Code example: Average

Jordi Bonet

iTIC http://ocw.itic.cat

February 2020

average.vhd (1/3)

```
  library ieee;
use ieee.std_logic_1164.all, ieee.numeric_std.all;
--use ieee.fixed_pkg.all;

entity average is
  generic (WIDTH: natural:=4); -- width of a,b,c,avg
  port( clk    : in std_logic;
        a,b,c  : in signed(WIDTH-1 downto 0);
        avg    : out signed(WIDTH-1 downto 0));
end;

architecture combinational of average is
  signal ab,abc : signed(a'length-1 downto 0);
begin
  ab<=a+b;
  abc<=ab+c;
  avg<=abc/3;
end;
```

average.vhd (2/3)

```
architecture pipeline of average is
  -- outputs of combinational circuits
  signal ab,abc,pre_avg : signed(a'length-1 downto 0);
  --outputs of registers
  signal saved_c,saved_ab,saved_abc :
                          signed(a'length-1 downto 0);

begin

  combinational_circuit_1: ab<=a+b;

  reg1:process(clk)
  begin
    if rising_edge(clk) then
      saved_ab<=ab;
      saved_c<=c;
    end if;
  end process;
```

average.vhd (3/3)

```
   combinational_circuit_2: abc<=saved_ab+saved_c;

  reg2:process(clk)
  begin
    if rising_edge(clk) then
      saved_abc<=abc;
    end if;
  end process;

  combinational_circuit_3: pre_avg<=saved_abc/3;

  reg3:process(clk)
  begin
    if rising_edge(clk) then
      avg<=pre_avg;
    end if;
  end process;

end;
```

average_tb.vhd (1/3)

```
library ieee;
use ieee.std_logic_1164.all, ieee.numeric_std.all;

entity average_tb is
  generic (t_WIDTH: natural:=4);
end;

architecture behav of average_tb is
  signal t_a , t_b , t_c, t_avg : signed(t_WIDTH-1 downto 0);
  signal t_clk : std_logic;
begin
  dut: entity work.average(combinational)
    generic map (WIDTH=>t_WIDTH)
    port map (a=>t_a, b=>t_b, c=>t_c, clk=>t_clk, avg=>t_avg);
```

average_tb.vhd (2/3)

```
--valid for all architectures (gtkwave)
abc_process: process
  procedure apply_test
    (p_t_a, p_t_b, p_t_c : in integer) is
  begin
    t_a<=to_signed(p_t_a,t_a'length);
    t_b<=to_signed(p_t_b,t_b'length);
    t_c<=to_signed(p_t_c,t_c'length);
    wait for 1 ms;
  end procedure;
begin
  apply_test(1,1,1);
  apply_test(6,0,0);
  apply_test(-1,0,0);
  apply_test(5,4,3);
  apply_test(5,3,-5);
  apply_test(7,7,-7);
  wait;
end process;
```

average_tb.vhd (3/3)

```
  --used in combinational architecture
assert_combinational_process: process
  --function returns boolean to use inside assert command;
  --impure functions can use global variables
  --wait not allowed inside a function
  impure function f_boolean
    (p_t_a,p_t_b,p_t_c,p_t_avg: integer)
    return boolean is
  begin
  return t_a=to_signed(p_t_a,t_a'length)
    and t_b=to_signed(p_t_b,t_b'length)
    and t_c=to_signed(p_t_c,t_c'length)
    and t_avg=to_signed(p_t_avg,t_avg'length);
    end function;
begin
  wait on t_a, t_b, t_c;
  wait for 100 us;
  assert
    f_boolean(1,1,1,1) or f_boolean(6,0,0,2) or
    f_boolean(-1,0,0,0) or f_boolean(5,4,3,4) or
    f_boolean(5,3,-5,1) or f_boolean(7,7,-7,2)
    report "bad combinational computation" severity warning;
end process;
```

average_pipeline_tb.vhd (1/3)

```
library ieee;
use ieee.std_logic_1164.all, ieee.numeric_std.all;

entity average_pipeline_tb is
  generic (t_WIDTH: natural:=5);
end;

architecture behav of average_pipeline_tb is
  signal t_a , t_b , t_c, t_avg : signed(t_WIDTH-1 downto 0);
  signal t_clk : std_logic;
begin
  dut: entity work.average(pipeline)
    generic map (WIDTH=>t_WIDTH)
    port map (a=>t_a, b=>t_b, c=>t_c, clk=>t_clk, avg=>t_avg);

  --NOT used in combinational architecture
  clk_process: process
  begin
    t_clk <= '0';
    wait for 300 us;
    for i in 1 to 20 loop
      t_clk <= not t_clk;  wait for 500 us;
    end loop;
    wait;
  end process;
```

average_pipeline_tb.vhd (2/3)

