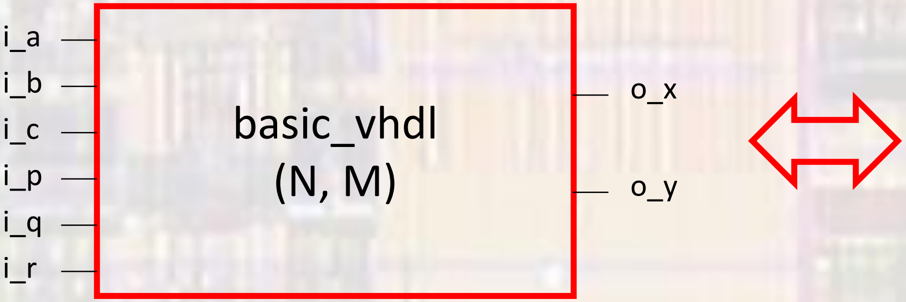
VHDL Entity

Last updated 12/19/24

VHDL Entity

- The Entity is the external view of the block
  - Inputs
  - Outputs
  - Parameters – generics
    - Allow 1 design for multiple variations
    - e.g. 4bit, 8bit, 16bit, 32bit adder

Just wires (ports) no interpretation
ALWAYS std_logic or std_logic_vector() for us



Generics

```
-- Entity definition
entity basic_vhdl is
      generic(   N:       positive := 8;
                 M:       positive := 16
              );
      port(   i_a:     in      std_logic;
              i_b:     in      std_logic;
              i_c :    in      std_logic;
              i_p:     in      std_logic_vector(N-1 downto 0);
              i_q:     in      std_logic_vector(N-1 downto 0);
              i_r:     in      std_logic_vector(M-1 downto 0);
              o_x:     out     std_logic;
              o_y:     out     std_logic_vector(M-1 downto 0)
          );
end entity;
```

Ports

VHDL Entity

- Format - block

```
-- Entity definition
entity block_name is




end entity ;
```

block_name must match the file name

VHDL Entity

- Format - ports

```
-- Entity definition
entity block_name is


  port(  pin_name :  direction   signal_type ;





          pin_name :  direction   signal_type
        );
end entity;
```

pin names will begin with i_ or o_
directions are input or output
signal type is std_logic or std_logic_vector

no ; on the last port in the list

VHDL Entity

- Format - generic

```
-- Entity definition
entity block_name is

   generic( generic_name :    generic_type := default_value ;
               generic_name :    generic_type := default_value
            );

  port(  pin_name :  direction   signal_type ;



               pin_name :  direction   signal_type
         );
end entity;
```

no ; on the last generic in the list

generic types can be natural, positive, integer

VHDL Entity

- Example

```
-- Entity definition
entity basic_vhdl is
    generic( N:      positive := 8;
             M:      positive := 16
           );
    port(  i_a:  in     std_logic;
           i_b:  in     std_logic;
           i_c : in     std_logic;
           i_p:  in     std_logic_vector(N-1 downto 0);
           i_q:  in     std_logic_vector(N-1 downto 0);
           i_r:  in     std_logic_vector(M-1 downto 0);
           o_x:  out    std_logic;
           o_y:  out    std_logic_vector(M-1 downto 0)
         );
end entity;
```