Counters - HDL

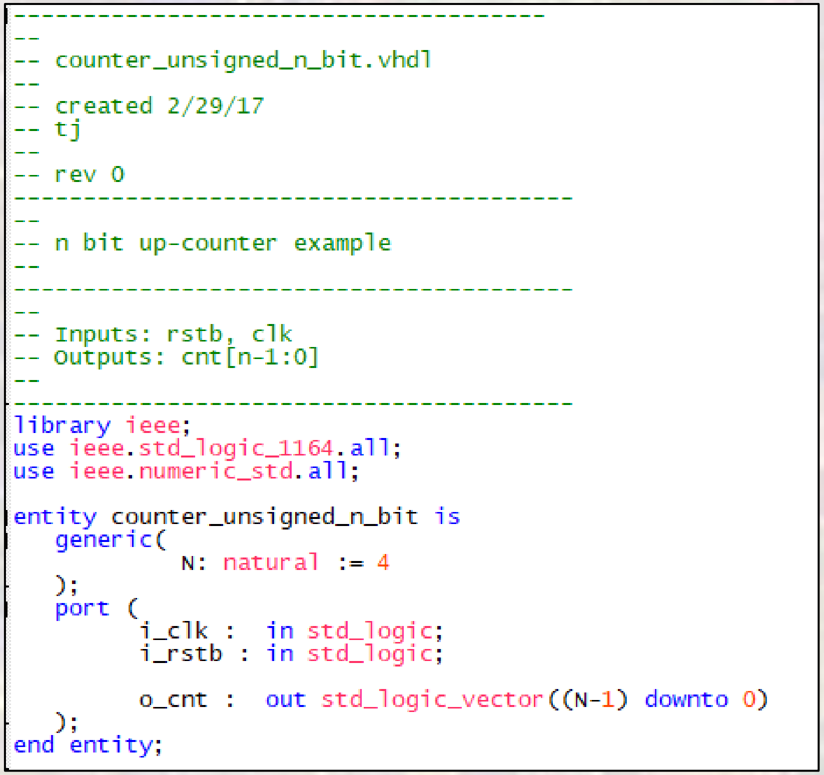
Last updated 1/18/21

Counters - HDL

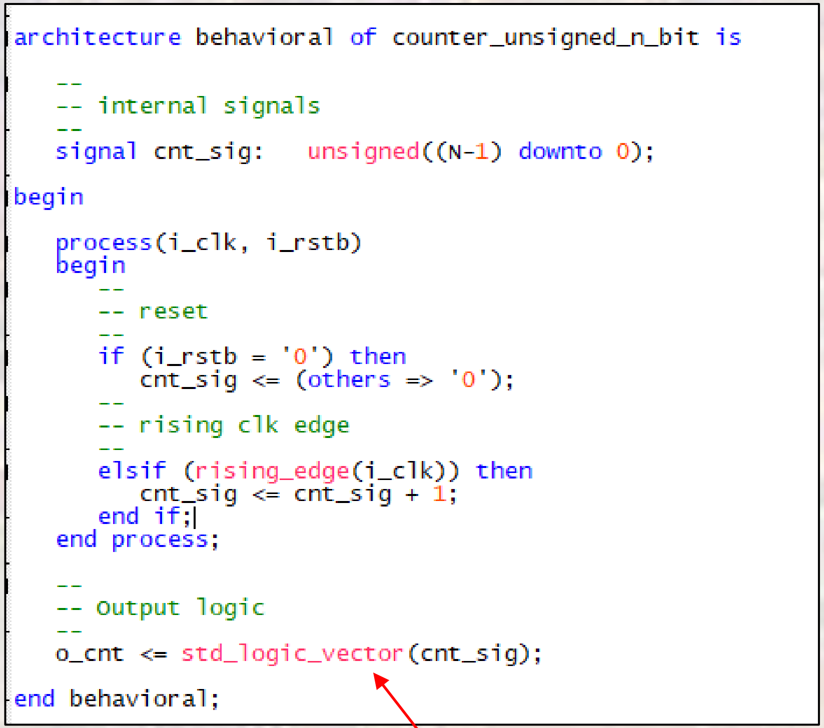
- Counter
  - Count in binary
  - 0000 → 0001 → 0010 → 0011 → 0100 … 1111 → 0000 …

Counters - HDL

- Counter - n bit – unsigned
  - Counts to 2^N-1 then back to 0



```
--------------------------------------
--
-- counter_unsigned_n_bit.vhdl
--
-- created 2/29/17
-- tj
--
-- rev 0
----------------------------------------
--
-- n bit up-counter example
--
----------------------------------------
--
-- Inputs: rstb, clk
-- Outputs: cnt[n-1:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter_unsigned_n_bit is
   generic(
            N: natural  := 4
   );
   port (
         i_clk  :  in std_logic;
         i_rstb :  in std_logic;

         o_cnt  :  out std_logic_vector((N-1) downto 0)
   );
end entity;
```

