

An Introduction to the Synopsys Design Compiler

Prepared by Li Li for ECE 368 (Instructor: Prof. Shantanu Dutt)

1. Besides all the source files, we need to write one more file named as “run” (you can name it as you like). In this file, these commands are needed:

```
target_library = lsi_10.db
```

```
symbol_library = lsi_10.sdb
```

```
link_library = "*" + target_library
```

```
synthetic_library = " "
```

```
read -format vhdl *.vhdl // * is the source file's name
```

```
read -format vhdl *.vhdl
```

```
//include all the necessary files
```

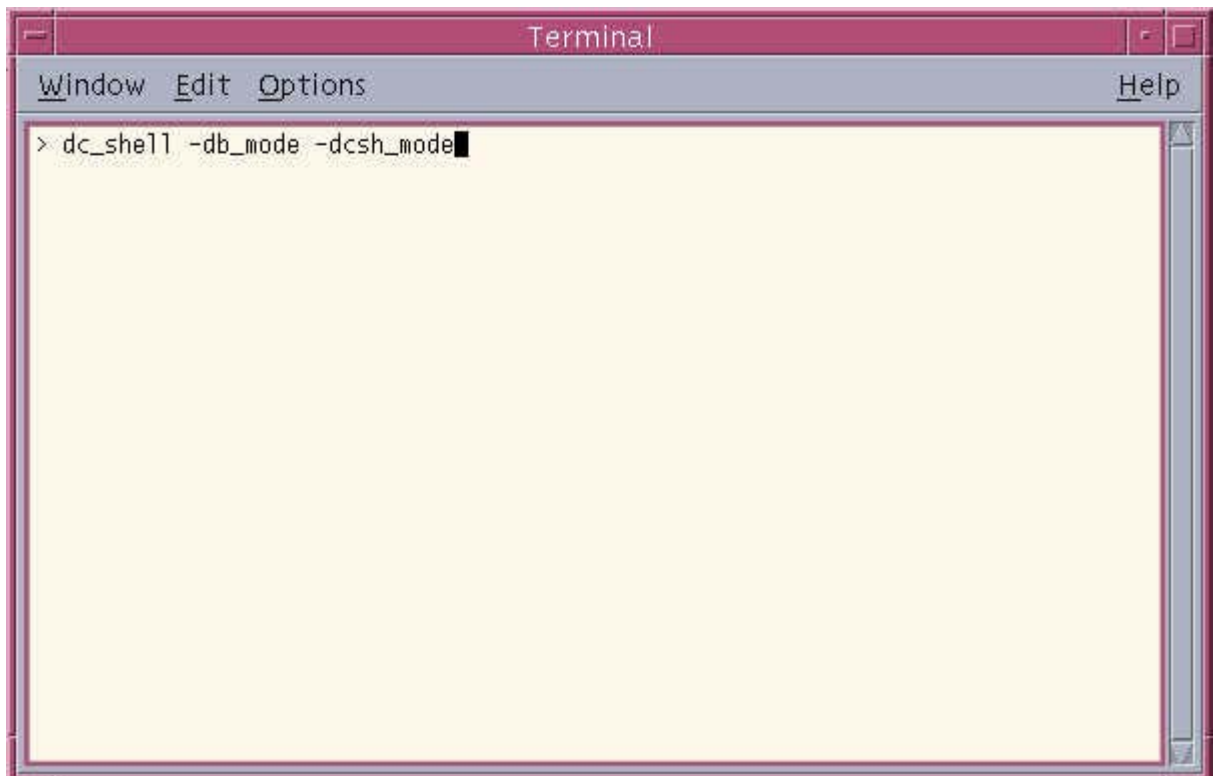
```
compile
```

```
write -format db -hierarchy -output **.db // the ** stands for the output file's name, you can name it as you like, .db is its file type
```

In this manual, we take bcd4to7seg project for example. The project should contain these files: and2.vhdl, and3.vhdl, and4.vhdl, or2.vhdl, or3.vhdl, or4.vhdl, my_not.vhdl, BCDto7seg.vhdl, run. Refer all these code at the end of this manual.

