HDL Memory - ROM

Last updated 1/26/21

HDL Memory - ROM

- HDL version of a ROM (16B)
  - Create a type for the array

```
type rom_type is array (0 to 15) of std_logic_vector (7 downto 0);
```

  - Specify the ROM contents

```
constant myROM:   rom_type:=(
                                "00000011",  -- addr 0
                                "10011001",  -- addr 1
                                "00101010",
                                 …
                                std_logic_vector(to_signed(95, 8)),
                                std_logic_vector(to_signed(-12, 8)),
                                "01100001",  -- addr 14
                                "01110010"   -- addr 15
        );
```

HDL Memory - ROM

- HDL version of a ROM (16B)
  - Read from the ROM
    - With the address as a std_logic_vector named addr

```
data <= myROM(to_integer(unsigned(addr)));
```

Convert to integer
(need an integer for index)

Need to know if signed or unsigned
to convert it to an integer

Convert SLV to unsigned
(no negative addresses)

HDL Memory - ROM

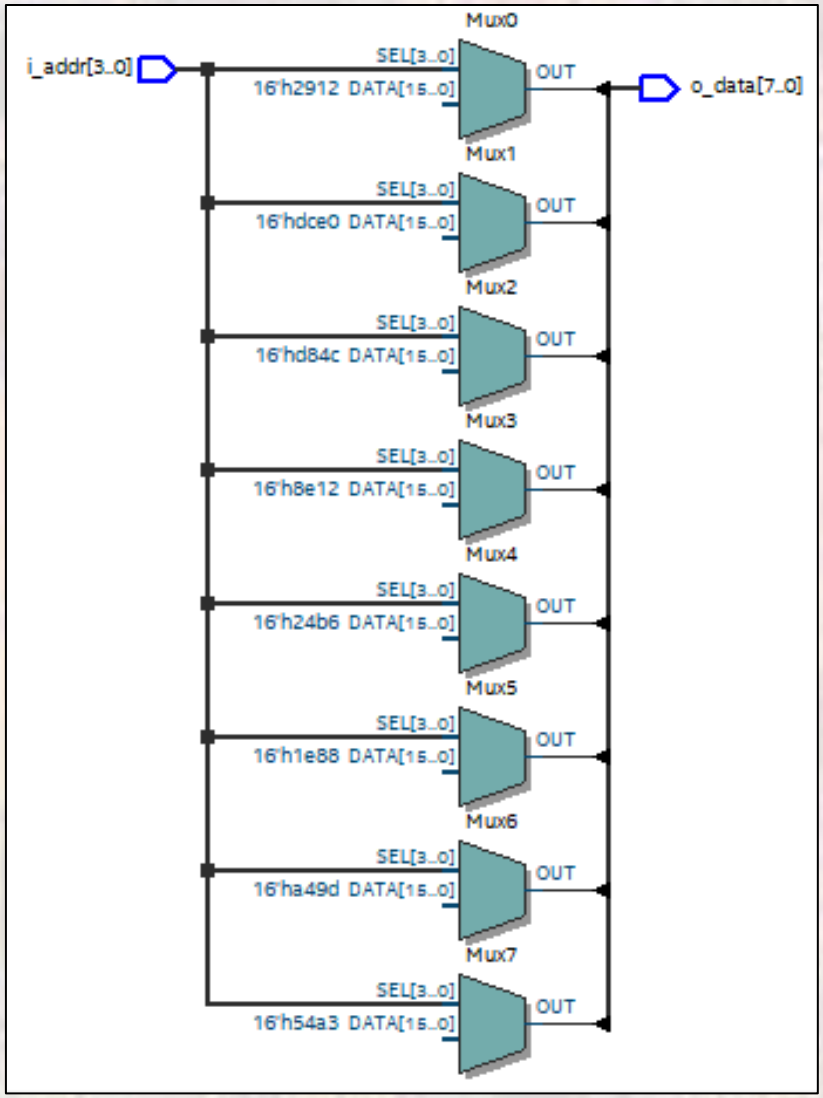
- ROM HDL

```
----------------------------------------
--
-- rom_async_16B.vhdl
--
-- created 4/25/17
-- tj
--
-- rev 0
----------------------------------------
--
-- 16B asynchronous ROM
--
----------------------------------------
--
-- Inputs:  addr
-- Outputs: data
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity rom_async_16B is
   port(
      i_addr: in std_logic_vector(3 downto 0);
      o_data: out std_logic_vector(7 downto 0)
   );
end;
```

```
architecture behavioral of rom_async_16B is

   type rom_type is array (0 to 15) of std_logic_vector (7 downto 0);

   -- ROM contents
   constant myROM: rom_type:=(
      "00000011",  -- addr 0
      "10011001",  -- addr 1
      "00101010",
      "00100110",
      "10011010",
      "01001001",
      "01100000",
      "01001111",
      "10000000",
      "00010100",
      std_logic_vector(to_signed(95, 8)),
      std_logic_vector(to_signed(-12, 8)),
      "01100101",
      "10001010",
      "01100001",  -- addr 14
      "01110010"   -- addr 15
   );

   begin

   -- read from ROM
   o_data <= myROM(to_integer(unsigned(i_addr)));

end behavioral;
```