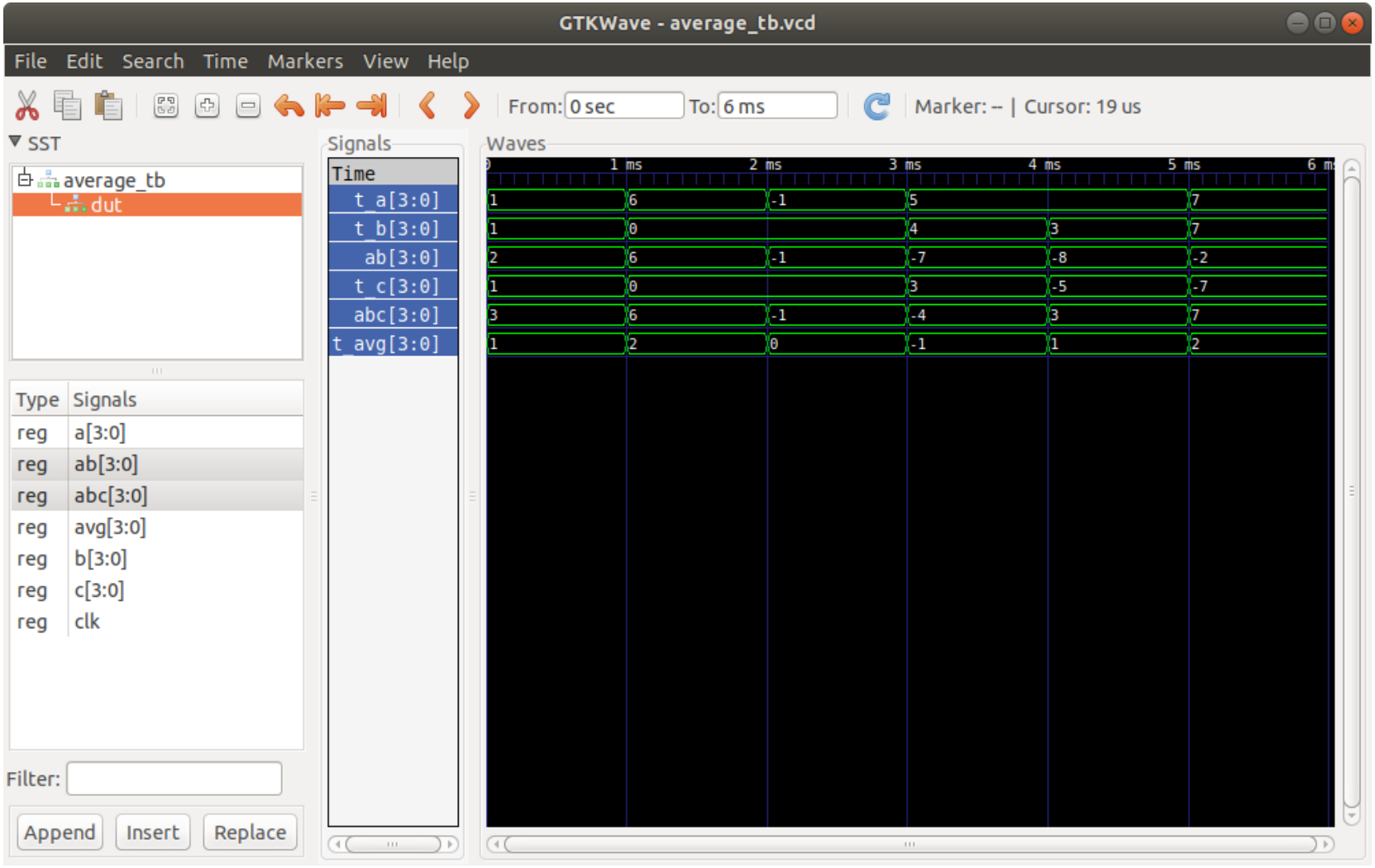
```
--valid for all architectures (gtkwave)
abc_process: process
  procedure apply_test
    (p_t_a, p_t_b, p_t_c : in integer) is
  begin
    t_a<=to_signed(p_t_a,t_a'length);
    t_b<=to_signed(p_t_b,t_b'length);
    t_c<=to_signed(p_t_c,t_c'length);
    wait for 1 ms;
  end procedure;
begin
  apply_test(1,1,1);
  apply_test(6,0,0);
  apply_test(-1,0,0);
  apply_test(5,4,3);
  apply_test(5,3,-5);
  apply_test(7,7,-7);
  wait;
end process;
```

