Hochschule Karlsruhe
University of Applied Sciences
Fakultät für Elektro- und Informationstechnik

Studiengang Elektro- und Informationstechnik (Bachelor)

Versuchsvorbereitung Labor Digitaltechnik

Versuch 8: Das letzte Gefecht

Jan Hoegen*

Erstellt am 11. Juni 2022
Betreuer: Prof. Dr.-Ing. Jan Bauer

Inhaltsverzeichnis

1 Analyse des Systementwurfs **2**
1.1 Erklärung des Top-Level-Systementwurfs . . . 2
1.2 Zählrichtung definieren . . . 2
1.3 Ein- und Ausfahrt detektieren durch Lichtschranken . . . 2
1.4 Ansteuerung der Einerstelle des Dekadenzählers . . . 3
1.5 Ansteuerung der Zehnerstelle des Dekadenzählers . . . 3
1.6 Beschriftung der Top-Level-Signale . . . 3

2 Verständnis des Automaten **3**
2.1 Moore oder Mealy? . . . 3
2.2 Vervollständigen der Simulation . . . 4

3 Auswahl der I/O-Pins **4**

4 Wissen transferieren **4**
4.1 Verständnis der Anlasswiederholsperre . . . 4
4.2 Transformation auf den Mealy-Automanten . . . 5

*Matrikelnumer: 82358. E-Mail: hoja1028@h-ka.de.

1 Analyse des Systementwurfs

1.1 Erklärung des Top-Level-Systementwurfs

Es werden die einzelnen Schaltblöcke des Top-Level-Systementwurfs erklärt.

input_synchronizer Stellt sicher, dass das Eingangssignal der Lichtschranken zwischen 0 V und V_{cc} liegt.

statemachine Das Signal der Lichtschranken wird analysiert und der aktuelle Zustand bestimmt. Daraufhin werden Zählrichtung und Zählimpuls an die Speicherblöcke *decade_counter* weitergeleitet.

decade_counter Hier werden der Wert der Einerstelle und der Zehnerstelle getrennt gespeichert. Erhält die Einerstelle einen Impuls von der *statemachine*, so wird um den Wert 1 hoch- bzw runtergezählt. Die Richtung wird durch das Signal *dir* bestimmt. Die Zehnerstelle wird durch ein Overflow- bzw. Underflowsignal der Einerstelle angesteuert.

sevenseg_decoder Schließlich wird der aktuelle Wert der Speicher decodiert, sodass die 7-Segment-Anzeige die zugehörige Dezimalziffer anzeigt.

1.2 Zählrichtung definieren

Wenn das Signal *dir* auf 1 gesetzt ist, wird vorwärts gezählt und um 1 inkrementiert. Bei $dir = 0$ wird abwärts gezählt.

1.3 Ein- und Ausfahrt detektieren durch Lichtschranken

Signalverlauf *opt_switch*: S2 S1.

Einfahrt	Ausfahrt
00	00
01	10
11	11
10	01
00	00

1.4 Ansteuerung der Einerstelle des Dekadenzählers

Der Dekadenzähler zählt immer dann in die vorgegebene Richtung, wenn eine positive Taktflanke anliegt und der Eingang *enable* auf 1 gesetzt ist.

Solange also $enable = 0$ gilt, wird der aktuelle Zustand beibehalten. Genau dann, wenn *statemachine* feststellt, dass das Aus- bzw. Einfahren beendet ist, wird *enable* für eine Periode auf 1 gesetzt. Der Speicher zählt somit um einen Wert weiter.

1.5 Ansteuerung der Zehnerstelle des Dekadenzählers

Wenn für die Einerstelle ein Overflow von 9 auf 0 oder ein Underflow von 0 auf 9 erfolgt, wird das Signal *ripple* des COUNTER_LOW für eine Periode auf 1 gesetzt. Dieses Signal wird an den *enable*-Eingang des COUNTER_HIGH angeschlossen. Dieser zählt also in genau diesem Fall um einen Wert weiter.

1.6 Beschriftung der Top-Level-Signale

Abbildung 1 zeigt die Beschriftung der Top-Level-Signale.

