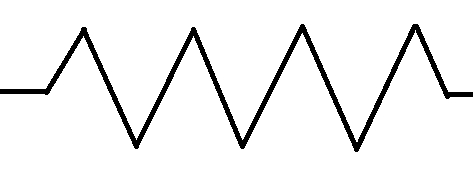
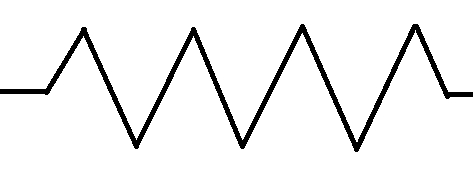
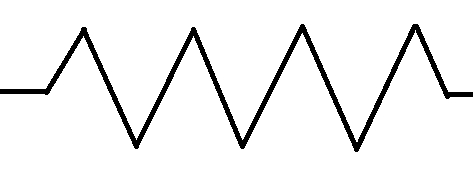
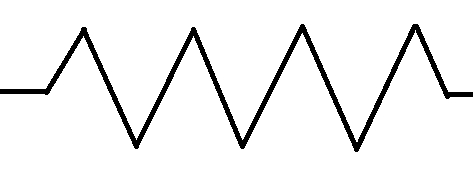
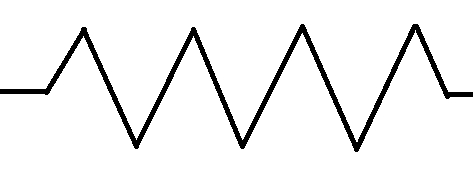
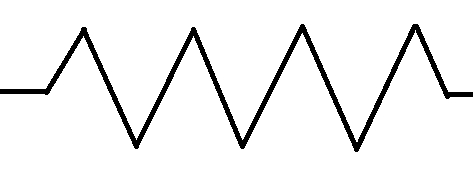
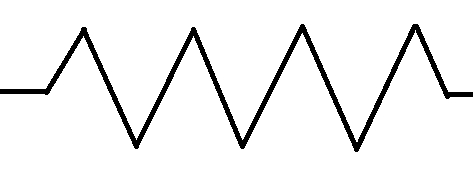
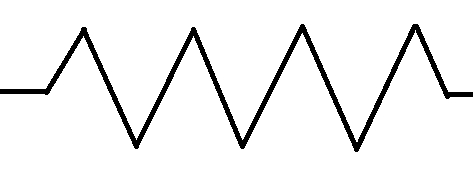
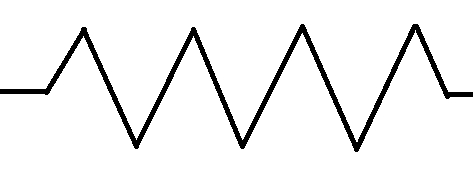
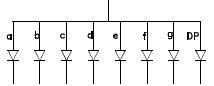
**Name: Xuanjiao Zhu Nr.3038674**

