# Counters

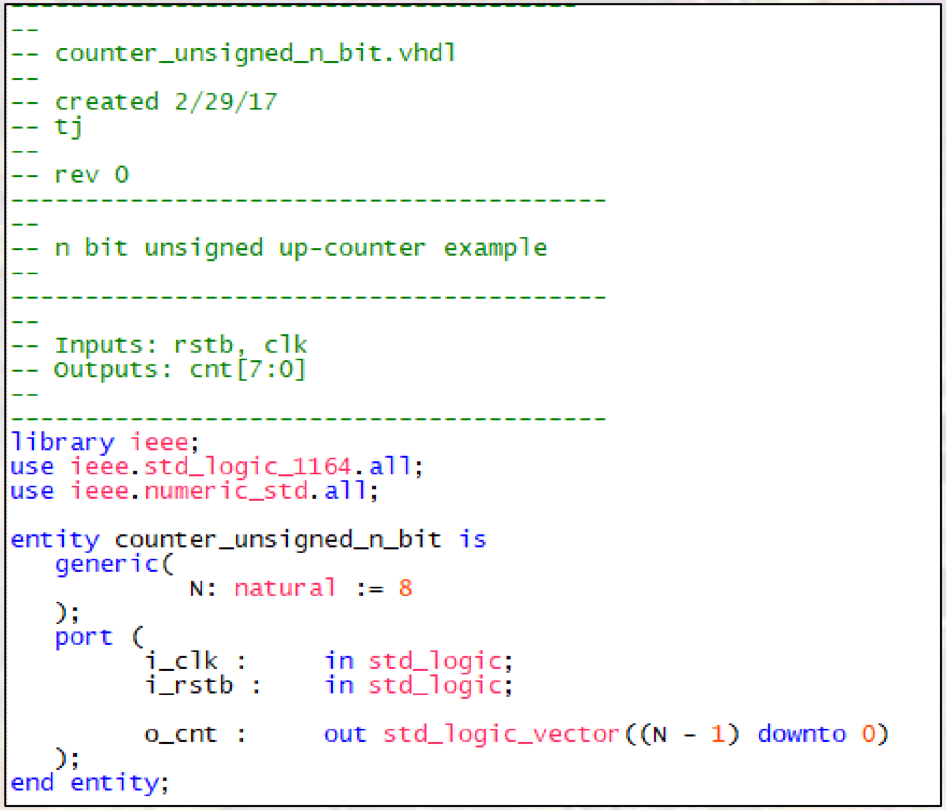
Last updated 2/8/24

# Counters

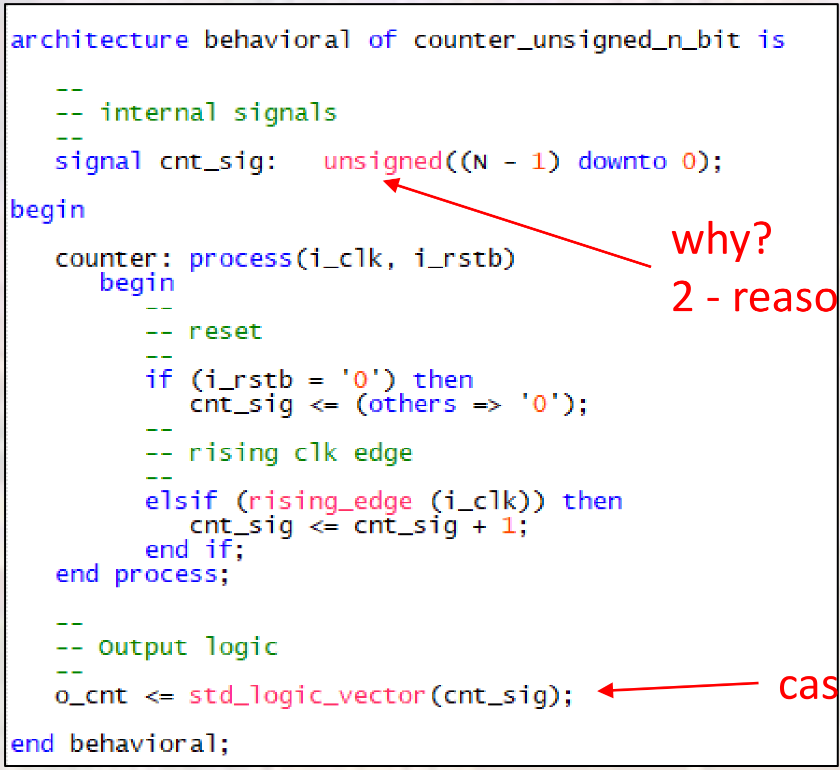
- Counters
  - Used as individual blocks
  - Used inside a block
    - Timers – delays, time-out, …
    - Collectors – cash, presses, …
  - Variations
    - Up, Down, Up-Down
    - Wrapping, non-wrapping
    - Modulo
    - Fixed size, n-bit
    - Fixed increment, m-increment

# Counters

- Counter - n bit - unsigned



```
---------------------------------------
--
-- counter_unsigned_n_bit.vhdl
--
-- created 2/29/17
-- tj
--
-- rev 0
----------------------------------------
--
-- n bit unsigned up-counter example
--
----------------------------------------
--
-- Inputs: rstb, clk
-- Outputs: cnt[7:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter_unsigned_n_bit is
   generic(
            N: natural := 8
   );
   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

   counter: 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;
```

why?