average_pipeline_tb.vhd (3/3)

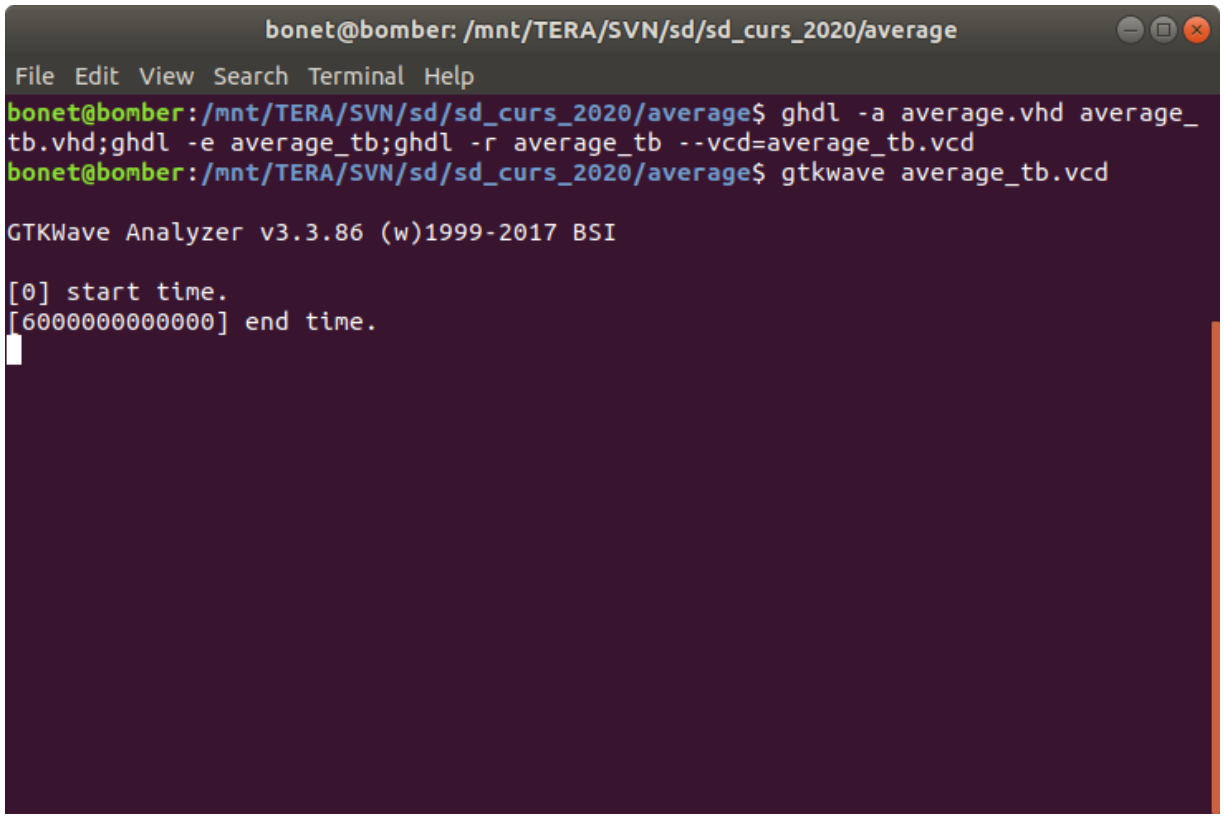
```
  --used in pipeline architecture
  assert_pipeline_process: process
    procedure p_assert
      (p_t_avg: in integer) is
    begin
      wait until rising_edge(t_clk);--3rd clk of nth data
      wait for 100 us;
      assert t_avg=to_signed(p_t_avg,t_avg'length)
      report "bad pipeline computation" severity warning;
    end procedure;
  begin
    wait until rising_edge(t_clk);--1st clk of 1st data
    wait until rising_edge(t_clk);--2nd clk of 1st data
    p_assert(1);--3rd rising_edge clk of 1st data
    p_assert(2);--3rd rising_edge clk of 2nd data
    p_assert(0);--3rd rising_edge clk of 3rd data
    p_assert(4);--3rd rising_edge clk of 4th data
    p_assert(1);--3rd rising_edge clk of 5th data
    p_assert(2);--3rd rising_edge clk of 6th data
    p_assert(2);-- repeating 6th data
    p_assert(3);-- repeating 6th data; error on purpose
    wait;
  end process;

end architecture;
```

