R·I·T

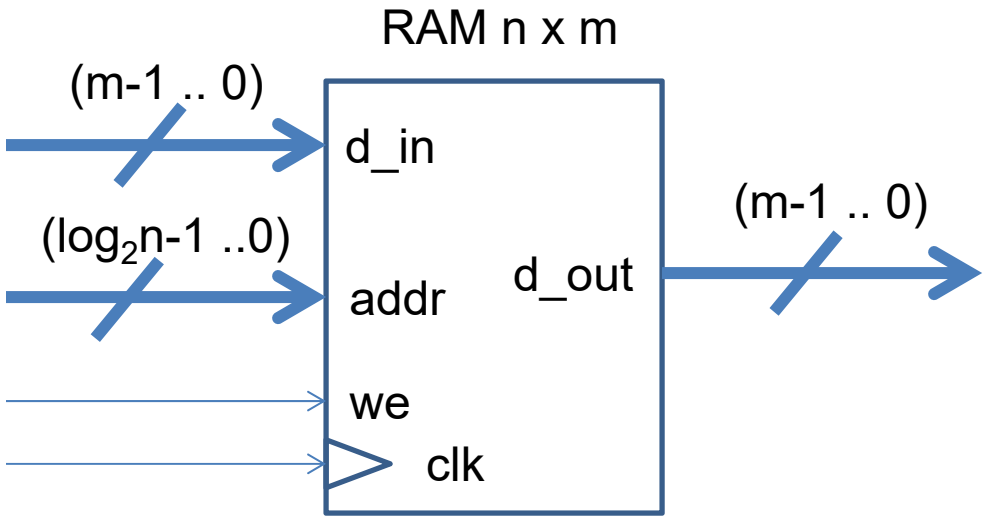
Memory

General notes and VHDL implementation

RAM Configuration

- An **n x m** RAM has n locations each of size m
  - **Example 4K x 128 :**
    - 4096 locations (1K = 1024)
    - The data in each location is 128 bits long
  - **m is the size of the d_in bus and the d_out bus**
  - **n and m have no relationship**
  - **The size of the address bus is $\log_2 n$ allowing each location to have a unique address**

Simple RAM Configuration



VHDL for simple RAM

```
entity raminfer is
   generic (addr_width  : integer := 6;
            data_width  : integer := 8);
   port(
         clk   : in std_logic;
         we    : in std_logic;
         addr  : in std_logic_vector((addr_width - 1) downto 0);
         d_in  : in std_logic_vector(data_width - 1 downto 0);
         d_out : out std_logic_vector(data_width - 1 downto 0));
end raminfer;

architecture rtl of raminfer is

type ram_type is array ((2**addr_width - 1) downto 0) of std_logic_vector (data_width - 1 downto 0);
signal RAM : ram_type;

begin
process(clk)
   begin
   if (clk'event and clk = '1') then
      if (we = '1') then
         RAM(to_integer(unsigned(addr))) <= d_in;
      end if;
      d_out <= RAM(to_integer(unsigned(addr)));
   end if;
 end process;
end rtl;
```

Question: what is on d_out during a write cycle?

You must know this!

- Consider a 64K x 32 RAM
  - **How many locations does it have?**
  - **How wide is the address input?**
  - **How wide are the data inputs and outputs?**

- How many locations can be addressed with address(10 downto 0)?