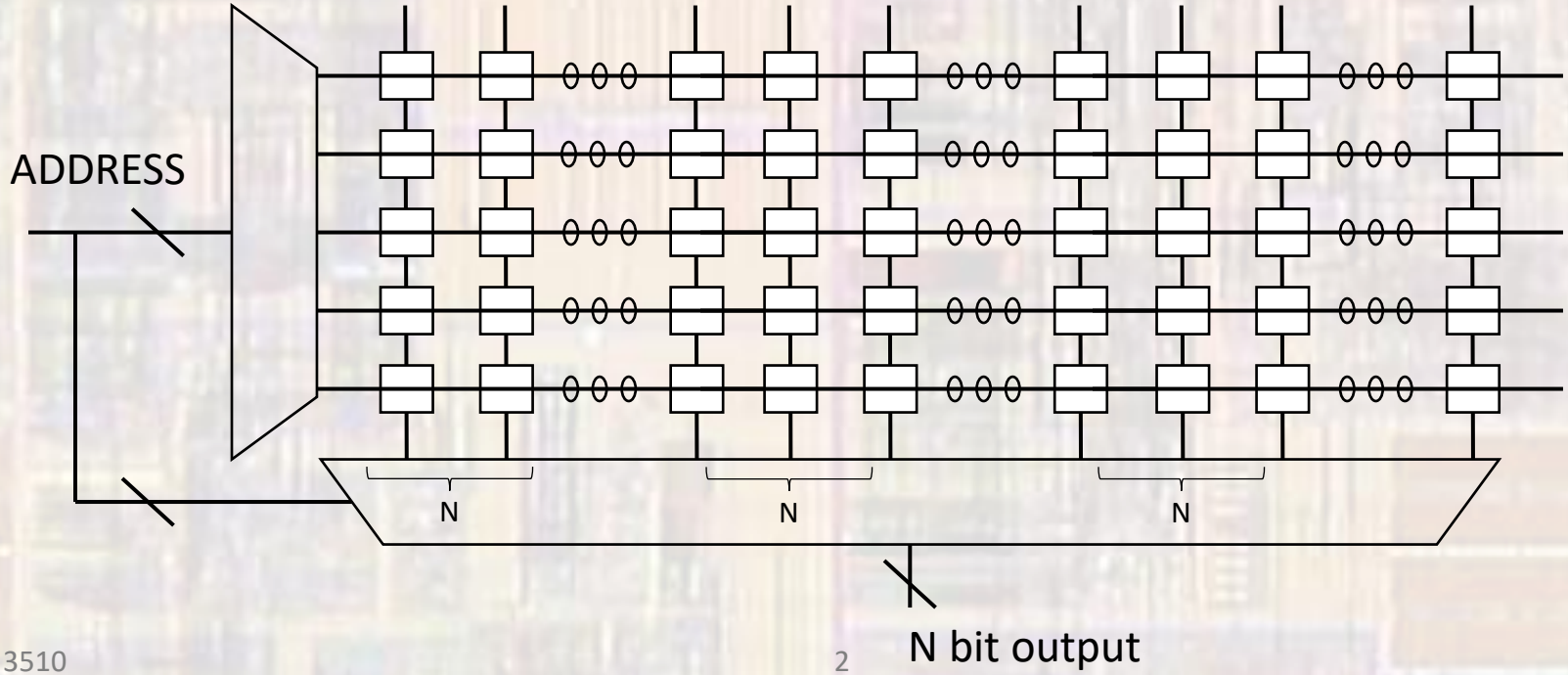
SRAM Model

Last updated 10/6/26

SRAM Model

- General Memory Topology
  - Array of n bit cells
  - Row decoder chooses 1 row
  - Column decoder chooses one N bit ‘column’
    - 1,4,8,16,32,64,128,… bits/column



© tj

SRAM Model

- Memory Descriptive Terminology
  - 16Mb in a x4
    - 16Mb total memory, each memory address provides 4 bits
  - 16Mb x 4
    - 64Mb total memory, each memory address provides 4 bits
    - Could be configured as 4 16MB memories, each providing 1b to the output

