



Program:

Course Number	2005
Course Title	EMBEDDED SYSTEMS ARCHITECTURE 1
Semester/Year	SUMMER/2019

Instructor	<b>Mohsen Salahi</b>
------------	----------------------

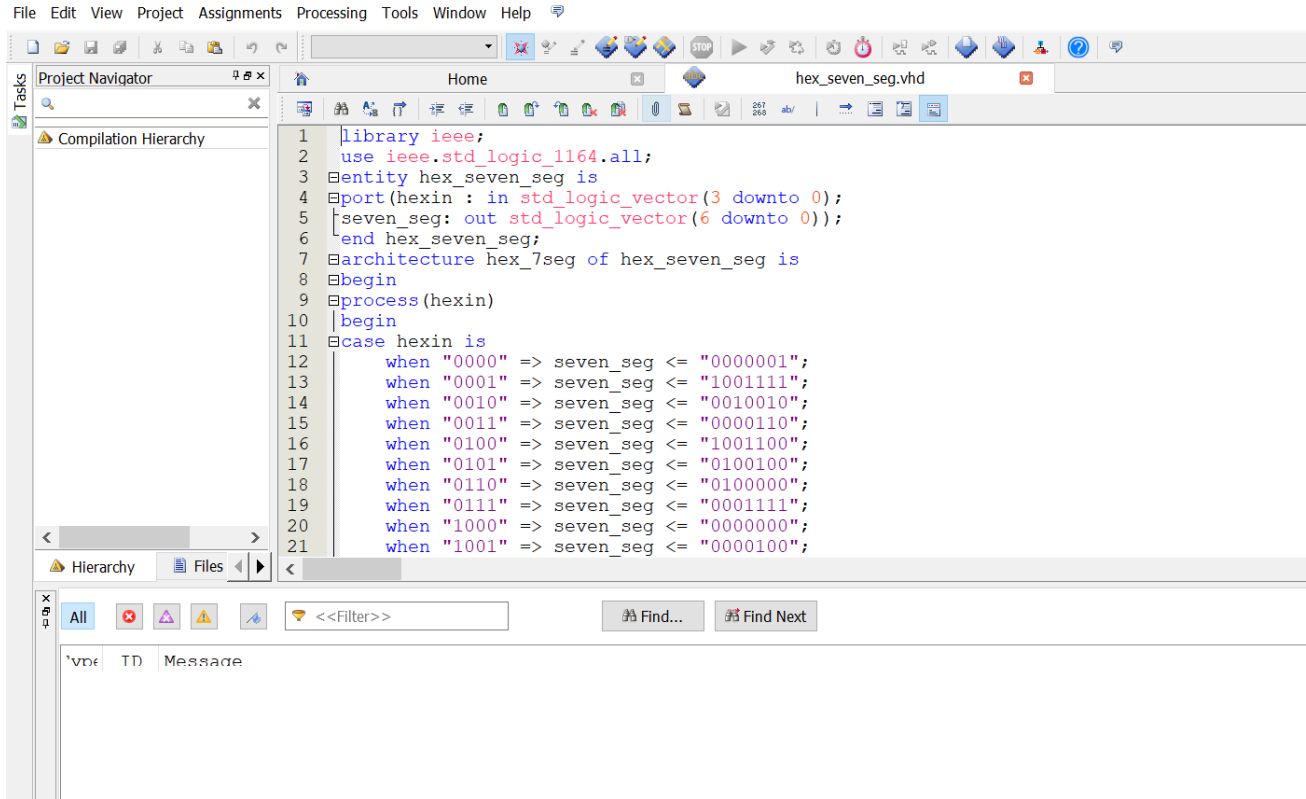
<b>Lab No.</b>	<b>2</b>
----------------	----------

Student Name	<b>CHANDRA SHEKARA REDDY SAGA</b>	<b>SRUTHY KRISHNAN</b>	<b>STEPHY BABY</b>
Student ID	C0747253	C0749122	C0753812
Signature*	<b>CSRS</b>	<b>SK</b>	<b>SB</b>

