

Prof. Dr. Rolf Drechsler (drechsle@informatik.uni-bremen.de)
Dipl. Inform. Daniel Große (grosse@informatik.uni-bremen.de)

Bremen, Juni 2005

FB 3 – Institut für Informatik
MZH, Raum 3510
Tel: 0421 218 7389

2. Praktische Aufgabe zur Vorlesung

Qualitätsorientierter Hardware-Entwurf

Thema:

In den vorliegenden Aufgaben geht es darum die Verifikationsmethode des *Bounded Model Checking (BMC)* näher kennen zu lernen. Auf der Website der Veranstaltung findet ihr das arbeitsgruppeninterne Tool *CheckSyC*, welches BMC für in *SystemC* geschriebene Schaltungen mittels temporaler Eigenschaften erlaubt. Es ist nun eure Aufgabe dieses Tool zur Verifikation einzusetzen.

Vorbereitung:

Nach dem Runterladen und Entpacken des Archivs (`praxis2_verification.zip`) in das Verzeichnis `verification/` wechseln und dort „`. setpath`“ (der Punkt und das Leerzeichen sind wichtig) aufrufen. Dadurch werden die benötigten Tools in den Pfad eingetragen.

Im Unterverzeichnis `0or/` befindet sich ein einfaches Verifikationsbeispiel. Der Ablauf der Verifikation und einige Hinweise sind in der README Datei beschrieben.

Aufgabe 1:

Ihr sollt zunächst die Rolle eines Verifikationsingenieurs übernehmen. Seine Aufgabe es ist die bereits kennen gelernte Schaltung der ALU gemäß der bekannten Spezifikation formal und unter zu Hilfenahme des Tools zu verifizieren. Dazu hier noch einmal die genaue Spezifikation für die Schaltung:

Die ALU nimmt zwei 4-Bit Zahlen a und b auf und führt mit ihnen eine von insgesamt acht verschiedenen Operation durch, die durch eine 3-Bit breite Select-Leitung sel ausgewählt wird. Das Ergebnis wird an den Ausgang r geschrieben. Die Operationen sollen wie folgt kodiert werden:

0. Logisches „Oder“
1. Logisches „Und“
2. Logisches „Exklusives Oder“
3. Identität der Zahl a
4. Addition
5. Konkatenation der Zahl an Eingang b von rechts an die Zahl an Eingang a
6. Multiplikation
7. Null

Verwendet die im Unterverzeichnis `1alu/` enthaltene Version der ALU, übersetzt das Modell und überprüft durch Formulieren entsprechender Eigenschaften, ob die gegebene Schaltung korrekt arbeitet, d.h. gemäß der Spezifikation. Gewährleistet, dass nach eurer Arbeit als Verantwortlicher der Verifikation eine korrekte Schaltung aus der Implementation synthetisiert werden kann. Haltet eure Arbeitsschritte und die dabei erlangten Erkenntnisse schriftlich in einer Art Arbeitsprotokoll fest.

Aufgabe 2:

Nun geht es um eine Schaltung die einen Modulo-8-Zähler realisiert (Unterverzeichnis `2counter/`). Überlegt euch, was der Zähler leisten sollte, um als korrekt zu gelten und haltet diese Überlegungen schriftlich fest. Definiert und formuliert anschließend Theoreme, die diese Eigenschaften überprüfen und vergewissert euch mit Hilfe des Tools, dass die Eigenschaften zutreffen.

Hinweise:

Die Aufgaben und Tools sind von der Website nur unter Eingabe eines Benutzernames und Passworts abrufbar. Benutzername und Passwort werden im Tutorium bekannt gegeben. Erweiterte Konstrukte der verwendeten Eigenschaftssprache werden im Tutorium besprochen.

(Abgabe bitte spätestens bis zum 28. Juni.)