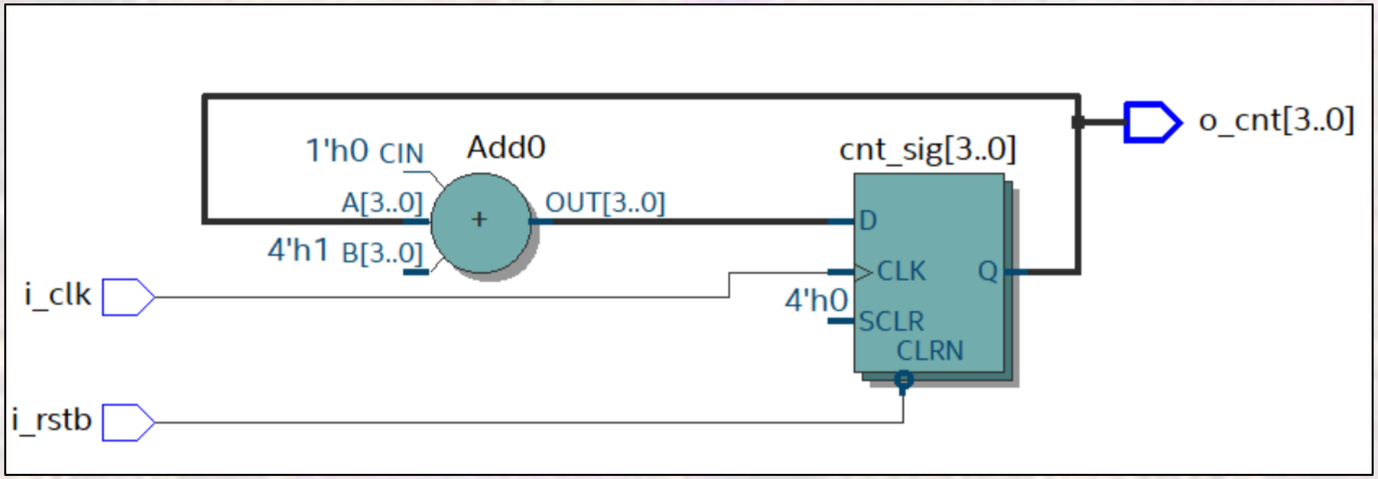


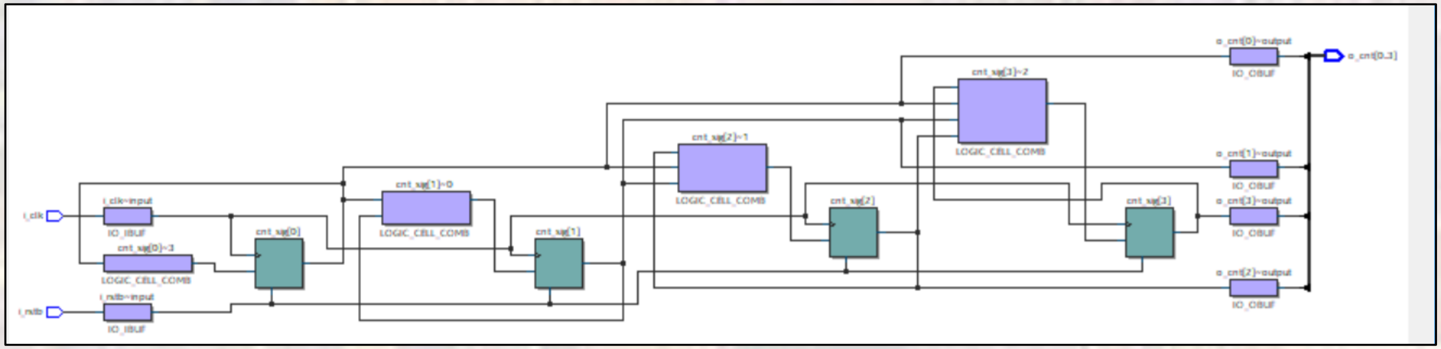
```
architecture behavioral of counter_unsigned_n_bit is

   --
   -- internal signals
   --
   signal cnt_sig:   unsigned((N-1) downto 0);

begin

   process(i_clk, i_rstb)
   begin
      --
      -- reset
      --
      if (i_rstb = '0') then
         cnt_sig <= (others => '0');
      --
      -- rising clk edge
      --
      elsif (rising_edge(i_clk)) then
         cnt_sig <= cnt_sig + 1;
      end if;
   end process;

   --
   -- Output logic
   --
   o_cnt <= std_logic_vector(cnt_sig);

end behavioral;
```

CAST

Counters - HDL

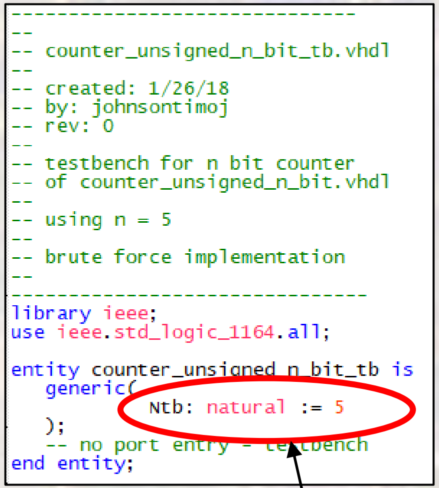
- Counter – n(4) bit - unsigned



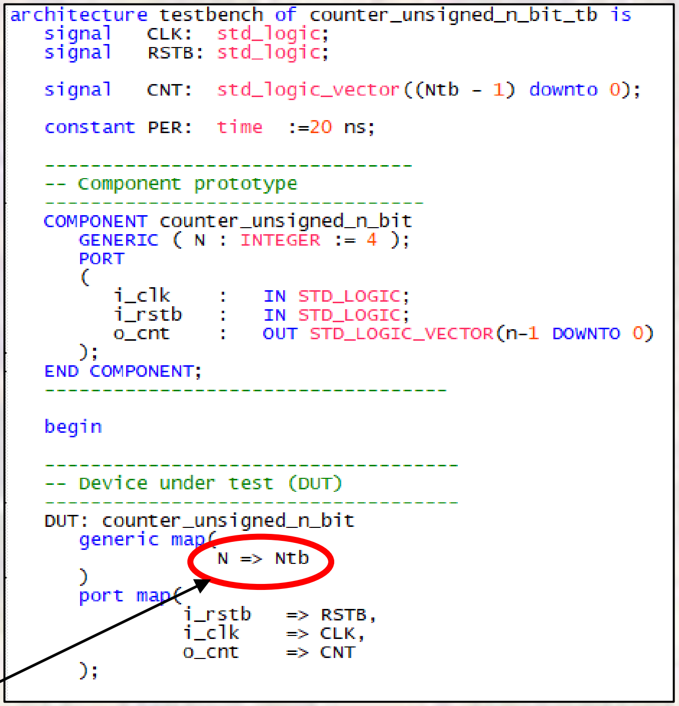


Counters - HDL

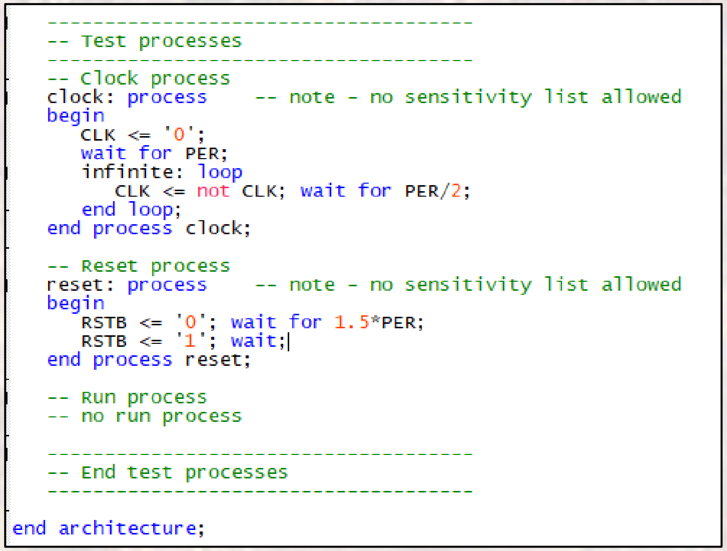
- Counter – n(5) bit – unsigned - testbench



```
------------------------------
--
-- counter_unsigned_n_bit_tb.vhdl
--
-- created: 1/26/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for n bit counter
-- of counter_unsigned_n_bit.vhdl
--
-- using n = 5
--
-- brute force implementation
--
-------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_unsigned_n_bit_tb is
   generic(
            Ntb: natural  := 5
   );
   -- no port entry - testbench
end entity;
```



```
architecture testbench of counter_unsigned_n_bit_tb is
   signal   CLK:  std_logic;
   signal   RSTB: std_logic;

   signal   CNT:  std_logic_vector((Ntb - 1) downto 0);

   constant PER:  time   :=20 ns;

   --------------------------------
   -- Component prototype
   ---------------------------------
   COMPONENT counter_unsigned_n_bit
      GENERIC ( N : INTEGER := 4 );
      PORT
      (
         i_clk    :   IN STD_LOGIC;
         i_rstb   :   IN STD_LOGIC;
         o_cnt    :   OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0)
      );
   END COMPONENT;
   -----------------------------------


   begin

   -----------------------------------
   -- Device under test (DUT)
   -----------------------------------
   DUT: counter_unsigned_n_bit
      generic map(
            N => Ntb
      )
      port map(
            i_rstb    => RSTB,
            i_clk     => CLK,
            o_cnt     => CNT
      );
```