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties.*

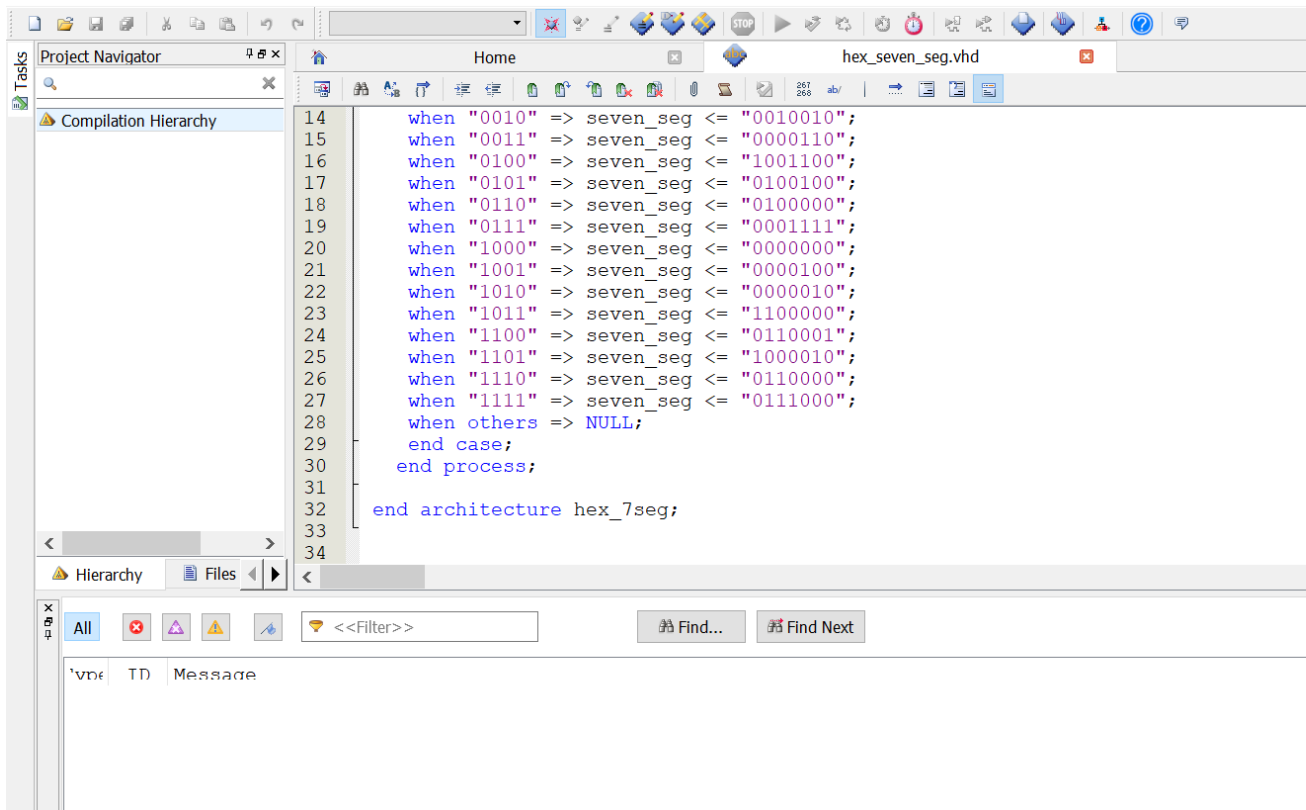
# SEVEN SEGMENT DISPLAY

## 1. PROGRAMMING VHDL



This screenshot shows the first 21 lines of the VHDL code in the 'hex\_seven\_seg.vhd' file. The code includes the IEEE standard logic library, defines the entity 'hex\_seven\_seg' with a 4-bit input 'hexin' and a 7-bit output 'seven\_seg', and begins the architecture 'hex\_7seg'. A process is defined for 'hexin' with a 'when' statement mapping 16 possible 4-bit hex values to their corresponding 7-bit segment outputs. The first four lines of the 'when' statement are visible in this screenshot.

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 entity hex_seven_seg is
4 port(hexin : in std_logic_vector(3 downto 0);
5      seven_seg: out std_logic_vector(6 downto 0));
6 end hex_seven_seg;
7 architecture hex_7seg of hex_seven_seg is
8 begin
9 process(hexin)
10 begin
11 case hexin is
12 when "0000" => seven_seg <= "0000001";
13 when "0001" => seven_seg <= "1001111";
14 when "0010" => seven_seg <= "0010010";
15 when "0011" => seven_seg <= "0000110";
16 when "0100" => seven_seg <= "1001100";
17 when "0101" => seven_seg <= "0100100";
18 when "0110" => seven_seg <= "0100000";
19 when "0111" => seven_seg <= "0001111";
20 when "1000" => seven_seg <= "0000000";
21 when "1001" => seven_seg <= "0000100";
```



This screenshot shows the continuation of the VHDL code from line 14 to line 34. It completes the 'when' statement with the remaining 12 hex values, adds an 'others' case for NULL, and ends the process and architecture. The code is as follows:

```
14 when "0010" => seven_seg <= "0010010";
15 when "0011" => seven_seg <= "0000110";
16 when "0100" => seven_seg <= "1001100";
17 when "0101" => seven_seg <= "0100100";
18 when "0110" => seven_seg <= "0100000";
19 when "0111" => seven_seg <= "0001111";
20 when "1000" => seven_seg <= "0000000";
21 when "1001" => seven_seg <= "0000100";
22 when "1010" => seven_seg <= "0000010";
23 when "1011" => seven_seg <= "1100000";
24 when "1100" => seven_seg <= "0110001";
25 when "1101" => seven_seg <= "1000010";
26 when "1110" => seven_seg <= "0110000";
27 when "1111" => seven_seg <= "0111000";
28 when others => NULL;
29 end case;
30 end process;
31
32 end architecture hex_7seg;
```

## 2. TEST BENCH

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY hex_seven_seg_vhd_tst IS
END hex_seven_seg_vhd_tst;
ARCHITECTURE hex_seven_seg_arch OF hex_seven_seg_vhd_tst IS
-- constants
-- signals
SIGNAL hexin : STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL seven_seg : STD_LOGIC_VECTOR(6 DOWNTO 0);
COMPONENT hex_seven_seg
    PORT (
        hexin : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
        seven_seg : BUFFER STD_LOGIC_VECTOR(6 DOWNTO 0)
    );
END COMPONENT;
BEGIN
    i1 : hex_seven_seg
    PORT MAP (
-- list connections between master ports and signals
        hexin => hexin,
        seven_seg => seven_seg
    );
    init : PROCESS
-- variable declarations
    BEGIN
        -- code that executes only once

    WAIT;
    END PROCESS init;
    always : PROCESS
-- optional sensitivity list
-- (    )
-- variable declarations
    BEGIN
        -- code executes for every event on sensitivity list
        hexin <="0000";
        wait for 100 ns;
        hexin <="0001";
        wait for 100 ns;
        hexin <="0010";
        wait for 100 ns;
        hexin <="0011";
        wait for 100 ns;
        hexin <="0100";
        wait for 100 ns;
        hexin <="0101";
        wait for 100 ns;
        hexin <="0110";
```

```

wait for 100 ns;
hexin <="0111";
wait for 100 ns;
hexin <="1000";
wait for 100 ns;
hexin <="1001";
wait for 100 ns;
hexin <="1010";
wait for 100 ns;
hexin <="1011";
wait for 100 ns;
hexin <="1100";
wait for 100 ns;
hexin <="1101";
wait for 100 ns;
hexin <="1110";
wait for 100 ns;
hexin <="1111";
wait for 100 ns;

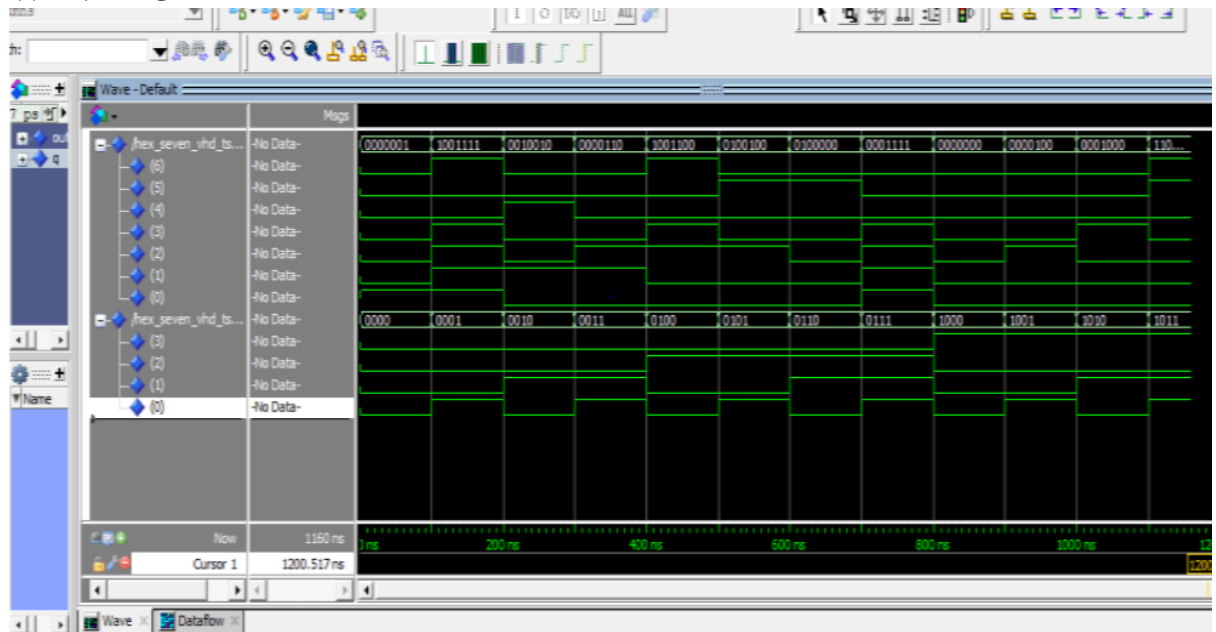
```

```

WAIT;
END PROCESS always;
END hex_seven_seg_arch;

```

### 3. WAVEFORM



#### 4. TRUTH TABLE

HEXADECIMAL	BINARY	SEVEN SEGMENT
0	0000	0000001
1	0001	1001111
2	0010	0010010
3	0011	0000110
4	0100	1001100
5	0101	0100100
6	0110	0100000
7	0111	0001111
8	1000	0000000
9	1001	0000100
a	1010	0000010
b	1011	1100000
c	1100	0110001
d	1101	1000010
e	1110	0110000
f	1111	0111000