Basispraktikum Hardwarenaher Systementwurf
WS 11/12
Arbeitsbericht 1

Sarah Lutteropp, Parto Karwat

14. November 2011

1 Vorwort

In diesem Aufgabenblatt ging es darum, einfache Schaltungen in VHDL zu beschreiben und zu testen, um mit der Sprache VHDL und den Simulationswerkzeugen (Modelsim, GHDL, GTKWave) vertraut zu werden.

2 Schaltungsentwurf mit VHDL

2.1 Allgemeine Vorgehensweise

Die VHDL-Dateien haben wir mithilfe des Texteditors Kate erstellt und bearbeitet. Besonders am Anfang hatten wir parallel zum Editorfenster die Folien der VHDL-Einführung geöffnet und uns von den Codebeispielen auf den Folien bei der Entwicklung unseres eigenen Quellcodes leiten lassen. Beim Coden haben wir uns alle paar Zeilen abgewechselt und simultan den Quelltext besprochen.

2.2 1.1 bis 1.4

Siehe angehängte Quelltexte.

2.3 Bonusaufgabe

Als Schaltung unserer Wahl haben wir eine Schaltung mit drei Eingängen und einem Ausgang implementiert, die folgendes leistet: Die Schaltung zählt einen Wert modulo 42 hoch und gibt ihn aus. Dabei haben die drei Eingänge folgende Funktion: Der erste Eingang dient dazu, die clk weiterzugeben, der zweite Eingang fungiert als Reset-Eingang und der dritte Eingang signalisiert, dass das Modul das Hochzählen pausieren soll. Ist der Reset-Eingang auf 0 gelegt, so gibt die Schaltung eine 0 aus. Ist dieser auf 1 gelegt, so zählt die Schaltung, falls sie gerade nicht pausieren soll, modulo 42 hoch. Hierzu

haben wir folgende weitere Bibliothek (nach einer Internetrecherche) eingebunden: ieee.numeric_std.all . Diese Bibliothek stellt u.a. Funktionalitäten zur Umwandlung eines Integer-Wertes in einen std_logic_vector zur Verfügung.
Die Bonusaufgabe ist in den Dateien bonus.vhd und bonus_tb.vhd gespeichert.

3 Simulation

3.1 Modelsim

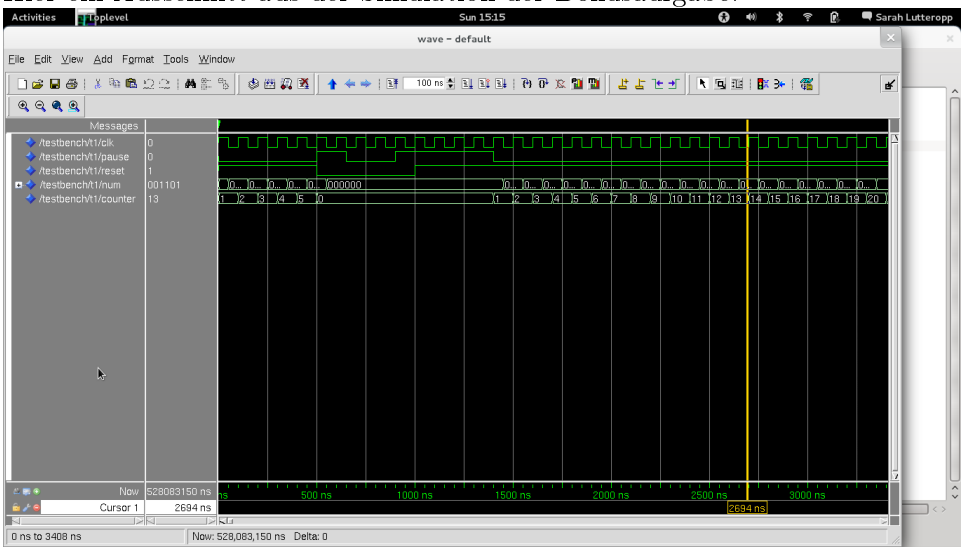
3.1.1 Direktes Setzen der Eingänge

Geht mit 'force', z.B.:

```
force -freeze sim:/in2_out1/a 0
```

3.1.2 Simulation mit Hilfe einer Testbench

Hier ein Ausschnitt aus der Simulation der Bonusaufgabe:



3.1.3 Skriptgesteuerte Ausführung

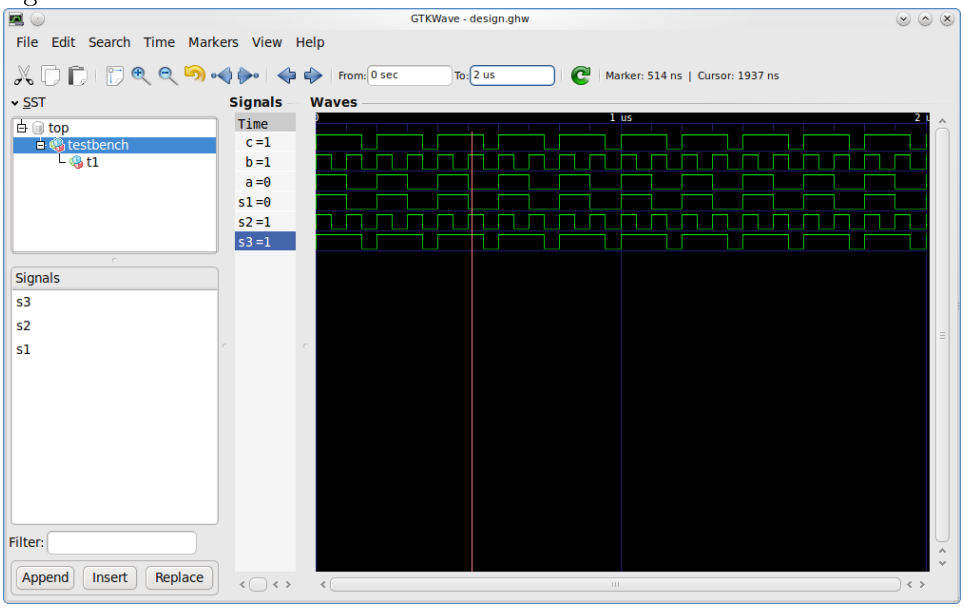
vsim -c -do script.do
Ein Beispielskript ist unter dem Dateinamen script.do zu finden.
... (hier sollten wir noch was hinzufügen)

3.2 GHDL und GTKWave

Um unsere Testbenches zu simulieren, haben wir folgende Befehle verwendet:

```
$ ghdl -i *.vhd \newline
$ ghdl -m NAME_DER_TESTBENCH
$ ghdl -r NAME_DER_TESTBENCH --wave=design.ghw --stop-time=2000ns
$ gtkwave design.ghw
```

Hier ein Beispiel für den Signalverlauf der zum Modul mynand zugehörigen Testbench bei GTKWave:



3.3 Vor- und Nachteile der verschiedenen Simulationsprogramme

3.3.1 Modelsim

Vorteile	Nachteile
• direktes Setzen der Eingänge möglich	• unpraktischer eingebauter Texteditor
• reichlich Dokumentation vorhanden	• erfordert Nutzungslizenz
• hoher Bekanntheitsgrad	• kostenpflichtig, closed source
• alles in einer Anwendung	• leicht gewöhnungsbedürftige Bedienung zu Anfang
• große Funktionsvielfalt	• mühsamere Installation

3.3.2 GHDL

Vorteile	Nachteile
• einfache Bedienung per Kommandozeile	• Bedienung nur über Kommandozeile
• kostenfrei, Open Source	• weniger Dokumentation vorhanden
• statisches Verhalten, keine Eingriffsmöglichkeit	• erfordert Testbenches
• andere Fehlermeldungen als Modelsim	• benötigt Configurations, falls mehrere Testbenches in einer Entity testbench enthalten sind

3.3.3 GTKWave

Vorteile	Nachteile
• GUI vorhanden	• weniger Einstellungsmöglichkeiten
• kostenfrei, Open Source	• weniger Dokumentation vorhanden
• übersichtliche Anzeige	• keine Möglichkeit, manuell Signale zu setzen und in die Simulation einzugreifen
• selbsterklärend	• nur reiner Viewer von von GHDL vorher generierter Datei
	• 'Zoom best fit' nicht automatisch eingestellt