© tj

SRAM Model

- Example – 16Mb in a x4 configuration

x4 means each column is 4 bits

or each address points to 4 bits

16Mb → 16,777,216 bits

16Mb in a x4 configuration → 4 bits / address → 4,194,304 - individual addresses

4,194,304 addresses → 22 address bits

x4 means 4 bit cells for every column

Assuming a square memory array and a square bit cell → 4 times as many rows as columns

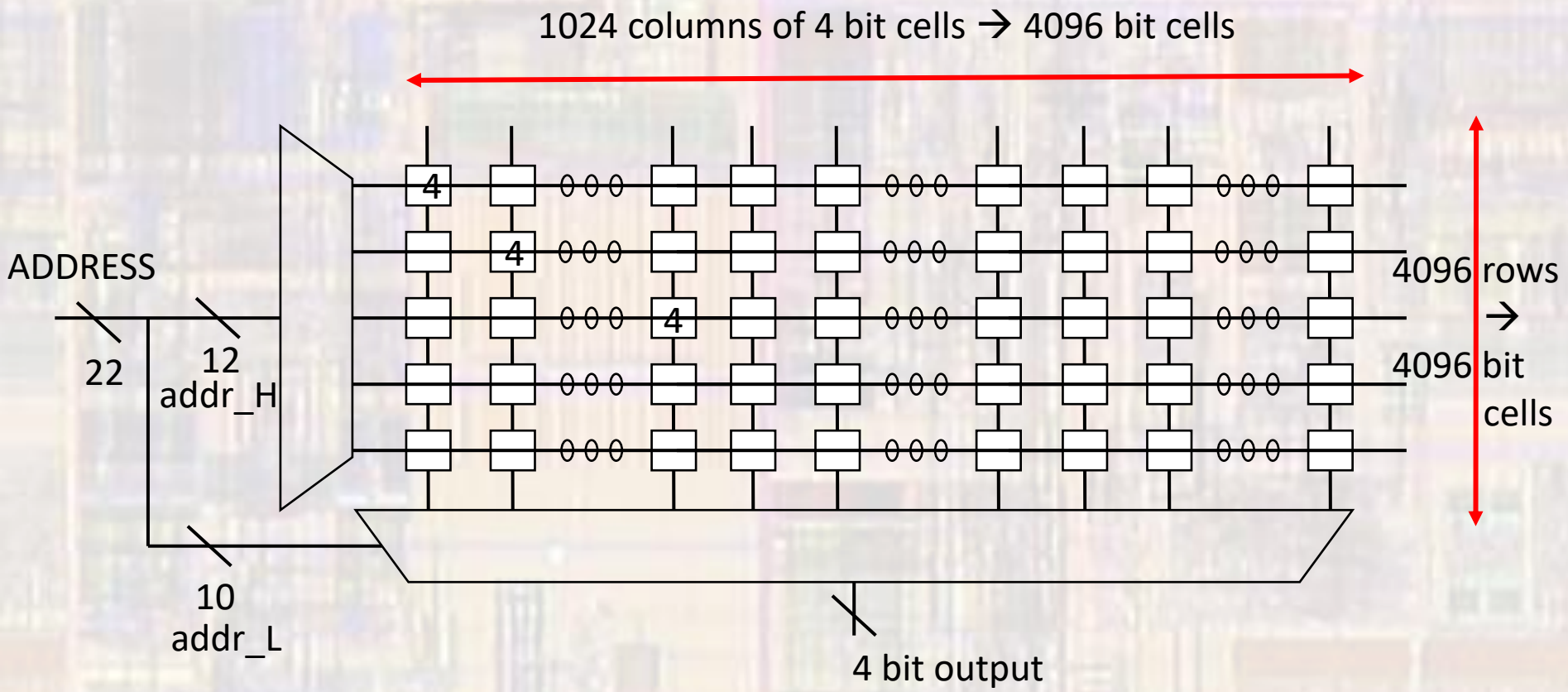
4 times as many rows as columns → 2 more row address bits than column address bits

