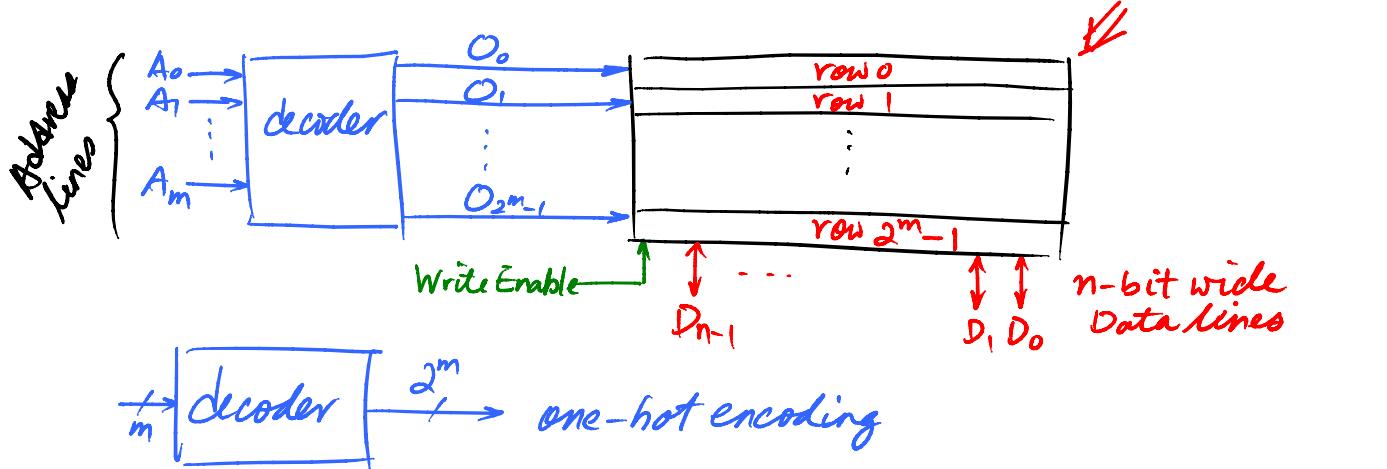


B.9 Memory and SRAM (static random access memory)

→ all memory units consist of two parts.



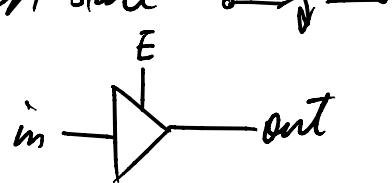
e.g. 3-bit address lines

A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

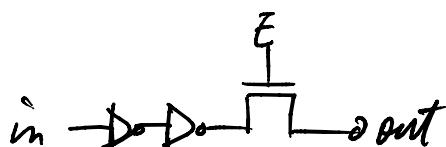
decoder output
you can address (pointing)
to 8 rows in the memory bank
a typical processor has 32
address lines, the addressable
rows are $2^{32} = 4,294,967,296$ (4G)

Memory cell array = storage cells

tri-state

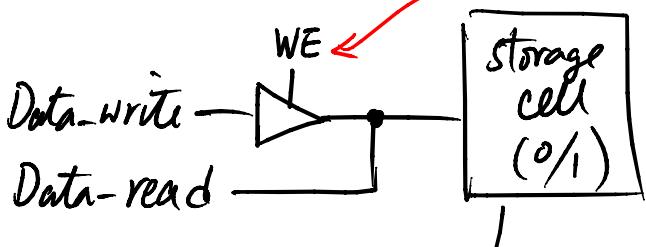


tri-state (logic values. 0, 1, Z) ↗ high impedance (no connection)



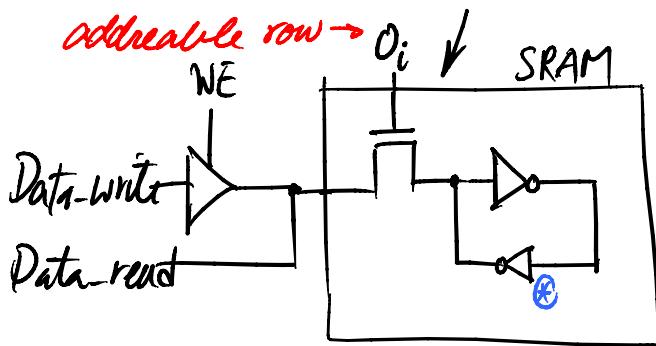
E in	out
0 0/1	Z (no connection)
1 0	0
1 1	1