Sieben Segmentanzeige

**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**



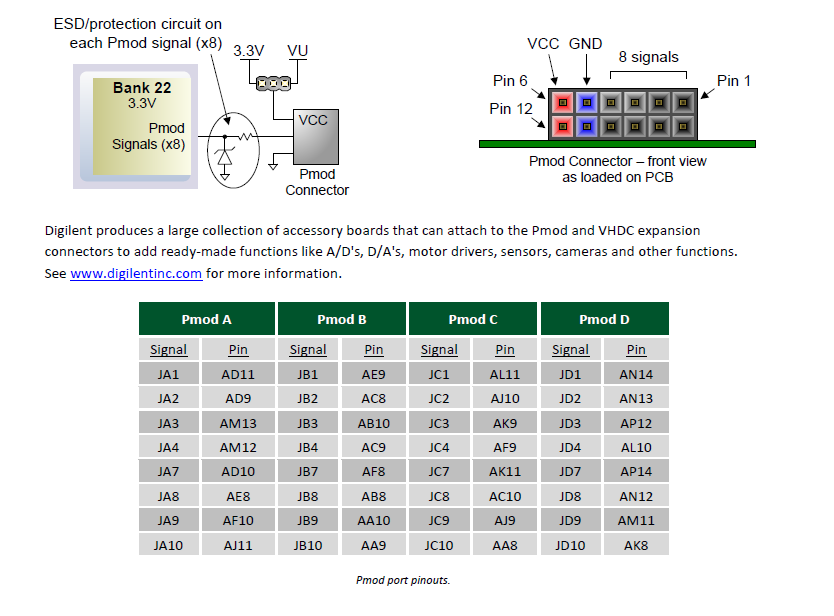
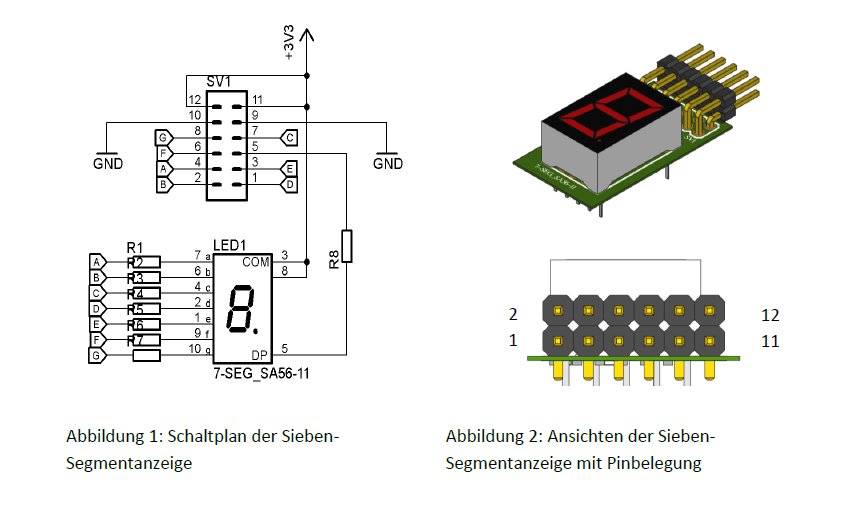
2.0V

A B C D E F G H

FPGA Pin

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

|  |  |
| --- | --- |
| **Logischem Pegel** | **LED Zustand** |
| **0** | **ON** |
| **1** | **OFF** |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **LED Char** | **Pin am SV1** | **Pin am PMOD** | **Name am PMOD** | **FPGA Pin** |
| **PMOD-A** | **A** | **4** | **2** | **JA2** | **AD9** |
| **B** | **2** | **1** | **JA1** | **AD11** |
| **C** | **7** | **10** | **JA10** | **AJ11** |
| **D** | **1** | **7** | **JA7** | **AD10** |
| **E** | **3** | **8** | **JA8** | **AE8** |
| **F** | **6** | **3** | **JA3** | **AM13** |
| **G** | **8** | **4** | **JA4** | **AM12** |
| **PMOD\_B** | **A** | **4** | **2** | **JB2** | **AC8** |
| **B** | **2** | **1** | **JB1** | **AE9** |
| **C** | **7** | **10** | **JB10** | **AA9** |
| **D** | **1** | **7** | **JB7** | **AF8** |
| **E** | **3** | **8** | **JB8** | **AB8** |
| **F** | **6** | **3** | **JB3** | **AB10** |
| **G** | **8** | **4** | **JB4** | **AC9** |

**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.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| **AUS** |  |  |  |  |  |  |  |
| **0** | **0(LED ON)** | **0** | **0** | **0** | **0** | **0** | **1** |
| **1** | **1(LED OFF)** | **0** | **0** | **1** | **1** | **1** | **1** |
| **2** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **3** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |
| **4** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |
| **5** | **0** | **1** | **0** | **0** | **1** | **0** | **0** |
| **6** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **7** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **8** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **9** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **A** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **B** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **C** | **0** | **1** | **1** | **0** | **0** | **0** | **1** |
| **D** | **1** | **0** | **0** | **0** | **0** | **1** | **0** |
| **E** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| **F** | **0** | **1** | **1** | **1** | **0** | **0** | **0** |

**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.**

parameter 0 0‘b0000001

**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?**

1. If else and Case structure
2. Multiplexer structure

......

if(!reset\_n) begin

out0<=LEER\_OUT;

out1<=LEER\_OUT;

end else begin

case(in)

ZERO: begin

out0<=ZERO\_OUT;

out1<=LEER\_OUT;

end

ONE: begin

out0<=ONE\_OUT;

out1<=LEER\_OUT;

end

......

default: begin

out0<=LEER\_OUT;

out1<=LEER\_OUT;

end

endcase

end

end