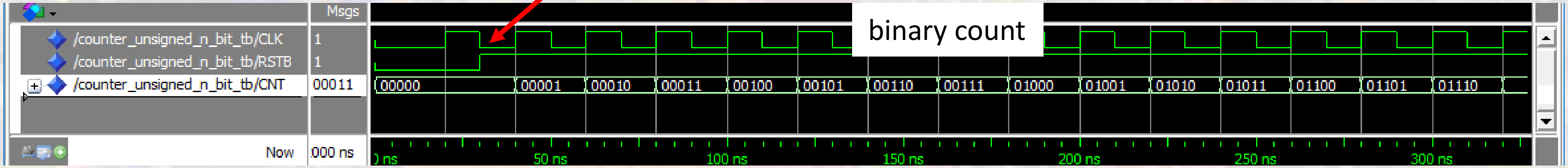
```
   -------------------------------------
   -- Test processes
   -------------------------------------
   -- Clock process
   clock: process      -- note - no sensitivity list allowed
   begin
      CLK <= '0';
      wait for PER;
      infinite: loop
         CLK <= not CLK; wait for PER/2;
      end loop;
   end process clock;

   -- Reset process
   reset: process      -- note - no sensitivity list allowed
   begin
      RSTB <= '0'; wait for 1.5*PER;
      RSTB <= '1'; wait;
   end process reset;

   -- Run process
   -- no run process


   -------------------------------------
   -- End test processes
   -------------------------------------

end architecture;
```

5 bit implementation

Counters - HDL

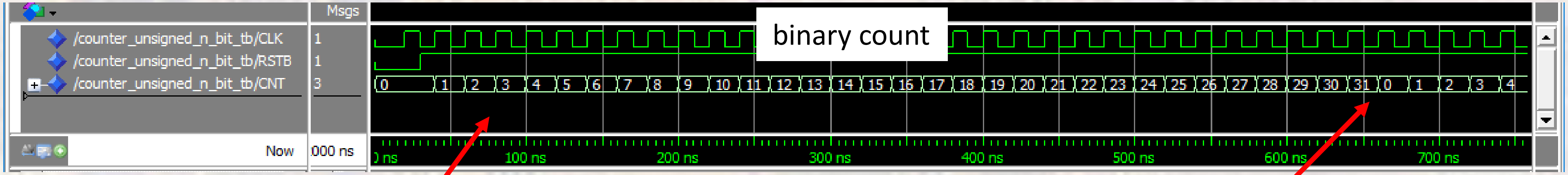
- Counter - n bit unsigned (5 bit)

reset



binary count

	Msgs
/counter_unsigned_n_bit_tb/CLK	1
/counter_unsigned_n_bit_tb/RSTB	1
/counter_unsigned_n_bit_tb/CNT	00011
Now	000 ns



binary count

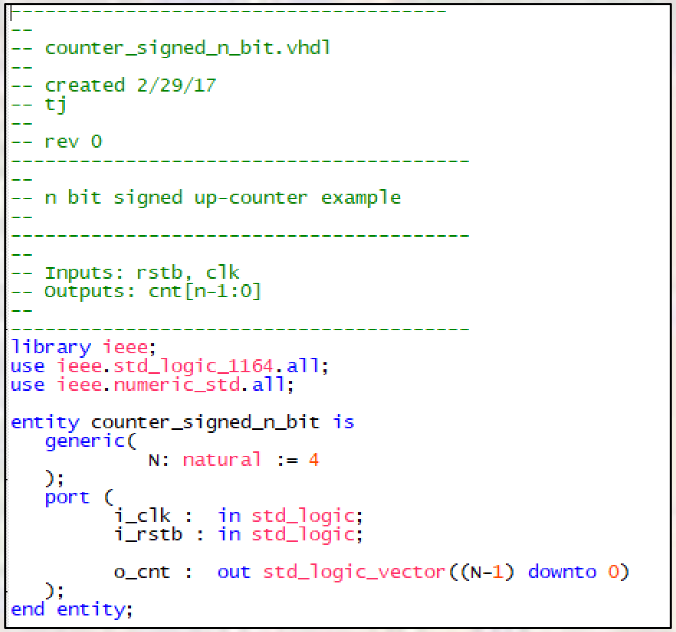
	Msgs
/counter_unsigned_n_bit_tb/CLK	1
/counter_unsigned_n_bit_tb/RSTB	1
/counter_unsigned_n_bit_tb/CNT	3
Now	000 ns

radix → unsigned

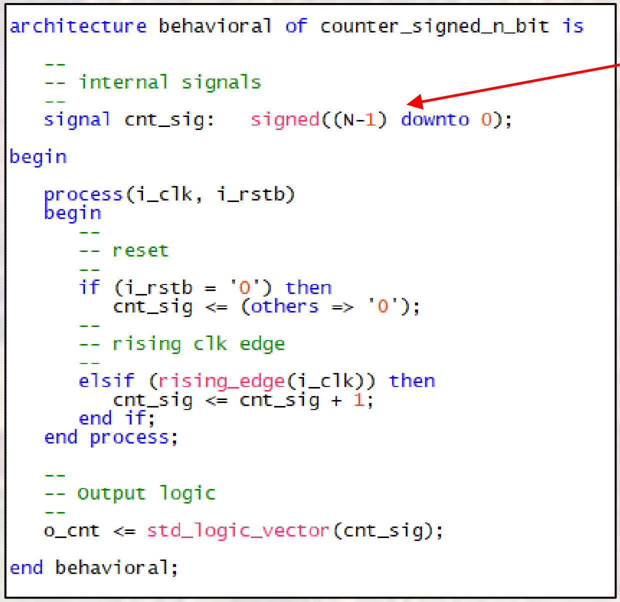
roll-over

Counters - HDL

- Counter - n bit – signed
  - Counts to $2^{N-1}-1$ then wraps to -2^{N-1} and back to 0



```
--------------------------------------
--
-- counter_signed_n_bit.vhdl
--
-- created 2/29/17
-- tj
--
-- rev 0
----------------------------------------
--
-- n bit signed up-counter example
--
----------------------------------------
--
-- Inputs: rstb, clk
-- Outputs: cnt[n-1:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter_signed_n_bit is
   generic(
            N: natural  := 4
   );
   port (
         i_clk :  in std_logic;
         i_rstb : in std_logic;

         o_cnt :  out std_logic_vector((N-1) downto 0)
   );
end entity;
```

