VHDL Generic Adders / Subtractors

Last updated 1/8/25

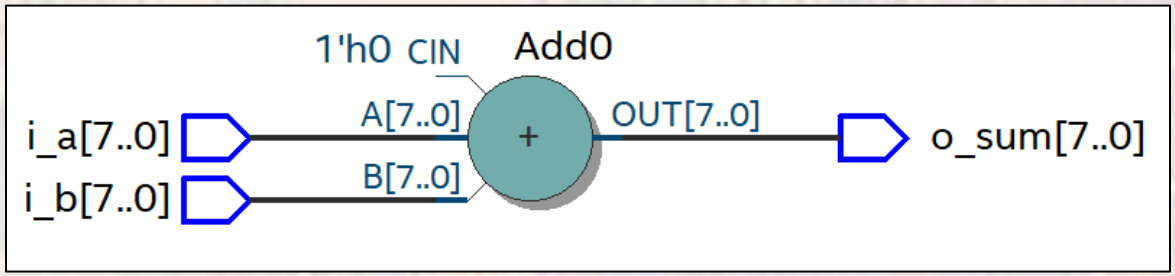
VHDL Generic Adders/Subtractors

- Make our VHDL more versatile using generics

```
----------------------------------
--
--  adder_nbit_signed.vhdl
--
--  by: johnsontimoj
--
--  created: 12/31/24
--
--  version: 0.0
--
----------------------------------
--
--  nbit signed adder
--
--  inputs: a, b
--
--  outputs: sum
--
----------------------------------
library IEEE;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity adder_nbit_signed is
   generic( N:     positive := 8
          );

   port( i_a:           in std_logic_vector((N-1) downto 0);
         i_b:           in std_logic_vector((N-1) downto 0);
         o_sum:         out std_logic_vector((N-1) downto 0)
       );
end entity;
```

default value

```
architecture behavioral of adder_nbit_signed is

   signal   a_sig:   signed((N-1) downto 0);
   signal   b_sig:   signed((N-1) downto 0);
   signal   sum_sig: signed((N-1) downto 0);

begin

   a_sig <= signed(i_a);
   b_sig <= signed(i_b);

   sum_sig <= a_sig + b_sig;

   o_sum <= std_logic_vector(sum_sig);

end architecture;
```



VHDL Generic Adders/Subtractors

- Using generics in structural VHDL

```
----------------------------------
--
--  adder_nbit_signed_3x.vhdl
--
--  by: johnsontimoj
--
--  created: 12/31/24
--
--  version: 0.0
--
----------------------------------
--
--  nbit signed adder intantiated 3x
--
--  inputs: a, b
--
--  outputs: sum
--
----------------------------------
library IEEE;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity adder_nbit_signed_3x is
   generic( X:     positive := 4;
            Y:     positive := 8;
            Z:     positive := 12
          );

   port( i_a1:          in std_logic_vector((X-1) downto 0);
         i_b1:          in std_logic_vector((X-1) downto 0);
         o_sum1:        out std_logic_vector((X-1) downto 0);

         i_a2:          in std_logic_vector((Y-1) downto 0);
         i_b2:          in std_logic_vector((Y-1) downto 0);
         o_sum2:        out std_logic_vector((Y-1) downto 0);

         i_a3:          in std_logic_vector((Z-1) downto 0);
         i_b3:          in std_logic_vector((Z-1) downto 0);
         o_sum3:        out std_logic_vector((Z-1) downto 0)
         );
   end entity;
```

```
architecture behavioral of adder_nbit_signed_3x is

   component adder_nbit_signed is
      generic( N:     positive := 8
             );

      port( i_a:           in std_logic_vector((N-1) downto 0);
            i_b:           in std_logic_vector((N-1) downto 0);
            o_sum:         out std_logic_vector((N-1) downto 0)
          );
   end component;


begin

   add4:    adder_nbit_signed
      generic map(
                  N => X
                  )
      port map (
                  i_a   => i_a1,
                  i_b   => i_b1,
                  o_sum => o_sum1
                );

   add8:    adder_nbit_signed
      generic map(
                  N => Y
                  )
      port map (
                  i_a   => i_a2,
                  i_b   => i_b2,
                  o_sum => o_sum2
                );

   add12:   adder_nbit_signed
      generic map(
                  N => Z
                  )
      port map (
                  i_a   => i_a3,
                  i_b   => i_b3,
                  o_sum => o_sum3
                );


end architecture;
```

generic adder instantiated 3 times with different sizes