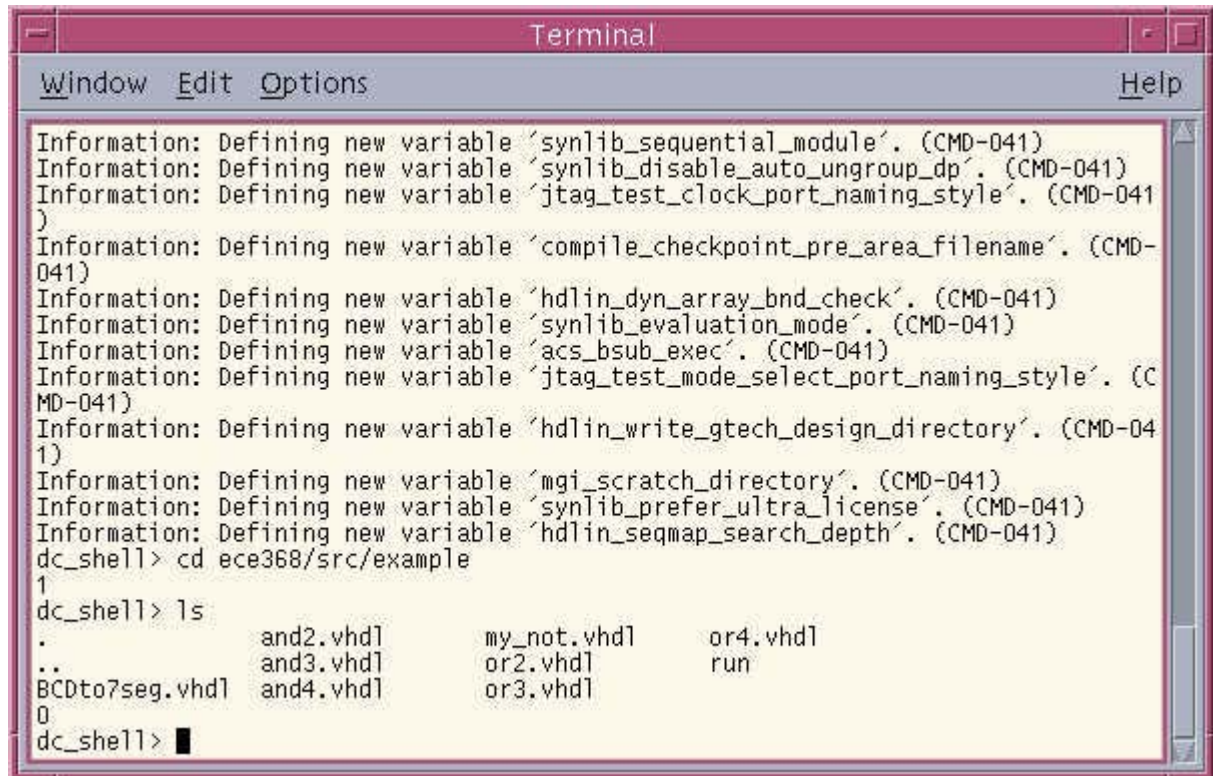
2. Compile the files in the dc_shell enviroment, just follow these steps:

a. In the terminal window, just type `dc_shell -db_mode -dcsh_mode`



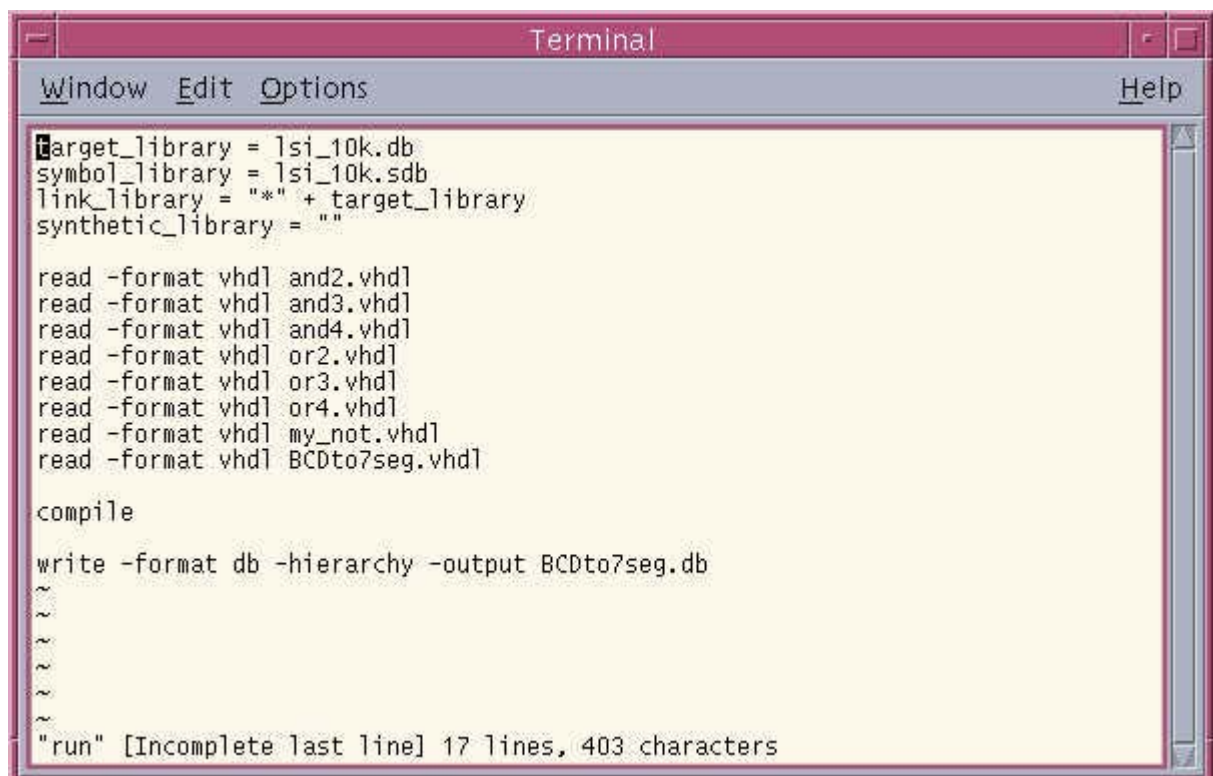
b. Now you are supposed in the dc_shell enviroment, first of all, ensure you are in the right directory which include all the files, in this example, all the files are in