tri-state in action

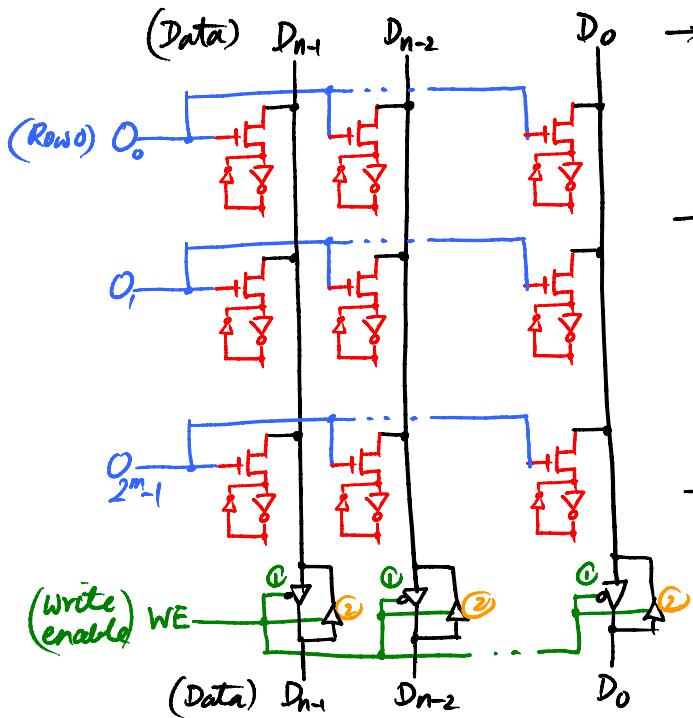


write-enable

if $WE=1$, Data-write gets written into the storage cell
if $WE=0$, Data in the storage cell can be read on Data-read

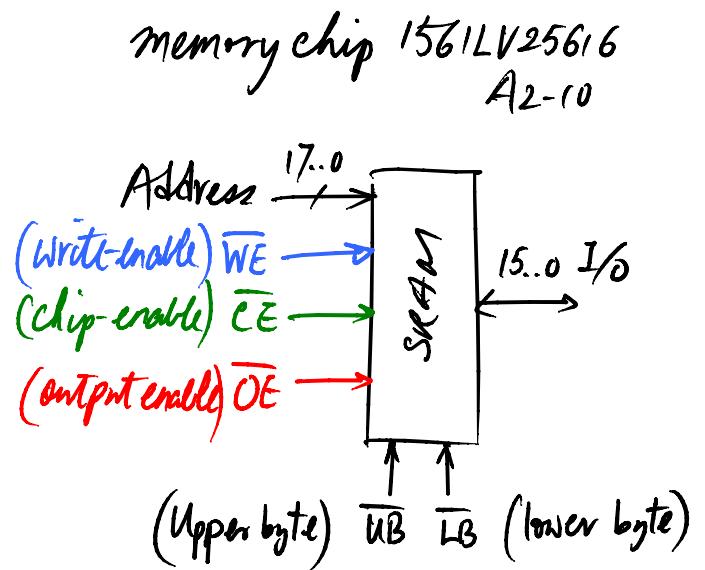
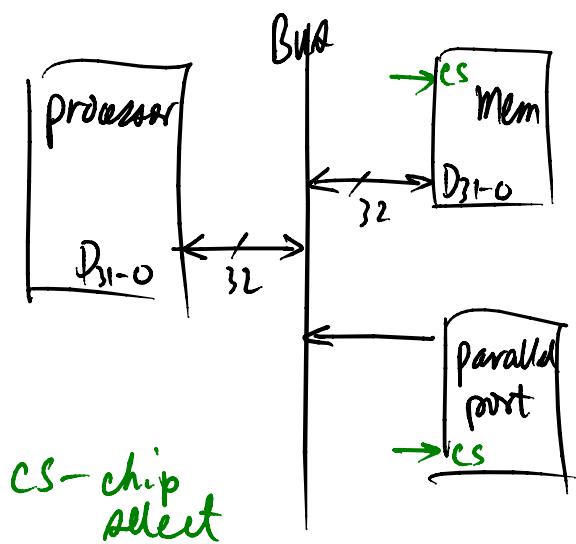


④ the "weaker" inverter so that data can be written into the storage cell (i.e. to be able to change value stored in the cell)



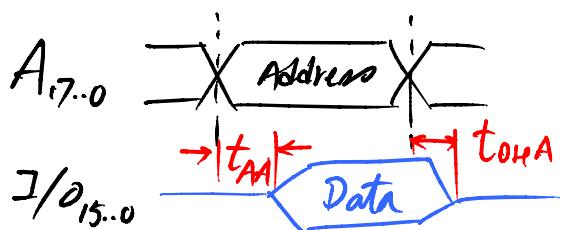
- to access row(i) in the memory array, you need to address it through ($O_i = 1$)
- if you wish to read what's stored in row(i) in the memory array, you need to set ($WE = 0$), which activates tri-state ①, hence allowing Data to be read on the Data lines
- if you wish to write new data into the memory array, you'll set ($WE = 1$) which activates tri-state ② allowing Data to be written into the memory array.