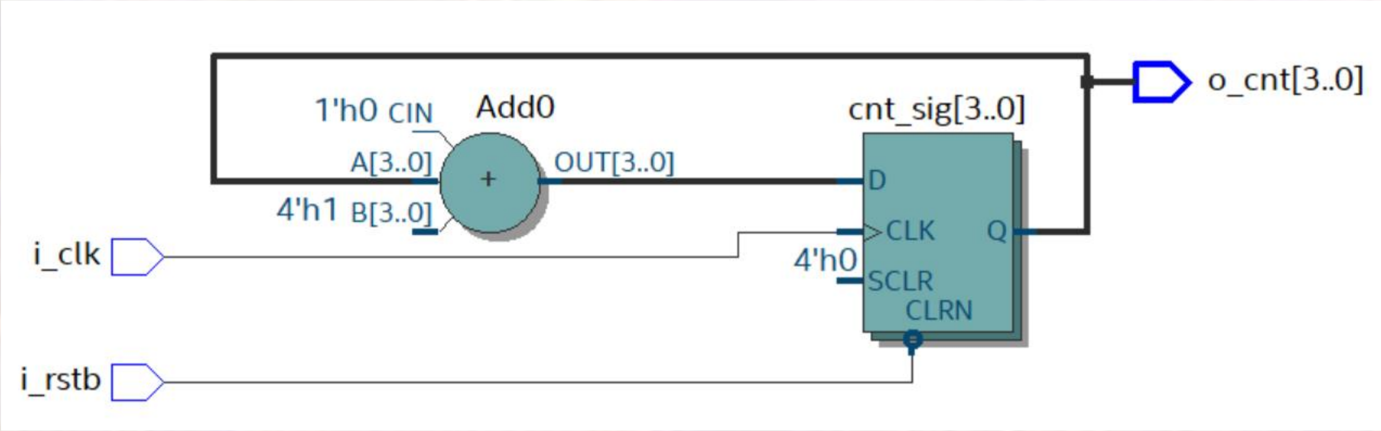


```
architecture behavioral of counter_signed_n_bit is

   --
   -- internal signals
   --
   signal cnt_sig:   signed((N-1) downto 0);

begin

   process(i_clk, i_rstb)
   begin
      --
      -- reset
      --
      if (i_rstb = '0') then
         cnt_sig <= (others => '0');
      --
      -- rising clk edge
      --
      elsif (rising_edge(i_clk)) then
         cnt_sig <= cnt_sig + 1;
      end if;
   end process;

   --
   -- output logic
   --
   o_cnt <= std_logic_vector(cnt_sig);

end behavioral;
```

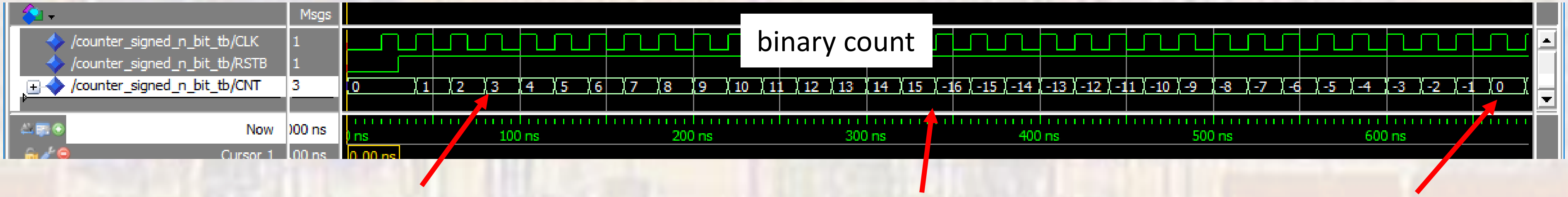
change

Counters - HDL

- Counter - n bit – signed (5 bit)



- No change in TB from unsigned version



binary count

	Msgs
/counter_signed_n_bit_tb/CLK	1
/counter_signed_n_bit_tb/RSTB	1
/counter_signed_n_bit_tb/CNT	3
Now	)00 ns
Cursor 1	)00 ns

radix → decimal

negative wrap

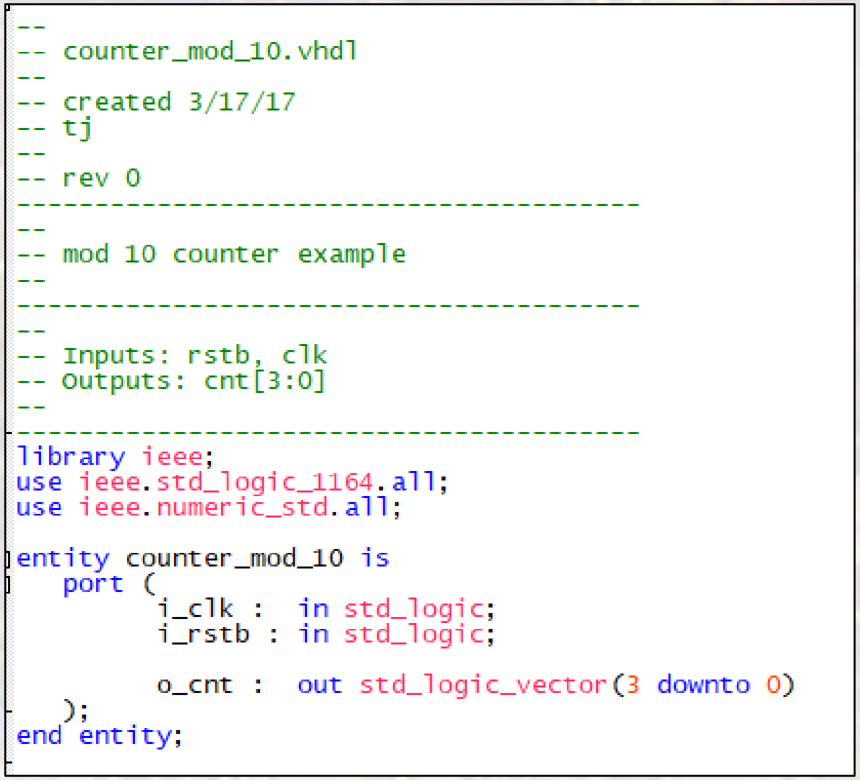
positive wrap

Counters - HDL

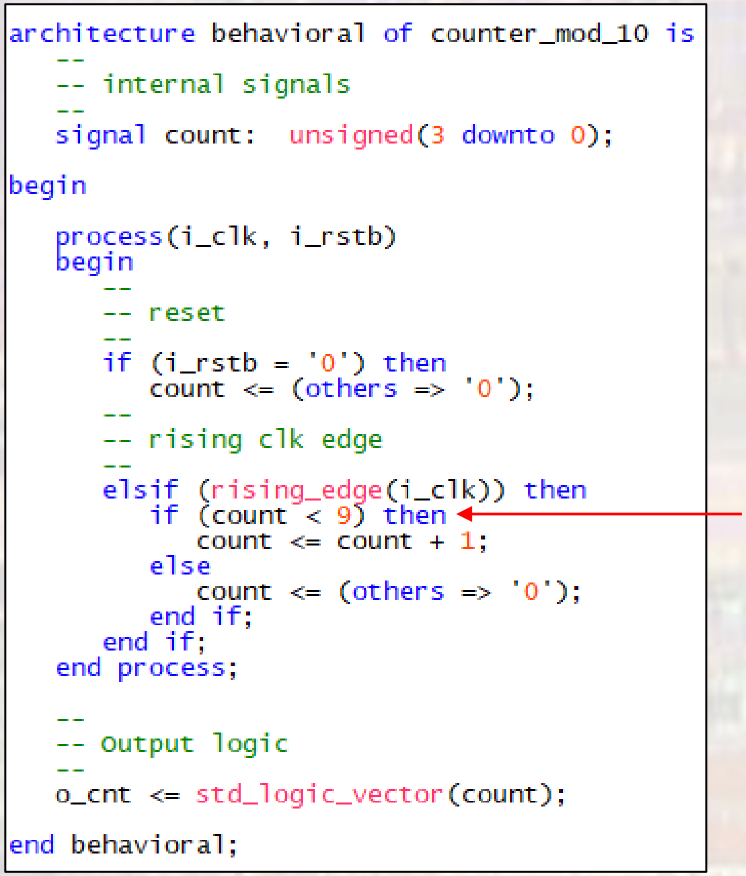
Mod 10 counter

Counters - HDL

- Mod 10 counter



```
--
-- counter_mod_10.vhdl
--
-- created 3/17/17
-- tj
--
-- rev 0
----------------------------------------
--
-- mod 10 counter example
--
----------------------------------------
--
-- Inputs: rstb, clk
-- Outputs: cnt[3:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter_mod_10 is
   port (
         i_clk  :  in std_logic;
         i_rstb : in std_logic;

         o_cnt  :  out std_logic_vector(3 downto 0)
   );
end entity;
```