22 address bits → 12 bits for row addresses and 10 bits for column addresses

© tj

SRAM Model

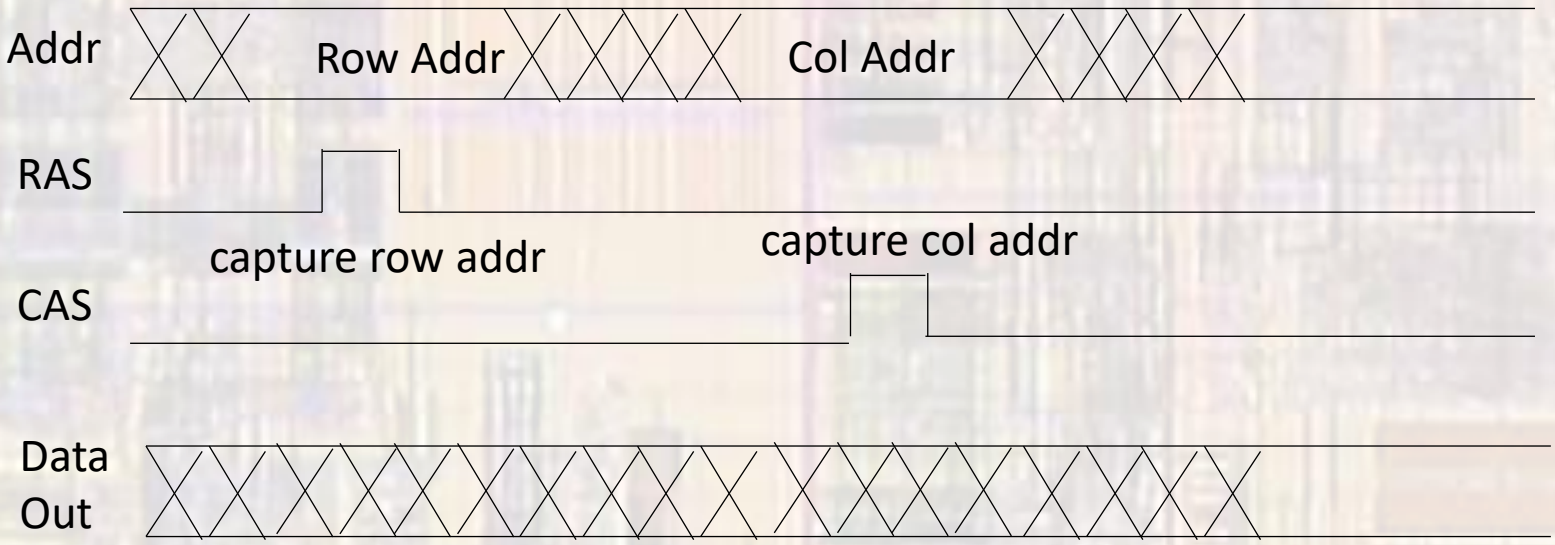
- Example – 16Mb in a x4 configuration



© tj

SRAM Model

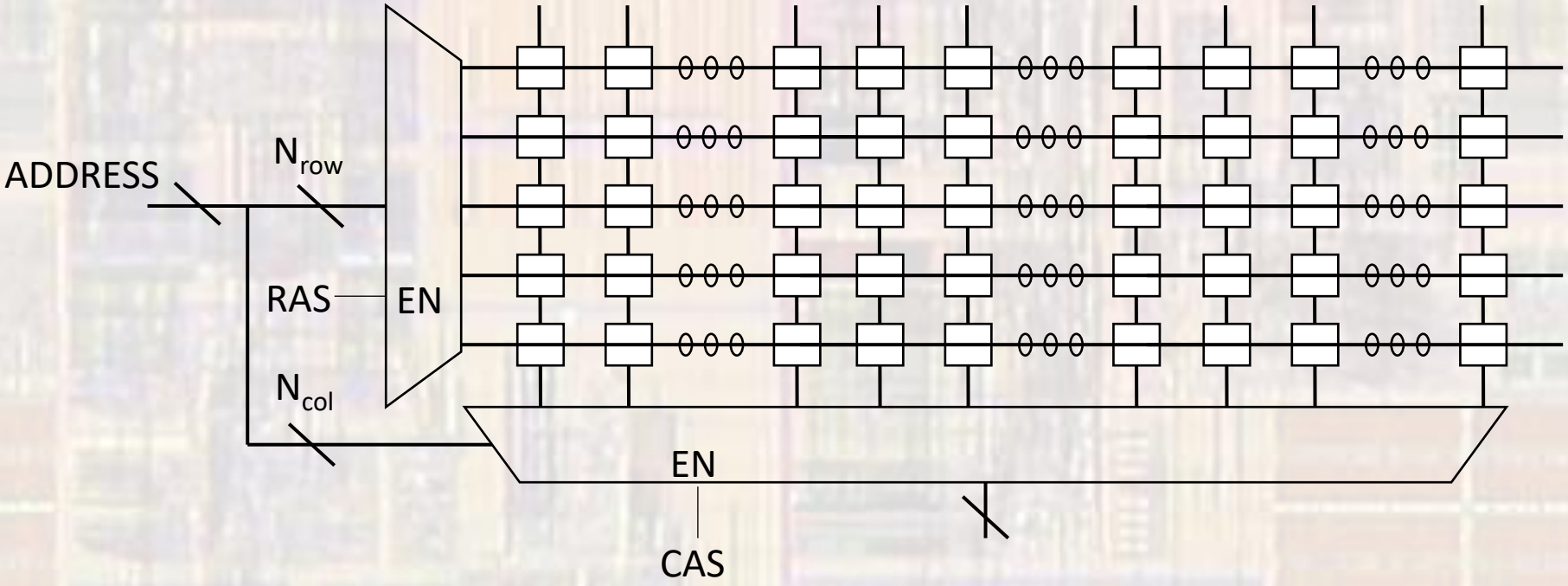
- Address Line Sharing
  - Reduce I/O by sharing the address inputs between Row addresses and Column addresses in time
