Steps of RTL compiler:

1) Make a folder which contains the following files:
    a. Verilog code (.v file)
    b. Script file (.tcl file)
    c. Constraints file (.sdc file)
    d. Technology file (.lib file)
2) Open terminal.
3) Go to folder which contains script file (.tcl file) using command "cd"
4) Enter into the cadence tool 1- csh. 2- Source /cad/cshrc.
5) Invoke the RTL compiler using command "rc –f script.tcl"
6) gui_show.
7) Report area > path. (.txt file)
8) Report timing > path. (.txt file)
9) Report power > path. (.txt file)

A) Verilog code (counter):

```
module counter(clk, reset_n, enable_n, data_o);
output reg [7:0] data_o;
input enable_n, clk, reset_n;

always @(posedge clk) begin
if (reset_n)  data_o <= 8'b0 ;
else if (enable_n) data_o <= data_o + 1;
end
endmodule
```

B) Script file:

```
Set_attribute lib_search_path ../lib/
Set_attribute hdl_search_path ../rtl/
Set_attribute library xyz.lib
Read_hdl counter.v
Elaborate counter
Read_sdc ../constraints/constraints.sdc
Synthesize –to_mapped
Write_hdl > counter_netlist.v
```

C) Constraints file:

```
Create_clock –name clk –period 10 –waveform {0 5} [get_ports "clk"]
Set_clock_transition –rise 0.1 [get_clocks "clk"]
Set_clock_transition –fall 0.1 [get_clocks "clk"]
```

D) Verilog code (ALU):

```
module ALU (input clk,input [15:0]read_data1,input [15:0] read_data2,input [3:0]ALUCtrl,output
reg [15:0]ALU_output);
always @(ALUCtrl,read_data1, read_data2) begin
case(ALUCtrl)
4'b0010: ALU_output = read_data1 + read_data2;
4'b0110: ALU_output = read_data1- read_data2;
4'b0000: ALU_output = read_data1 & read_data2;
4'b0001: ALU_output = read_data1 | read_data2;
4'b0011: ALU_output = read_data1 * read_data2;
end
endcase
endmodule
```