smolao / Digital-electronics-1

Code Issues Pull requests Actions Projects Security Insights

main

Digital-electronics-1 / Labs / 04-segment / README4.md

Ondřej Smola splnil

History

0 contributors

Raw Blame

270 lines (202 sloc) | 7.03 KB

Lab-4 - Seven-segment display decoder (Ondřej Smola - 217628)

1st - Preparation task

Nexys A7 board

Switch je napojen na PIN

Switch	Pin
U13	AN7
K2	AN6
T14	AN5
P14	AN4
J14	AN3

Switch	Pin
T9	AN2
J18	AN1
J17	AN0
T10	CA
R10	CB
K16	CC
K13	CD
P15	CE
T11	CF
L18	CG
H15	DP

Decoder truth table

Hex	Inputs	A	B	C	D	E	F	G
0	0000	0	0	0	0	0	0	1
1	0001	1	0	0	1	1	1	1
2	0010	0	0	1	0	0	1	0
3	0011	0	0	0	0	1	1	0
4	0100	1	0	0	1	1	0	0
5	0101	0	1	0	0	1	0	0
6	0110	0	1	0	0	0	0	0
7	0111	0	0	0	1	1	1	1
8	1000	0	0	0	0	0	0	0
9	1001	0	0	0	0	1	0	0
A	1010	0	0	0	1	0	0	0

Hex	Inputs	A	B	C	D	E	F	G
b	1011	1	1	0	0	0	0	0
C	1100	0	1	1	0	0	0	1
d	1101	1	0	0	0	0	1	0
E	1110	0	1	1	0	0	0	0
F	1111	0	1	1	1	0	0	0

2nd - Seven-segment display decoder

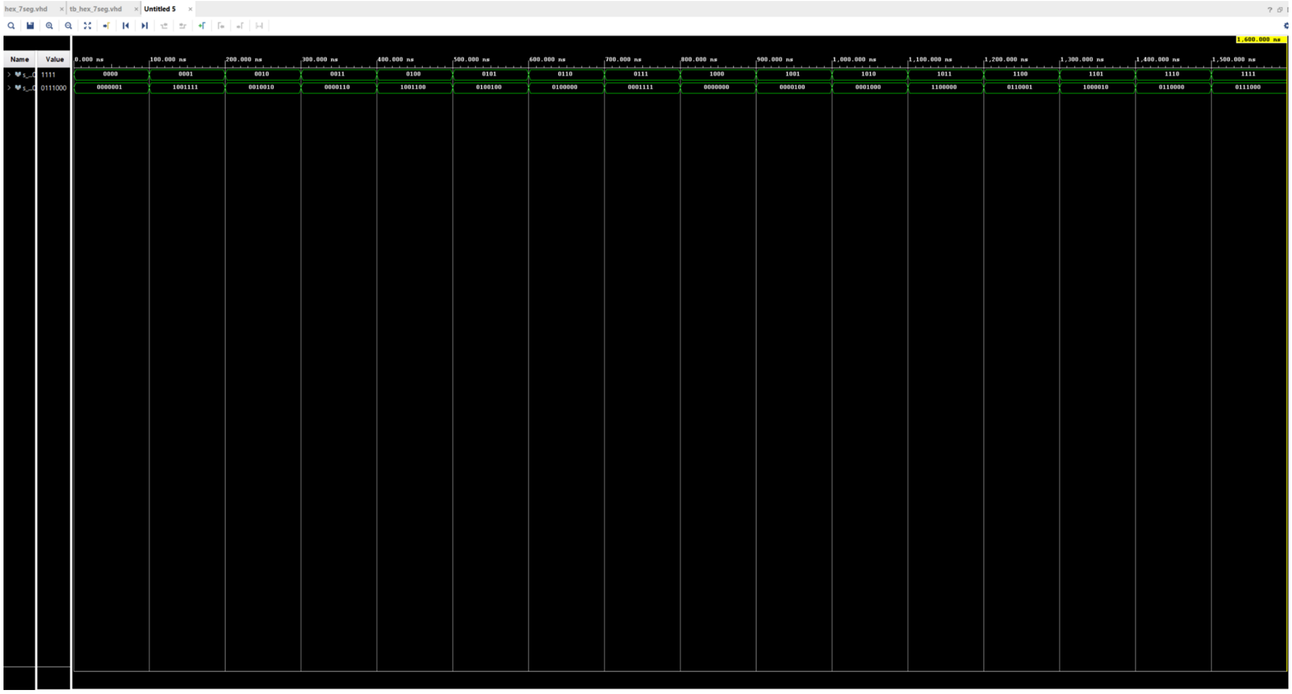
Listing of VHDL architecture (hex_7seg.vhd)

```
architecture Behavioral of hex_7seg is

begin
--------------------------------------------------------------------
    -- p_7seg_decoder:
    -- A combinational process for 7-segment display decoder.
    -- Any time "hex_i" is changed, the process is "executed".
    -- Output pin seg_o(6) corresponds to segment A, seg_o(5) to B, etc.
    --------------------------------------------------------------------
    p_7seg_decoder : process(hex_i)
    begin
        case hex_i is
            when "0000" =>
                seg_o <= "0000001";     -- 0
            when "0001" =>
                seg_o <= "1001111";     -- 1
            when "0010" =>
                seg_o <= "0010010";     -- 2
            when "0011" =>
                seg_o <= "0000110";     -- 3
            when "0100" =>
                seg_o <= "1001100";     -- 4
            when "0101" =>
                seg_o <= "0100100";     -- 5
            when "0110" =>
                seg_o <= "0100000";     -- 6
            when "0111" =>
                seg_o <= "0001111";     -- 7
            when "1000" =>
                seg_o <= "0000000";     -- 8
            when "1001" =>
                seg_o <= "0000100";     -- 9
            when "1010" =>
                seg_o <= "0001000";     -- A
            when "1011" =>
                seg_o <= "1100000";     -- b
            when "1100" =>
                seg_o <= "0110001";     -- C
            when "1101" =>
                seg_o <= "1000010";     -- d
            when "1110" =>
                seg_o <= "0110000";     -- E
            when others =>
                seg_o <= "0111000";     -- F

        end case;
    end process p_7seg_decoder;

end Behavioral;
```

