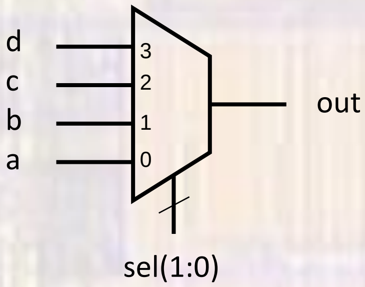
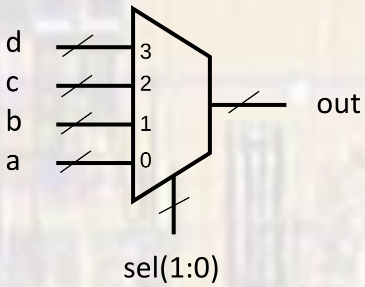
Multiplexors

Last updated 8/22/24

Multiplexor

- A multiplexor selects one of several inputs and routes it to the output
  - The inputs and output can be single wires or busses

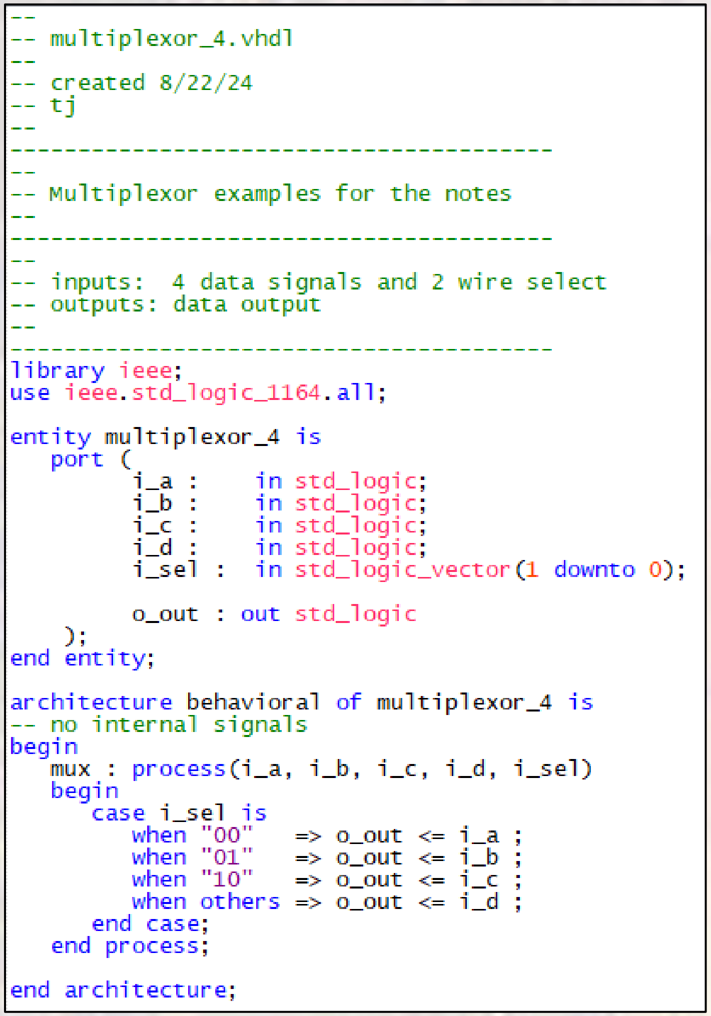




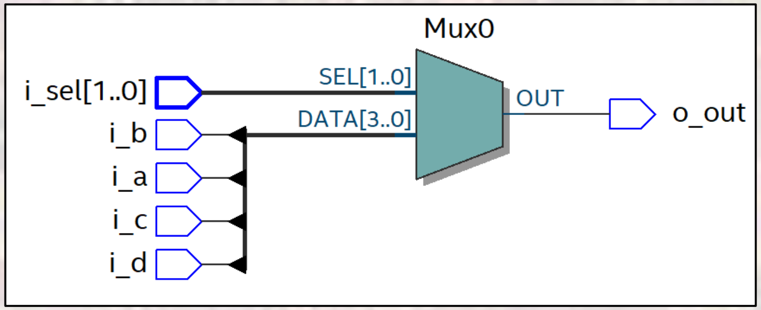
sel1	sel0	out
0	0	a
0	1	b
1	0	c
1	1	d

Multiplexor

- 4 input multiplexor



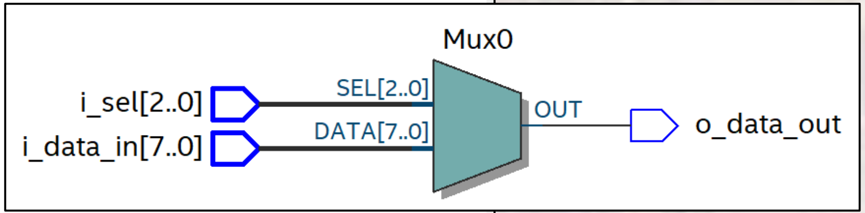
```
--
-- multiplexor_4.vhdl
--
-- created 8/22/24
-- tj
--
----------------------------------------
--
-- Multiplexor examples for the notes
--
----------------------------------------
--
-- inputs:  4 data signals and 2 wire select
-- outputs: data output
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity multiplexor_4 is
   port (
         i_a :    in std_logic;
         i_b :    in std_logic;
         i_c :    in std_logic;
         i_d :    in std_logic;
         i_sel  :  in std_logic_vector(1 downto 0);

         o_out : out std_logic
     );
end entity;

architecture behavioral of multiplexor_4 is
-- no internal signals
begin
   mux : process(i_a, i_b, i_c, i_d, i_sel)
   begin
