Power Estimation using PTPX:

1. Run post-synthesis simulation and generate the .vcd files from your testbench.

   To dump the .vcd file, I used the below dumpvars & dumpfile commands after initial begin statement in the testbench: (Replace with your design and instance names)

   ```
   initial begin
           $dumpfile("syscluster_inv_tb.vcd");
           $dumpvars(0, testbench.sys0);
   ```

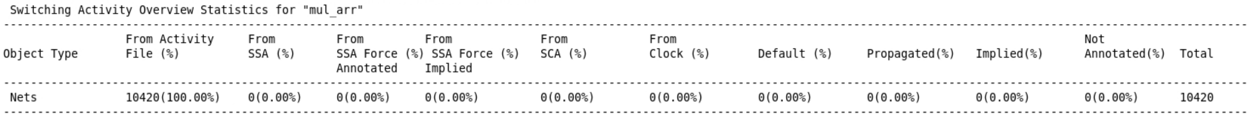
   Post-syn simulation can be run by adding a rule 'dve_syn' to the Makefile used for simulation:

   ```
   dve_syn:  $(HEADERS) $(SYNFILES) $(TESTBENCH)

   $(VCS) $(HEADERS) $(TESTBENCH) $(SYNFILES) $(LIB) +define+SYNTH_TEST -o syn_simv -R -gui
   ```

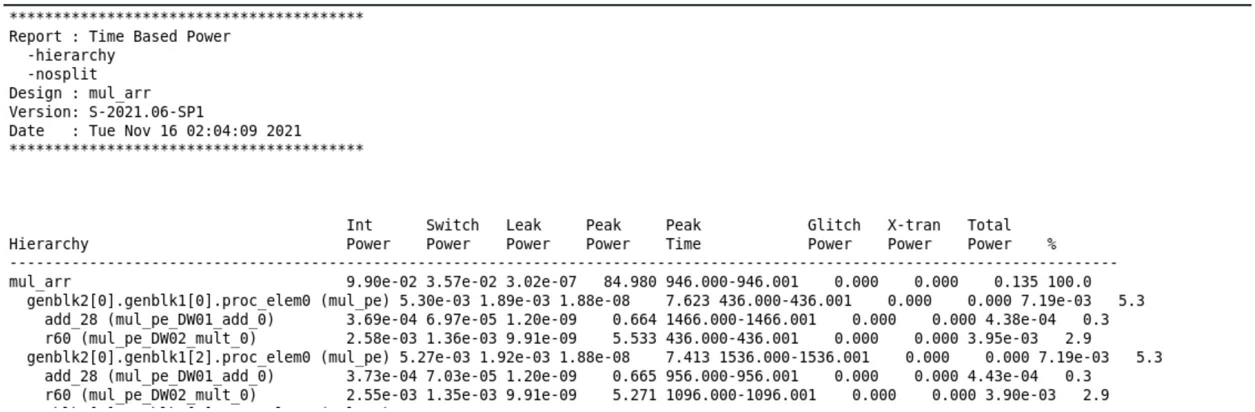
   To annote the delays, sdf annotation can be used.

2. Create a directory for eg. power/ and copy envset.tcl, pp.tcl and the Makefile in this directory.
3. Add the libraries and change the pointers of your files in the envset.tcl
4. Add the START_TIME and END_TIME in the envset.tcl for the power simulation based on your testbench (where you want to measure power).
5. Run 'make pp' to invoke pt_shell and run PTPX.
6. In the primetime log file, check the annotation report from report_switching_activity. Ideally, you should see good switching acitivity % and there shouldn't be 'not annotated' nets.



```
 Switching Activity Overview Statistics for "mul_arr"
---------------------------------------------------------------------------------------------------------------------------------------------------------------
                  From Activity  From        From          From            From           From                                                       Not
Object Type       File (%)       SSA (%)     SSA Force (%) SSA Force (%)   SCA (%)        Clock (%)      Default (%)    Propagated(%)  Implied(%)     Annotated(%)  Total
                                             Annotated     Implied
---------------------------------------------------------------------------------------------------------------------------------------------------------------
 Nets             10420(100.00%) 0(0.00%)    0(0.00%)      0(0.00%)        0(0.00%)       0(0.00%)       0(0.00%)       0(0.00%)       0(0.00%)       0(0.00%)      10420
---------------------------------------------------------------------------------------------------------------------------------------------------------------
```

7. Check the report dumped for the power metrics:



```
****************************************
Report : Time Based Power
  -hierarchy
  -nosplit
Design : mul_arr
Version: S-2021.06-SP1
Date   : Tue Nov 16 02:04:09 2021
****************************************



                                      Int      Switch   Leak     Peak     Peak             Glitch   X-tran   Total
Hierarchy                             Power    Power    Power    Power    Time             Power    Power    Power    %
--------------------------------------------------------------------------------------------------------------------------------
mul_arr                               9.90e-02 3.57e-02 3.02e-07   84.980 946.000-946.001    0.000    0.000    0.135 100.0
  genblk2[0].genblk1[0].proc_elem0 (mul_pe) 5.30e-03 1.89e-03 1.88e-08    7.623 436.000-436.001    0.000    0.000 7.19e-03   5.3
    add_28 (mul_pe_DW01_add_0)        3.69e-04 6.97e-05 1.20e-09    0.664 1466.000-1466.001    0.000    0.000 4.38e-04   0.3
    r60 (mul_pe_DW02_mult_0)          2.58e-03 1.36e-03 9.91e-09    5.533 436.000-436.001    0.000    0.000 3.95e-03   2.9
  genblk2[0].genblk1[2].proc_elem0 (mul_pe) 5.27e-03 1.92e-03 1.88e-08    7.413 1536.000-1536.001    0.000    0.000 7.19e-03   5.3
    add_28 (mul_pe_DW01_add_0)        3.73e-04 7.03e-05 1.20e-09    0.665 956.000-956.001    0.000    0.000 4.43e-04   0.3
    r60 (mul_pe_DW02_mult_0)          2.55e-03 1.35e-03 9.91e-09    5.271 1096.000-1096.001    0.000    0.000 3.90e-03   2.9
```