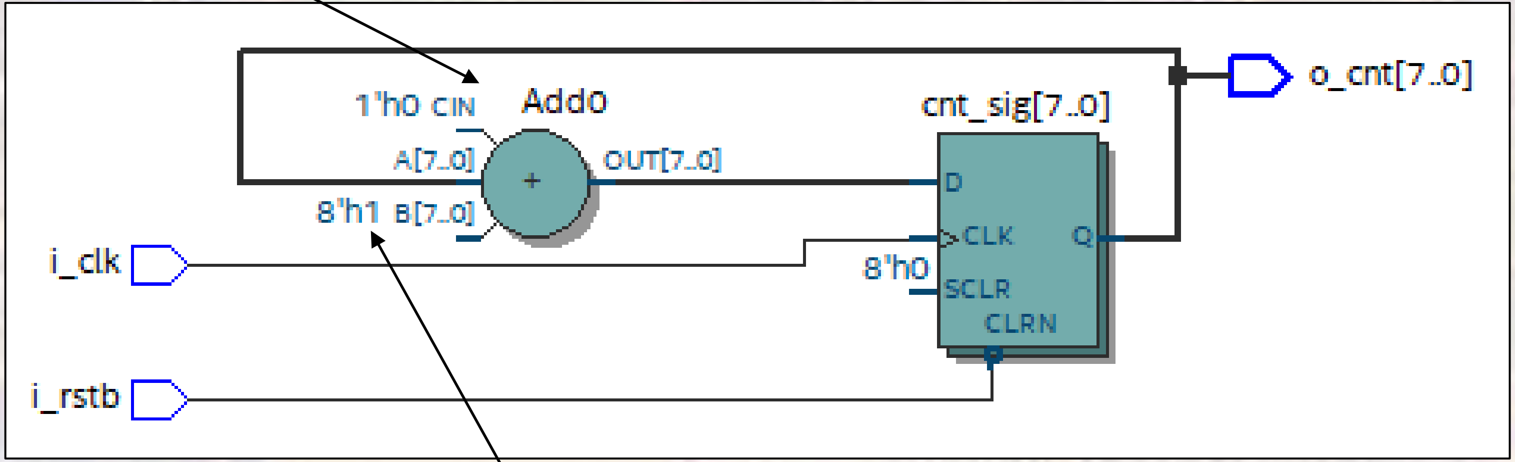
2 - reasons

cast

# Counters

- Counter - n bit – unsigned – (default)

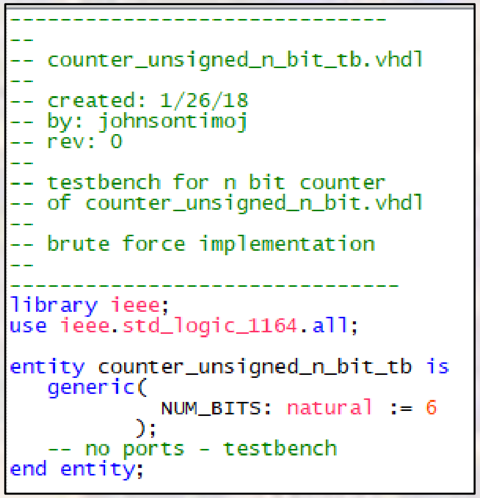
adds 1



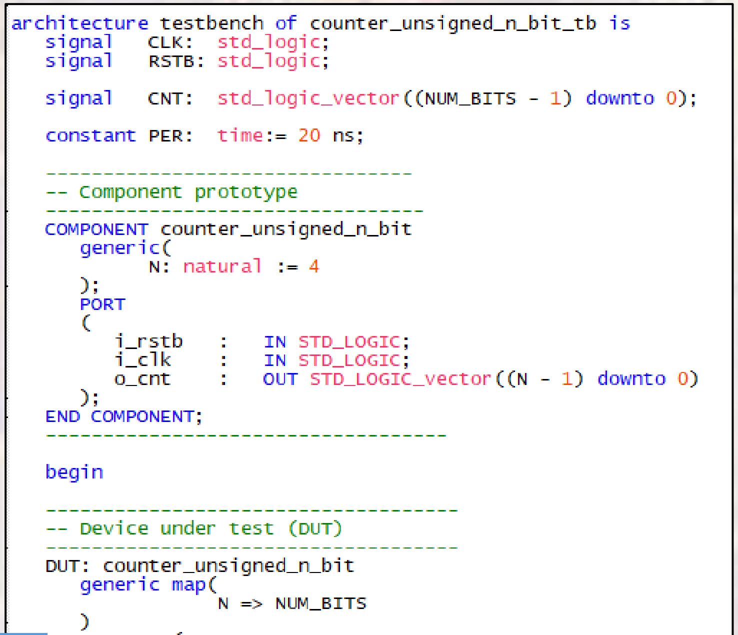
Format
num_bits ‘ format value
8’h1 → 8 bits, hex, value 1

# Counters

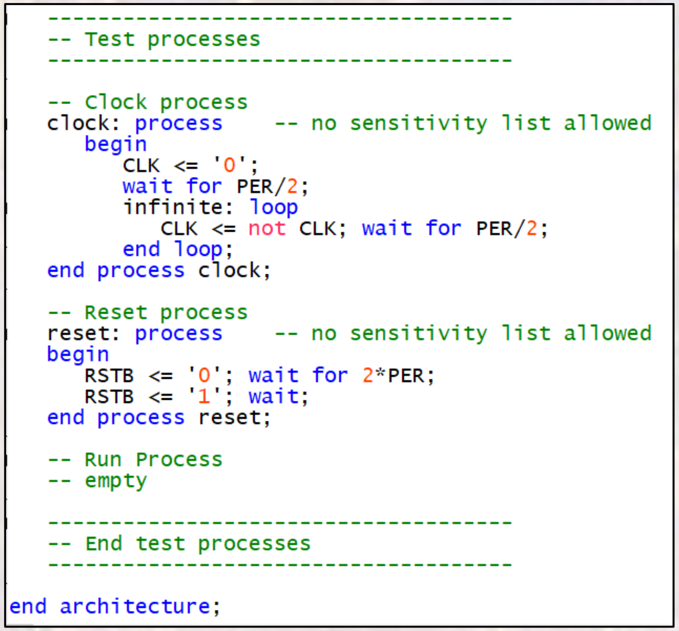
- Counter - n bit – unsigned
  - Testbench - 6 bit version