Listing of VHDL stimulus process from testbench (tb_hex_7seg.vhd)

```
p_stimulus : process
    begin
        -- Report a note at the beginning of stimulus process
        report "Stimulus process started" severity note;


        -- First test values
        s_hex <= "0000"; wait for 100 ns;

        s_hex <= "0001"; wait for 100 ns;

        s_hex <= "0010"; wait for 100 ns;

        s_hex <= "0011"; wait for 100 ns;

        s_hex <= "0100"; wait for 100 ns;

        s_hex <= "0101"; wait for 100 ns;

        s_hex <= "0110"; wait for 100 ns;

        s_hex <= "0111"; wait for 100 ns;

        s_hex <= "1000"; wait for 100 ns;

        s_hex <= "1001"; wait for 100 ns;

        s_hex <= "1010"; wait for 100 ns;
        s_hex <= "1011"; wait for 100 ns;

        s_hex <= "1100"; wait for 100 ns;

        s_hex <= "1101"; wait for 100 ns;

        s_hex <= "1110"; wait for 100 ns;

        s_hex <= "1111"; wait for 100 ns;




        -- Report a note at the end of stimulus process
        report "Stimulus process finished" severity note;
        wait;
end process p_stimulus;
```

Screenshot with simulated waweforms



Listing VHDL code from source file top.vhd

```
entity top is
    Port
    (
    SW  : in STD_LOGIC_VECTOR (4 - 1 downto 0);   -- Input binary data
    CA  : out STD_LOGIC;                          -- Cathods
    CB  : out STD_LOGIC;
    CC  : out STD_LOGIC;
    CD  : out STD_LOGIC;
    CE  : out STD_LOGIC;
    CF  : out STD_LOGIC;
    CG  : out STD_LOGIC;

    LED : out STD_LOGIC_VECTOR (8 - 1 downto 0);   -- LED indicators
    AN  : out STD_LOGIC_VECTOR (8 - 1 downto 0)    -- Common anode signals to individ

    );
end top;

architecture Behavioral of top is

begin

    -- Instance (copy) of hex_7seg entity
    hex2seg : entity work.hex_7seg
        port map(
            hex_i     => SW,
            seg_o(6)  => CA,
            seg_o(5)  => CB,
            seg_o(4)  => CC,
            seg_o(3)  => CD,
            seg_o(2)  => CE,
            seg_o(1)  => CF,
            seg_o(0)  => CG
        );
```

3rd - LED(7:4) indicators

Truth table and listing of VHDL code for LEDs(7:4)

Truth table

Hex	Inputs	LED4	LED5	LED6	LED7
0	0000	1	0	0	0
1	0001	0	0	1	1
2	0010	0	0	0	1
3	0011	0	0	1	0
4	0100	0	0	0	1

