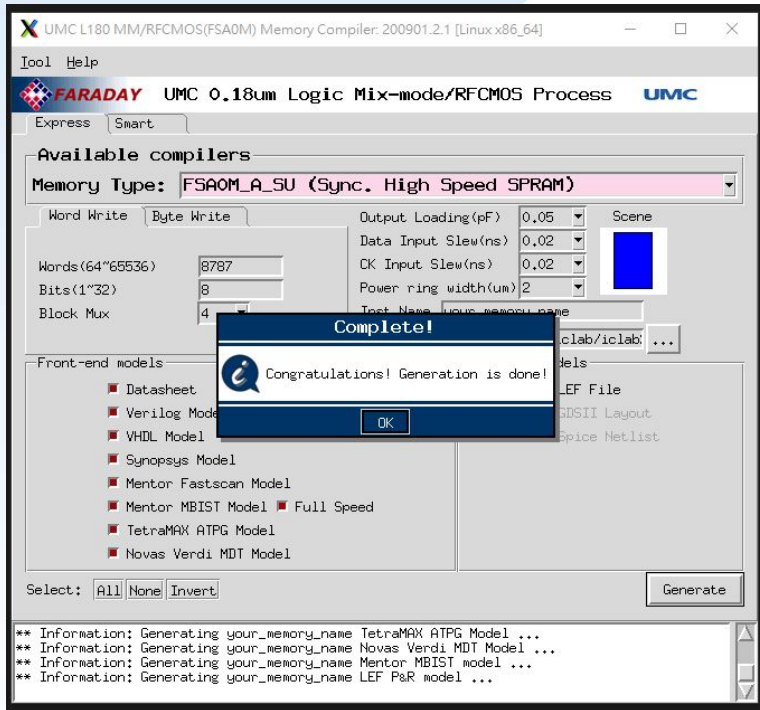
Memory Compiler GUI steps

- ✓ Step 6. >> memaker
- ✓ Step 7. Choose your memory (single port/dual port)
- ✓ Step 8. Set output loading to 0.05 pF
- ✓ Step 9. You can change your memory's name



Memory Compiler GUI steps

✓ Step 10. Click All and generate the memory



```
-rw-r--r-- 1 iclab200 iclab 8140 Oct 13 16:40 your_name.ds
-rw-r--r-- 1 iclab200 iclab 2937 Oct 13 16:40 your_name.fastscan
-rw-r--r-- 1 iclab200 iclab 96552 Oct 13 16:40 your_name.lef
-rw-r--r-- 1 iclab200 iclab 3781 Oct 13 16:40 your_name.mbist
-rw-r--r-- 1 iclab200 iclab 2291 Oct 13 16:40 your_name.mdt
-rw-r--r-- 1 iclab200 iclab 3304 Oct 13 16:40 your_name.tmax
-rw-r--r-- 1 iclab200 iclab 25554 Oct 13 16:40 your_name.v
-rw-r--r-- 1 iclab200 iclab 46618 Oct 13 16:40 your_name.vhd
-rw-r--r-- 1 iclab200 iclab 90421 Oct 13 16:40 your_name_BC.lib
-rw-r--r-- 1 iclab200 iclab 90413 Oct 13 16:40 your_name_TC.lib
-rw-r--r-- 1 iclab200 iclab 90420 Oct 13 16:40 your_name_WC.lib
```


Memory Compiler GUI steps

- ✓ Step 11. >> lc_shell
- ✓ Step 12. >> read_lib your_name_WC.lib
- ✓ Step 13. >>
write_lib your_name_WC -output your_name_WC.db
- ✓ Step 14. Edit your filelist.f

```
TESTBED.v  
../04_MEM/MYMEMORY.v  
../04_MEM/MEMORY2.v
```



Memory Compiler GUI steps

- ✓ Step 15. Put .db and .v file into your 04_MEM file
- ✓ Step 16. Open 02_SYN/.synopsys_dc.setup and add your .db file into the link_library and target_library

```
set company "iclab"
set desinger "Student"

set search_path      " ./ \
    ../01_RTL        \
    ../04_MEM        \
    ~iclabTA01/UMC018_CBDK/CIC/SynopsysDC/db/ \
    ~iclabTA01/UMC018_CBDK/CIC/Sdb/          \
    /usr/cad/synopsys/synthesis/cur/libraries/syn/ \
    /usr/cad/synopsys/synthesis/cur/dw "

set target_library    " fsa0m_a_generic_core_sslp62v125c.db \
    fsa0m_a_generic_core_fflp98vm40c.db \
    fsa0m_a_t33_generic_io_sslp62v125c.db \
    fsa0m_a_t33_generic_io_ttlp8v25c.db"

set link_library      " * $target_library dw_foundation.sldb standard.sldb "
set symbol_library    " *.sdb "
set synthetic_library " dw_foundation.sldb "

set verilogout_no_tri true
set hdlin_enable_presto_for_vhdl "TRUE"
set sh_enable_line_editing true
history keep 100
alias h history
```

Add your .db file here



Remind!

- ✓ When using IP, information in lib file belong to certain module name, so **modifying module name in v file is forbidden.**

```
module MEMORY2 (A0,A1,A2,A3,A4,A5,A6,A7,A8,DO0,DO1,DO2,DO3,DO4,
                DO5,DO6,DO7,DI0,DI1,DI2,DI3,DI4,DI5,DI6,DI7,
                CK,WEB,OE, CS);

    `define      TRUE                (1'b1)
    `define      FALSE              (1'b0)

    parameter    SYN_CS              = `TRUE;
    parameter    NO_SER_TOH         = `TRUE;
    parameter    AddressSize        = 9;
    parameter    Bits               = 8;
    parameter    Words              = 400;
    parameter    Bytes              = 1;
    parameter    AspectRatio        = 1;
    parameter    TOH                = (78:112:185);

    output       DO0,DO1,DO2,DO3,DO4,DO5,DO6,DO7;
    input        DI0,DI1,DI2,DI3,DI4,DI5,DI6,DI7;
    input        A0,A1,A2,A3,A4,A5,A6,A7,A8;
    input        WEB;
    input        CK;
    input        CS;
    input        OE;
```

can not modify

