Praktikum

Entwurf digitaler Systeme für FPGAs

Versuch 2

Sieben-Segmentanzeige

Betreuer:
Tobias Kleinfeld, Raum BB 811, 0203-379-1090, tobias.kleinfeld@uni-due.de
Pascal Raffelberg, Raum BB 818, 0203-379-2819, pascal.raffelberg@uni-due.de

Teilnehmer:

_____

Gruppen-Nr.

_____

Matr. Nr.:

_____

Versuchsdatum:
Versuchsbeginn/ende:

Bescheinigung der Teilnahme
Testat erteilt / nicht erteilt

Unterschrift: Datum:

Duisburg, den 15.05.2017 T. Kleinfeld

1 Einleitung

Der Inhalt dieses Versuchs besteht darin, einen Controller für die Ansteuerung von Sieben-Segmentanzeigen zu implementieren. Sieben-Segmentanzeigen stellen sehr preiswerte Anzeigeelemente dar. Für einfache Anwendungen wie Uhrzeit- und Temperaturanzeige verwendet man sie nach wie vor. Um die Ansteuerung der einzelnen Segmente zu vereinfachen soll auf dem FPGA ein Controller implementiert werden, der in der Lage ist ein Datennibble[1] in die zugehörige Zahl umzusetzen.

Das Board stellt 8 Hardwareschalter für den Nutzer bereit. Eine Gruppe von je 4 Schaltern soll nun als Eingang für das Datennibble dienen und über seine Schalterstellungen die Binärzahlen repräsentieren. Ein weiterer Schalter, der für die Darstellung nicht verwendet wird, kann dazu benutzt werden, um zwischen den Anzeigemodi der Zahlen von 0 bis 15 im Hexadezimalsystem und Dezimalsystem zu wechseln.

Die Platine, bestehend aus Sieben-Segmentanzeige, Vorwiderständen und Pin-Header SV1, wird nachfolgend als Sieben-Segment-Modul bezeichnet. Der zugehörige Schaltplan ist in Abbildung 1 angegeben. Der Anschluss an das FPGA-Board erfolgt über den Pin-Header SV1. Die Symbole A bis G stellen Labels dar und verknüpfen identische Bezeichner miteinander.

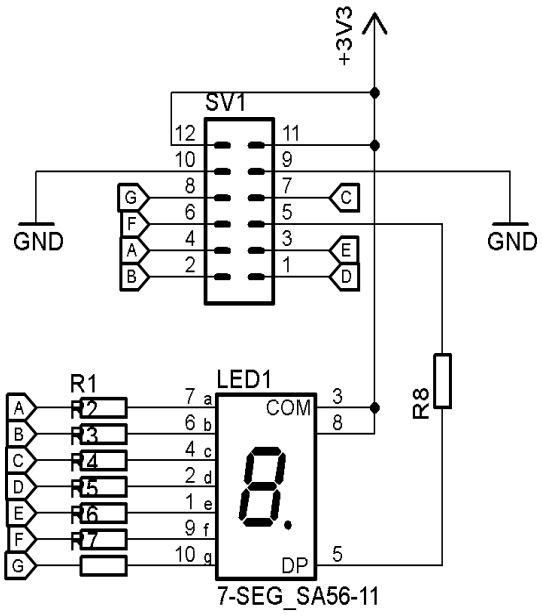

Abbildung 1: Schaltplan der Sieben-Segmentanzeige

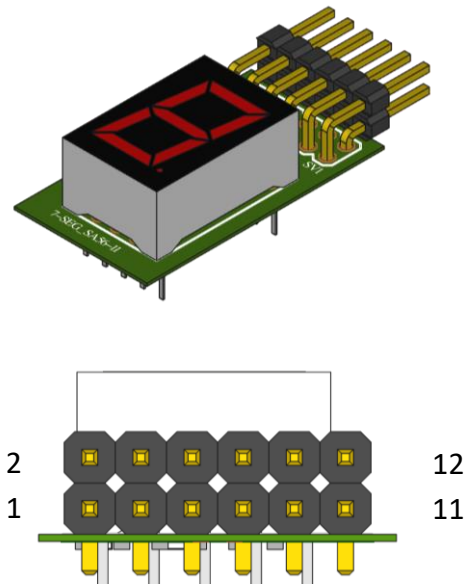

Abbildung 2: Ansichten der Sieben-Segmentanzeige mit Pinbelegung

[1] [1] Ein „nibble“ entspricht einem halben Byte (Halbbyte) und somit 4Bit.

2 Vorbereitungsaufgaben

Die Sieben-Segmentanzeige ist aus einzelnen LEDs aufgebaut, welche über die Anschlüsse **A** bis **G** angesteuert werden. Hierbei ist zu beachten, dass die LEDs als **„Common-Anode“** verschaltet sind (Siehe 7seg_Manual.pdf in moodle).