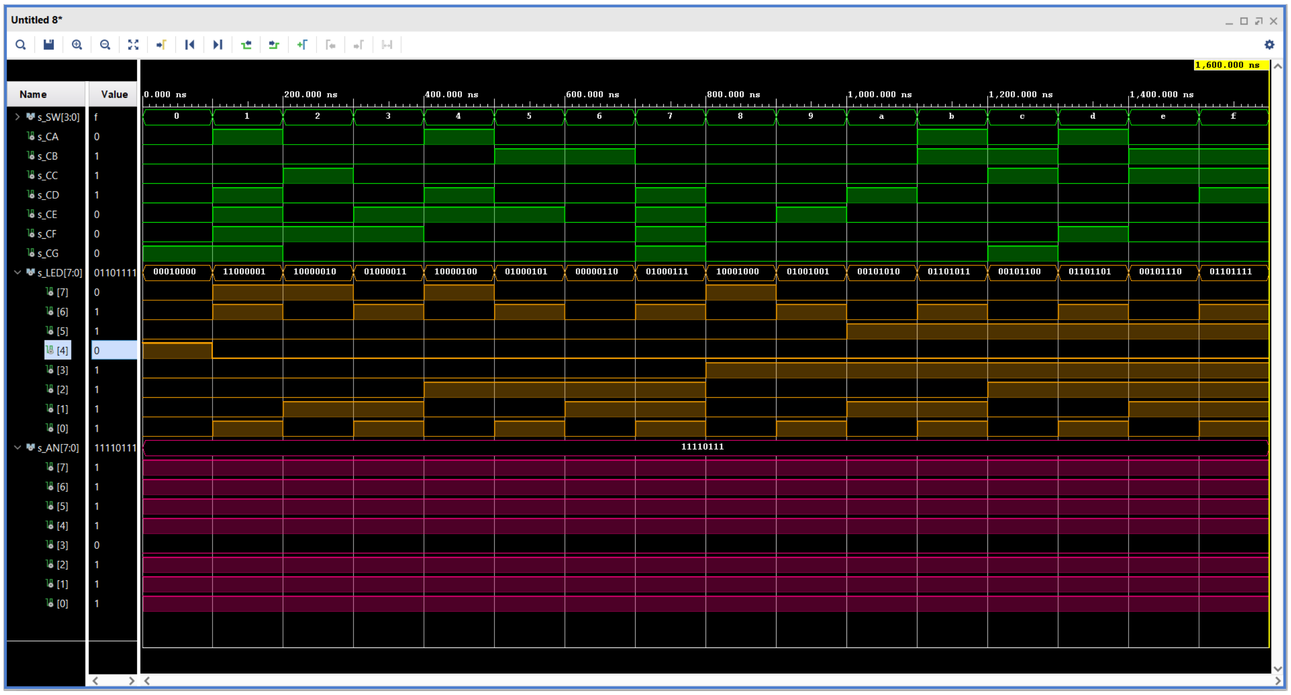
Hex	Inputs	LED4	LED5	LED6	LED7
5	0101	0	0	1	0
6	0110	0	0	0	0
7	0111	0	0	1	0
8	1000	0	0	0	1
9	1001	0	0	1	0
A	1010	0	1	0	0
b	1011	0	1	1	0
C	1100	0	1	0	0
d	1101	0	1	1	0
E	1110	0	1	0	0
F	1111	0	1	1	0

VHDL code for LEDS (7:4)

```
-- Connect one common anode to 3.3V
    AN <= b"1111_0111";

    -- Display input value on LEDs
    LED(4 - 1 downto 0) <= SW;

    -- LED(7:4) indicators
    -- Turn LED(4) on if input value is equal to 0, ie "0000"
    LED(4) <= '1' when (SW = "0000") else '0';

    -- Turn LED(5) on if input value is greater than "1001", ie 9
    LED(5) <= '1' when (SW = "1001") else '0';

    -- Turn LED(6) on if input value is odd, ie 1, 3, 5, ...
    LED(6) <= '1' when (SW = "0001") else
              '1' when (SW = "0011") else
              '1' when (SW = "0101") else
              '1' when (SW = "0111") else
              '1' when (SW = "1001") else
              '1' when (SW = "1011") else
              '1' when (SW = "1101") else
              '1' when (SW = "1111") else '0';

    -- Turn LED(7) on if input value is a power of two, ie 1, 2, 4, or 8
```

```
        LED(7) <= '1' when (SW = "0001") else
                  '1' when (SW = "0010") else
                  '1' when (SW = "0100") else
                  '1' when (SW = "1000") else '0';
```

Screenshot of simulated time waweforms


Untitled 8*
Name
Value
0.000 ns
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1,000.000 ns
1,200.000 ns
1,400.000 ns
1,600.000 ns
s_SW[3:0] f
0 1 2 3 4 5 6 7 8 9 a b c d e f
s_CA 0
s_CB 1
s_CC 1
s_CD 1
s_CE 0
s_CF 0
s_CG 0
s_LED[7:0] 01101111
00010000 11000001 10000010 01000011 10000100 01000101 00000110 01000111 10001000 01001001 00101010 01101011 00101100 01101101 00101110 01101111
[7] 0
[6] 1
[5] 1
[4] 0
[3] 1
[2] 1
[1] 1
[0] 1
s_AN[7:0] 11110111
11110111
[7] 1
[6] 1
[5] 1
[4] 1
[3] 0
[2] 1
[1] 1
[0] 1