  - RAS – Row Address Strobe (current address is for row decode)
  - CAS – Column Address Strobe (current address is for column decode)



© tj

SRAM Model

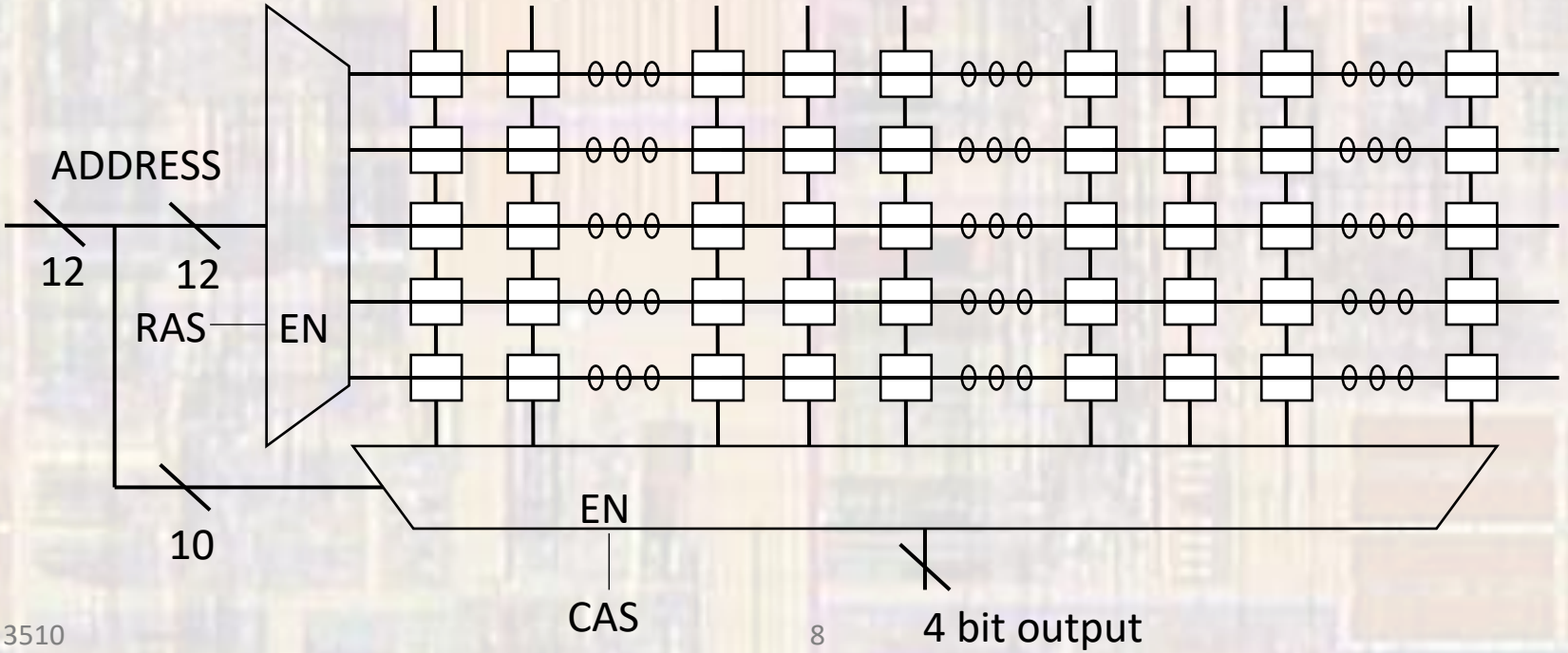
- Address Line Sharing
  - Reduce I/O by sharing the address inputs between Row addresses and Column addresses – in time
  - Number of address bits is the greater of the number of row address bits and column address bits



