

VHDL Adder Subtractor Example

Last updated 3/25/26

VHDL Adder / Subtractor Examples

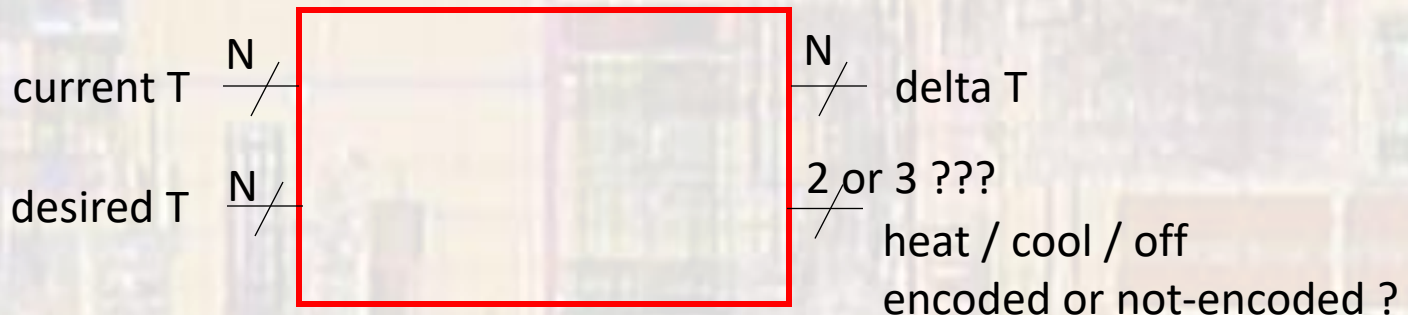
- You work for Google and are designing hardware for a new high precision factory thermostat.

Inputs: Current temperature (? bits)

Desired temperature (? bits)

Outputs: Delta temperature (? bits)