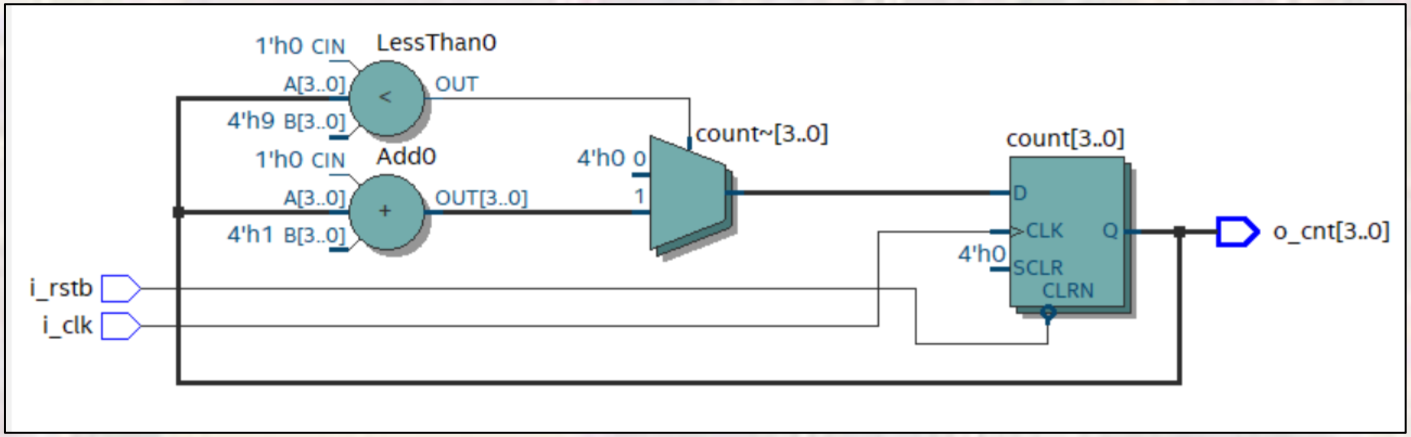


```
architecture behavioral of counter_mod_10 is
   --
   -- internal signals
   --
   signal count:  unsigned(3 downto 0);

begin

   process(i_clk, i_rstb)
   begin
      --
      -- reset
      --
      if (i_rstb = '0') then
         count <= (others => '0');
      --
      -- rising clk edge
      --
      elsif (rising_edge(i_clk)) then
         if (count < 9) then
            count <= count + 1;
         else
            count <= (others => '0');
         end if;
      end if;
   end process;

   --
   -- Output logic
   --
   o_cnt <= std_logic_vector(count);

end behavioral;
```

test

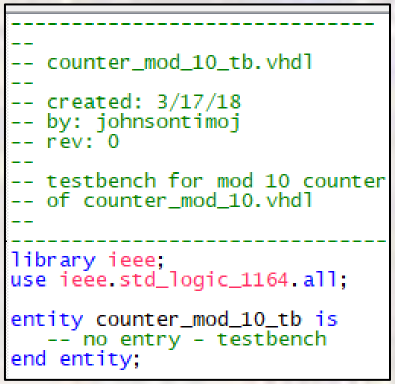
Counters - HDL

- Mod 10 counter

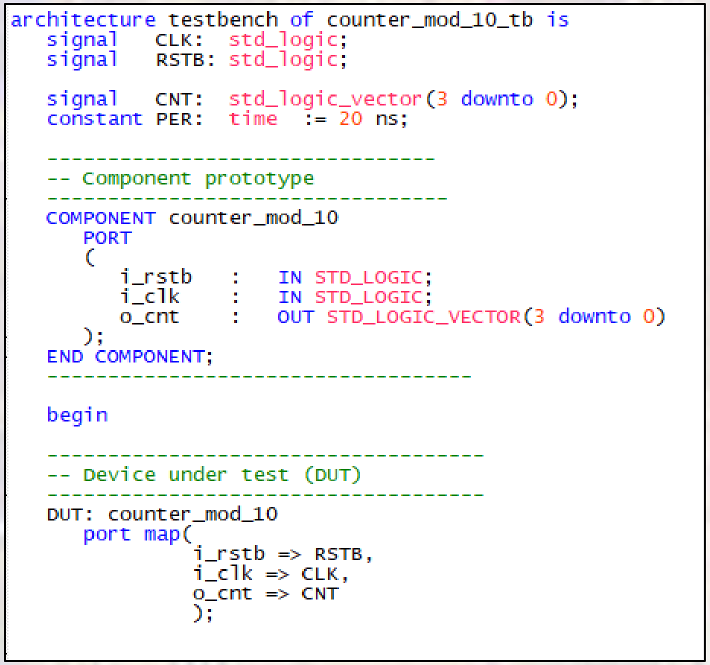


Counters - HDL

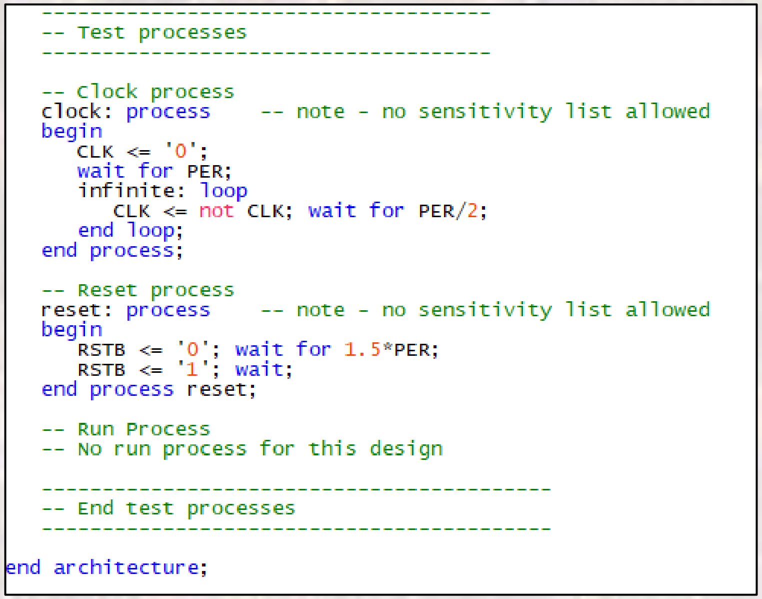
- Mod 10 counter - testbench



```
------------------------------
--
-- counter_mod_10_tb.vhdl
--
-- created: 3/17/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for mod 10 counter
-- of counter_mod_10.vhdl
--
------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_mod_10_tb is
     -- no entry - testbench
end entity;
```



