

Dateiname: StA032_Lorenz_J

Titel:

Entwicklung eines Transaktionsinterfaces für eine C++-Testbench basierend auf SystemC2.0

Bearbeiter:

Jens Lorenz

Text der Kurzfassung:

Die Studienarbeit hat gezeigt, dass man mit wenig „Handgriffen“ SystemC zum Verifizieren von HDL-Designs effizient nutzen kann. Es ermöglicht auf eine einfache Weise unterschiedliche, abstrakte Testmodelle, so genannte Testcases, zu entwickeln.

Das in der Aufgabenstellung erwähnte DUT wurde entworfen und hat seinen Funktionstest bestanden. Einzelne Module des DUTs wurden reziprok zur Top-Down-Methode in Verilog getestet und anschließend zusammengefügt. Das Topmodul wurde anschließend mittels der vorhandenen CTB verifiziert. Das fehlerfreie DUT machte es daraufhin möglich, die selbstgeschriebene SystemC-Testbench zu verifizieren.

Das Design selbst erkennt ein FAW in allen der drei vorgestellten optischen Übertragungsprotokolle. Somit kann das DUT in der Diplomarbeit eingesetzt werden, um die Anbindung an das Emulationssystem mit Hilfe eines bekannten und einfachen Designs durchzuführen. Für die Performancetestreihe wird dieses Testdesign durch einen wesentlichen komplexeren Block ersetzt.

Kern der Aufgabe war es, die CTB so umzuschreiben, dass man statt eines Verilog Testcase ein SystemC Testcase verwendet. Durch Komplikationen beim Kompilieren eines Beispieltests in SystemC mit implementierter CTB-Bibliothek, konnte die Aufgabe noch nicht fertig gestellt werden. Letztendlich beweist die selbstgeschriebene Testbench die Funktionstüchtigkeit der Testumgebung.

Die TB besitzt eine standardisierte μ C-Interfacesteuerung, einen Sende- und einen Empfangsmanager. Das μ C-Interface ermöglicht den Zugriff auf die internen Register des DUT. Der Sende- und der Empfangsmanager besitzen vier Funktionen, welche dem Test-Ingenieur unterschiedliche Möglichkeiten eröffnen, auf den Test einfach Einfluss zu nehmen.

Nicht näher besprochen wurde die Verwendung der Bibliothek SCV und die von Cadence entwickelte Bibliothek CVE. Beide ermöglichen es einfach und schnell durch den erweiterten Funktionsumfang komplexe Tests zu erstellen. Denkbar wäre es, mit einem Befehl aus der CVE-Bibliothek, den Inhalt der Payload des Sende- oder Empfangsmanagers mit Zufallszahlen zu füllen.