Heat / Cool / Off



VHDL Adder / Subtractor Examples

- First attempt

```
-----
-- thermostat_factory_precision_fail.vhdl
-- by: johnsontimoj
-- created: 13/25/26
-- version: 0.0
-----
-- class example of a precision thermostat block
-- inputs: temp_current, temp_desired
-- outputs: temp_delta, heat, cool, off (active high)
-----
-- does not account for wrapping of signed numbers
-----
library IEEE;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity thermostat_factory_precision_fail is
  generic(
    N: positive := 4
  );
  port(
    i_T_current: in std_logic_vector((N - 1) downto 0);
    i_T_desired: in std_logic_vector((N - 1) downto 0);
    o_T_delta: out std_logic_vector((N - 1) downto 0);
    o_heat: out std_logic;
    o_cool: out std_logic;
    o_off: out std_logic
  );
end entity;
```

```
architecture behavioral of thermostat_factory_precision_fail is
  signal t_current_sig: signed((N - 1) downto 0);
  signal t_desired_sig: signed((N - 1) downto 0);
  signal t_delta_sig: signed((N - 1) downto 0);
  signal delta_pos_neg_zero_sig: signed(1 downto 0);

begin
  -- Type conversion
  --
  t_current_sig <= signed(i_T_current);
  t_desired_sig <= signed(i_T_desired);

  -- calculate the difference (pos if desired is > current)
  --
  t_delta_sig <= t_desired_sig - t_current_sig;

  -- determine heat/cool/off state
  --
  delta_pos_neg_zero_sig <=
    "01" when (t_delta_sig > 0) else
    "10" when (t_delta_sig < 0) else
    "00" when (t_delta_sig = 0) else
    "00" ;

  -- output signals
  --
  with delta_pos_neg_zero_sig select o_heat <=
    '1' when "01",
    '0' when others;

  with delta_pos_neg_zero_sig select o_cool <=
    '1' when "10",
    '0' when others;

  with delta_pos_neg_zero_sig select o_off <=
    '1' when "00",
    '0' when others;

  o_T_delta <= std_logic_vector(t_delta_sig);

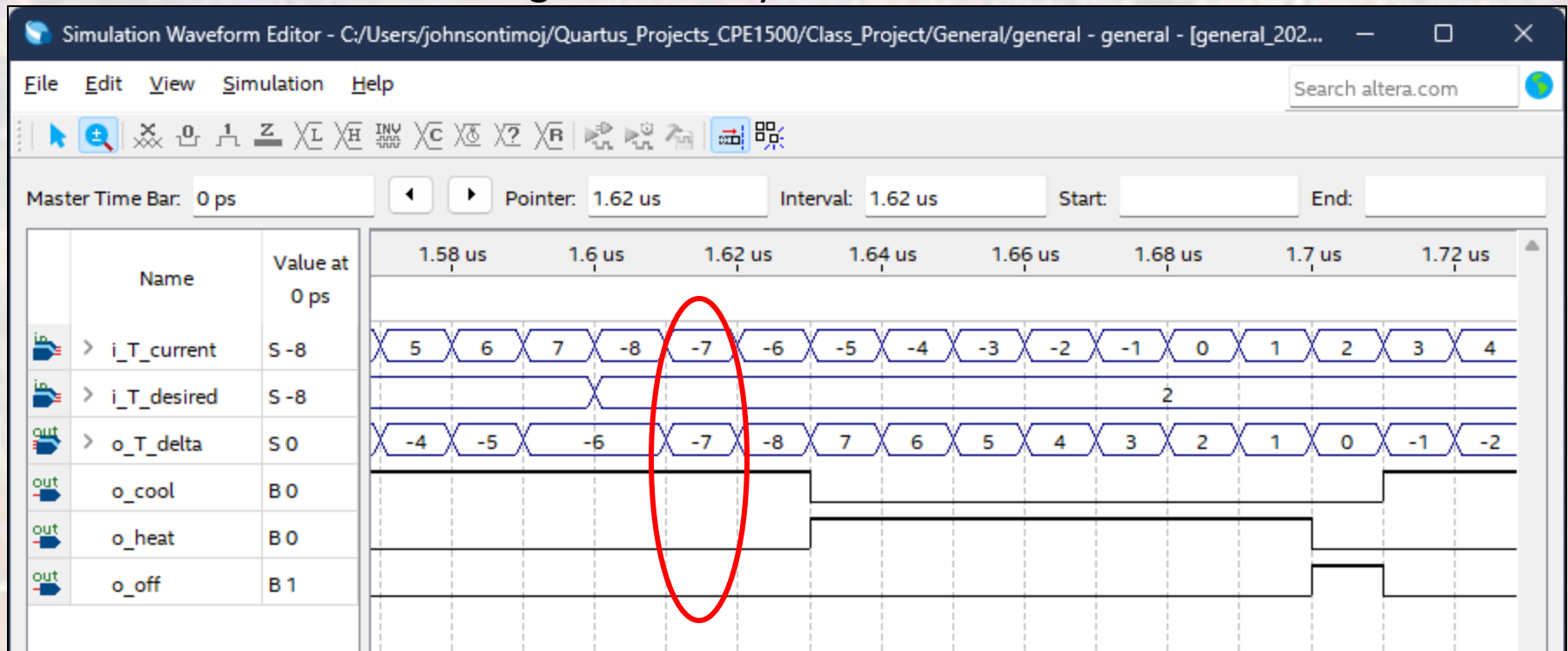
end architecture;
```

VHDL Adder / Subtractor Examples

- First attempt - **Fails**

- current = -7
- desired = 2
- $\text{delta} = 2 - (-7) = 9 \rightarrow \text{heat}$