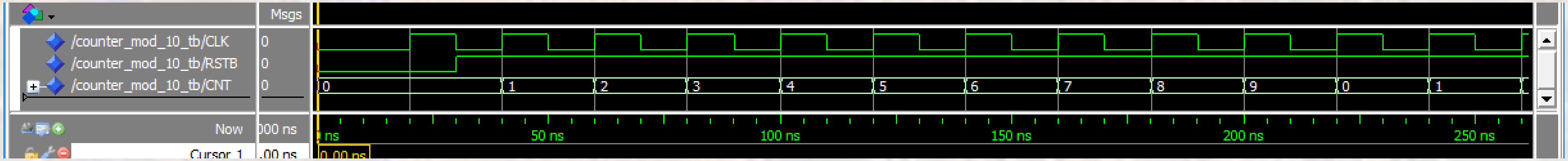
```
architecture testbench of counter_mod_10_tb is
   signal   CLK:  std_logic;
   signal   RSTB: std_logic;

   signal   CNT:  std_logic_vector(3 downto 0);
   constant PER:  time   := 20 ns;

   --------------------------------
   -- Component prototype
   ---------------------------------
   COMPONENT counter_mod_10
      PORT
      (
         i_rstb    :   IN STD_LOGIC;
         i_clk     :   IN STD_LOGIC;
         o_cnt     :   OUT STD_LOGIC_VECTOR(3 downto 0)
      );
   END COMPONENT;
   -----------------------------------

   begin

   ------------------------------------
   -- Device under test (DUT)
   ------------------------------------
   DUT: counter_mod_10
      port map(
               i_rstb => RSTB,
               i_clk => CLK,
               o_cnt => CNT
               );
```



```
   -------------------------------------
   -- Test processes
   -------------------------------------

   -- Clock process
   clock: process    -- note - no sensitivity list allowed
   begin
      CLK <= '0';
      wait for PER;
      infinite: loop
         CLK <= not CLK; wait for PER/2;
      end loop;
   end process;

   -- Reset process
   reset: process    -- note - no sensitivity list allowed
   begin
      RSTB <= '0'; wait for 1.5*PER;
      RSTB <= '1'; wait;
   end process reset;

   -- Run Process
   -- No run process for this design

   ------------------------------------------
   -- End test processes
   ------------------------------------------

end architecture;
```