What do we need tri-state buffers?



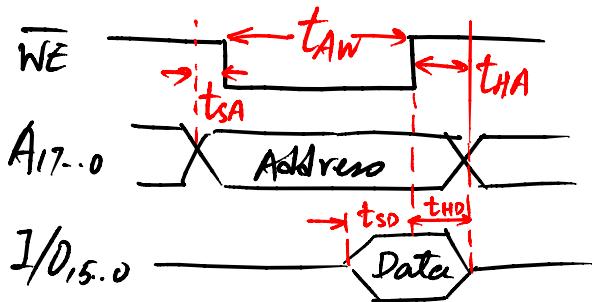
To use the chip, some timing requirements must be followed.

Read a row of data $\overline{CE} = \overline{OE} = 0, \overline{WE} = 1$



t_{AA} - address access time (10 ns)
 t_{OHA} - output hold address

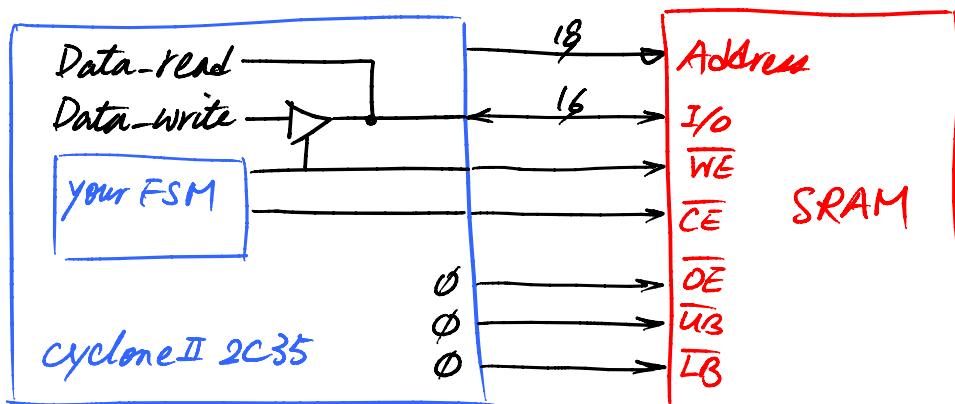
Write a row of data ($\overline{CE} = 0, \overline{WE} = 0$)



t_{SA} - setup address time (1 ns)
 t_{AW} - address width time (8 ns)
 t_{OHA} - hold address (0 ns)
 t_{SD} - setup data time (6 ns)
 t_{HD} - hold data time (0 ns)

Connection of SRAM chip to the FPGA

- each pin on the SRAM chip is connected to the FPGA chip
- each pin has a name like **SRAM-DQ**, **SRAM-Address [17..0]**, **SRAM-WE-N**



How to describe Tri-state buffer (\rightarrow) in verilog?

inout [15:0] SRAM-DQ;

assign Data-read = SRAM-DQ;

assign SRAM-DQ = (**SRAM-WE-N** == 0)? Data-write : 16'bZ;

"high impedance state"

SRAM (Static Random Access Memory)



DRAM (Dynamic Random Access Memory)

memory cell →

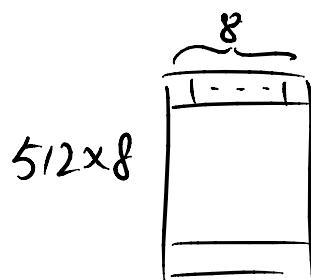
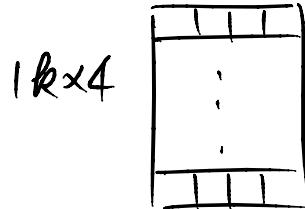
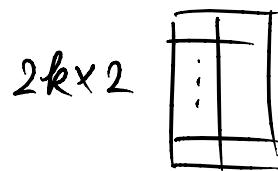
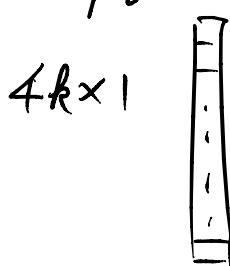


capacitor

Using memory inside the FPGA chip

(Cyclone II 2C35) memory block is called M4K (4096 cells)

Configure it into different aspect ratios



Quartus II, use mega wizard to configure

SRAM on DE2 board vs SRAM we talked about

2 differences

1. DataIn and Dataout are separate wires (no tri-state needed)
2. Address, DataIn, and write are all stored in registers (FF inside the M4K block)