`ece368/src/example/`, so we type the following command to go to the directory:
`cd ece368/src/example.`



```
Terminal
Window Edit Options Help
Information: Defining new variable 'synlib_sequential_module'. (CMD-041)
Information: Defining new variable 'synlib_disable_auto_ungroup_dp'. (CMD-041)
Information: Defining new variable 'jtag_test_clock_port_naming_style'. (CMD-041)
)
Information: Defining new variable 'compile_checkpoint_pre_area_filename'. (CMD-041)
Information: Defining new variable 'hdlin_dyn_array_bnd_check'. (CMD-041)
Information: Defining new variable 'synlib_evaluation_mode'. (CMD-041)
Information: Defining new variable 'acs_bsub_exec'. (CMD-041)
Information: Defining new variable 'jtag_test_mode_select_port_naming_style'. (C
MD-041)
Information: Defining new variable 'hdlin_write_gtech_design_directory'. (CMD-04
1)
Information: Defining new variable 'mgi_scratch_directory'. (CMD-041)
Information: Defining new variable 'synlib_prefer_ultra_license'. (CMD-041)
Information: Defining new variable 'hdlin_seqmap_search_depth'. (CMD-041)
dc_shell> cd ece368/src/example
1
dc_shell> ls
.                and2.vhdl        my_not.vhdl      or4.vhdl
..               and3.vhdl        or2.vhdl         run
BCDto7seg.vhdl  and4.vhdl        or3.vhdl
0
dc_shell> █
```

The content of “run” file is like this:



```
Terminal
Window Edit Options Help
target_library = lsi_10k.db
symbol_library = lsi_10k.sdb
link_library = "*" + target_library
synthetic_library = ""

read -format vhd1 and2.vhdl
read -format vhd1 and3.vhdl
read -format vhd1 and4.vhdl
read -format vhd1 or2.vhdl
read -format vhd1 or3.vhdl
read -format vhd1 or4.vhdl
read -format vhd1 my_not.vhdl
read -format vhd1 BCDto7seg.vhdl

compile

write -format db -hierarchy -output BCDto7seg.db
~
~
~
~
~
"run" [Incomplete last line] 17 lines, 403 characters
```

When you are in the correct directory, type the following command to compile all the vhd1 files in design compiler: *include run*

```

Terminal
Window Edit Options Help
Information: Defining new variable 'synlib_sequential_module'. (CMD-041)
Information: Defining new variable 'synlib_disable_auto_ungroup_dp'. (CMD-041)
Information: Defining new variable 'jtag_test_clock_port_naming_style'. (CMD-041)
)
Information: Defining new variable 'compile_checkpoint_pre_area_filename'. (CMD-041)
Information: Defining new variable 'hdlin_dyn_array_bnd_check'. (CMD-041)
Information: Defining new variable 'synlib_evaluation_mode'. (CMD-041)
Information: Defining new variable 'acs_bsub_exec'. (CMD-041)
Information: Defining new variable 'jtag_test_mode_select_port_naming_style'. (CMD-041)
Information: Defining new variable 'hdlin_write_gtech_design_directory'. (CMD-041)
Information: Defining new variable 'mgi_scratch_directory'. (CMD-041)
Information: Defining new variable 'synlib_prefer_ultra_license'. (CMD-041)
Information: Defining new variable 'hdlin_seqmap_search_depth'. (CMD-041)
dc_shell> cd ece368/src/example
1
dc_shell> ls
.          and2.vhdl      my_not.vhdl   or4.vhdl
..         and3.vhdl      or2.vhdl      run
BCDto7seg.vhdl and4.vhdl    or3.vhdl
0
dc_shell> include run

```

After successfully compiled, you will get the output file `**db` (BCDto7seg.db in this example) in your current directory.