      case i_sel is
         when "00"    => o_out <= i_a ;
         when "01"    => o_out <= i_b ;
         when "10"    => o_out <= i_c ;
         when others => o_out <= i_d ;
      end case;
   end process;

end architecture;
```



Multiplexor

- n input multiplexor – N must be a power of 2

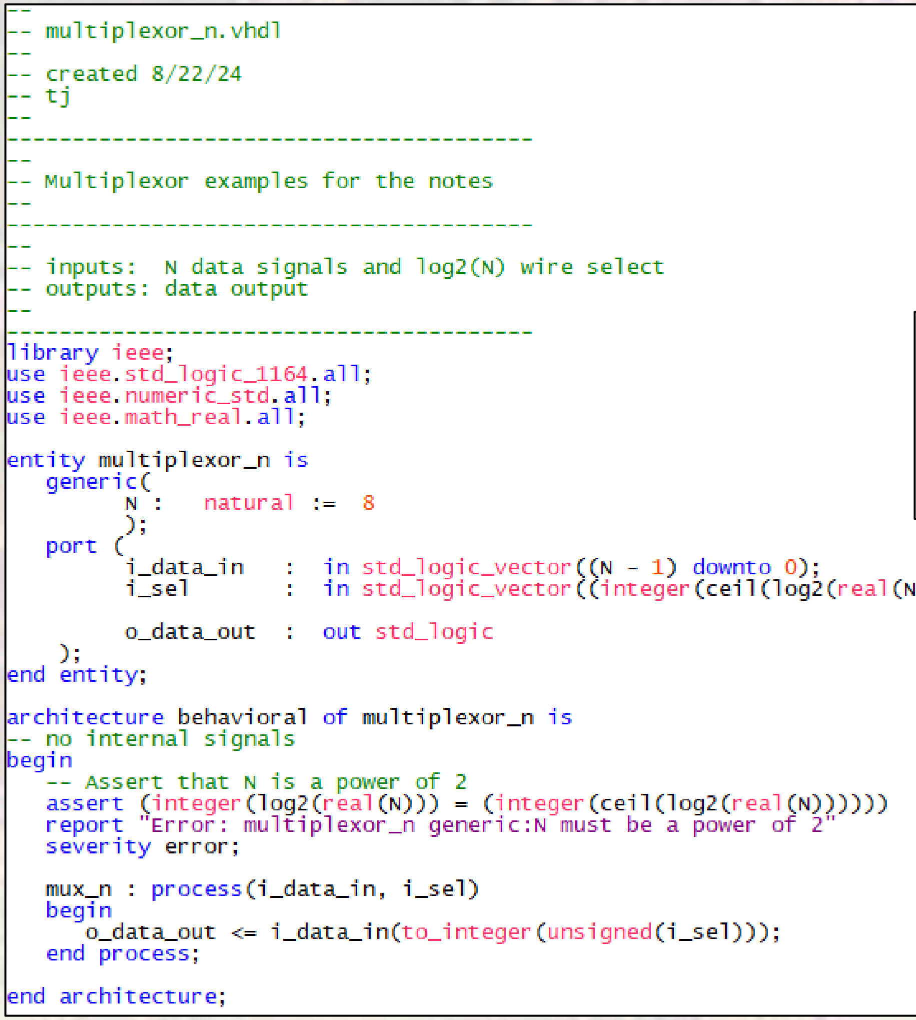
```
--
-- multiplexor_n.vhdl
--
-- created 8/22/24
-- tj
--
----------------------------------------
--
-- Multiplexor examples for the notes
--
----------------------------------------
--
-- inputs:  N data signals and log2(N) wire select
-- outputs: data output
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

entity multiplexor_n is
   generic(
         N :   natural :=  8
         );
   port (
         i_data_in    :  in std_logic_vector((N - 1) downto 0);
         i_sel        :  in std_logic_vector((integer(ceil(log2(real(N)))) - 1) downto 0);

         o_data_out   :  out std_logic
    );
end entity;

architecture behavioral of multiplexor_n is
-- no internal signals
begin
   mux_n : process(i_data_in, i_sel)
   begin
      o_data_out <= i_data_in(to_integer(unsigned(i_sel)));
   end process;

end architecture;
```