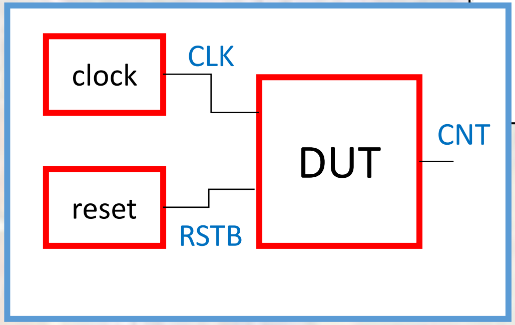
```
------------------------------
--
-- counter_unsigned_n_bit_tb.vhdl
--
-- created: 1/26/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for n bit counter
-- of counter_unsigned_n_bit.vhdl
--
-- brute force implementation
--
-------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_unsigned_n_bit_tb is
   generic(
            NUM_BITS: natural := 6
         );
   -- no ports - 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((NUM_BITS - 1) downto 0);

   constant PER:  time:= 20 ns;

   -------------------------------
   -- Component prototype
   --------------------------------
   COMPONENT counter_unsigned_n_bit
      generic(
            N: natural := 4
      );
      PORT
      (
         i_rstb    :   IN STD_LOGIC;
         i_clk     :   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 => NUM_BITS
      )
      port map(
               i_rstb => RSTB,
               i_clk => CLK,
               o_cnt => CNT
      );
```



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

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

   -- Reset process
   reset: process    -- no sensitivity list allowed
   begin
      RSTB <= '0'; wait for 2*PER;
      RSTB <= '1'; wait;
   end process reset;