Counters - HDL

- Mod 10 counter - verification

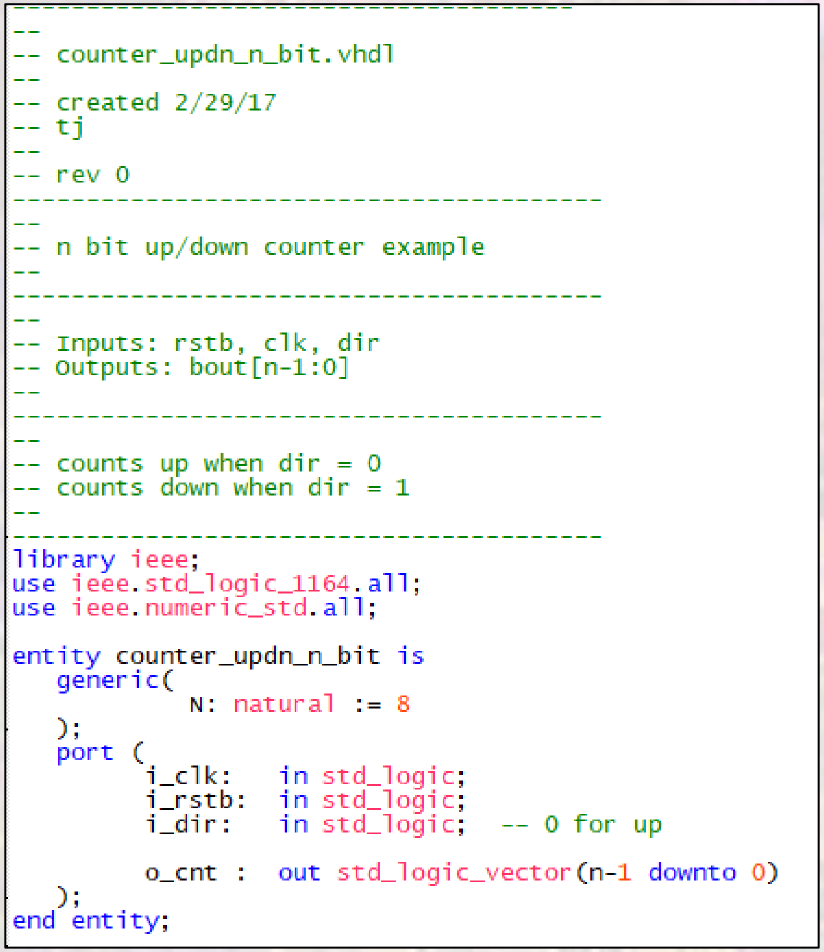


Counters - HDL

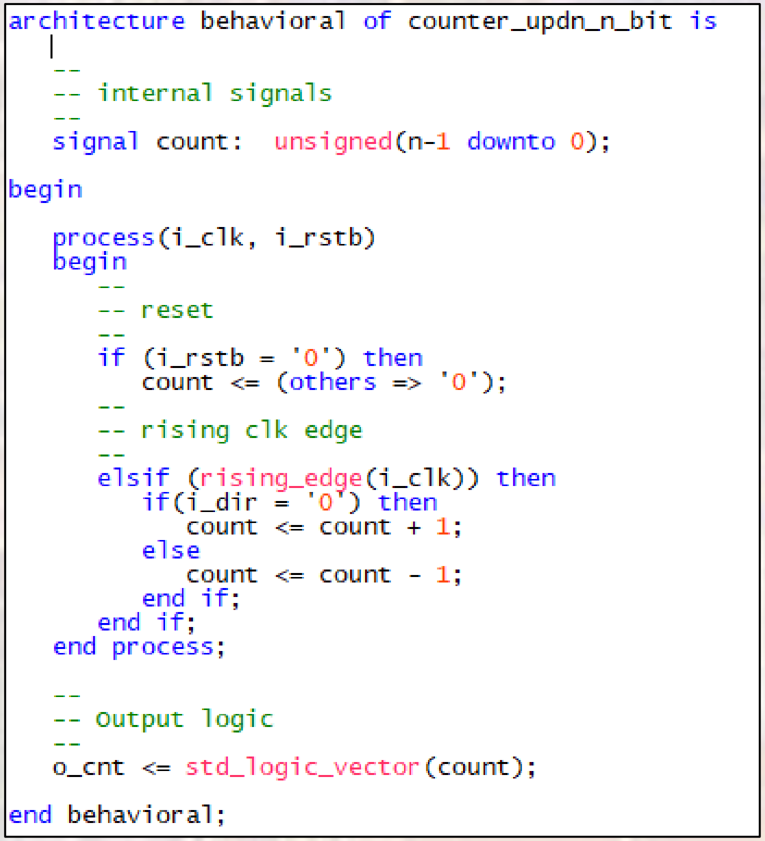
Up/Down counter

Counters - HDL

- up/down counter



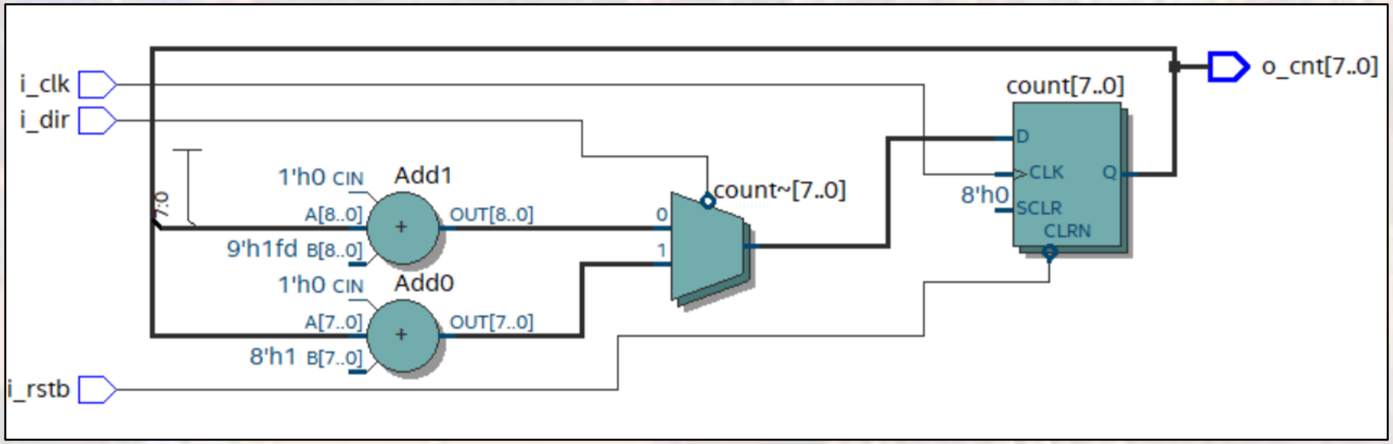
```vhdl
----------------------------------------
--
-- counter_updn_n_bit.vhdl
--
-- created 2/29/17
-- tj
--
-- rev 0
----------------------------------------
--
-- n bit up/down counter example
--
----------------------------------------
--
-- Inputs: rstb, clk, dir
-- Outputs: bout[n-1:0]
--
----------------------------------------
--
-- counts up when dir = 0
-- counts down when dir = 1
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter_updn_n_bit is
   generic(
            N: natural := 8
   );
   port (
         i_clk:   in std_logic;
         i_rstb:  in std_logic;
         i_dir:   in std_logic;   -- 0 for up

         o_cnt :  out std_logic_vector(n-1 downto 0)
   );
end entity;
```



```vhdl
architecture behavioral of counter_updn_n_bit is

   --
   -- internal signals
   --
   signal count:  unsigned(n-1 downto 0);

begin

   process(i_clk, i_rstb)
   begin
      --
      -- reset
      --
      if (i_rstb = '0') then
         count <= (others => '0');
      --
      -- rising clk edge
      --
      elsif (rising_edge(i_clk)) then
         if(i_dir = '0') then
            count <= count + 1;
         else
            count <= count - 1;
         end if;
      end if;
   end process;

   --
   -- Output logic
   --
   o_cnt <= std_logic_vector(count);

end behavioral;
```

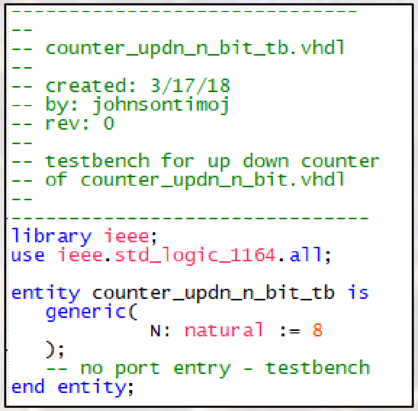
Counters - HDL

- up/down counter

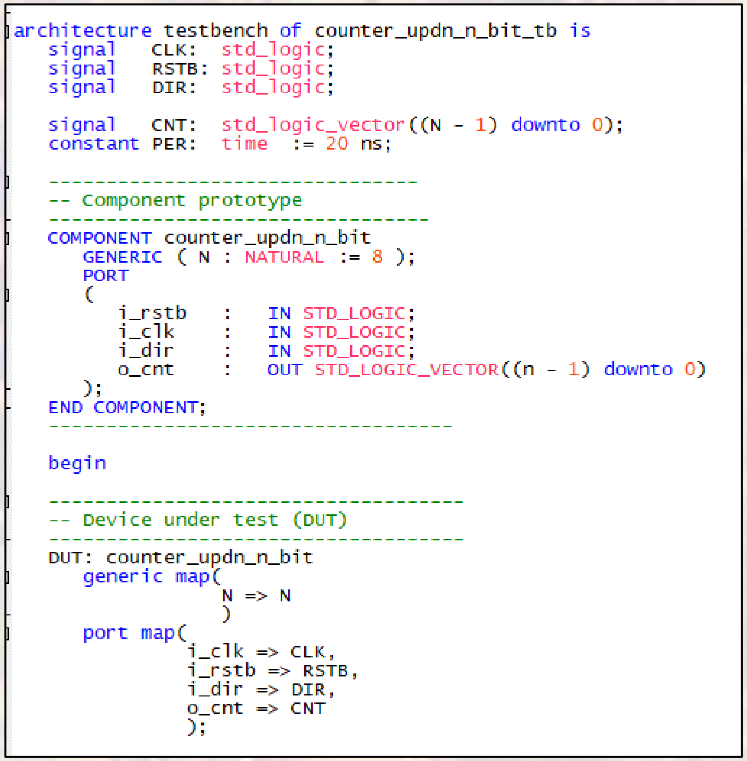


Counters - HDL

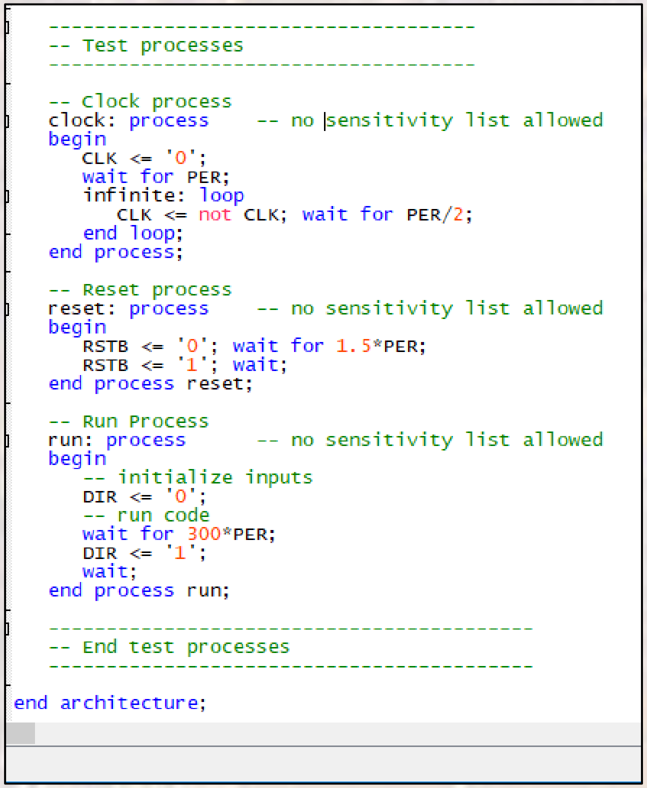
- up/down counter - testbench



```
--------------------------------
--
-- counter_updn_n_bit_tb.vhdl
--
-- created: 3/17/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for up down counter
-- of counter_updn_n_bit.vhdl
--
---------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_updn_n_bit_tb is
   generic(
            N: natural := 8
   );
   -- no port entry - testbench
end entity;
```



