

# BUILDING A ROM WITH XILINX WEBPACK

1. Open your project.
2. Project -> New Source.
  - a. Select VHDL module
  - b. File Name: "InstROM"
  - c. Select Next
  - d. Port Name "InstAddress", in, MSB=7,LSB=0 (for a 256-entry ROM)
  - e. Port Name "InstOut", out, MSB=7,LSB=0
  - f. Select Next
  - g. Select Finish
3. Edit the VHDL code in InstROM.vhd, so it looks like this:

---

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity InstROM is
    Port ( InstAddress : in std_logic_vector(7 downto 0);
          InstOut      : out std_logic_vector(7 downto 0));
end InstROM;

architecture Behavioral of InstROM is

    type ROM_Array is array (0 to 255)
        of std_logic_vector(7 downto 0);

    constant Content: ROM_Array := (
        B"00100100",
        B"00000001",
        B"00010010",
        B"11111111",
        B"00000011",
        others => B"00000000"    );

begin

    InstOut <= Content(conv_integer(InstAddress));

end Behavioral;
```

---

This is for a 256-entry ROM (8-bit PC), 8 bits wide (8 bit instructions). The values in red are those that would change for a different sized ROM. Notice that all the inputs are binary. You should also be able to specify them in hex (see RAM code). Your ROM contents will be different (and much longer!). This puts hex 0x24 (binary 00100100) in location 0, 0x101 in location 1, 0x012 in location 2, etc.

4. In "Sources" Window, click InstROM.vhd.
5. In "Processes" window, click "Check Syntax" (in "Synthesize XST") – correct errors.
6. In "Processes" window, click "Create Schematic Symbol" (in "Design Entry Utilities")
7. Open Schematic in which you want to include your ROM.

8. "instrom" should now be added to your library of symbols. You may want to edit its shape to make it look nicer.
9. Test it alone to make sure it is reading the right values, with the bits in the order you expect, etc.