but 9 in 4 bit signed is really -7 \rightarrow cool -- **incorrect**



VHDL Adder / Subtractor Examples

- Second attempt
 - Need to make the delta signal big enough to hold
 - $(-8) - 7$ to $7 - (-8) \rightarrow 1$ bit bigger

```
-----
-- thermostat_factory_precision.vhdl
-- by: johnsontimoj
-- created: 13/25/26
-- version: 0.0
-----
-- class example of a precision thermostat block
-- inputs: temp_current, temp_desired
-- outputs: temp_delta, heat, cool, off (active high)
-----
-- corrects for signed number wrapping by making the delta signal 1 bit larger
-- which prevents wrapping when doing the subtraction
-----
library IEEE;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity thermostat_factory_precision is
  generic(
    N: positive := 4
  );
  port(
    i_T_current: in std_logic_vector((N - 1) downto 0);
    i_T_desired: in std_logic_vector((N - 1) downto 0);
    o_T_delta: out std_logic_vector(((N - 1) + 1) downto 0);
    o_heat: out std_logic;
    o_cool: out std_logic;
    o_off: out std_logic
  );
end entity;
```

```
architecture behavioral of thermostat_factory_precision is
  signal t_current_sig: signed((N - 1) downto 0);
  signal t_desired_sig: signed((N - 1) downto 0);
  signal t_delta_sig: signed(((N - 1) + 1) downto 0); -- account for not wanting to wrap
  signal delta_pos_neg_zero_sig: signed(1 downto 0);

begin
  -- Type conversion
  --
  t_current_sig <= signed(i_T_current);
  t_desired_sig <= signed(i_T_desired);

  -- calculate the difference (pos if desired is > current)
  -- must resize before subtraction
  --
  t_delta_sig <= resize(t_desired_sig, (N + 1)) - resize(t_current_sig, (N + 1));

  -- determine heat/cool/off state
  --
  delta_pos_neg_zero_sig <=
    "01" when (t_delta_sig > 0) else
    "10" when (t_delta_sig < 0) else
    "00" when (t_delta_sig = 0) else
    "00" ;

  -- output signals
  --
  with delta_pos_neg_zero_sig select o_heat <=
    '1' when "01",
    '0' when others;

  with delta_pos_neg_zero_sig select o_cool <=
    '1' when "10",
    '0' when others;

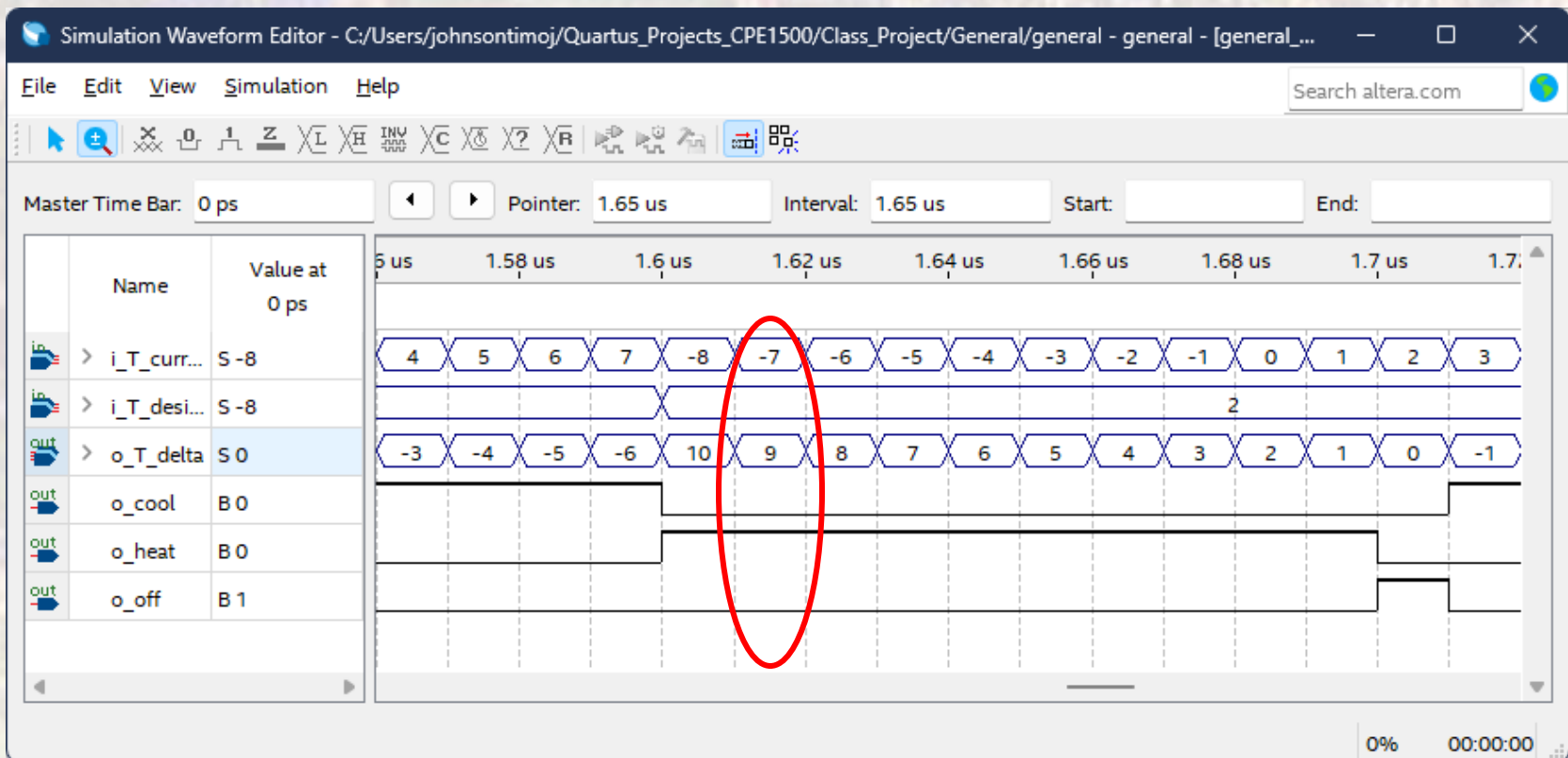
  with delta_pos_neg_zero_sig select o_off <=
    '1' when "00",
    '0' when others;

  o_T_delta <= std_logic_vector(t_delta_sig);

end architecture;
```