© tj

SRAM Model

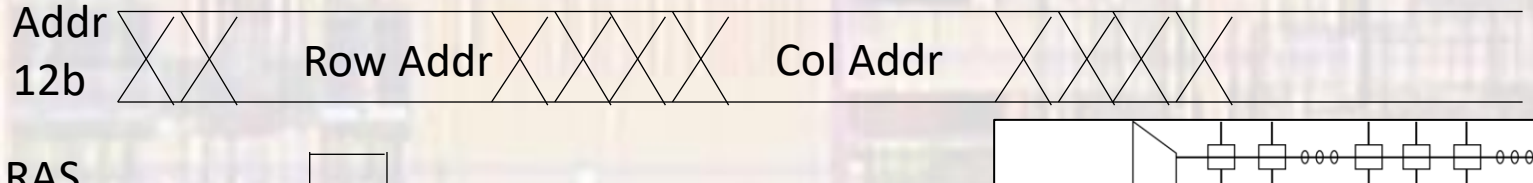
- Address Line Sharing
  - Reduce I/O by sharing the address inputs between Row addresses and Column addresses
  - 22 bit addr → 12 bit addr + RAS and CAS → 14 wires
  - RAS – Row Address Strobe (current address is for row decode)
  - CAS – Column Address Strobe (current address is for column decode)

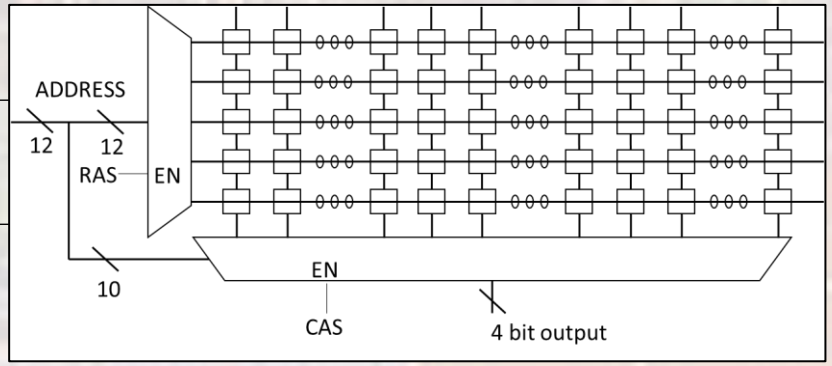


© tj

SRAM Model

- Address Line Sharing
  - Reduce I/O by sharing the address inputs between Row addresses and Column addresses
  - 22 bit addr → 12 bit addr + RAS and CAS → 14 wires
  - RAS – Row Address Strobe (current address is for row decode)
  - CAS – Column Address Strobe (current address is for column decode)





