

Technische Universität Ilmenau

Sommersemester 2017

Praktikum Technische Informatik
Übersicht zu Versuch 3 für Basic Engineering School

Stand:
20. August 2017

1 Einleitung

Im Rahmen der Basic Engineering School wird der dritte Praktikumsversuch im Fach Technische Informatik mithilfe des Autonomen Miniaturtransporters (AMT) durchgeführt. Im Folgenden werden die zu realisierenden Komponenten und das dazu nötige Vorgehen kurz erläutert.

2 Allgemeine Hinweise zum Umgang mit den Quartus Vorgaben

2.1 Vorgaben

Zum Erfüllen der Aufgaben wird ein **Archiv** zur Verfügung gestellt. Dieses Enthält die *Quartus Projekte*, die bearbeitet werden sollen und darüber hinaus zwei fertige Beispielprojekte die den Umgang mit der Software und verschiedenen Möglichkeiten aufzeigen können. Dazu steht das Dokument **QuartusEinführung** zur Verfügung das den *Umgang mit der Quartus Software im Allgemeinen* erklärt. Außerdem sind dort Erklärungen zu den Beispielen zu finden. Im Archiv ist jedem Ordner eine **read.me**-Datei zugeordnet, die *Funktion und Struktur* des jeweiligen erläutern.

2.2 Bearbeitung

Alle zu bearbeitenden Dateien sind aufgrund der Definition ihrer Schnittstellen bereits vorhanden. Sie dürfen ausschließlich in dem Projekt bearbeitet werden, zu dem sie gehören, da sonst Probleme mit mehrfach existierenden Blöcken auftreten können.

Die alten Versionen der Dateien sollen jeweils durch die bearbeiteten ersetzt werden, da sich ihre Benennung nicht verändern darf. Die in ihnen bestehende Logik ist ausschließlich zu erweitern, beziehungsweise die Automaten grafen mit dem Grafen Editor (State Machine Wizard) unter *Edit an existing State Machine Design* zu bearbeiten.

Nachdem eine Datei bearbeitet wurde, sind folgende Schritte auszuführen:

- Falls Graf vorliegt: *VHDL erzeugen*
- Aus der Block/Schematic-Datei oder der VHDL-Datei *Symbol-Datei erzeugen oder aktualisieren*
- In allen Dateien – auch in anderen Projekten – die bearbeiteten Symbole einbinden, *Symbole und Blöcke aktualisieren*
- Prüfen, dass – wo verwendet – an veränderten Blöcken Ein- und Ausgangssignale korrekt verbunden sind. Vertauschungen können zu Fehlern führen, die nur schwer zu finden sind.

Erklärungen zu den Schritten sind dem Dokument **QuartusEinführung**-Dokument zu entnehmen.

3 Aufgabenstellung

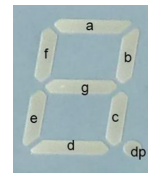
3.1 BCD-Zähler

Als erstes soll ein Automat erstellt werden, der binär codiert von 0 bis 9 zählt. Dazu macht man sich eine Eigenschaft der Automaten zu Nutzen und verwendet ausschließlich den Clock Eingang. Realisiert werden soll der Entwurf im vorgegebenen *IP-Cores.qpf* Projekt. Dieses enthält bereits eine Vorgabe für den Zähler. Diese ist mithilfe des Grafen Editors zu vervollständigen.

Letztlich zählen die Zähler die Anzahl der Kreuzungen an den links oder rechts abgebogen, beziehungsweise geradeaus gefahren wurde.

3.2 7-Segment-Decoder

Anschließend sollen die BCD-Zahlen über die Anzeigen am AMT ausgegeben werden. Dazu ist ein Decoder nötig, der die Binärzahlen in logische Werte (1 und 0) für die Ansteuerung einzelnen Segmente der Anzeigen umwandelt. Die ermittelte Logik ist ebenfalls im *IP-Cores.qpf* Projekt in die vorgegebene Datei einzufügen. Hierbei ist darauf zu achten, dass die Segmente low-active sind (logisch 0 → Segment leuchtet; logisch 1 → Segment dunkel).



3.3 Fahrstrecke

Abschießend soll ein Automat erstellt werden, der die gruppenspezifische Wegstrecke beschreibt und die erforderlichen Fahrhinweisungen an den Fahrautomaten übergibt. Dazu ist im *AMT.qpf* Projekt ebenfalls ein Automat mit Ein- und Ausgängen erstellt worden. Dieser kann mithilfe des Grafen Editors vervollständigt werden. Die Logik des Fahrstreckenautomaten ist aus dem der Fahrsteuerung herzuleiten. Die Dokumentation dazu ist im Dokument **Digitalsteuerung_AMT** zu finden. Dort wird auch die Wahl der verschiedenen Fahrmodi beschrieben.