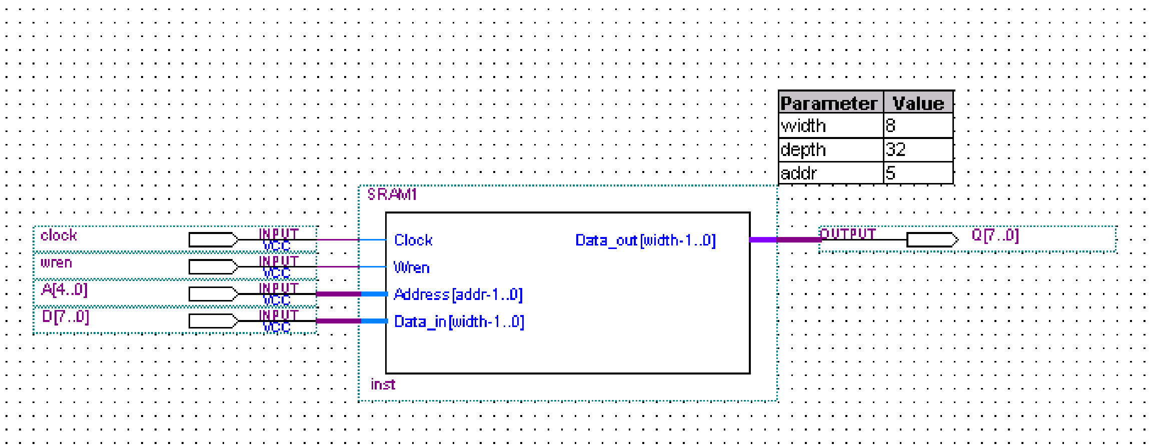
CSC 343 LAB

32*8 SRAM design

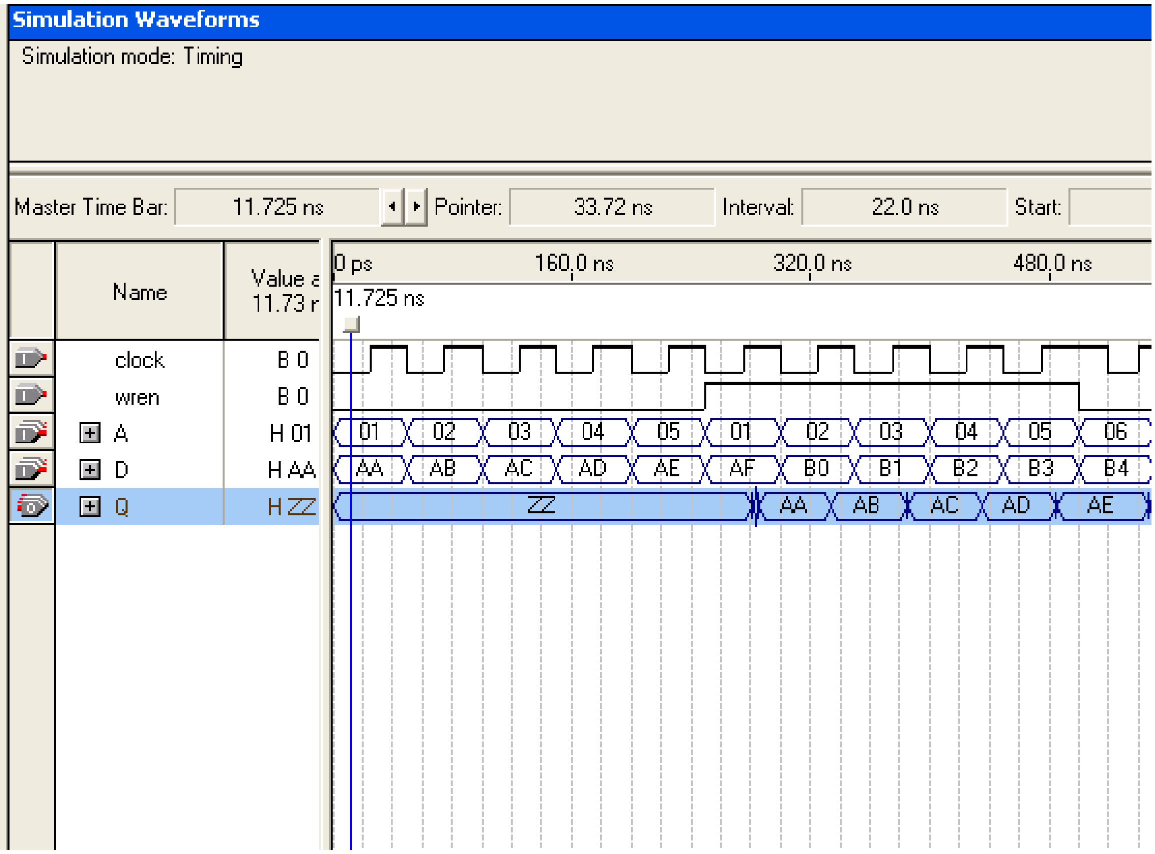
This is a 32*8 RAM(sram1.vhd). It contains 32 eight-bit words, which are accessed using a five-bit address port(A[4..0]), a eight-bit data input port(D[7..0]), a eight-bit date output port(Q[7..0]), a write/Read control input(wren), and a clock.
When wren is low and clock comes (positive trigger), data from D will be stored into the memory with the corresponding address. When wren is high and clock comes, data will be read from the memory with the corresponding address into port Q.



```
==================SRAM1.vhd=====================
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity SRAM1 is
generic(    width:      integer:=8;
            depth:      integer:=32;
            addr: integer:=5);
port(      Clock:         in std_logic;
           Wren:          in std_logic;
           Address:       in std_logic_vector(addr-1 downto 0);
           Data_in:       in std_logic_vector(width-1 downto 0);
           Data_out:      out std_logic_vector(width-1 downto 0)
    );
end SRAM1;


architecture behav of SRAM1 is
```

```
type ram_type is array (0 to depth-1) of
        std_logic_vector(width-1 downto 0);
signal tmp_ram: ram_type;

begin
    process(Clock, Wren)
    begin
      if (Clock'event and Clock='1') then
            if Wren='1' then
                -- buildin function conv_integer change the type
                -- from std_logic_vector to integer
                Data_out <= tmp_ram(conv_integer(Address));
            elsif Wren='0' then
                tmp_ram(conv_integer(Address)) <= Data_in;
                Data_out <= (Data_out'range => 'Z');
            end if;
        end if;
    end process;
end behav;
```

The output of 32*8 SRAM:



Simulation Waveforms

Simulation mode: Timing

Master Time Bar: 11.725 ns Pointer: 33.72 ns Interval: 22.0 ns Start:

Name	Value a 11.73 r
clock	B 0
wren	B 0
A	H 01
D	H AA
Q	H ZZ