   -- Run Process
   -- empty

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

end architecture;
```

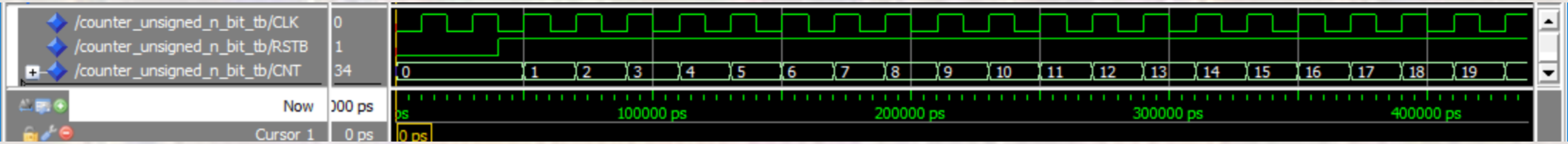


# Counters

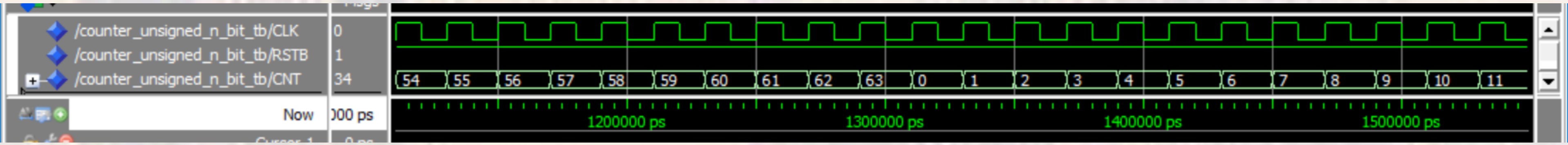
- Counter - n bit – unsigned
  - Testbench - 6 bit version

reset

counting



wrap



# Counters

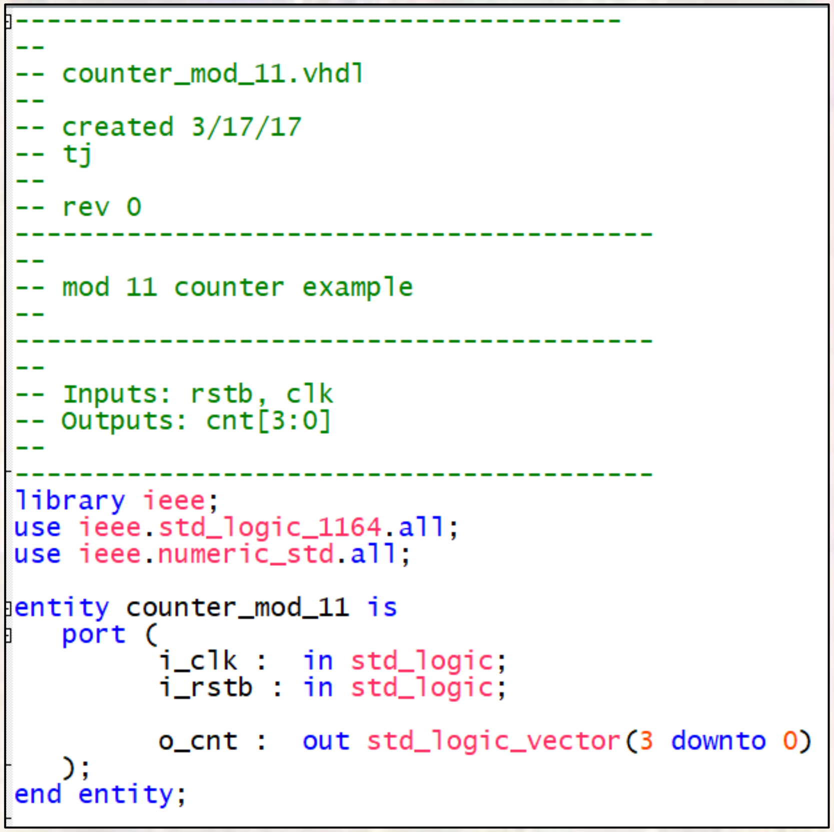
## Mod counter

MOD x counts x times

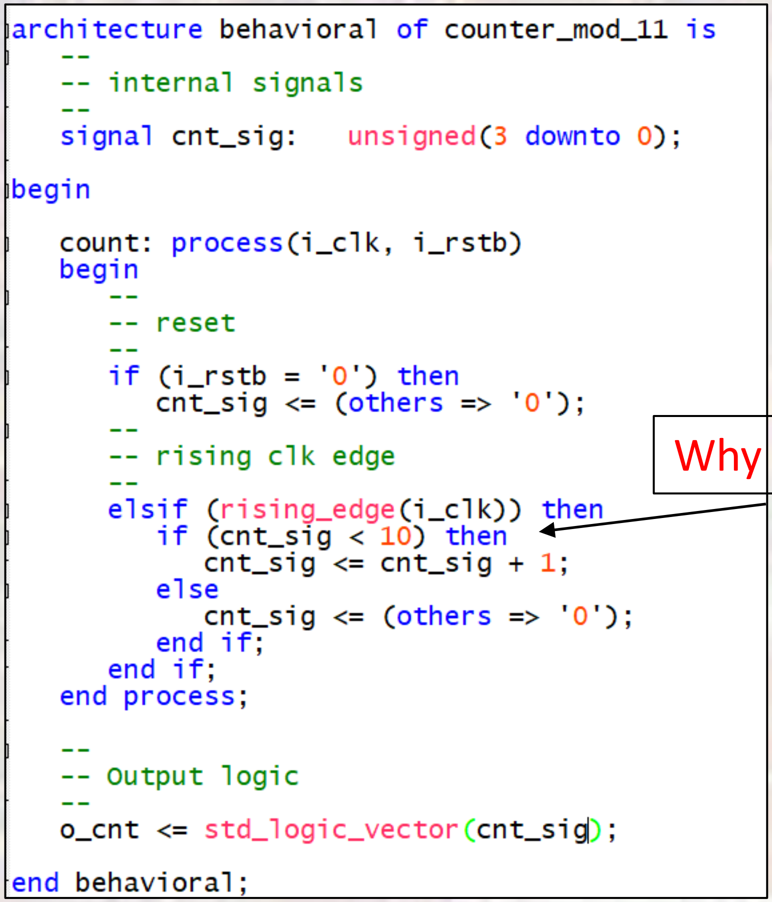
Range 0 – (x – 1)

# Counters

- Mod 11 counter
  - Mod 11 only needs to compare 4 bits, no need to optimize the roll over test



```
--------------------------------------
--
-- counter_mod_11.vhdl
--
-- created 3/17/17
-- tj
--
-- rev 0
----------------------------------------
--
-- mod 11 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_11 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_11 is
   --
   -- internal signals
   --
   signal cnt_sig:   unsigned(3 downto 0);

begin

   count: 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
         if (cnt_sig < 10) then
            cnt_sig <= cnt_sig + 1;
         else
            cnt_sig <= (others => '0');
         end if;
      end if;
   end process;