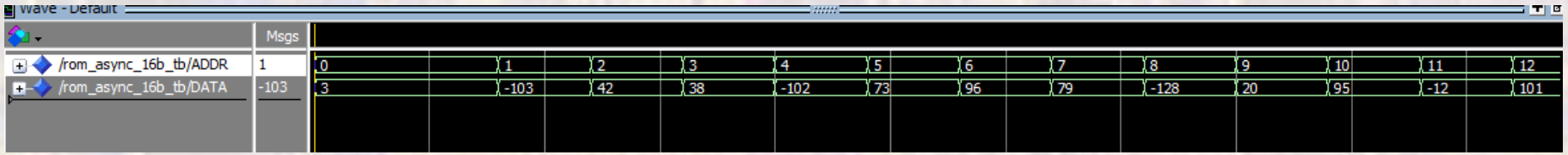
HDL Memory - ROM

- ROM HDL



HDL Memory - ROM

- ROM HDL



In real life this may have lots of glitches as the address bits change → use a synchronous version

HDL Memory - ROM

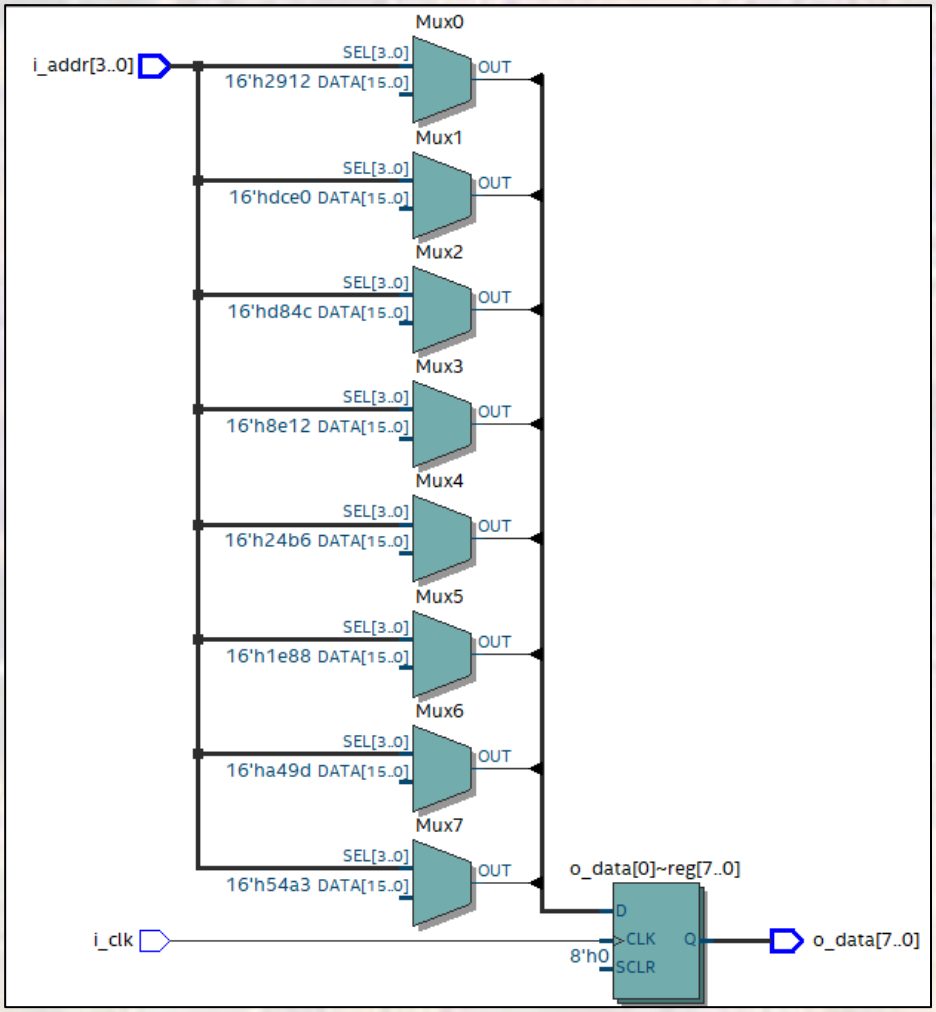
- ROM HDL - sync

```
----------------------------------------
--
-- rom_sync_16B.vhdl
--
-- created 4/25/17
-- tj
--
-- rev 0
----------------------------------------
--
-- 16B synchronous ROM
--
----------------------------------------
--
-- Inputs:  clk, addr
-- Outputs: data
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity rom_sync_16B is
    port(
        i_clk:   in std_logic;
        i_addr:  in std_logic_vector(3 downto 0);
        o_data:  out std_logic_vector(7 downto 0)
    );
end;
```

```
architecture behavioral of rom_sync_16B is

    type rom_type is array (0 to 15) of std_logic_vector (7 downto 0);

    -- ROM contents
    constant myROM: rom_type:=(
        "00000011",   -- addr 0
        "10011001",   -- addr 1
        "00101010",
        "00100110",
        "10011010",
        "01001001",
        "01100000",
        "01001111",
        "10000000",
        "00010100",
        std_logic_vector(to_signed(95, 8)),
        std_logic_vector(to_signed(-12, 8)),
        "01100101",
        "10001010",
        "01100001",   -- addr 14
        "01110010"    -- addr 15
    );

    begin

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

        -- Output signals

end behavioral;
```