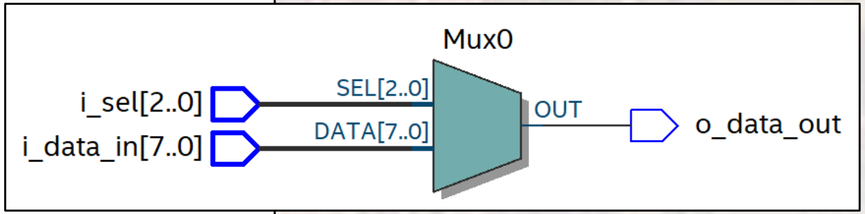
although the data inputs are configured as a vector – they would be connected as individual wires

Multiplexor

- n input multiplexor – check if N is a power of 2



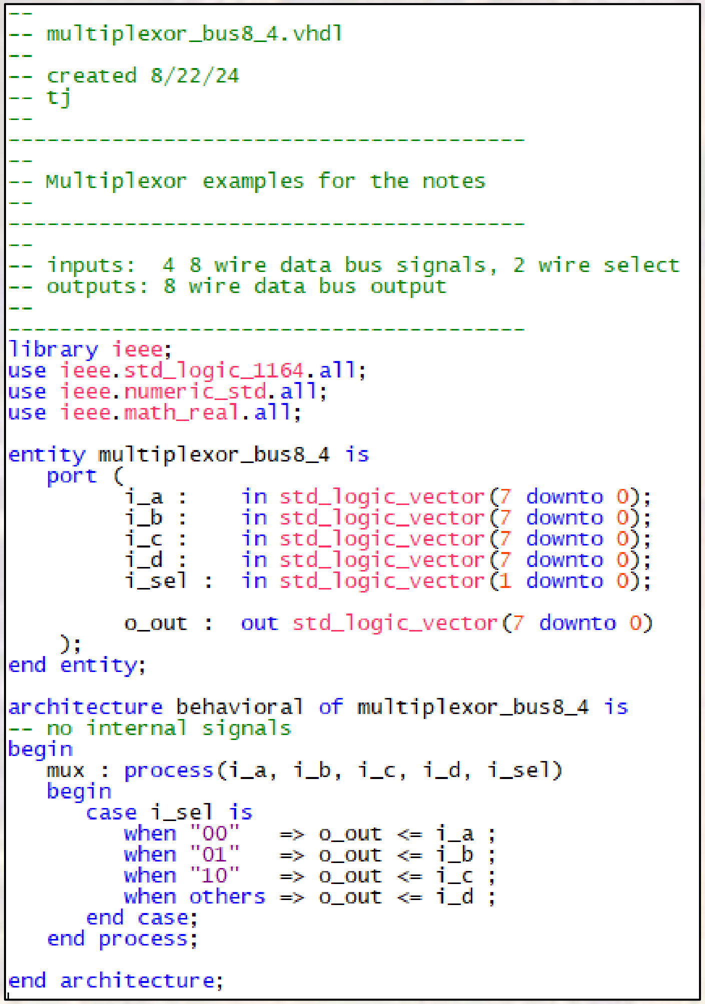
```vhdl
--
-- multiplexor_n.vhdl
--
-- created 8/22/24
-- tj
--
----------------------------------------
--
-- Multiplexor examples for the notes
--
----------------------------------------
--
-- inputs:  N data signals and log2(N) wire select
-- outputs: data output
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

entity multiplexor_n is
   generic(
         N :   natural :=  8
         );
   port (
         i_data_in   :  in std_logic_vector((N - 1) downto 0);
         i_sel       :  in std_logic_vector((integer(ceil(log2(real(N)))) - 1) downto 0);

         o_data_out  :  out std_logic
     );
end entity;

architecture behavioral of multiplexor_n is
-- no internal signals
begin
   -- Assert that N is a power of 2
   assert (integer(log2(real(N))) = (integer(ceil(log2(real(N))))))
   report "Error: multiplexor_n generic:N must be a power of 2"
   severity error;

   mux_n : process(i_data_in, i_sel)
   begin
      o_data_out <= i_data_in(to_integer(unsigned(i_sel)));
   end process;

end architecture;
```