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

end behavioral;
```

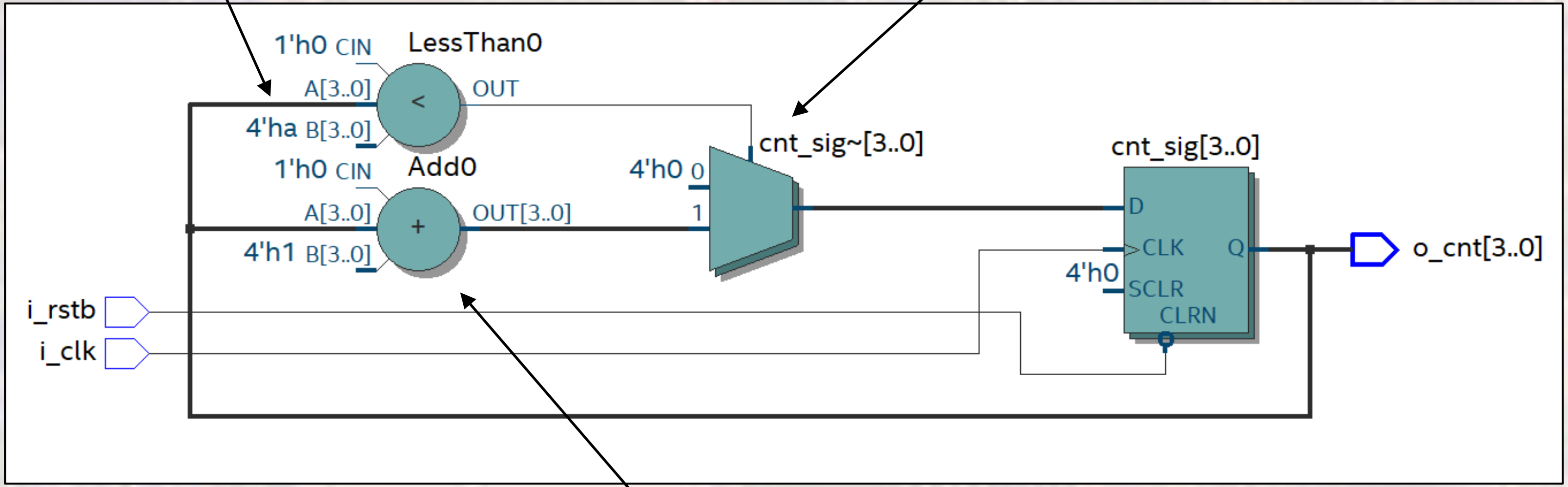
Why < 10

# Counters

- Mod 11 counter

Special adder (subtracts) and check for < 0
cnt – 10 (4bit, hex a)

control selection of
cnt+1 or 0

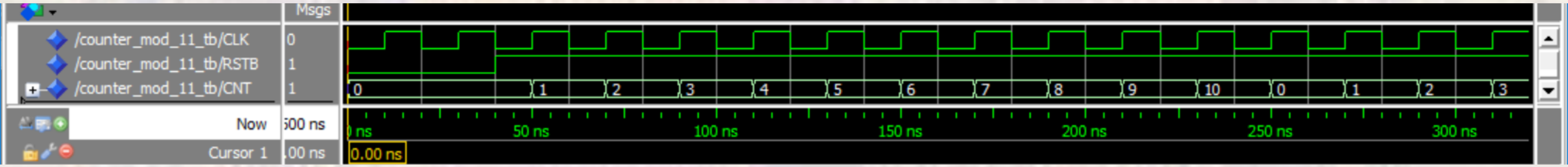


add 1

# Counters

- Mod 11 counter
  - Testbench results

reset counting mod11 wrap

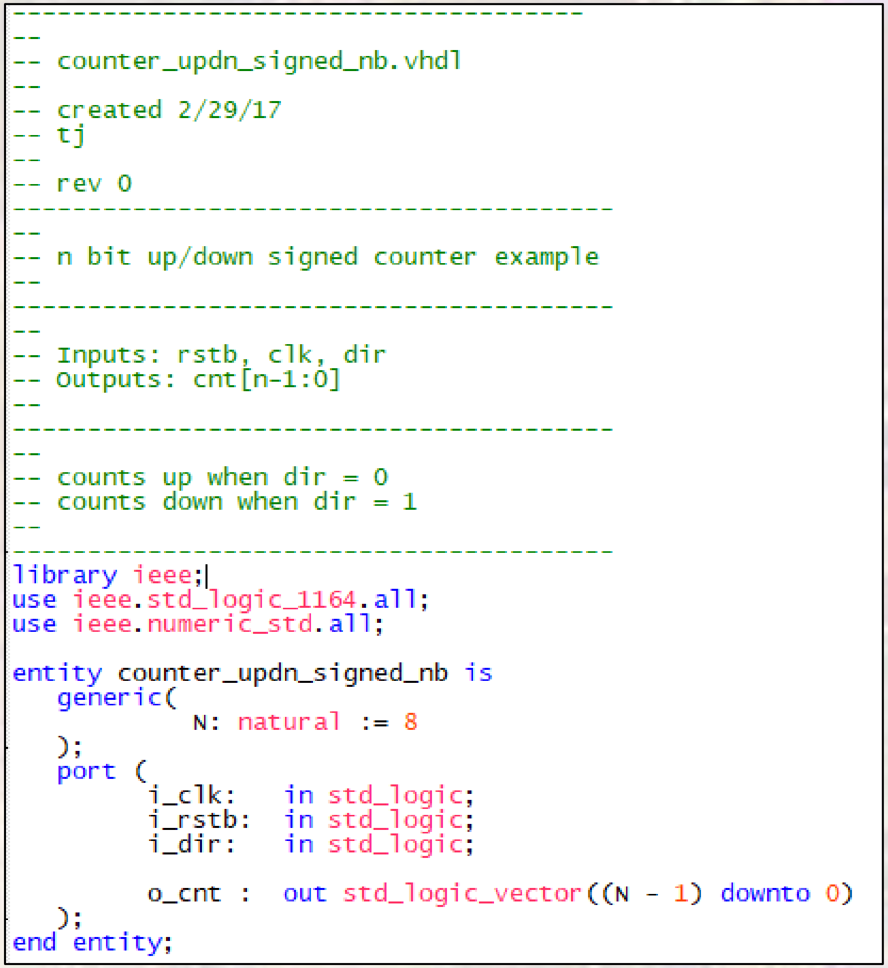


# Counters

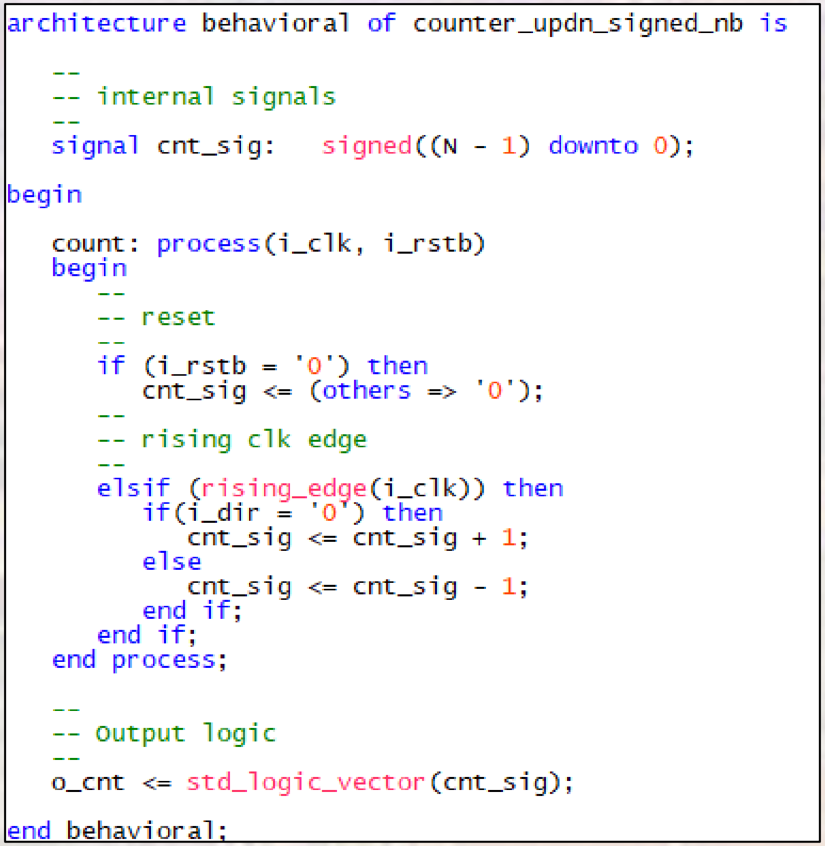
Up/Down counter

# Counters

- up/down signed n-bit counter