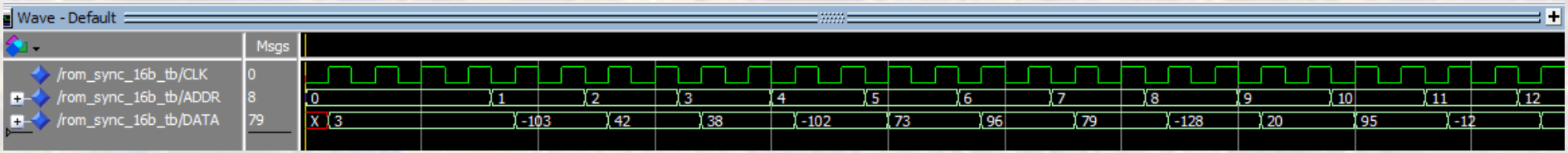
HDL Memory - ROM

- ROM HDL - sync



HDL Memory - ROM

- ROM HDL - sync



Address is stable when clock comes along

HDL Memory - ROM

- Reading from a file
  - Special attribute for memory files
    - ram_init_file
  - Tell the system to use the file for myROM

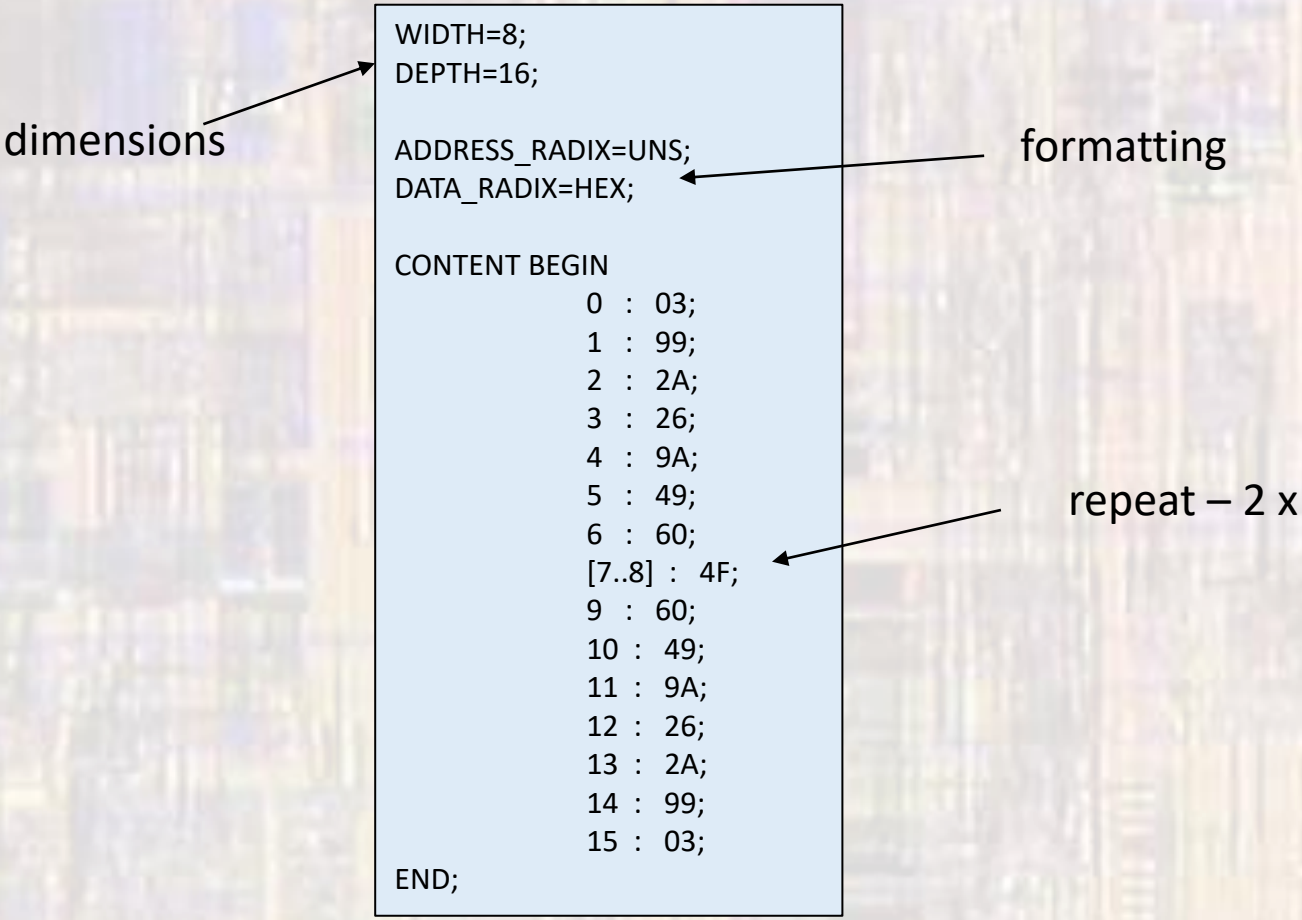
```
ATTRIBUTE ram_init_file: string;
ATTRIBUTE ram_init_file of myROM: signal is "rom_init.mif";
```

windows file

  - File must be added to the project

HDL Memory - ROM

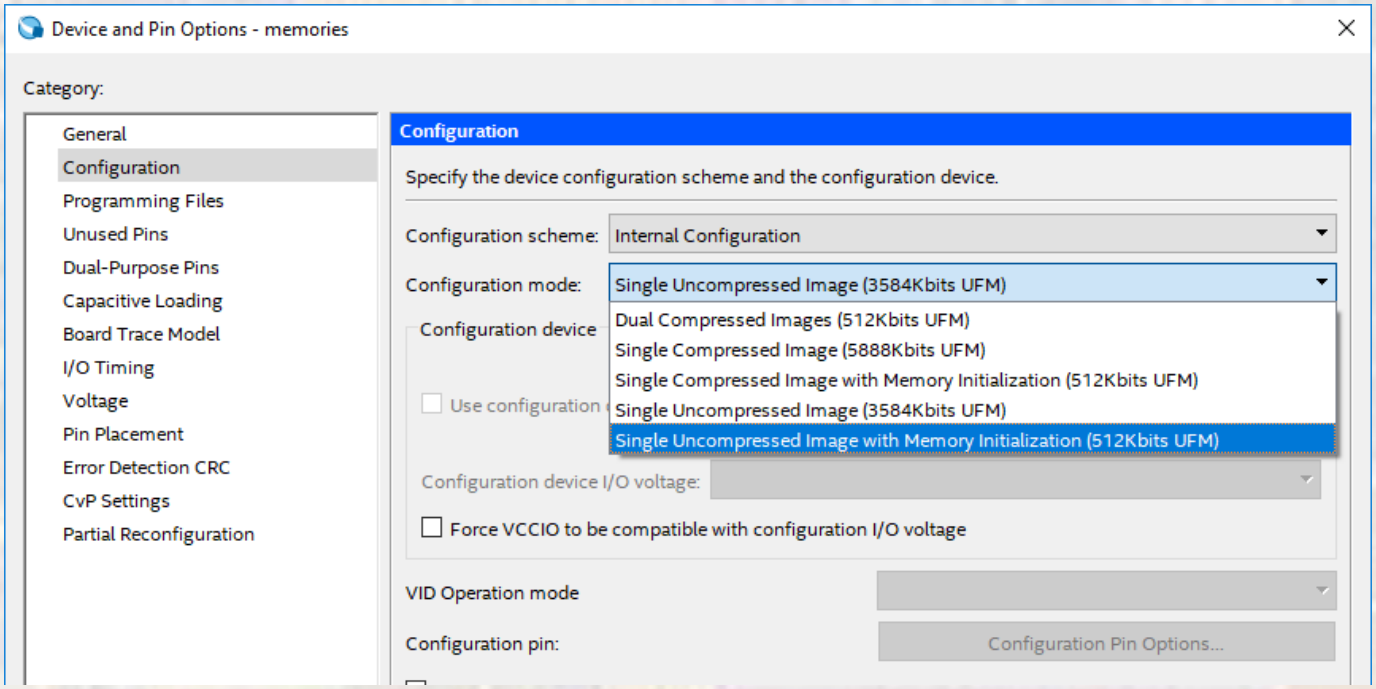
- Reading from a file
  - MIF file format