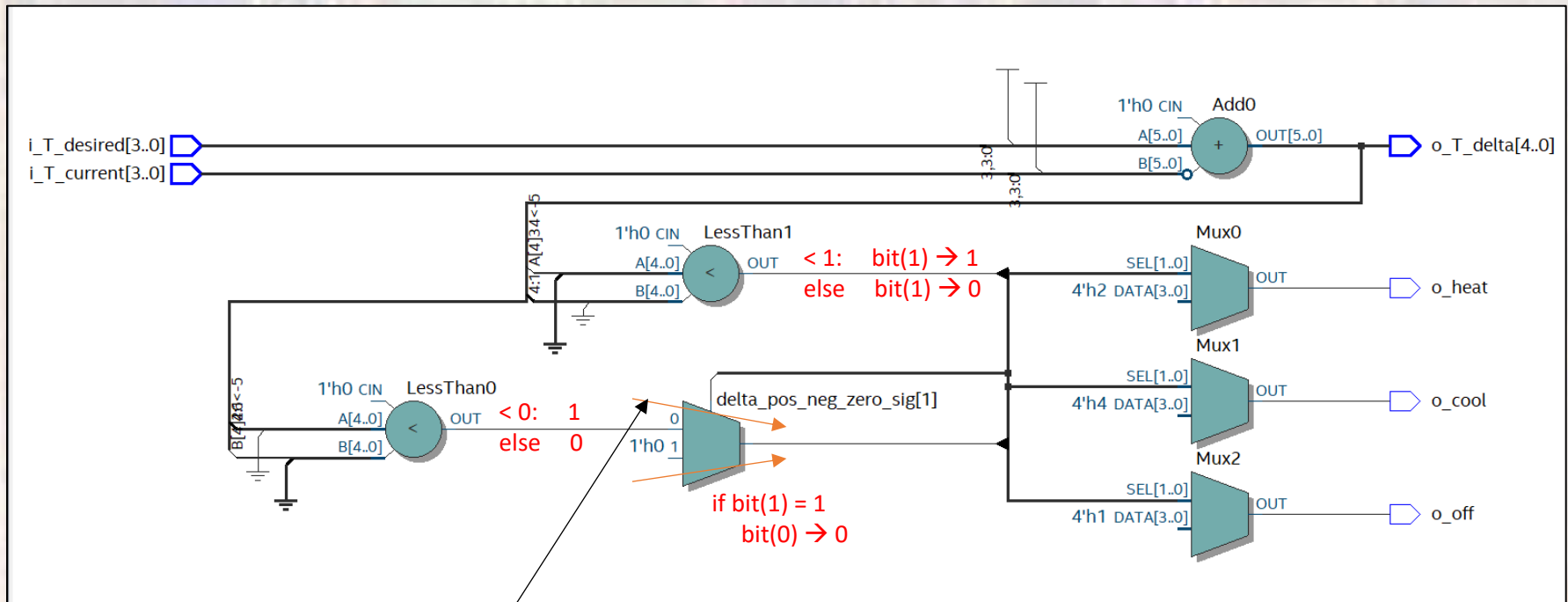
VHDL Adder / Subtractor Examples

- Second attempt
 - current = -7
 - desired = 2
 - delta = $2 - (-7) = 9 \rightarrow$ heat - correct



VHDL Adder / Subtractor Examples

- RTL (mathematical model)



if bit(1) = 0
 if <0
 bit(0) → 1 not <1 and not <0 → greater than 0
 if not <0
 bit(0) → 0 not <0 and <1 → must be 0