ROM Constants

Last updated 7/21/23

ROM Constants

- Setting ROM values using constants
  - Create a constant signal and initialize with the ROM values
    - Note – this is an exception to our requirement not to initialize values

    signal name: type := (address/value list) ;
  - Specify the address and value

    …

    Address => Value

    …
  - User others to set unused locations

    others => Value

ROM Constants

- Mux example

```
--
-- rom_muxbased_constants.vhdl
--
-- created 4/25/17
-- tj
--
-- rev 0
----------------------------------------
--
-- Mux based rom with constants for values
--
----------------------------------------
-- inputs: addr
-- outputs: data
----------------------------------------

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

entity rom_muxbased_constants is
   generic(
      mem_width:  positive := 16;
      mem_depth:  positive := 16
   );
   port(
         i_addr:  in    std_logic_vector(((integer(ceil(log2(real(mem_depth))))) - 1) downto 0);
         o_data:  out   std_logic_vector((mem_width - 1) downto 0)
   );
end entity;
```

```
architecture behavioral of rom_muxbased_constants is

   -- ROM structure
   type rom_type is array (0 to (mem_depth - 1)) of std_logic_vector ((mem_width - 1) downto 0);

-- ROM contents
   constant my_ROM: rom_type:=(
      0 =>  X"C010",
      1 =>  X"C04A",
      2 =>  X"5180",
      3 =>  X"02C0",
      4 =>  X"4640",

      8 =>  X"2E40",
      9 =>  X"6B00",
      10 => X"F000",

      others => X"F000"
   );

begin
   o_data <= my_ROM(to_integer(unsigned(i_addr)));

end architecture;
```

ROM Constants

- Inferred example

```
----------------------------------------
--
-- rom_inferred_constants.vhdl
--
-- created 4/25/17
-- tj
--
-- rev 0
----------------------------------------
--
-- Inferred rom with constants for values
--
----------------------------------------
-- inputs: clk, addr
-- outputs: data
----------------------------------------

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

entity rom_inferred_constants is
   generic(
      mem_width:  positive := 16;
      mem_depth:  positive := 16
   );
   port(
         i_clk:   in    std_logic;
         i_addr:  in    std_logic_vector(((integer(ceil(log2(real(mem_depth))))) - 1) downto 0);
         o_data:  out   std_logic_vector((mem_width - 1) downto 0)
   );
end entity;
```

```
architecture behavioral of rom_inferred_constants is

   -- ROM structure
   type rom_type is array (0 to (mem_depth - 1)) of std_logic_vector ((mem_width - 1) downto 0);

-- ROM contents
   signal my_ROM: rom_type:=(
      0 =>  X"C010",
      1 =>  X"C04A",
      2 =>  X"5180",
      3 =>  X"02C0",
      4 =>  X"4640",

      8 =>  X"2E40",
      9 =>  X"6B00",
      10 => X"F000",

      others => X"F000"
   );

begin

   process (i_clk)
      begin
         if (rising_edge(i_clk)) then
            o_data <= my_ROM(to_integer(unsigned(i_addr)));
         end if;
      end process;

end architecture;
```