© tj

SRAM Model

- Entity

```
--
-- sram_memory.vhdl
--
-- created 10/5/26 by tj
--
------------------------------------------------
-- Simple RAS/CAS memory
--
-- Generics: row address width, col address width, data width
--             row address width must be greater than column address width
--
-- inputs: clk, rstb(for registers only), rasN, casN,
--          weN, addr in, data in
-- outputs: data out, row active
--
------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity sram_memory is
    generic(
            ROW_ADDR_WIDTH:    positive := 6;
            COL_ADDR_WIDTH:    positive := 4;
            DATA_WIDTH:        positive := 16
        );
    port(
          i_clk:             in std_logic;
          i_rstb:            in std_logic;
          i_ras_n:           in std_logic;
          i_cas_n:           in std_logic;
          i_we_n:            in std_logic;
          i_addr_in:         in std_logic_vector((ROW_ADDR_WIDTH - 1) downto 0);
          i_data_in:         in std_logic_vector((DATA_WIDTH - 1) downto 0);
          o_data_out:        out std_logic_vector((DATA_WIDTH - 1) downto 0);
          o_row_active:      out std_logic
          );
end entity;
```


© tj

SRAM Model

- Internal Signals

```
architecture behavioral of sram_memory is

   constant NUM_COLS:   positive := (2 ** COL_ADDR_WIDTH);
   constant NUM_WORDS:  positive := ((2 ** ROW_ADDR_WIDTH) * (2 ** COL_ADDR_WIDTH));

   -- memory block
   type sram_type is array(0 to (NUM_WORDS - 1)) of std_logic_vector((DATA_WIDTH - 1) downto 0);
   signal ram: sram_type;

   -- address registers
   signal row_addr_sig: unsigned((ROW_ADDR_WIDTH - 1) downto 0);
   signal col_addr_sig: unsigned((COL_ADDR_WIDTH - 1) downto 0);

   signal data_out_valid_sig: std_logic;

begin
```


© tj

SRAM Model

- Row/Col Address Registers

```
-- row addr reg
process(i_rstb, i_ras_n)
begin
   -- register reset, NOT memory reset
   if(i_rstb = '0') then
      row_addr_sig <= (others => '0');
   elsif(falling_edge(i_ras_n)) then
      row_addr_sig <= unsigned(i_addr_in((ROW_ADDR_WIDTH - 1) downto 0));
   end if;
end process;
```

```
-- col addr reg
process(i_rstb, i_ras_n, i_cas_n)
begin
   -- register reset, NOT memory reset
   if(i_rstb = '0') then
      col_addr_sig <= (others => '0');
   elsif(falling_edge(i_cas_n)) then
      col_addr_sig <= unsigned(i_addr_in((COL_ADDR_WIDTH - 1) downto 0));
   end if;
end process;
```


