FH-OÖ Hagenberg/ESD
Metrikorientierter Hardwareentwurf, WS 2015
Rainer Findenig, Markus Lindorfer, Markus Schutti © 2008 (R 2395)

3. Übung: PROL16: Scan Test

Name(n): Punkte:

1 Einbau der Scan-Kette

Wie in der letzten Übung konzeptionell erlernt, sollen Sie nun Scankette in Ihren PROL16 eingebauen. Wie bereits erwähnt, läuft dies in der Praxis natürlich automatisiert ab; abgesehen von einer kleinen Code-Änderung werden Sie also nur das Syntheseskript verändern.

Zuerst wird Synopsys instruiert, scanfähige FFs zu verwenden, indem den Parameter `-scan` beim `compile`-Kommando hinzugefügt wird:

```
compile -scan
```

Danach kann spezifiziert werden, wie die Scankette einzubauen ist:

```
set_scan_configuration -style multiplexed_flip_flop
set_scan_configuration -chain_count 1
set_dft_signal -view existing_dft -type Reset \
        -port [get_port {res_i}] -active_state 0
set_dft_signal -view existing_dft -type ScanClock \
        -port [get_port {clk_i}] -timing [list 45 55]
set_dft_signal -view existing_dft -type ScanDataIn \
        -port [get_port {mem_data_i[0]}]
set_dft_signal -view existing_dft -type ScanDataOut \
        -port [get_port {mem_data_o[0]}]
set_dft_signal -view existing_dft -type ScanEnable \
        -port [get_port {scan_enable_i}]
set_dft_signal -view existing_dft -type Constant \
        -port [get_port {test_mode_i}] -active_state 1
create_test_protocol
```

Es empfiehlt sich, vor dem eigentlichen Einbau der Scankette zu kontrollieren, ob eventuell DfT-Verletzungen vorhanden sind:

```
dft_drc > ../doc/${design}.pre_scan.drc
preview_dft -show all > ../doc/${design}.pre_scan.preview
```

Wenn dieser Design Rule Check (DRC) fehlerfrei durchläuft, kann die Kette eingebaut werden:

```
set_dft_insertion_configuration -synthesis_optimization none
set_dft_insertion_configuration -map_effort low
```

```
insert_dft
```

Das Ergebnis können Sie aus den folgenden Reports entnehmen:

```
dft_drc > ../doc/${design}.drc
report_scan_path -view existing_dft -chain all \
        > ../doc/${design}.scan_report
report_dft_signal -view existing_dft           \
        >> ../doc/${design}.scan_report
report_dft_configuration                       \
        >> ../doc/${design}.scan_report
```

Sie werden feststellen, dass sich (mindestens) zwei Probleme im Entwurf befinden: über `mem_oe_no` und `mem_we_no` wird das Taktsignal an Primärausgänge geleitet. Dies ist eine DfT-Verletzung, für den Entwurf allerdings notwendig. Daher wird nun der spezielle Eingang `test_mode_i` verwendet: wenn `test_mode_i=1` sollen die beiden Ausgänge nicht mit dem Takt verknüpft, sondern direkt ausgegeben werden.

Zum Schluss sollen eine Netzliste und eine STIL-Datei ausgegeben werden: beide Dateien dienen als Eingabe für das ATPG-Tool. Die STIL-Datei können Sie mit dem folgenden Befehl erzeugen:

```
write_test_protocol -out ${design}.spf
```

Beachten Sie, dass Sie das Syntheseskript derart anpassen müssen, dass die Netzliste statt in VHDL in Verilog ausgegeben wird, da Synopsys nach dem Einfügen einer Scan-Kette oft fehlerhaften VHDL-Code erzeugt. In diese Netzliste müssen Sie zusätzlich noch die benötigten Include-Dateien einfügen:

```
`include "/eda/ams/verilog/c35b3/c35_CORELIB.v"
`include "/eda/ams/verilog/udp.v"
```

Der für den Einbau der Scankette relevante Teil des Synthesescriptes steht Ihnen im Verzeichnis `$MHE3_HOME/templates/synopsys` als Datei `insert_scan.tcl` bereit.

2 Golden Simulation

Nun soll noch einmal der gesamte Entwurf, inkl. eingebauter Scankette und SDF-Annotation, simuliert werden, um zu verifizieren dass durch den Einbau der Scankette keine Fehler im Entwurf entstanden sind.