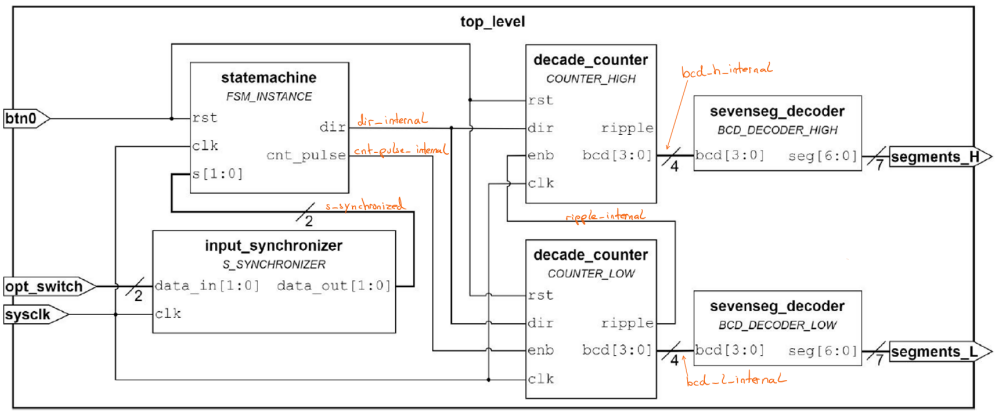

Abbildung 1: Beschriftung der Top-Level-Signale

2 Verständnis des Automaten

2.1 Moore oder Mealy?

Da der Ausgang vom aktuellen Zustand un dem Eingang abhängt, liegt ein Mealy-Automat vor.

2.2 Vervollständigen der Simulation

Da S2 das MSB ist, muss Simulation 1 zu Szenario B gehören und Simulation 2 zu Szenario A. Abbildung 2 zeigt die vervollständigte Simulation.

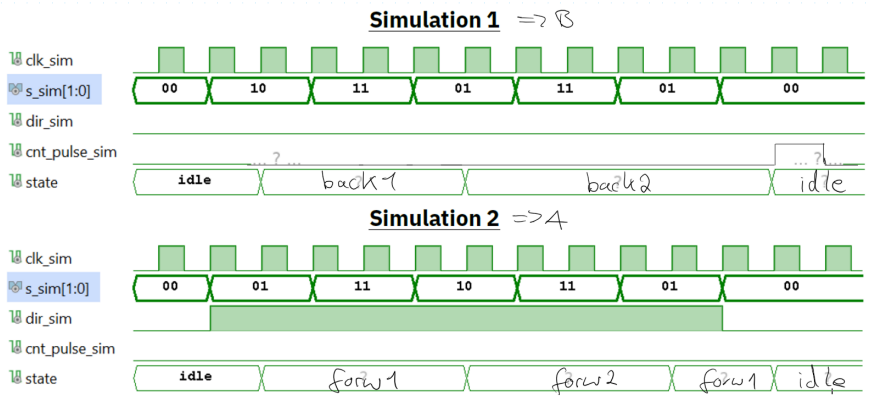

Abbildung 2: Vervollständigen der Simulation

3 Auswahl der I/O-Pins

Abbildung 3 zeigt die Pinbelegung des FPGA.

4 Wissen transferieren

4.1 Verständnis der Anlasswiederholsperre

Die Zustände werden als Variablentyp in VHDL definiert. Damit lassen sie sich auf leicht verständliche Weise referenzieren und in case-Statements verwenden.

Indem zwei Prozesse verwendet, werden, lassen sich Speicher und Kombinatorik getrennt voneinander betrachten. Im kombinatorischen Teil werden durch ÜSN und ASN festgelegt, welchen Wert der nächste Zustand haben wird. Im sequentiellen Prozess wird festgelegt, wann der Zustandswechsel erfolgt.

In einem case-Statement können die Zustände, die vorher als Variablentypen definiert wurden, als einzelne Fälle betrachtet werden. Durch If-Anweisungen werden die Eingangssignale überprüft und der entspreche Output gesetzt. Außerdem wird damit der nächste Zustand festgelegt.

warum case gtu geeignet?

wie werden übergange realisiert? wie viele verzweigungen gibt es

4.2 Transformation auf den Mealy-Automanten

wie viele when fälle

wie viele if zweige für jeden zustand

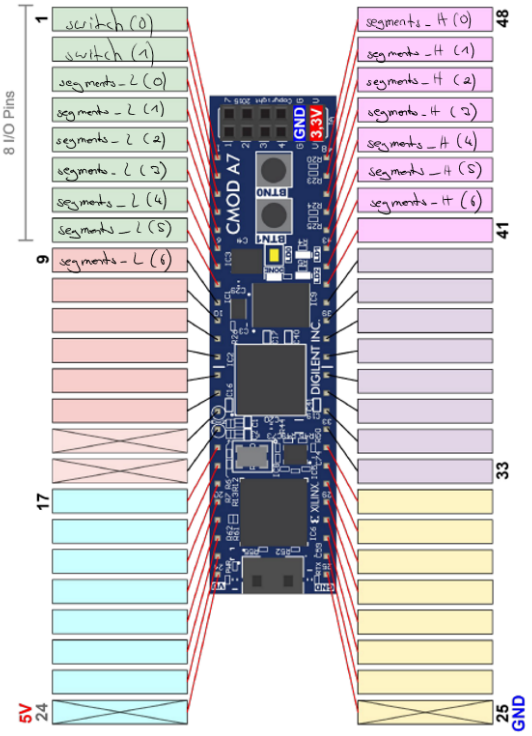

Abbildung 3: Pinbelegung des FPGA