ghdl : average_tb.vhd, width=4

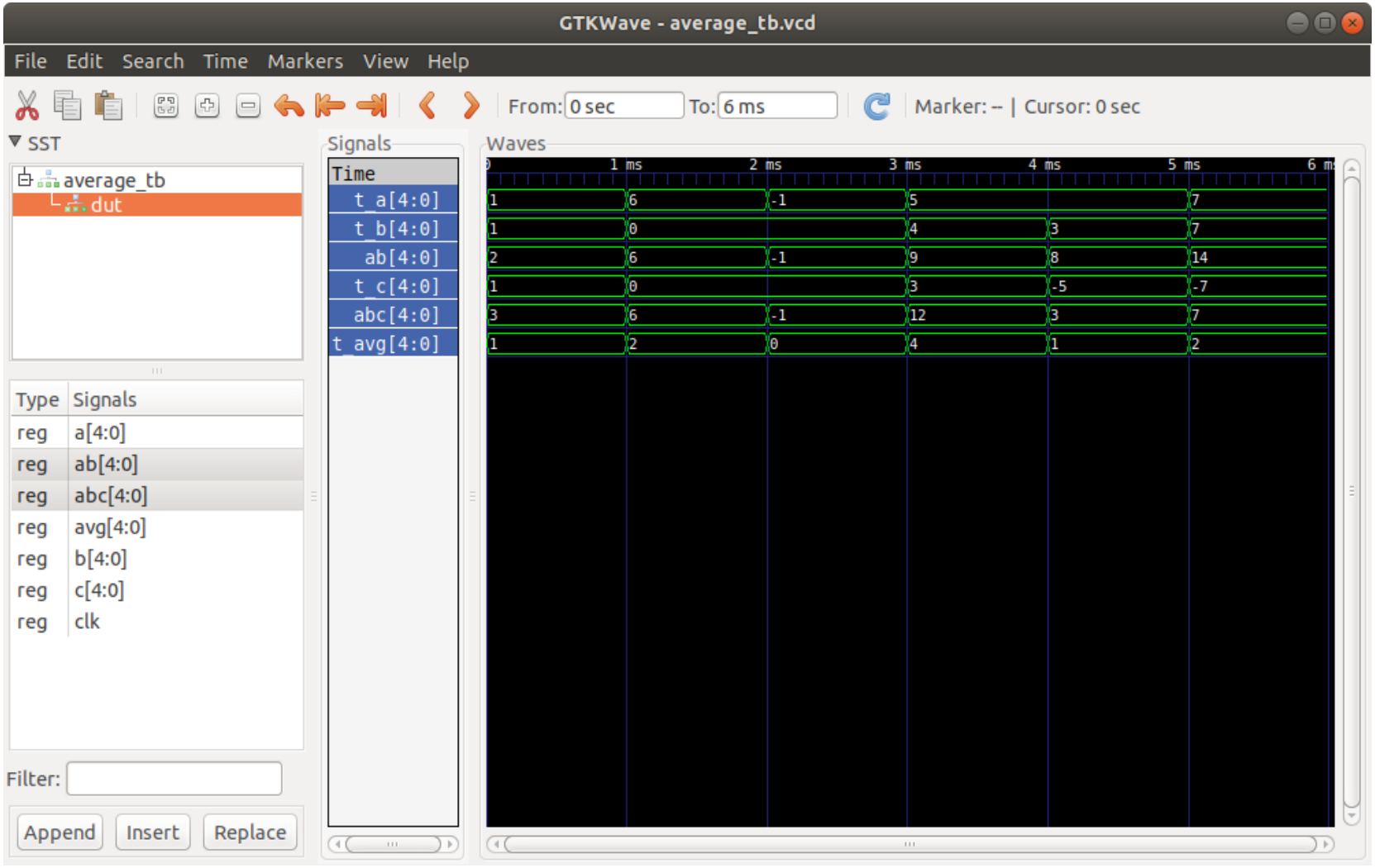
gtkwave : average_tb.vhd, width=4



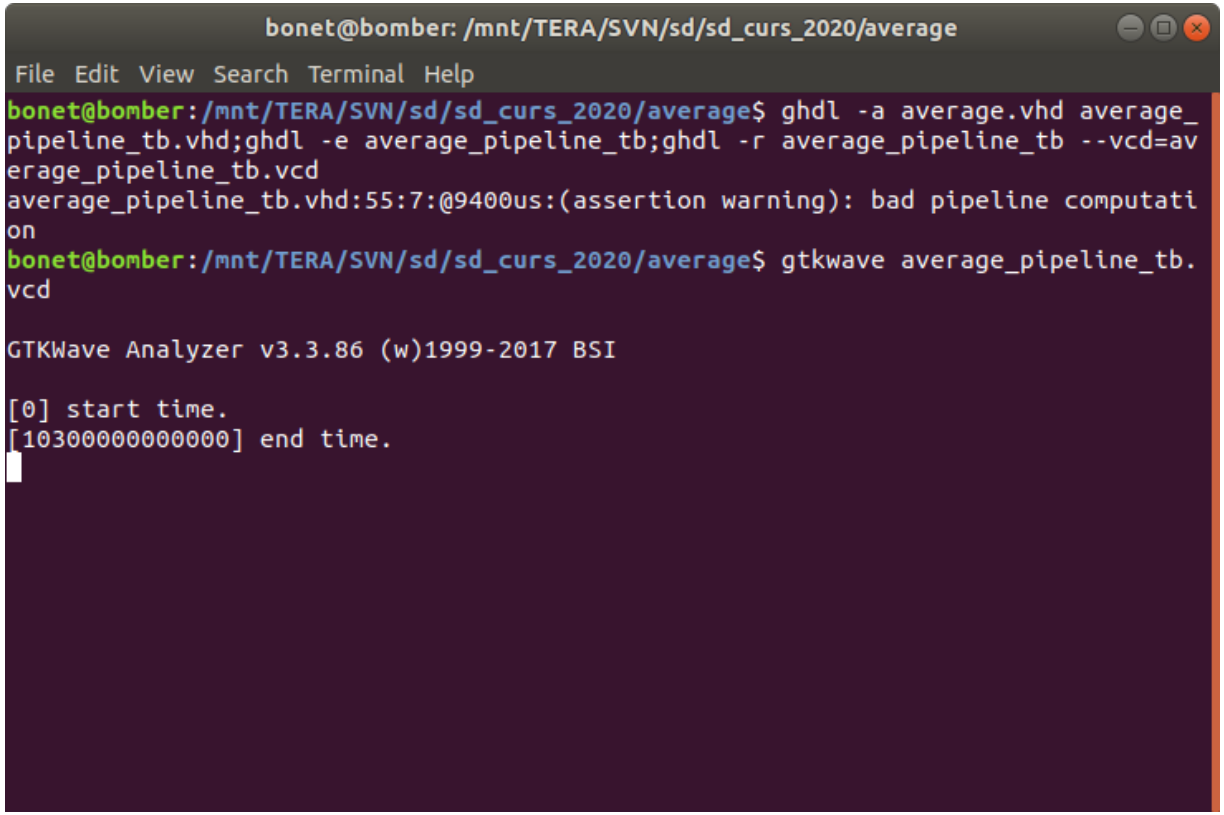
ghdl : average_tb.vhd, width=5



gtkwave : average_tb.vhd, width=5



ghdl : average_tb_pipeline.vhd, width=5



gtkwave : average_tb_pipeline.vhd, width=5