



Einführung zum Praktikumsversuch

Getting Started With *DE1-SoC* Board Using *VHDL*

Prof. Dr. Martin J. W. Schubert, Electronics Laboratory,
OTH Regensburg, Regensburg, Germany

Einführung zum Praktikumsversuch

Getting Started With *DE1-SoC* Board Using *VHDL*

Kurzfassung. Dieses Dokument ist eine Hinführung zum Praktikumsversuch "*Getting Started With DE1-SoC Board Using VHDL*" im Elektroniklabor (Raum S081) der OTH Regensburg. Es unterstützt Studierende ohne Vorkenntnisse mit digitaler Hardware und / oder der Hardware-Beschreibungssprache *VHDL*.

1 Einleitung

Auffinden der Unterlagen

Zum Auffinden der Versuchsanleitung gehen Sie bitte mit einem Internet-Browser auf die Homepage von Prof. Schubert, indem Sie entweder mit einer Internet-Suchmaschine den Text "OTH Prof Schubert" suchen oder direkt die URL <https://hps.hs-regensburg.de/~scm39115/> eingeben. Von dort aus folgen Sie den Links → Ausbildungsangebot → Praktika → Praktika mit DE1-SoC-Boards → [Getting Started with DE1SoC Board.pdf](#).

Aufenthalt im Labor

Im Elektroniklabor bekommen Sie eine Belehrung über die Laborordnung und müssen unterschreiben, dass Sie diese akzeptieren. Wichtigste Punkte sind:

- Seien Sie aus Sicherheitsgründen nie allein im Labor.
- Seien Sie vorsichtig mit hohen Spannungen (im Elektroniklabor nur in den Steckdosen).
- Verwenden Sie Geräte nur bestimmungsgemäß.
- Lebensmittel – auch Getränke – sind im Labor nicht erlaubt.
- Lassen Sie einen offenen Account nicht unbeaufsichtigt, verwenden Sie Screen-Lock, z.B. mittels gleichzeitigem Drücken der Tasten *CTRL+Alt+Entf*, dann → *Sperren*.

Bereitgestellte Labortische und Materialien

Setzen Sie sich an eine der 10 Werkbänke des Elektroniklabors. Unter jeder Werkbank steht ein Rollwagen mit 4 Schubladen. In der obersten Schublade befindet sich ein Voltmeter, das Sie verwenden dürfen, alles Andere, was Sie benötigen, finden Sie in der 2. Schublade. Wenn etwas fehlt fragen Sie Ihren Betreuer, bitte kannibalisieren Sie nicht andere Werkbänke oder Schubladen.

2 Grundlagenwissen zum Versuch

Was ist ein *FPGA*?

Das Akronym *FPGA* steht für "*Field Programmable Gate Array*". Dies ist ein „im Feld“ – also bei der Anwendung – programmierbarer Logikbaustein. So wie ein Motor nicht allein verwendet werden kann sondern nur eingebettet in ein Auto, so ist auch das *FPGA* in ein Board eingebettet, welches es mit Strom versorgt und Anschlüsse zur Außenwelt anbietet.

Was ist *VHDL*?

Das Akronym *HDL* steht für "*Hardware Description Language*". Computer – auch Mikrocomputer – werden mit Programmiersprachen programmiert. Zur Programmierung von digitaler Hardware (wie beispielsweise einem *FPGA*) verwendet man eine *Hardware Description Language* oder kurz *HDL*. Weltweit haben sich 2 *HDLs* durchgesetzt: *VerilogHDL* dominiert im amerikanischen und asiatischen Raum, *VHDL* in Europa.

Programmieren mit *VHDL* in diesem Praktikum

In diesem Praktikum genügt es den gegebenen *VHDL*-Quelltext gemäß Anleitung zu übernehmen, zu kompilieren und in das *FPGA* zu laden. Dann wird das *FPGA* programmiert und es sollte erst einmal etwas passieren, allerdings mit einem offensichtlichen Fehler, der im *VHDL*-Quelltext relativ leicht zu korrigieren ist. Wer das tun will sollte Folgendes verstehen:

VHDL allgemein:

- Text nach einem doppelten Minuszeichen ist Kommentar, z. B.: `-- dies ist Kommentar`
- "`<=`" ist ein sog. Treiber, z.B.: `a<=b; -- Signal a den Wert von Signal b zuweisen`
- Das Semikolon beendet eine Anweisung, nicht das Zeilenende, z.B.: `a<=b; b<=c; c<=d;`
- Der Operator **NOT** invertiert Signale: Aus '0' wird '1', und aus '1' wird '0', z.B.: `y<= NOT a;`
- Ein Bitvektor besteht aus mehreren parallelen Bits, dargestellt in "...". Beispiel: "1010101".

Wissen für die in der Versuchsanleitung geforderte VHDL-Code-Korrektur:

- Die Signale `sw (#)`, `#=0...9`, kann man mit den schwarzen Schiebeschaltern (rück)setzen.
- Die Signale `key (#)`, `#=0...3`, kann man durch Drücken der 4 sog. Push-Buttons setzen.
- Die Signale `hex0`, `hex1...hex5` sind Bitvektoren mit 7 Bits, auf dem Board als 7 parallele Leitungen realisiert, wobei Signal `hex#` die 7-Segment-Ziffer `#` antreibt, `#=0...5`.
- `c_seg7 (x)`, `x=0...15`, sind Bitvektoren mit je 7 Bits im *VHDL*-Code, wobei auf `c_seg7 (x)` die Bits für jene Segmente auf logisch '1' gesetzt sind, welche auf einer 7-Segment-Ziffer leuchten müssen, um die hexadezimale Ziffer `x` anzuzeigen: `x= 0...9, A, b, C, d, E, F`.
Beispiele: `c_seg7 (0) = "1111110"`, `c_seg7 (1) = "0110000"`, `c_seg7 (8) = "1111111"`.
Beispiel: `hex1 <= c_seg7 (5)` ; zeigt auf der Ziffer 1 der 7-Segment-Anzeige eine 5 an.
- Der **NOT**-Operator vor einem Bitvektor invertiert alle seine Bits. Beispiel:
`hex1 <= NOT c_seg7 (5)` ; -- alle 7 Bits werden invertiert zugewiesen

3 Zusammenfassung

Dieses Dokument ist eine kurze Hinführung zum Versuch "*Getting Started With DE1-SoC Board Using VHDL*" im Elektroniklabor der *OTH Regensburg*.