library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity teht2 is

Port (

start\_stop\_button : in STD\_LOGIC; -- Start/Stop-painike

resetti\_button : in STD\_LOGIC; -- Nollauspainike

clk : in STD\_LOGIC; -- Kellosignaali

segmentit\_sekunnit : out STD\_LOGIC\_VECTOR(6 downto 0); -- 7-segmenttinäyttö sekunneille

segmentit\_kymmenet : out STD\_LOGIC\_VECTOR(6 downto 0) -- 7-segmenttinäyttö kymmenille

);

end teht2;

architecture Behavioral of teht2 is

signal sekunti\_counter : natural range 0 to 9 := 0;

signal kymmenys\_counter : natural range 0 to 9 := 0;

signal clk\_counter : natural range 0 to 50000000 := 0; -- 50 MHz kello

signal running : std\_logic := '0'; -- Laskurin käynnistystila ('1' = käynnissä, '0' = pysäytetty)

signal last\_button\_state : std\_logic := '0';

signal last\_button\_state2 : std\_logic := '0';

signal resetting : std\_logic := '0';

begin

-- Prosessi, joka hallitsee start/stop-painikkeen toimintaa ja running tilan toimintaa

process(clk)

begin

if rising\_edge(clk) then

if resetting = '1' then

running <= '0';

else

-- Jos painiketta painetaan (nouseva reuna) ja se oli aiemmin vapautettu

if start\_stop\_button = '1' and last\_button\_state = '0' then

running <= not running; -- Vaihdetaan laskurin tila (käynnissä/pysäytetty)

end if;

-- Päivitetään painikkeen edellinen tila

last\_button\_state <= start\_stop\_button;

end if;

end if;

end process;

-- prosessi joka hallitsee nollauspainikkeen ja nollaustilan toimintaa

process(clk)

begin

if rising\_edge(clk) then

-- Jos painiketta painetaan (nouseva reuna) ja se oli aiemmin vapautettu

if resetti\_button = '1' and last\_button\_state2 = '0' then

resetting <= '1';

else

resetting <= '0'; -- Nollataan resetting, kun resettiä ei paineta

end if;

-- Päivitetään painikkeen edellinen tila

last\_button\_state2 <= resetti\_button;

end if;

-- lisätään tämä jotta laskuri pysähtyy nollaan kun se saavuttaa 60.

if kymmenys\_counter = 6 then

resetting <= '1';

end if;

end process;

process(clk)

begin

if rising\_edge(clk) then

if resetting = '1' then

sekunti\_counter <= 0;

kymmenys\_counter <= 0;

else

if running = '1' then -- Laskuri käynnissä vain, jos running = '1'

clk\_counter <= clk\_counter + 1;

if clk\_counter >= 50000000 then

clk\_counter <= 0;

-- Päivitetään sekuntilaskuri

if sekunti\_counter = 9 then

sekunti\_counter <= 0;

-- Päivitetään kymmenyslaskuri

if kymmenys\_counter = 6 then

kymmenys\_counter <= 0;

else

kymmenys\_counter <= kymmenys\_counter + 1;

end if;

else

sekunti\_counter <= sekunti\_counter + 1;

end if;

end if;

end if;

end if;

end if;

end process;

-- 7-segmenttinäytön päivitys sekuntilaskurin perusteella

process(clk)

begin

case sekunti\_counter is

when 0 => segmentit\_sekunnit <= "0000001"; -- 0

when 1 => segmentit\_sekunnit <= "1001111"; -- 1

when 2 => segmentit\_sekunnit <= "0010010"; -- 2

when 3 => segmentit\_sekunnit <= "0000110"; -- 3

when 4 => segmentit\_sekunnit <= "1001100"; -- 4

when 5 => segmentit\_sekunnit <= "0100100"; -- 5

when 6 => segmentit\_sekunnit <= "0100000"; -- 6

when 7 => segmentit\_sekunnit <= "0001111"; -- 7

when 8 => segmentit\_sekunnit <= "0000000"; -- 8

when 9 => segmentit\_sekunnit <= "0000100"; -- 9

when others => segmentit\_sekunnit <= "1111111"; -- Kaikki segmentit pois päältä

end case;

case kymmenys\_counter is

when 0 => segmentit\_kymmenet <= "0000001"; -- 0

when 1 => segmentit\_kymmenet <= "1001111"; -- 1

when 2 => segmentit\_kymmenet <= "0010010"; -- 2

when 3 => segmentit\_kymmenet <= "0000110"; -- 3

when 4 => segmentit\_kymmenet <= "1001100"; -- 4

when 5 => segmentit\_kymmenet <= "0100100"; -- 5

when others => segmentit\_kymmenet <= "1111111"; -- Kaikki segmentit pois päältä

end case;

end process;

end Behavioral;

Demo video:

[74812239300\_\_DD3AF43B-473A-42EB-8D8F-914535CB68F5.MOV](https://1drv.ms/v/c/2b2980717d4b73cc/EbPNwCUKmNdCi85YuppBoIwBaGvxkqeGza5PyFTTARmccA?e=yMrwQU)