Tutorial 1.a: Test pattern generation for combinational circuit

Objective: Generate the test pattern for combinational circuit.

Note: Tessent accepts verilog code only in gate level modeling.

Invoke tessent tool with *atpg.do* file

```
>> tessent  -shell  -dofile  atpg.do  –logfile  log  –replace
```

1. Set context for test pattern generation which will decide the operation of Tessent tool
   ```
   >>set_context  patterns  –scan
   ```

2. Read verilog file *c1.v*
   ```
   >>read_verilog  c1.v
   ```

3. Set the top module name as current design name
   ```
   >>set_current_design  c1
   ```

4. Set the mode of operation. The three different modes are available in Tessent namely *setup, insertion and analysis*. In setup mode, the required verilog files and library files are included. The scan insertion is done during insertion mode. The test pattern generation and fault simulation is performed during analysis mode.
   ```
   >>set_system_mode  analysis
   ```

5. Select the fault type. Stuck fault model is the default fault model.
   ```
   >>add_faults  –all
   ```

6. Generate the test patterns for selected fault model
   ```
   >>create_patterns
   ```

7. Generate the report which contains fault coverage and list of faults selected
   ```
   >>report_faults  –class  DS
   ```

8. Write the test pattern in .ascii and verilog formats for readability and fault simulation.

   ```
   >> write_patterns  ./c1_pattern.ascii   –ascii   –parallel   –replace
   >> write_patterns  ./c1_pattern.v   –verilog   –parallel   -replace
   ```

Tutorial 1.b: Fault simulation

Objective: Activate the fault at any particular node of circuit *c1*. For stuck at one (zero) fault, insert OR (AND) gate to the corresponding node in RTL code (*c1.v*). Perform the fault simulation

Fault injection:

Invoke tessent tool with *fault_sim.do* dofile

```
>> tessent  -shell   -dofile   fault_sim.do   –logfile   log_fault   –replace
```

1. Set the context for fault simulation

   ```
   >>set_context  patterns  –scan
   ```

2. Read the fault injected verilog file *c1.v*

   ```
   >>read_verilog  c1.v
   ```

3. Set the top module name as current design name

   ```
   >>set_current_design  c1
   ```

   set_system_mode analysis - then alone the add fault will works

4. Select the fault type. Else stuck fault model is the default fault model in Tessent

   ```
   >>add_faults  –all
   ```

5. Generate the test patterns for selected fault model

   ```
   >>create_patterns
   ```

6. Generate the report which contains fault coverage and list of faults selected

   ```
   >>report_faults   –class DS
   >>report_statistics
   ```

7. Write the test pattern in *.ascii* and verilog formats for readability and fault simulation.

```
>> write_patterns   ./c1_pattern1.ascii  –ascii –parallel   –replace
>> write_patterns  ./c1_pattern1.v   –verilog   –parallel   –replace
```

Fault simulation with external test pattern:

9. Repeat the steps upto 4 from fault injection

10. Generate the test pattern separately using ATPG algorithm or BIST circuit. This external test patterns can be included in .ascii file without affecting the ASCII file format.

```
>>set_pattern_source  external c1_external.ascii
```

Or

```
>>read_patterns c1_external.ascii
```

11.Simulate the external test pattern set

```
>>simulate_patterns   -store_patterns   all
```

12. Report the fault coverage and detected fault list

```
>>report_statistics
>>report_fault  -class  DS
```