```vhdl
--------------------------------------
--
-- counter_updn_signed_nb.vhdl
--
-- created 2/29/17
-- tj
--
-- rev 0
----------------------------------------
--
-- n bit up/down signed counter example
--
----------------------------------------
--
-- Inputs: rstb, clk, dir
-- Outputs: cnt[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_signed_nb is
   generic(
            N: natural := 8
   );
   port (
         i_clk:   in std_logic;
         i_rstb:  in std_logic;
         i_dir:   in std_logic;

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



```vhdl
architecture behavioral of counter_updn_signed_nb is

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

begin

   count: 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
         if(i_dir = '0') then
            cnt_sig <= cnt_sig + 1;
         else
            cnt_sig <= cnt_sig - 1;
         end if;
      end if;
   end process;

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

end behavioral;
```

# Counters

- up/down signed n-bit counter (default 8 bit)

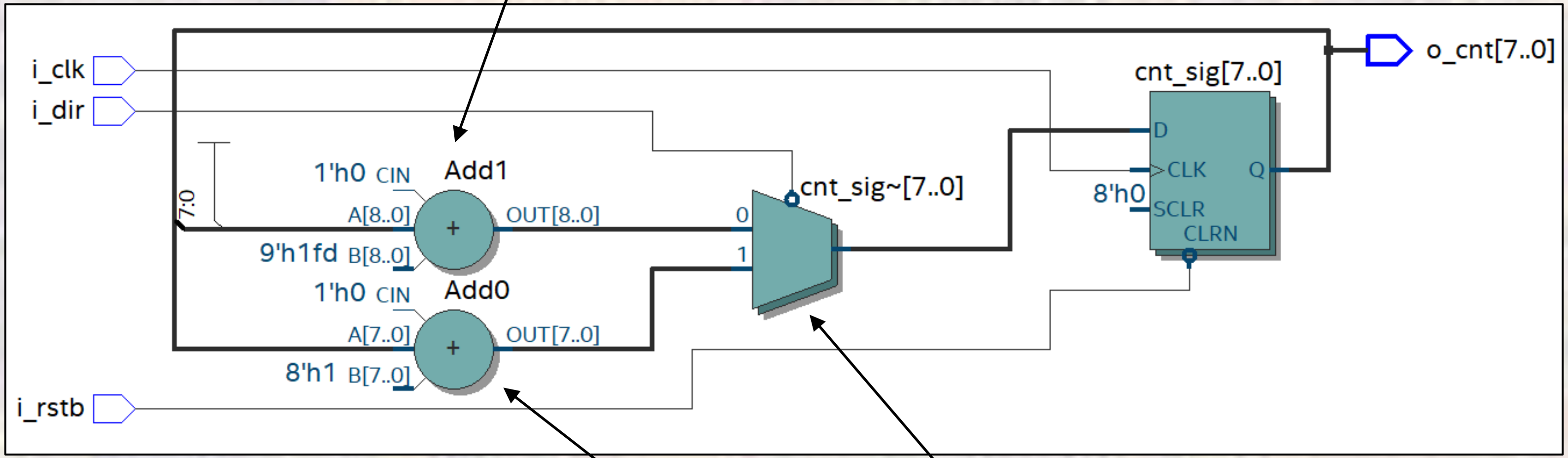
subtract 1

|  |  |  |
|---|---|---|
| cnt & 1 | → | a bcde fgh1 |
| 1fd | → | + 1 1111 1101 |
|  |  | n opqr stuv |

$\underbrace{\text{n opqr stuv}}_{\text{cnt - 1}}$

subtract 1

|  |  |
|---|---|
| Shift and concatenate 1 | 2x + 1 |
| Add -3 | + -3 |
|  | 2x – 2 |
| Only select upper bits → /2 | x - 1 |