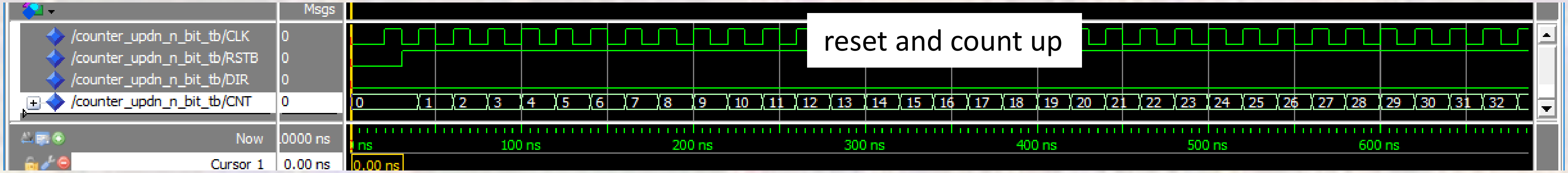
```
architecture testbench of counter_updn_n_bit_tb is
   signal   CLK:  std_logic;
   signal   RSTB: std_logic;
   signal   DIR:  std_logic;

   signal   CNT:  std_logic_vector((N - 1) downto 0);
   constant PER:  time  := 20 ns;

   -----------------------------------
   -- Component prototype
   ------------------------------------
   COMPONENT counter_updn_n_bit
      GENERIC ( N : NATURAL := 8 );
      PORT
      (
         i_rstb    :    IN STD_LOGIC;
         i_clk     :    IN STD_LOGIC;
         i_dir     :    IN STD_LOGIC;
         o_cnt     :    OUT STD_LOGIC_VECTOR((n - 1) downto 0)
      );
   END COMPONENT;
   -------------------------------------

   begin

   -------------------------------------
   -- Device under test (DUT)
   -------------------------------------
   DUT: counter_updn_n_bit
      generic map(
                  N => N
                  )
      port map(
               i_clk => CLK,
               i_rstb => RSTB,
               i_dir => DIR,
               o_cnt => CNT
               );
```

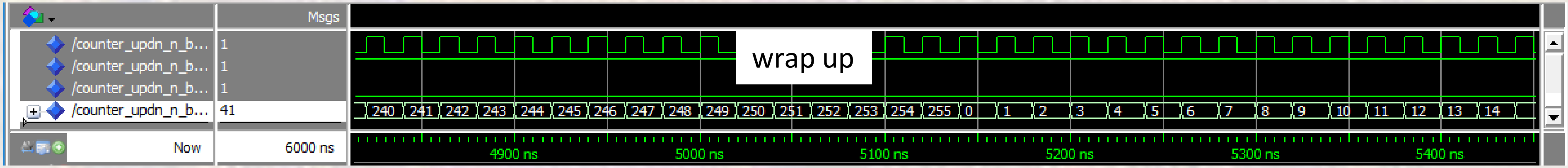


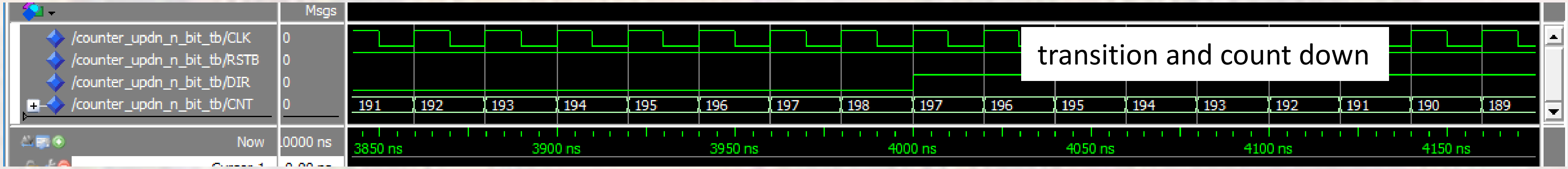
```
   -------------------------------------
   -- Test processes
   -------------------------------------

   -- Clock process
   clock: process     -- no sensitivity list allowed
   begin
      CLK <= '0';
      wait for PER;
      infinite: loop
         CLK <= not CLK; wait for PER/2;
      end loop;
   end process;

   -- Reset process
   reset: process     -- no sensitivity list allowed
   begin
      RSTB <= '0'; wait for 1.5*PER;
      RSTB <= '1'; wait;
   end process reset;

   -- Run Process
   run: process       -- no sensitivity list allowed
   begin
      -- initialize inputs
      DIR <= '0';
      -- run code
      wait for 300*PER;
      DIR <= '1';
      wait;
   end process run;

   ----------------------------------------
   -- End test processes
   ----------------------------------------

end architecture;
```

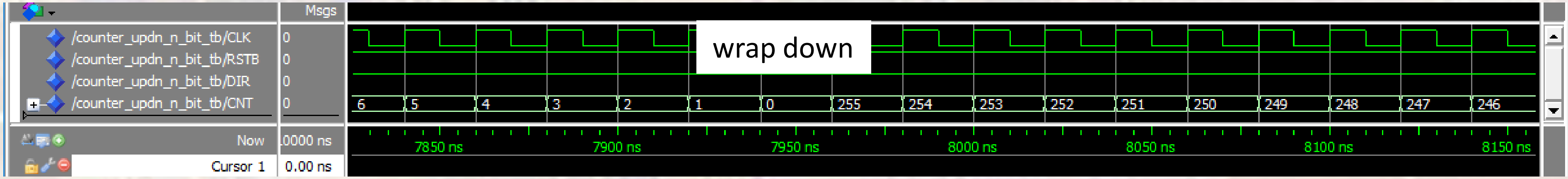
Counters - HDL

- up/down counter - verification


reset and count up
Msgs
/counter_updn_n_bit_tb/CLK 0
/counter_updn_n_bit_tb/RSTB 0
/counter_updn_n_bit_tb/DIR 0
/counter_updn_n_bit_tb/CNT 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Now 0000 ns
Cursor 1 0.00 ns
ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns
0.00 ns


wrap up


transition and count down
Msgs
/counter_updn_n_bit_tb/CLK 0
/counter_updn_n_bit_tb/RSTB 0
/counter_updn_n_bit_tb/DIR 0
/counter_updn_n_bit_tb/CNT 0
191 192 193 194 195 196 197 198 197 196 195 194 193 192 191 190 189
Now 0000 ns
3850 ns 3900 ns 3950 ns 4000 ns 4050 ns 4100 ns 4150 ns


wrap down
Msgs
/counter_updn_n_bit_tb/CLK 0
/counter_updn_n_bit_tb/RSTB 0
/counter_updn_n_bit_tb/DIR 0
/counter_updn_n_bit_tb/CNT 0
6 5 4 3 2 1 0 255 254 253 252 251 250 249 248 247 246
Now 0000 ns
Cursor 1 0.00 ns
7850 ns 7900 ns 7950 ns 8000 ns 8050 ns 8100 ns 8150 ns