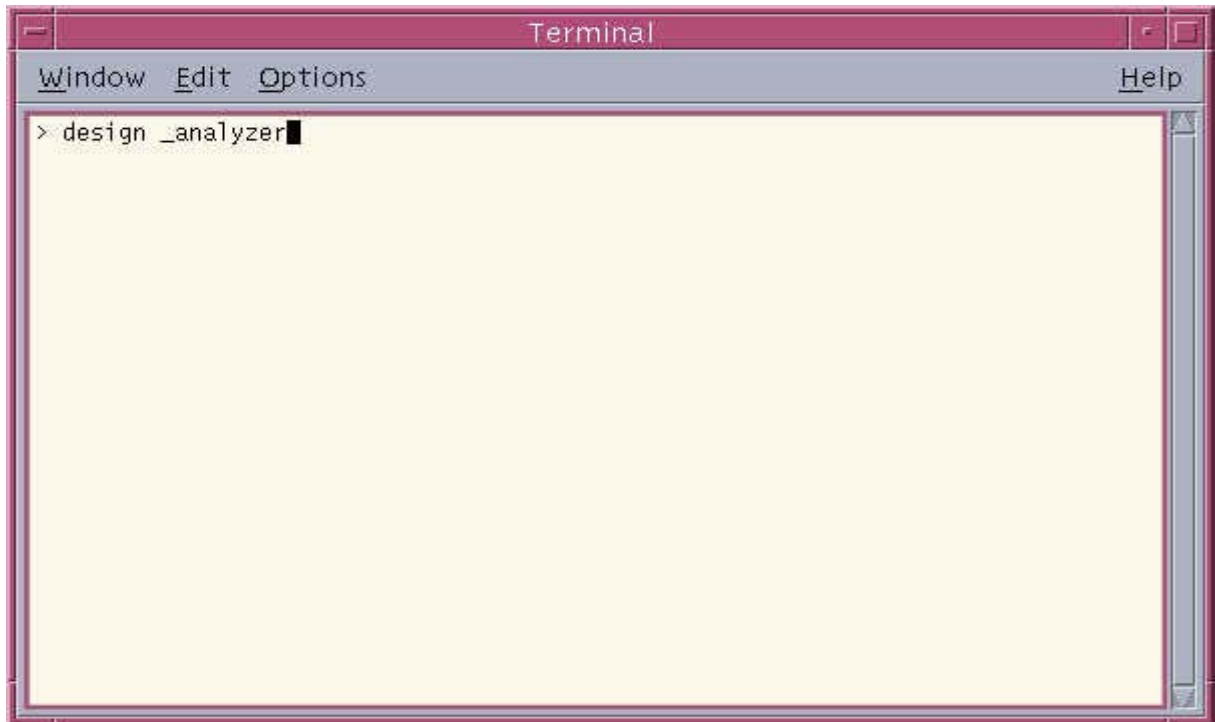
```

Terminal
Window Edit Options Help
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
0:00:10    58.0    0.00    0.0    0.0
-----
Optimization Complete
-----
Transferring design 'BCDto7seg' to database 'BCDto7seg.db'
Current design is 'BCDto7seg'.
1
write -format db -hierarchy -output BCDto7seg.dbWriting to file /home3/grad3/111
2/ece368/src/example/BCDto7seg.db
1
1
dc_shell>

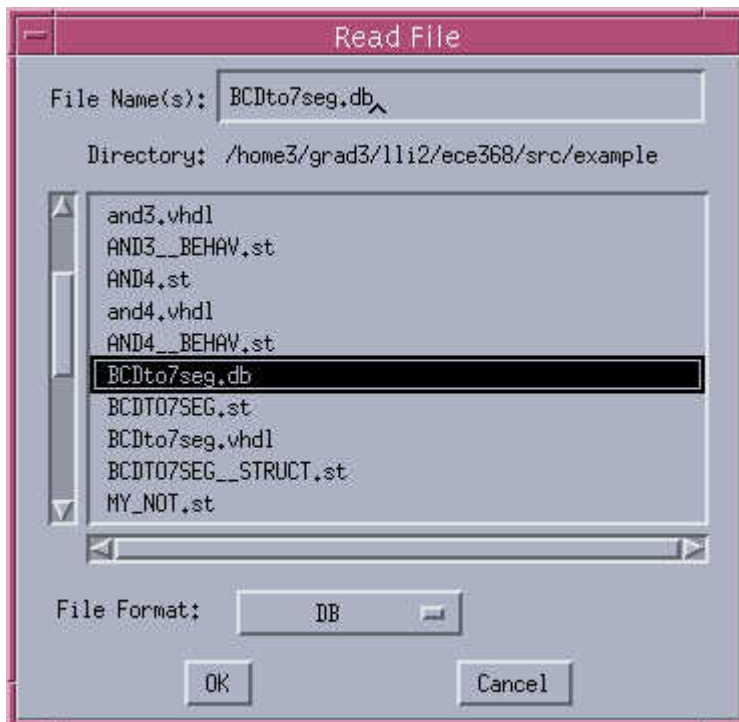
```