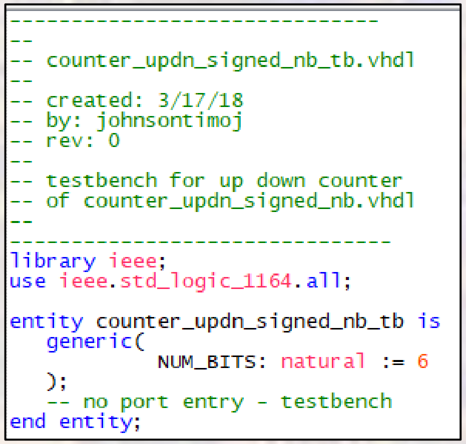


add 1

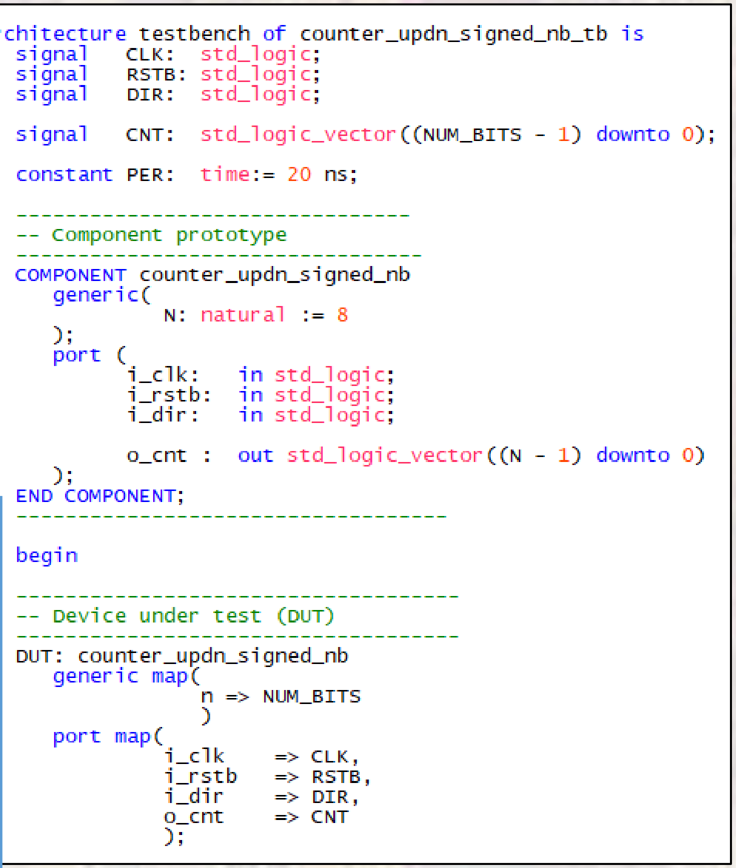
select + or -

# Counters

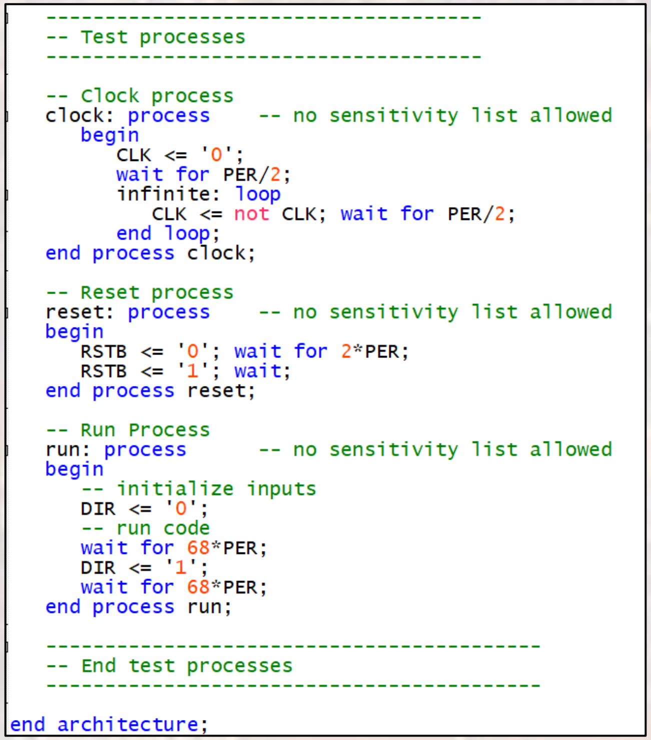
- up/down signed n-bit counter
  - Testbench - 6 bit test



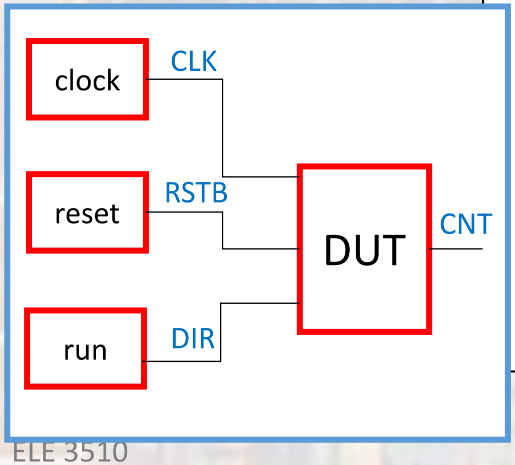
```vhdl
-----------------------------
--
-- counter_updn_signed_nb_tb.vhdl
--
-- created: 3/17/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for up down counter
-- of counter_updn_signed_nb.vhdl
--
------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_updn_signed_nb_tb is
   generic(
            NUM_BITS: natural  := 6
   );
   -- no port entry - testbench
end entity;
```



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

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

   constant PER:  time:= 20 ns;