HDL Memory - ROM

- Reading from a file
  - Assignments → Devices → Device and Pin Options → configuration → Configuration Mode
  - select

Single Uncompressed Image with Memory Initialization



HDL Memory - ROM

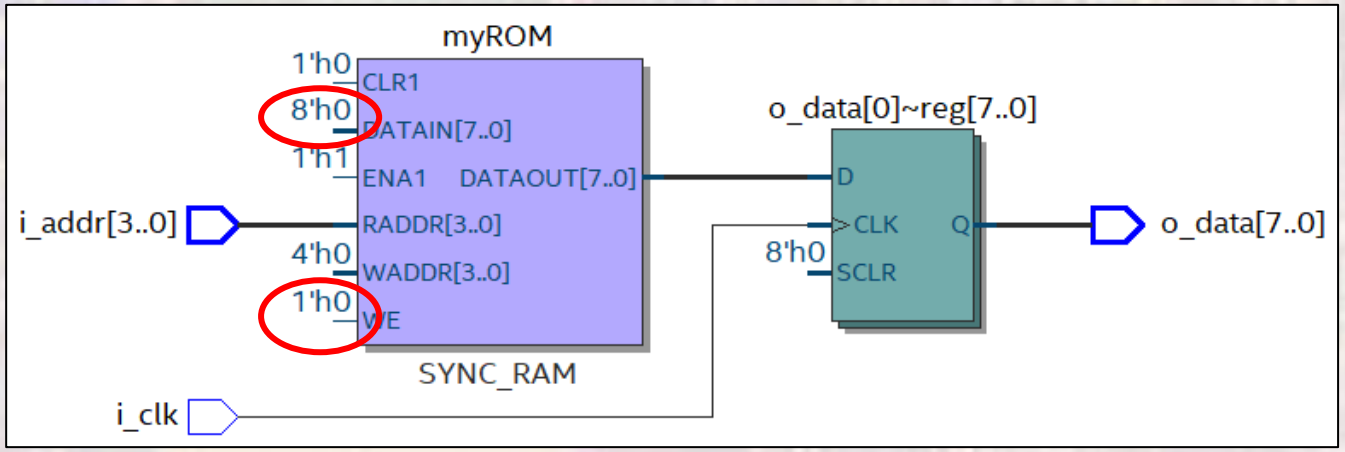
- ROM HDL - file

```
----------------------------------------
--
-- rom_sync_16B_wfile.vhdl
--
-- created 4/25/17
-- tj
--
-- rev 0
----------------------------------------
--
-- 16B synchronous ROM loaded from file
--
----------------------------------------
--
-- Inputs:  clk, addr
-- Outputs: data
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity rom_sync_16B_wfile is
   port(
      i_clk:   in std_logic;
      i_addr:  in std_logic_vector(3 downto 0);
      o_data:  out std_logic_vector(7 downto 0)
   );
end;
```

```
architecture behavioral of rom_sync_16B_wfile is

   type rom_type is array (0 to 15) of std_logic_vector (7 downto 0);

   -- ROM
   signal myROM: rom_type;

   -- ROM contents
   ATTRIBUTE ram_init_file: string;
   ATTRIBUTE ram_init_file of myROM: signal is "rom_init.mif";


   begin

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

      -- Output signals

end behavioral;
```

HDL Memory - ROM

- ROM HDL
  - Tool instantiates the ROM as a sync_RAM



HDL Memory - ROM

- ROM HDL
  - Simulation does not support the concept of an external .mif file for loading the ROM
  - Required for DE10 implementations