© tj

SRAM Model

- R/W Process

```
   -- memory read/write process
   process(i_clk, row_addr_sig, col_addr_sig, i_ras_n, i_cas_n, i_we_n, ram, i_data_in)
   begin
      if(rising_edge(i_clk)) then
         if(i_cas_n = '0') then
            o_data_out <= ram((to_integer(row_addr_sig) * NUM_COLS) + to_integer(col_addr_sig));
            if(i_we_n = '0') then
               ram((to_integer(row_addr_sig) * NUM_COLS) + to_integer(col_addr_sig)) <= i_data_in;
            end if;
         end if;
      end if;
   end process;

   -- output logic
   o_row_active <= not i_ras_n;

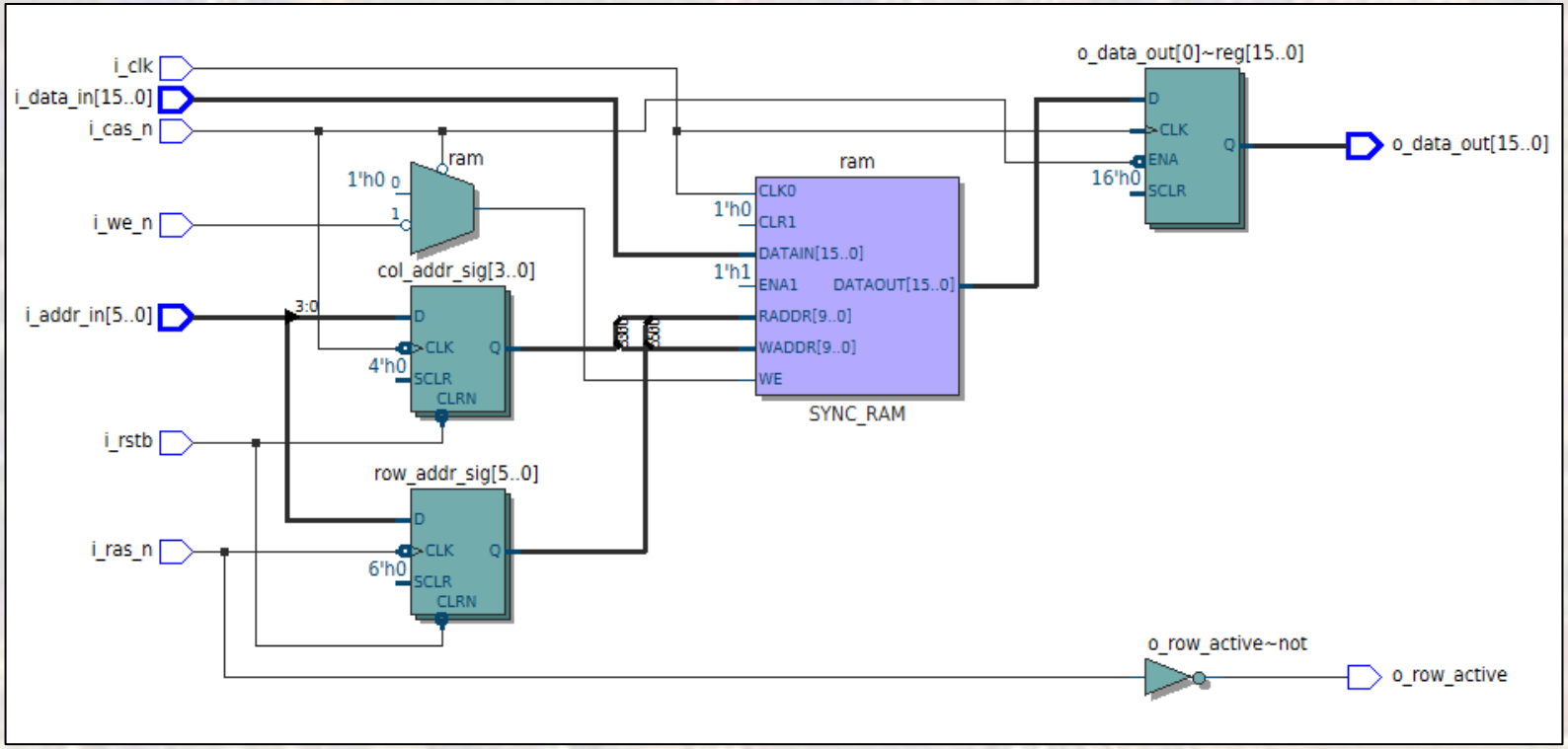
end architecture;
```


© tj

SRAM Model

- RTL

Flow Status	Successful - Tue Oct 6 13:04:09 2026
Quartus Prime Version	25.1std.0 Build 1129 10/21/2025 SC Lite Edition
Revision Name	sram_controller
Top-level Entity Name	sram_memory
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	12 / 49,760 (< 1 %)
Total registers	10
Total pins	44 / 360 (12 %)
Total virtual pins	0
Total memory bits	16,384 / 1,677,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)



 © tj