1. add_atpg_primitives

Use this command to create logical functions that may be constrained. Constraining these functions may be useful in avoiding undesirable conditions in patterns generated by the ATPG process.

```
DRC> add_atpg_primitives MY_ATPG_PRIM equiv \
   {BLASTER/MAIN/CPU/TP/CYCL/CDEC/U1936/in1 \
   BLASTER/MAIN/ALU_CORE/TP/CYCL/CDEC/U1936/in1 \
   BLASTER/MAIN/ALU_CORE/TP/CYCL/CDEC/U16/in2 \
   BLASTER/MAIN/ALU_CORE/TP/CYCL/CDEC/U13/in0 }
Gate with ID#=20201 has been added to the ATPG primitive list.
DRC> report_atpg_primitives -all
name           id#    type      inputs
------------   -----  -------   ----------------------------
MY_ATPG_PRIM   20201  EQUIV     861 990 1232 723
```

2. analyze_buses

Use this command to analyze potential problems associated with buses.

```
TEST-T> report_buses -contention fail
             contention    zstate   #drivers
  gate_id  status/capture  status  strong/weak  behavior_data
  -------  --------------  ------  -----------  --------------------
 Warning: Requested report contained no entries. (M13)
```

3. analyze_compressors

The analyze_compressors command enables you to run ATPG by creating a virtual compressor. You can select parameters for the compressor and the decompressor and run ATPG. Use the analyze_compressorscommand to perform a what-if analysis and explore the benefit of scan-compression on your design. This command deletes the internal patterns after an analyze compressors run. You must run ATPG in scan mode separately and compare results to the scan mode.

4. get_licenses

Use this command to manually check out selected licenses. Normally, this is not required because licenses are automatically checked out as needed.

BUILD> get_license test-faultsim
BUILD> report_licenses
test-fault-max
test-faultsim

5. gui_stop

After you have started displaying the TetraMAX GUI (using the gui_start command), enter the following command to exit the GUI:

gui_stop

6. test

This command changes the session to TEST command mode, where fault simulation and test generation can be performed. To enter TEST command mode, the DRC process must be successfully completed.

DRC> test

Begin scan design rules checking...
Begin simulating sg0 load_unload proc...
Begin clock rules checking...

.
.
.

DRC Summary Report

Warning: Rule V16 (miscounted formal parameters) failed 2 times.
Warning: Rule S19 (nonscan cell disturb) failed 2 times.
There were 4 violations that occurred during DRC process.
Design rules checking was successful, CPU time=0.66 sec.

---

7. report_feedback_paths

Use this command to get a report on the feedback paths in the design.

```
TEST> report_feedback_paths -all
id# #gates #sources sensitization_status
--- ------ -------- --------------------
 0    2        1      pass
```

8. report_power

Use this command to report a variety of power data. For related details, see "Power-Aware ATPG " in the TetraMAX User Guide.

```
TEST-T> report_power -per_pattern -percentage
Error: Invalid pattern source (random). (M74)
```

9. set_toggle_weights

This command is used in the power-aware ATPG flow to specify separate toggle weights for both shift and capture cycles. By default, each toggle or transition at a flip-flop (FF) is scored as a 1. This scoring only considers the flip-flop and does not take into account the fan-out of the flip-flop. By using toggle weighting, you can place an integer number, 1 or higher, onto those flip-flops to represent larger fan-out nets and cones of logic.

```
RC-T> set_toggle_weights TCAM/ff1 -weight 5 -shift -capture
```

10. set_delay

Use this command to specify options for transition fault and path delay fault test generation and fault simulation.

```
TEST> set_delay -max_tmgn 80%
```