The following is showing how to view and print the hardware synthesis result:

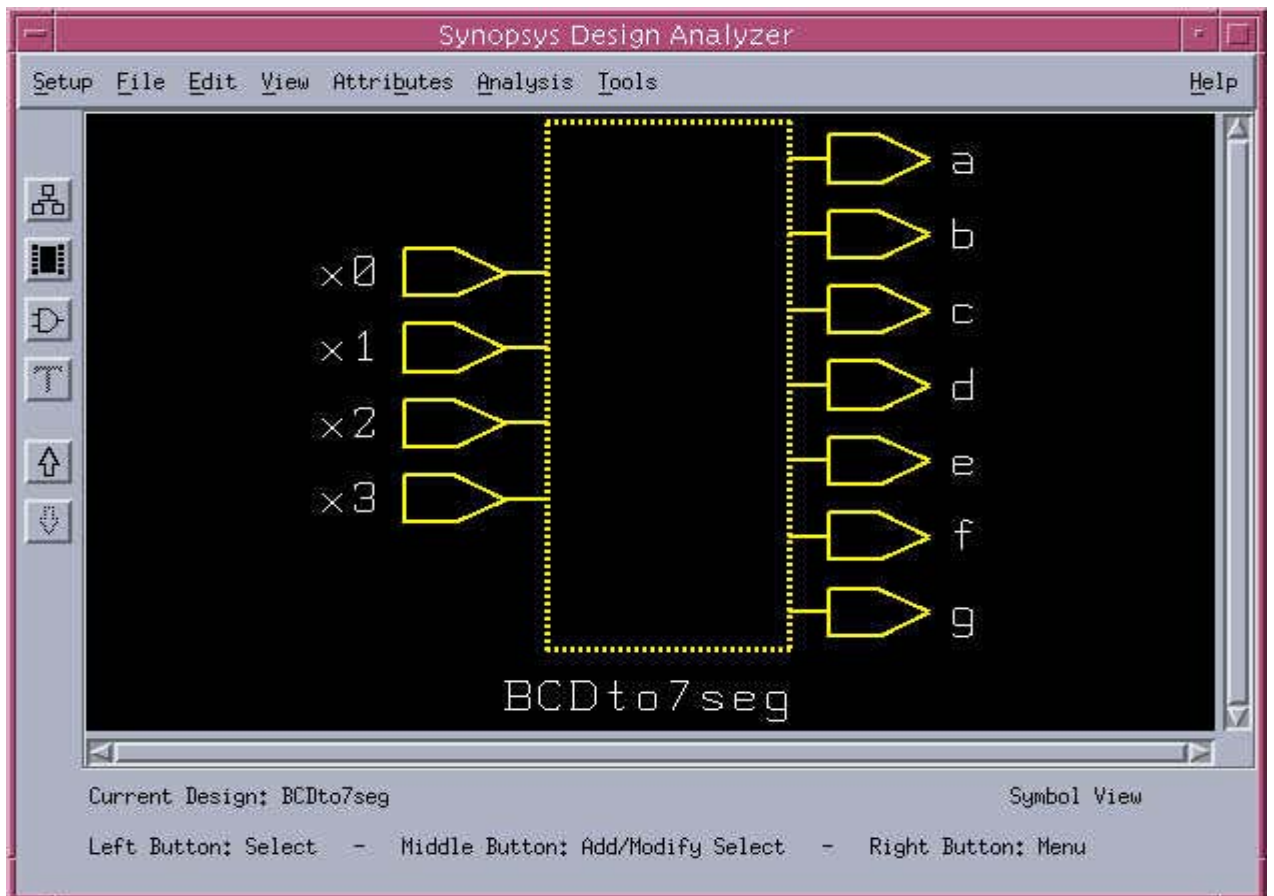