1. Suchen Sie im „7-Seg Manual“ das elektrische Schaltbild und geben Sie dieses an. Zeichnen Sie zusätzlich den Anschluss für die Versorgungspannung, die Vorwiderstände und die Anschlüsse zum FPGA ein.

2. Die LEDs sollen durch logische Zustände geschaltet werden. Geben Sie die Zugehörigkeit von logischem Pegel und LED Zustand an. Beachten Sie die „Common-Anode“ Verschaltung.

Die Sieben-Segmentanzeigen werden an den **PMOD-Schnittstellen** des FPGA-Boards betrieben. In der Betriebsanleitung des FPGA-Boards[2] findet Sie hierzu eine Tabelle mit den Pin-Zuordnungen der Board-Hardware zum FPGA. Hier finden Sie ebenfalls die Pin-Zuordnung für die Hardwareschalter SW0 bis SW8.

3. Erstellen Sie anhand von Tabelle 1 das Pin-Mapping für zwei Sieben-Segmentanzeigen an PMOD-A und PMOD-B.

4. Zahlen und Buchstaben können durch eine Kombination der LEDs A bis G dargestellt werden. Erstellen Sie eine Tabelle nach dem Schema von Tabelle 2 für alle darstellbaren Zeichen.

In Verilog kann die „parameter“-Syntax genutzt werden um konstanten Werten einen ALIAS zuzuordnen. Durch diese Maßnahme kann der Quelltext lesbarer und wartbarer gemacht werde.

5. Geben Sie den Verilog-Code für einen ALIAS „ZERO“ an, welcher das Mapping auf den LED-Bit-Vektor (A bis G) für das Zeichen „0“ realisiert.

6. Der Controller soll Anhand der Eingangs-Schalterstellung dem Ausgang einen zugehörigen Wert liefern. Welche Verilog-Konstrukte stehen Ihnen allgemein zur Verfügung um Entscheidungszweige zu realisieren? Welche Struktur eignet sich für den genannten Fall am besten?

[2] Reference Manual in moodle

3 Versuchsdurchführung

3.1 Programmierung

Erstellen Sie ein Projekt mit dem Namen „Sieben_Segmenanazeige.v“. Das Modul soll taktsynchron arbeiten und über einen „low-active-reset“ (reset_n) wieder in den Initialisierungszustand zurückwechseln.
Mit Hilfe der Schalter 0 bis 3 soll die darzustellende Zahl kodiert und auf den zwei Segmentanzeigen dargestellt werden:

- 0 bis 15 für dezimale Darstellung
- 0 bis F für hexadezimale Darstellung

Ein weiterer nicht belegter Schalter, zum Beispiel Schalter 7, soll verwendet werden um zwischen einer dezimalen und einer hexadezimalen Darstellung der Zahl, die sich aus dem „Schaltercode“ ergibt, wechseln zu können. Für die dezimale Darstellung der Zahl steht Ihnen die zweite Sieben-Segmentanzeige zur Verfügung.

3.2 Simulationsaufgabe

Testen Sie mit Hilfe einer Testbench ihren Programmcode. Schreiben Sie eine Testbench, die alle Zeichen durchtestet und vergleichen Sie das Ergebnis mit der Tabelle aus den Vorbereitungsaufgaben.

3.3 Implementation & Test

Schließen Sie die Sieben-Segmentanzeigen an die gewählten PMOS an. Implementieren Sie anschließend ihr Programm auf dem FPGA. Überprüfen Sie die einzelnen Zeichen und das Umschalten der zwei Betriebsmodi.

4 Optionale Zusatzaufgabe: BCD

Darstellung einer zweistelligen Dezimalzahl über ein high-nibble[3] und ein low-nibble, die jeweils BCD-codiert sind. Im vorangegangen Versuchsteil haben wir gesehen, dass alle Zahlen von 0 bis 9 mit Hilfe von 4-Bits dargestellt werden können. Dieses Verfahren nennt sich „Binary-Coded-Decimal“-Codierverfahren. Mit zwei Sieben-Segmentanzeigen und den Schaltern 0 bis 7 können somit alle Zahlen von 0 bis 99 dargestellt werden. Programmieren Sie dazu den Verilog-Code und testen sie die Funktion mit Hilfe der Anzeigeelemente.

5 Anhang

	LED Char	Pin am SV1	Pin am PMOD	Name am PMOD	FPGA Pin
PMOD-A	A				
	B				
	C				
	D				
	E				
	F				
	G				
PMOD-B	A				
	B				
	C				
	D				
	E				
	F				
	G				

Tabelle 1: Pin-Mapping FPGA zu LEDs

	A	B	C	D	E	F	G
AUS							
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
A							
B							
C							
D							
E							
F							

Tabelle 2: Mapping von LEDs zu Zahlen