Fabulec / Digital-electronics-1

MIT License

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security

main Digital-electronics-1 / 04-segment /

Fabulec De1 ... 13 minutes ago History

..

Image	13 minutes ago
README.md	13 minutes ago

README.md

Digital-electronics-1

https://github.com/Fabulec/Digital-electronics-1

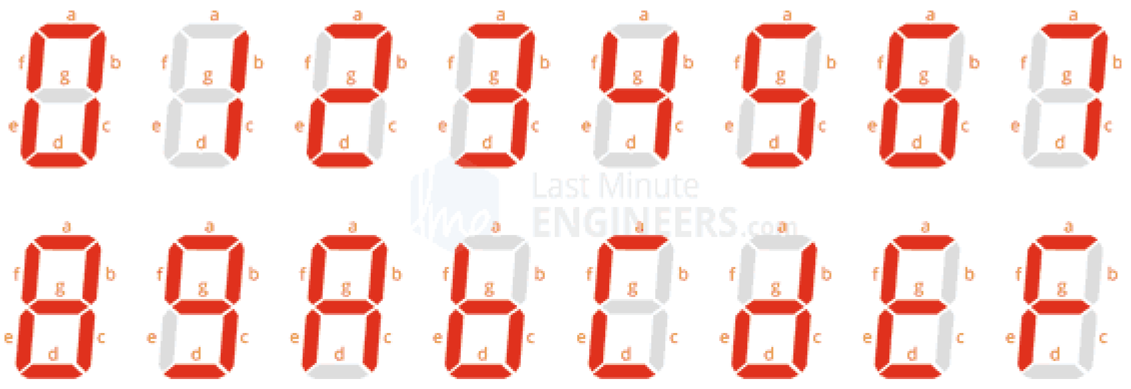
Lab 4: Seven-segment display decoder:

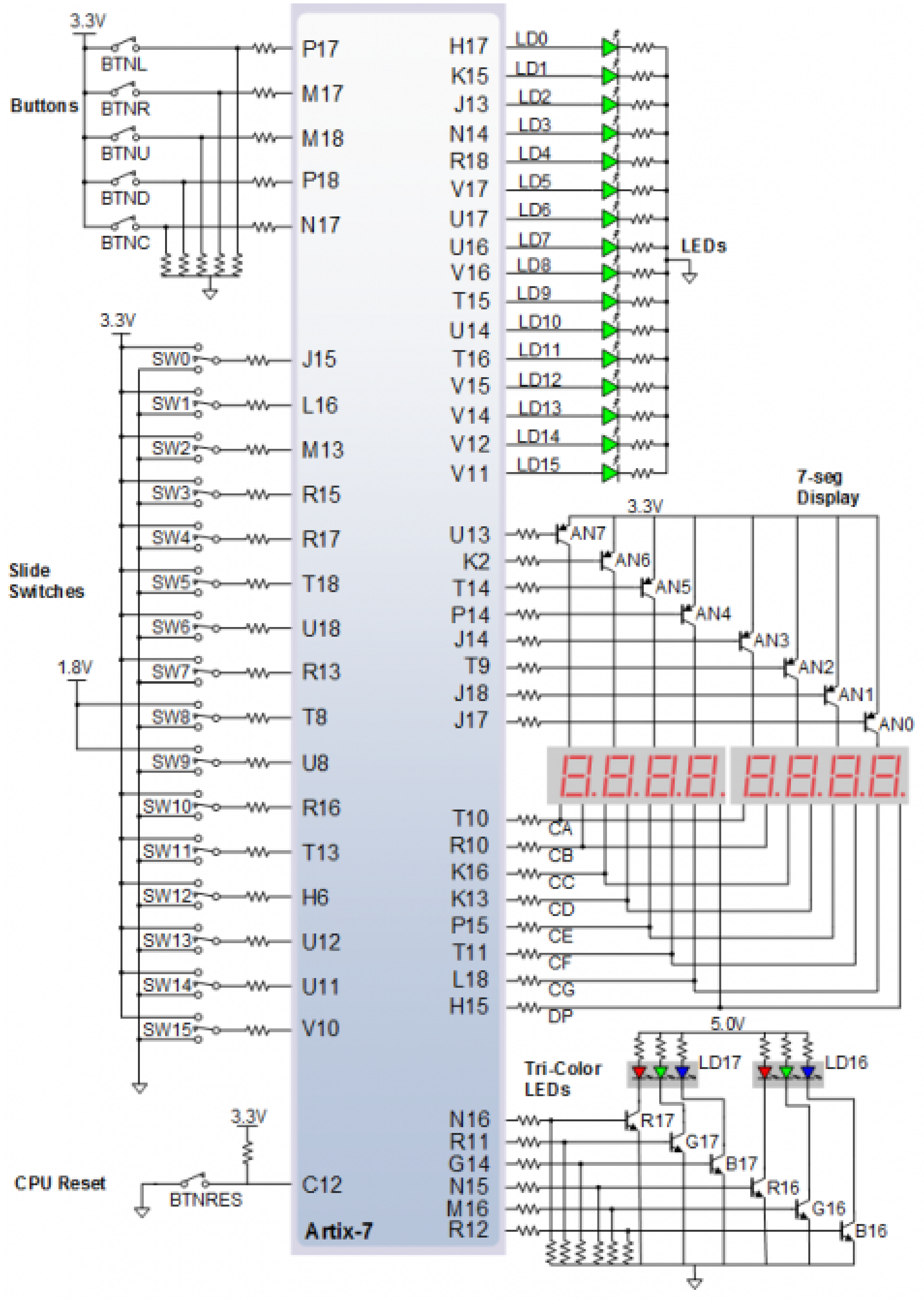
1. Preparation tasks:

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

Hex	Inputs	A	B	C	D	E	F	G
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	1	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
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





2. Seven-segment display decode:

Listing of VHDL architecture from source file:

```vhdl
architecture Behavioral of hex_7seg is

begin
```

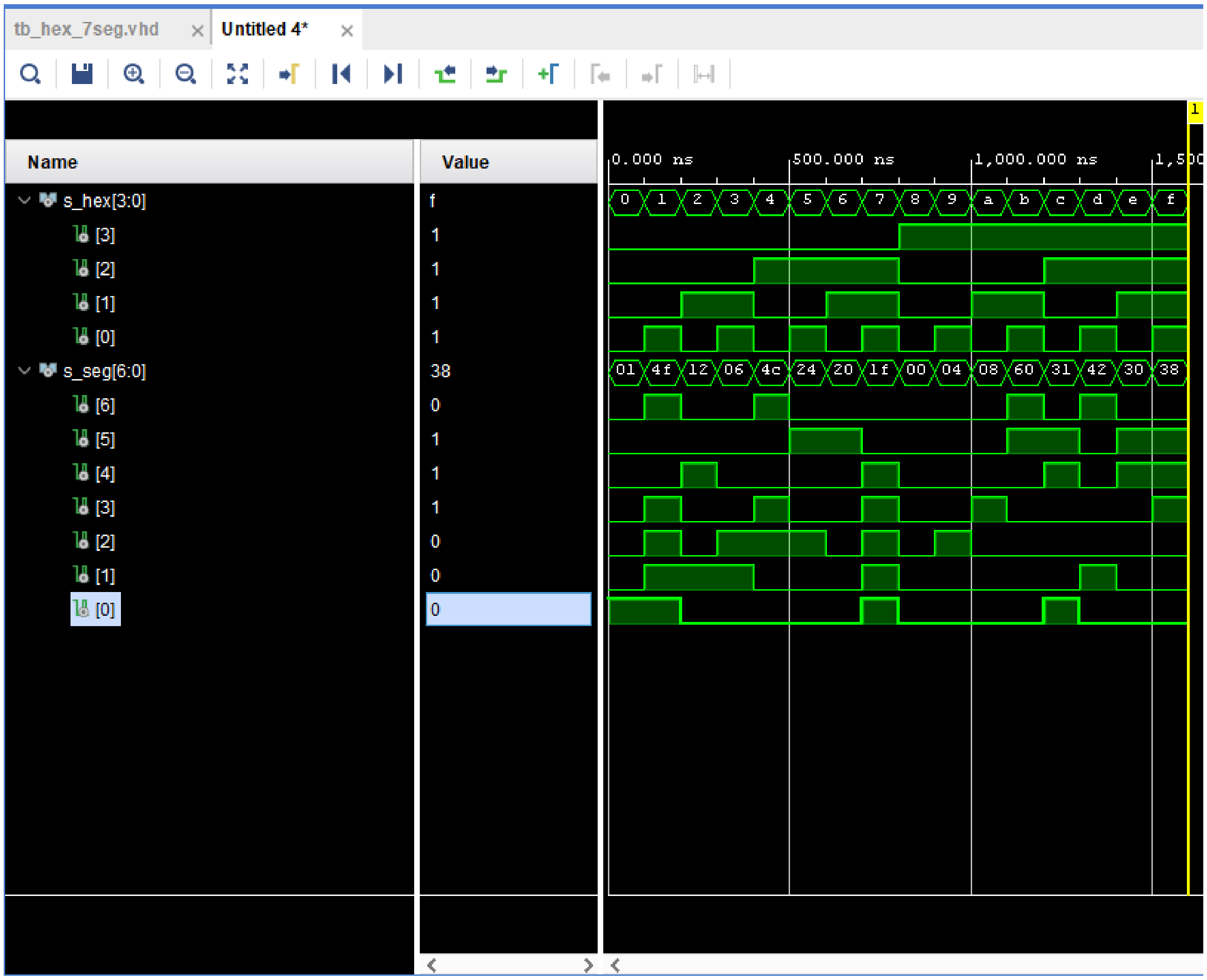
```
    p_7seg_decoder : process(hex_i)
    begin

        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 <= "0011111";     -- 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;
```