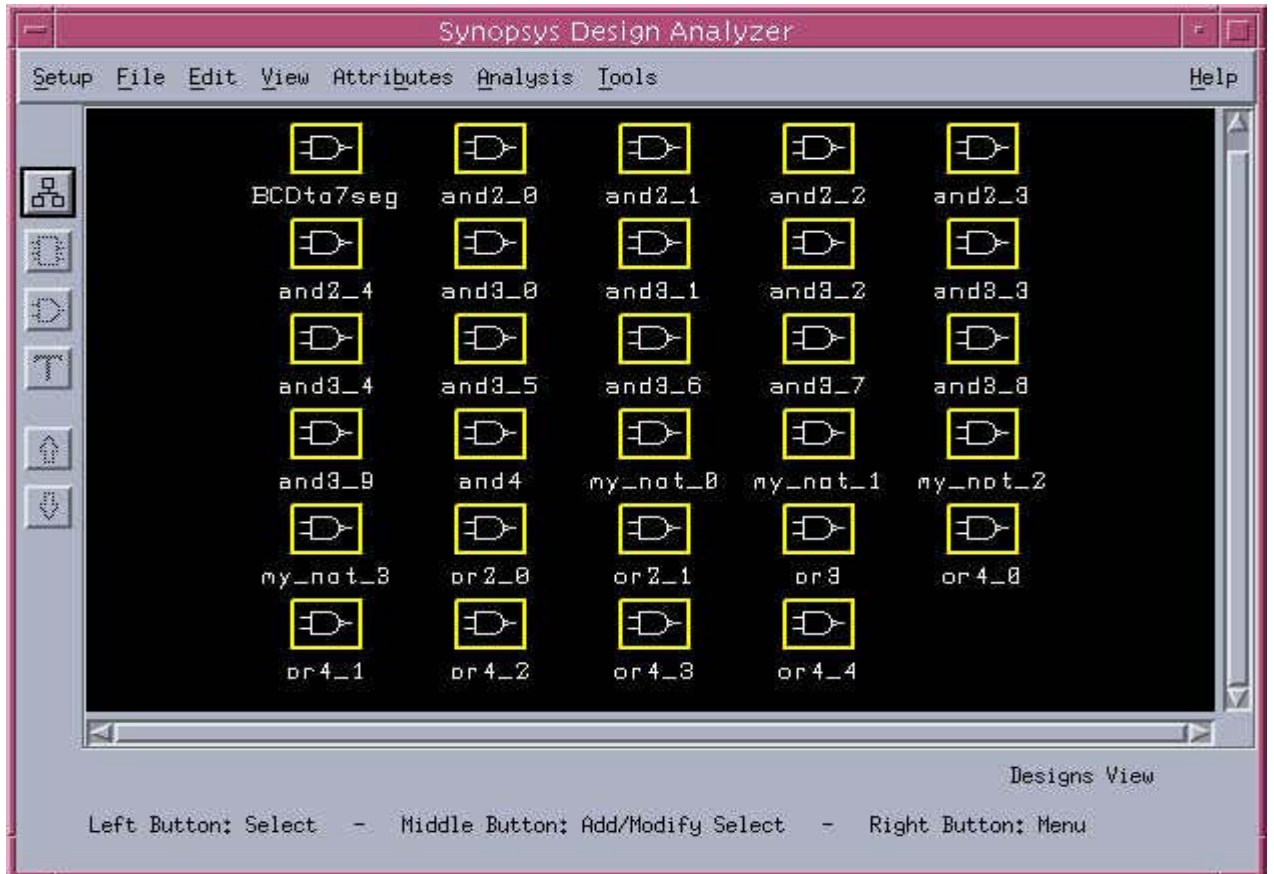
3. In the terminal (not in `dc_shell`) window, type `design_analyzer` to start design analyzer;