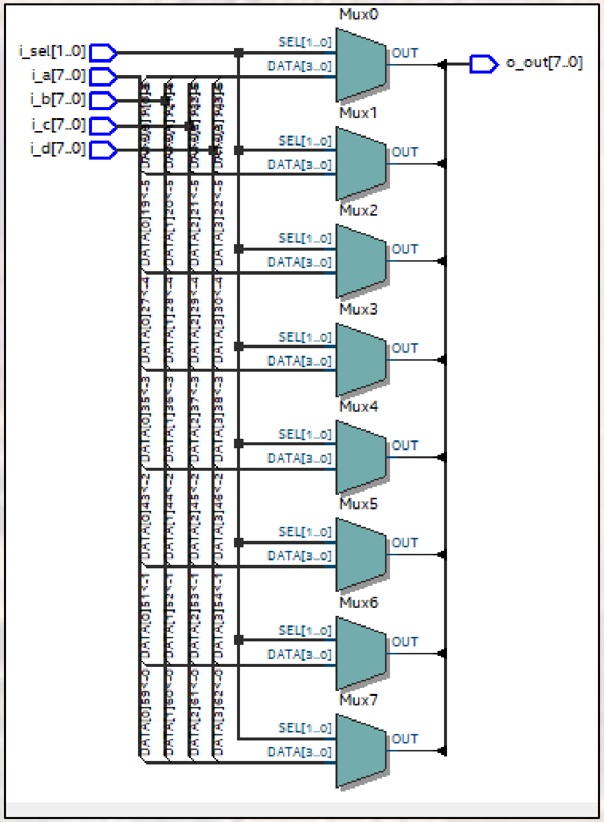
Create a compiler error if N is NOT a power of 2

Multiplexor

- bus multiplexor – 8 wire, 4 way



```
--
-- multiplexor_bus8_4.vhdl
--
-- created 8/22/24
-- tj
--
----------------------------------------
--
-- Multiplexor examples for the notes
--
----------------------------------------
--
-- inputs:  4 8 wire data bus signals, 2 wire select
-- outputs: 8 wire data bus output
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

entity multiplexor_bus8_4 is
   port (
         i_a :     in std_logic_vector(7 downto 0);
         i_b :     in std_logic_vector(7 downto 0);
         i_c :     in std_logic_vector(7 downto 0);
         i_d :     in std_logic_vector(7 downto 0);
         i_sel :   in std_logic_vector(1 downto 0);

         o_out :   out std_logic_vector(7 downto 0)
    );
end entity;

architecture behavioral of multiplexor_bus8_4 is
-- no internal signals
begin
   mux : process(i_a, i_b, i_c, i_d, i_sel)
   begin
      case i_sel is
         when "00"    => o_out <= i_a ;
         when "01"    => o_out <= i_b ;
         when "10"    => o_out <= i_c ;
         when others  => o_out <= i_d ;
      end case;
   end process;

end architecture;
```



Instead of implementing this as 4, 8 wire muxes the compiler implemented it as 8, 4 wire muxes each managing 1 wire of the 8 wire inputs

Multiplexor

- bus multiplexor – M wire, N way

see if you can do this