Listing of VHDL stimulus process from testbench file:

```
architecture Behavioral of tb_hex_7seg is

    signal s_hex        : std_logic_vector(4 - 1 downto 0);
    signal s_seg        : std_logic_vector(7 - 1 downto 0);

begin

    uut_hex_7seg : entity work.hex_7seg
```

```
            port map(
                hex_i    => s_hex,
                seg_o    => s_seg
                );

                p_stimulus : process
                begin

                    report "Stimulus process started" severity note;


                    s_hex <= "0000"; wait for 100ns;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                    report "Stimulus process finished" severity note;
                    wait;

        end process p_stimulus;
    end Behavioral;
```

Screenshot with simulated time waveforms:



Listing of VHDL code from source file:

```
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
        );
```

3. LED indicators:

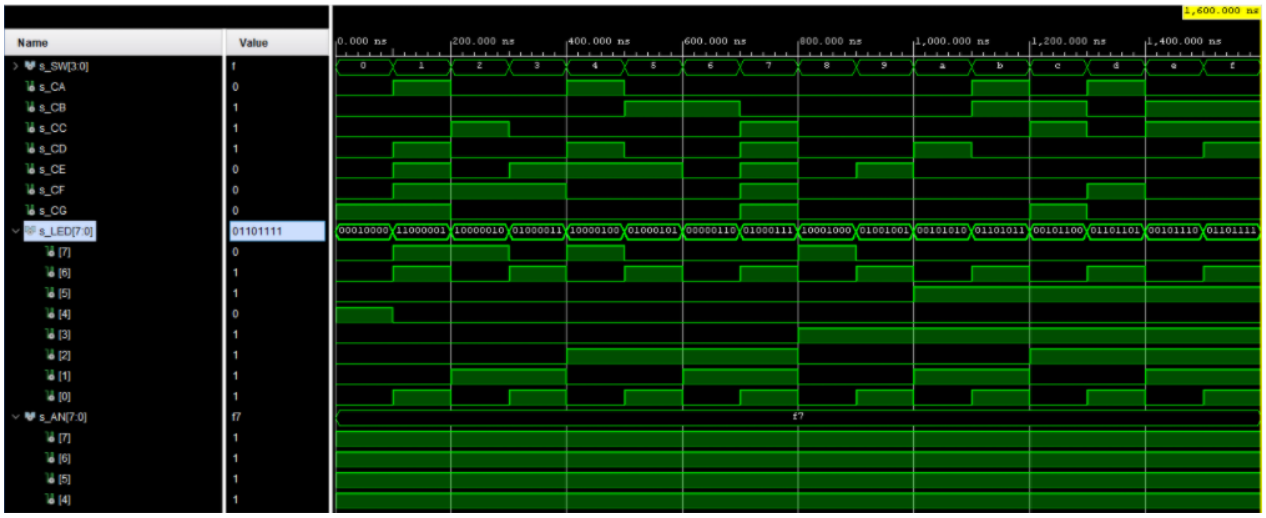
Truth Table:

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
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

Listing of VHDL code for LEDs:

```
-- Display input value
    LED(3 downto 0) <= [illegible]

    LED(4)  <= '1' when (SW = "0000") else '0';

    LED(5)  <= '1' when (SW > "1001") else '0';
    LED(6)  <= SW(0);

    LED(7)  <= '1' when (SW = "0001" or SW = "0010" or SW = "0100" or SW = "1000") e
```

Screenshot with simulated time waveforms:



Name
Value
s_SW[3:0] 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
[7] 0
[6] 1
[5] 1
[4] 0
[3] 1
[2] 1
[1] 1
[0] 1
s_AN[7:0] f7
[7] 1
[6] 1
[5] 1
[4] 1
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
0 1 2 3 4 5 6 7 8 9 a b c d e f
00010000 11000001 10000010 01000011 10000100 01000101 00000110 01000111 10001000 01001001 00101010 01101011 00101100 01101101 00101110 01101111
f7