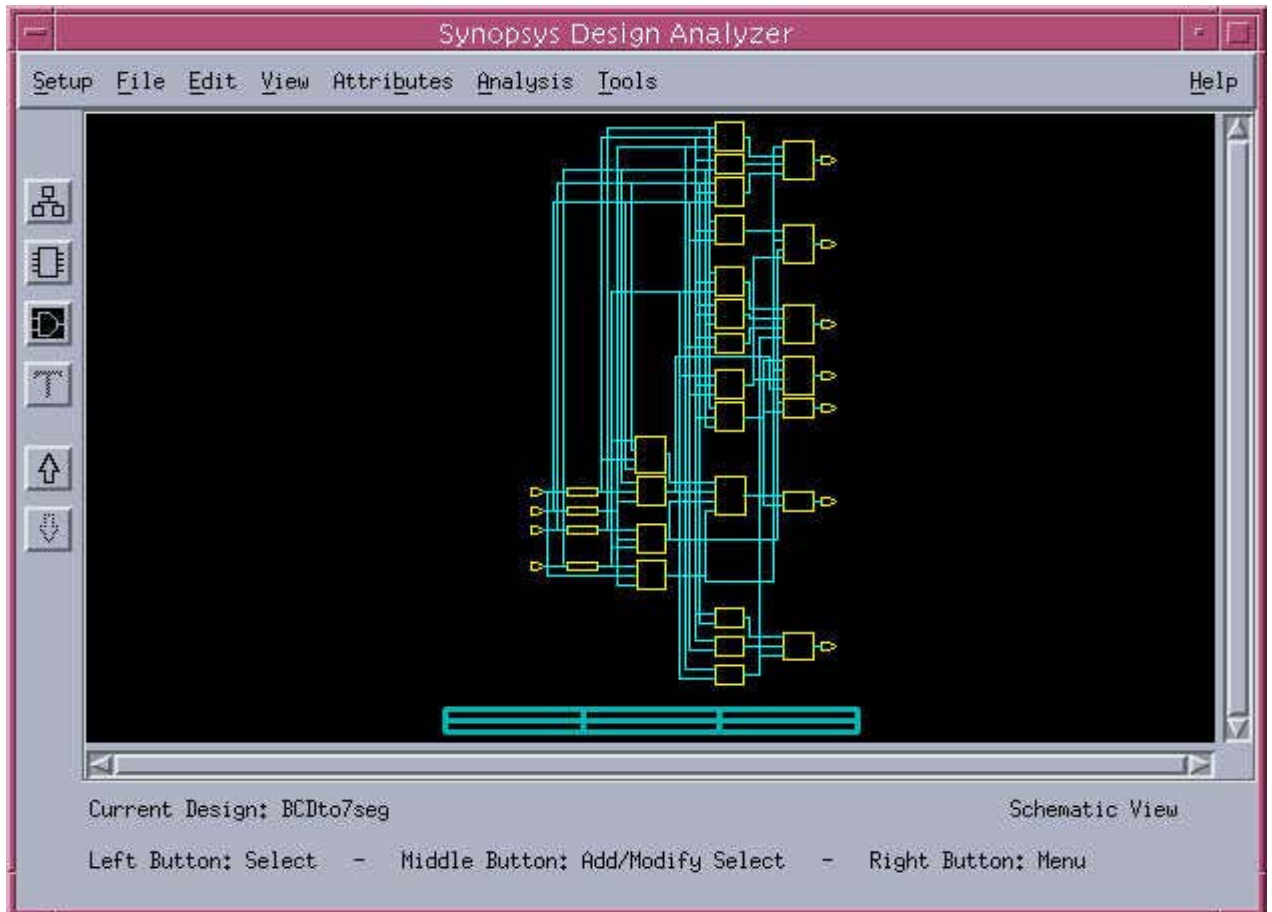


4. When a window (Synopsys Design Analyzer) prompts up, in the file menu, choose *Read...*, then choose your output file ***db*. then click OK.

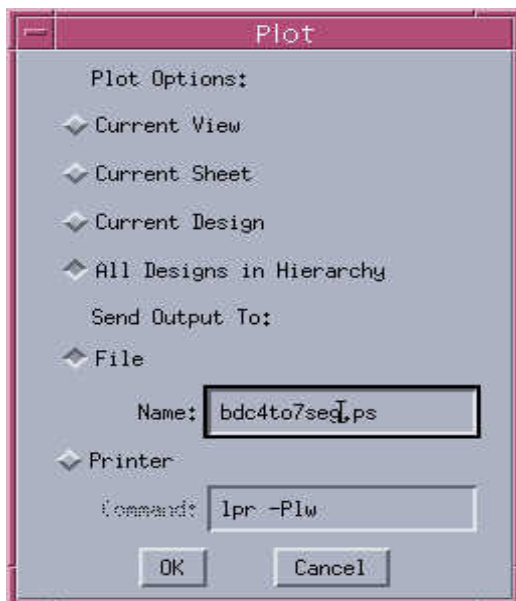


5. Now you should be able to view the hierarchy of the design. Double click the symbol can make you go down to a lower level.





6. If you want to print the result, in the file menu, choose *plot...* As you can see, you have 4 options to choose: Current View, Current Sheet, Current Design, All Designs in Hierarchy, choose any one you need. Then choose the *File* option, rename your output file as ****.ps.*(bcd4to7seg.ps in the example) Then you will find the file in the root directory.



Appendix:

ece368/src/example/and2.vhdl:

```
entity and2 is
    port(x,y:in bit;
          z:out bit);
end entity and2;
```

architecture behav of and2 is

```
begin
    z <= '1' after 6 ns when x = '1' and y = '1'
        else '0' after 6 ns;
```

```
end architecture behav;
```

ece368/src/example/and3.vhdl

```
entity and3 is
    port(x,y,p:in bit;
          z:out bit);
end entity and3;
```

architecture behav of and3 is

```
begin
    z <= '1' after 9 ns when x = '1' and y = '1' and p = '1'
        else '0' after 9 ns;
```

```
end architecture behav;
```

ece368/src/example/and4.vhdl

```
entity and4 is
    port(x,y,p,q:in bit;
          z:out bit);
end entity and4;
```

architecture behav of and4 is

```
begin
    z <= '1' after 12 ns when x = '1' and y = '1' and p = '1' and q = '1'
        else '0' after 12 ns;
```

```
end architecture behav;
```

ece368/src/example/or2.vhdl

```
entity or2 is
```

```
    port(x,y:in bit;
          z:out bit);
end entity or2;
```

architecture behav of or2 is

```
begin
    z <= '0' after 6 ns when x = '0' and y = '0'
        else '1' after 6 ns;

end architecture behav;
```

ece368/src/example/or3.vhdl

```
entity or3 is
    port(x,y,p:in bit;
          z:out bit);
end entity or3;
```