   --------------------------------
   -- Component prototype
   ---------------------------------
   COMPONENT counter_updn_signed_nb
      generic(
               N: natural  := 8
      );
      port (
            i_clk:   in std_logic;
            i_rstb:  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_signed_nb
      generic map(
               n => NUM_BITS
               )

      port map(
               i_clk    => CLK,
               i_rstb   => RSTB,
               i_dir    => DIR,
               o_cnt    => CNT
               );
```



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

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

   -- Reset process
   reset: process    -- no sensitivity list allowed
   begin
      RSTB <= '0'; wait for 2*PER;
      RSTB <= '1'; wait;
   end process reset;

   -- Run Process
   run: process      -- no sensitivity list allowed
   begin
      -- initialize inputs
      DIR <= '0';
      -- run code
      wait for 68*PER;
      DIR <= '1';
      wait for 68*PER;
   end process run;

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

end architecture;
```

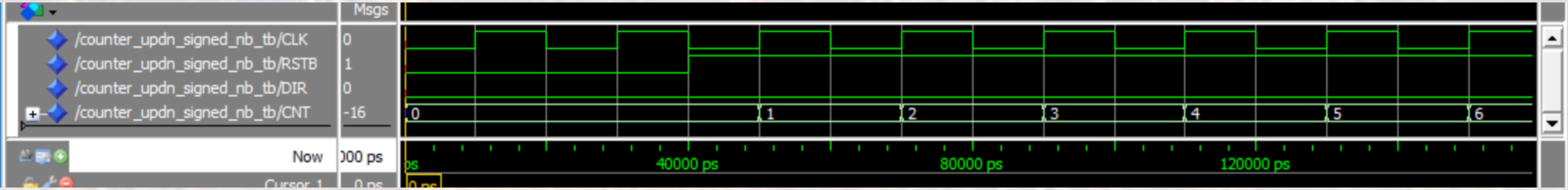


# Counters

- up/down signed n-bit counter
  - Testbench - 6 bit

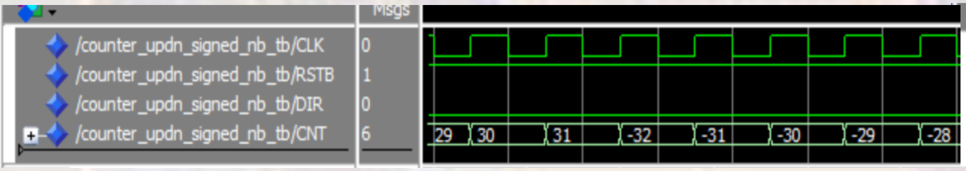
reset

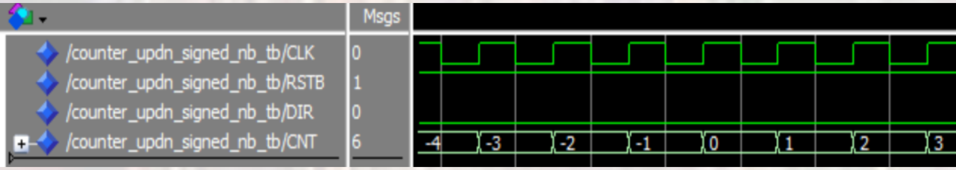
counting up


Msgs
/counter_updn_signed_nb_tb/CLK 0
/counter_updn_signed_nb_tb/RSTB 1
/counter_updn_signed_nb_tb/DIR 0
/counter_updn_signed_nb_tb/CNT -16
0
1
2
3
4
5
6
Now 000 ps
ps
40000 ps
80000 ps
120000 ps
Cursor 1 0 ns
0 ns

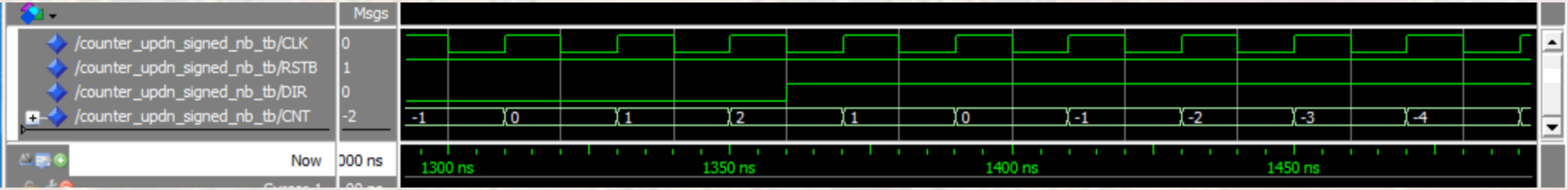
transition to negative

transition to positive


Msgs
/counter_updn_signed_nb_tb/CLK 0
/counter_updn_signed_nb_tb/RSTB 1
/counter_updn_signed_nb_tb/DIR 0
/counter_updn_signed_nb_tb/CNT 6
29
30
31
-32
-31
-30
-29
-28


Msgs
/counter_updn_signed_nb_tb/CLK 0
/counter_updn_signed_nb_tb/RSTB 1
/counter_updn_signed_nb_tb/DIR 0
/counter_updn_signed_nb_tb/CNT 6
-4
-3
-2
-1
0
1
2
3

counting down


Msgs
/counter_updn_signed_nb_tb/CLK 0
/counter_updn_signed_nb_tb/RSTB 1
/counter_updn_signed_nb_tb/DIR 0
/counter_updn_signed_nb_tb/CNT -2
-1
0
1
2
1
0
-1
-2
-3
-4
Now 000 ns
1300 ns
1350 ns
1400 ns
1450 ns