

4. Übung: PROL16: Automated Test Pattern Generation

Name(n):

Punkte:

In der letzten Übung haben Sie mit Hilfe des Synopsys Design Compilers eine Scan-Chain in Ihren PROL16 eingebaut. Auf Basis dieser Scan-Chain sollen in dieser Übung mit Hilfe des ATPG-Werkzeugs Synopsys TetraMAX (t_{max}) Testvektoren für *Single-Stuck-At*-Fehler erstellt werden.

1 Theorie

- Gute Einführungen in das Thema ATPG finden Sie beispielsweise in [Men08, Kapitel 2] und [Syn08b, Kapitel 10]. Erklären Sie die folgenden Fehlerklassen:

- Detected
- Posdet (Mentor)/Possibly Detected (Synopsys)
- Untestable/Undetectable
- ATPG Untestable
- Undetected/Not Detected

2 ATPG in der Praxis

Machen Sie sich mit den Grundlagen des ATPG-Design-Flows vertraut [Syn08b, Kapitel 4]. Stellen Sie sicher, dass Sie alle benötigten Dateien zur Verfügung haben: die Netzliste und die STIL-Datei werden vom Design Compiler erzeugt, die Bibliotheken (in [Syn08b] als *Models* bezeichnet) stehen Ihnen im Verzeichnis `$AMS_DIR/verilog/c35b3/` zur Verfügung.

Erzeugen Sie nun die Testmuster für Ihren Entwurf. Das folgende Skript fasst die Schritte aus [Syn08b, Kapitel 4] zusammen:

```
1 read netlist netlist/prol16.v
2 read netlist $AMS_DIR/verilog/c35b3/*.v
3 report modules -summary
4 report modules -error
5 run build_model cpu
6 run drc prol16.spf
7 set faults -fault_coverage
8 set faults -summary verbose
```

```

9| add faults -all
10| set patterns internal
11| run atpg -random
12| report summaries > ../doc/atpg_results.rep
13| write pattern pfile.stil -format stil99 -replace
14| write faults ../doc/atpg_fault_list.rep -all -replace

```

- Interpretieren Sie die Ergebnisse, die in die Datei `atpg_results.rep` ausgegeben werden! Beschreiben Sie in diesem Zusammenhang auch die beiden Coverage-Werte (siehe [Syn08b, Kapitel 10]).
-

3 Simulation der Testmuster

TetraMAX erstellt in der Datei `pfile_stildpv.v` eine Verilog-DPV-Testbench [Syn08a] (DPV: *Direct Pattern Validation*), mit der Sie die erstellten Testvektoren simulieren können. Führen Sie eine Simulation mit dieser Testbench durch¹. Beachten Sie, dass die erstellte Testbench eine PLI-Bibliothek verwendet und Sie die Simulation daher mit dem Parameter `-pli` starten müssen:

```

1| vlog ../syn/netlist/pro116.v
2| vlog ../syn/pfile_stildpv.v
3| vsim -pli /eda/etc/synopsys/libstildpv/libstildpv.so cpu_test
4| run -all

```

Genauere Informationen zum Erstellen dieser PLI-Bibliothek können Sie [Syn08a, Kapitel 2] entnehmen.

- Erstellen und simulieren Sie nun zwei fehlerhafte Versionen der Netzliste: die erste Version soll einen Single-Stuck-At-Fehler enthalten, der in der Fehlerklasse *Detected by Simulation* liegt, die zweite einen nicht-detektierbaren Fehler. Dokumentieren Sie die Auswahl und die Vorgehensweise und interpretieren Sie die Ergebnisse.
-

Literatur

[Men08] Mentor Graphics Corporation. *Scan and ATPG Process Guide*, 2008.

[Syn08a] Synopsys, Inc. *Test Pattern Validation User Guide*, September 2008.

[Syn08b] Synopsys, Inc. *TetraMAX ATPG User Guide*, September 2008.

¹Wenn Sie die Signale Ihres DUTs im Wave-Fenster betrachten, können Sie anhand des Takts und des Eingangs `scan_enable_i` den Scan-Vorgang beobachten.