architecture behav of or3 is

```
begin
    z <= '0' after 9 ns when x = '0' and y = '0' and p = '0'
        else '1' after 9 ns;

end architecture behav;
```

ece368/src/example/or4.vhdl

```
entity or4 is
    port(x,y,p,q:in bit;
          z:out bit);
end entity or4;
```

architecture behav of or4 is

```
begin
    z <= '0' after 12 ns when x = '0' and y = '0' and p = '0' and q = '0'
        else '1' after 12 ns;

end architecture behav;
```

ece368/src/example/my_not.vhdl

```
entity my_not is
    port(x:in bit;
          z:out bit);
```

```
end entity my_not;

architecture behav of my_not is

begin
    z <= '1' after 3 ns when x = '0'
        else '0' after 3 ns;

end architecture behav;
```

ece368/src/example/BCDto7seg.vhdl

```
entity BCDto7seg is
    port(x3,x2,x1,x0: in bit;
        a,b,c,d,e,f,g: out bit);
end BCDto7seg;
```

```
architecture struct of BCDto7seg is
```

```
    component and2 is
        port(x,y:in bit;
            z:out bit);
    end component;
```

```
    component and3 is
        port(x,y,p:in bit;
            z:out bit);
    end component;
```

```
    component and4 is
        port(x,y,p,q:in bit;
            z:out bit);
    end component;
```

```
    component or2 is
        port(x,y:in bit;
            z:out bit);
    end component;
```

```
    component or3 is
        port(x,y,p:in bit;
            z:out bit);
    end component;
```

```
    component or4 is
```

```

        port(x,y,p,q:in bit;
              z:out bit);
end component;

component my_not is
    port(x:in bit;
          z:out bit);
end component;

signal internal_sig : bit_vector(16 downto 0);
signal not_x3, not_x2, not_x1, not_x0 : bit;

begin

    not1: my_not port map(x3, not_x3);
    not2: my_not port map(x2, not_x2);
    not3: my_not port map(x1, not_x1);
    not4: my_not port map(x0, not_x0);

    and2_0: and2 port map(not_x3, x1, internal_sig(0));
    and2_1: and2 port map(not_x3, not_x2, internal_sig(1));
    and2_2: and2 port map(not_x2, not_x1, internal_sig(2));
    and2_3: and2 port map(not_x3, x2, internal_sig(3));
    and2_4: and2 port map(not_x3, x0, internal_sig(16));

    and3_0: and3 port map(not_x3, x2, x0, internal_sig(4));
    and3_1: and3 port map(x3, not_x2, not_x1, internal_sig(5));
    and3_2: and3 port map(not_x3, not_x2, not_x0, internal_sig(6));
    and3_3: and3 port map(not_x2, not_x1, not_x0, internal_sig(7));
    and3_4: and3 port map(not_x3, x2, not_x1, internal_sig(8));
    and3_5: and3 port map(not_x3, x2, not_x0, internal_sig(9));
    and3_6: and3 port map(not_x3, not_x1, not_x0, internal_sig(10));
    and3_7: and3 port map(not_x3, x1, x0, internal_sig(11));
    and3_8: and3 port map(not_x3, not_x2, x1, internal_sig(12));
    and3_9: and3 port map(not_x3, x1, not_x0, internal_sig(13));

    and4_0: and4 port map(not_x3, x2, not_x1, x0, internal_sig(14));

    or4_0: or4 port map(internal_sig(0), internal_sig(4), internal_sig(5),
internal_sig(6), a);
    or4_1: or4 port map(internal_sig(7), internal_sig(8), internal_sig(9),
internal_sig(5), b);
    or4_2: or4 port map(internal_sig(1), internal_sig(2), internal_sig(10),
internal_sig(11), c);

```

```
    or4_3: or4 port map(internal_sig(8), internal_sig(12), internal_sig(13),
internal_sig(5), d);
    or2_0: or2 port map(internal_sig(7), internal_sig(13), e);
    or4_4: or4 port map(internal_sig(7), internal_sig(5), internal_sig(14),
internal_sig(12), internal_sig(15));
    or2_1: or2 port map(internal_sig(15), internal_sig(13), f);
    or3_0: or3 port map(internal_sig(3), internal_sig(2), internal_sig(16), g);

end architecture struct;
```

ece368/src/example/run

```
target_library = lsi_10k.db
symbol_library = lsi_10k.sdb
link_library = "*" + target_library
synthetic_library = ""
```

```
read -format vhdl and2.vhdl
read -format vhdl and3.vhdl
read -format vhdl and4.vhdl
read -format vhdl or2.vhdl
read -format vhdl or3.vhdl
read -format vhdl or4.vhdl
read -format vhdl my_not.vhdl
read -format vhdl BCDto7seg.vhdl
```

compile

```
write -format db -hierarchy -output BCDto7seg.db
```