

Aufgabe 1

(10 + 20 Punkte)

Lernziele: Machen Sie sich mit der Entwicklungsumgebung vertraut. Sie besteht aus dem graphischen VHDL-Compiler/-Simulator *ModelSim* von Mentor Graphics und aus der VHDL-Synthesesoftware *ISE* von Xilinx. In dieser Aufgabe lernen Sie auch:

- Implementierung von VHDL-Komponenten bei vorgegebener Schnittstelle,
- synthesegerechte Verhaltensbeschreibung einfacher Schaltnetze mit Hilfe selektierter und bedingter Signalzuweisungen,
- synthesegerechte Verhaltensbeschreibung einfacher Zähler mit Hilfe von Prozeß- und IF-Anweisungen und
- Strukturbeschreibung im hierarchischen Entwurf durch Instanzierung von Komponenten

Als Vorbereitung zur Lösung dieser Aufgabe machen Sie sich mit den Kapiteln 3 und 4 aus dem *Spartan-3 FPGA Starter Kit Board User Guide* (ug130.pdf) vertraut.

Hinweise:

- Vorgegebene Schnittstellen oder bereitgestellte Komponenten dürfen weder geändert noch durch andere ersetzt werden.
- Der Takt in synchronen Systemen ist eine „unantastbare“ Größe und darf auf keinen Fall über Gattern mit anderen Signalen verknüpft werden. So etwas ist ein schlechter Programmierstil und Hinweis auf mangelnde Kenntnisse der Digitaltechnik. Der Takt darf nur direkt an die Clock-Eingänge von Flipflops angeschlossen werden.

Aufgabenstellung.

Es ist ein sog. BCD-Transcoder zu entwickeln, der eine 4-stellige BCD-Zahl auf einer 4-stelligen 7-Segmentanzeige darstellt. Der Transcoder ist in VHDL zu modellieren, zu simulieren (eine Testumgebung steht zur Verfügung) und für den FPGA-Baustein XC3S200-4FT256 auf dem Entwicklungsboard zu synthetisieren.

Die Schnittstelle des BCD-Transcoders:

```
ENTITY hex4x7seg IS
  GENERIC(RSTDEF: std_logic := '0');
  PORT(rst: IN std_logic; -- reset
        clk: IN std_logic; -- clock (rising edge)
        en: IN std_logic; -- enable, active high
        swrst: IN std_logic; -- software reset
        data: IN std_logic_vector(15 DOWNTO 0); -- data input
        dpin: IN std_logic_vector( 3 DOWNTO 0); -- 4 decimal point
        an: OUT std_logic_vector( 3 DOWNTO 0); -- 4 digit enable signals
        dp: OUT std_logic; -- 1 decimal point output
        seg: OUT std_logic_vector( 7 DOWNTO 1)); -- 7 FPGA connections to display
END hex4x7seg;
```

Im Bild 1 ist die Belegung einer 7-Segmentanzeige mit ihren einzelnen Segmenten dargestellt. Diese Segmente sind hier anders als in der Literatur, wo sie mit den Buchstaben *a* bis *g* markiert sind, mit den Ziffern *1* bis *7* durchnummeriert. Außerdem

ist in diesem Bild zu sehen, welche Segmente leuchten (markiert mit Schwarz) und welche ausgeschaltet bleiben (markiert mit Hellgrau), und zwar in der Abhängigkeit vom „erweiterten“ BCD-Wert ($0000_2..1111_2$), z.B. für den BCD-Wert 0101_2 sollen die Segmente (1, 2, 4, 5, 7) leuchten und die anderen (3, 6) dunkel bleiben.

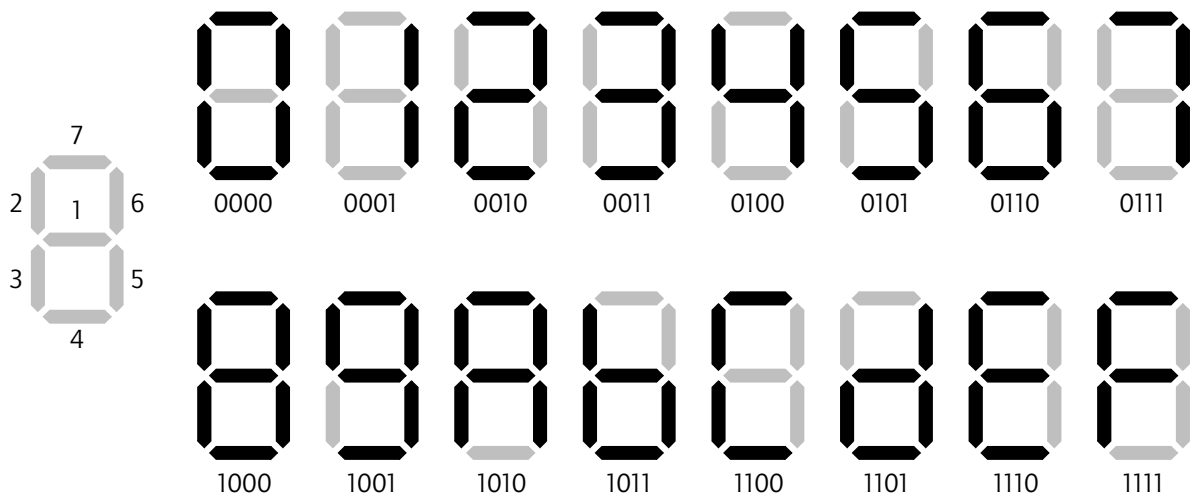


Bild 1: Darstellung von erweiterten BCD-Werten $0000_2..1111_2$ in der 7-Segmentanzeige.

Aufbau und Funktionsweise. Diese Schaltung, dessen Blockschaltbild in Bild 3 zu sehen ist, lässt sich aus sechs elementaren Komponenten zusammenbauen. Der Frequenzteiler ist ein synchroner Modulo- 2^{14} -Zähler, der die Eingangsfrequenz von 50 MHz auf ca. 3 kHz reduziert. In periodischen Zeitabständen (ca. $328 \mu s$) generiert er ein Enable-Signal, mit dem ein synchroner Modulo-4-Zähler über seinen Enable-Eingang gesteuert wird. Die in diesem Zähler generierte Zählsequenz wird in einem 1-aus-4-Decoder dazu verwendet, vier nicht überlappende Phasensignale ($an_3..an_0$) zur Ansteuerung jeweils einer Stelle in der 4-stelligen 7-Segmentanzeige zu erzeugen, so wie das im Bild 2 dargestellt ist.

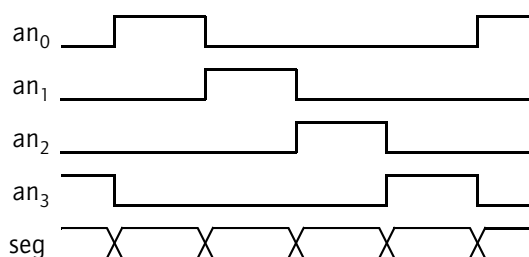


Bild 2: Impulsplan in „positiver“ Logik mit vier nicht überlappenden Phasensignalen.

Diese Zählsequenz dient gleichzeitig dazu, mit Hilfe eines 1-aus-4-Multiplexers die acht Schiebeschalter $sw_7..sw_0$ in zwei 4-Bit-Blöcken für den Decodiertvorgang auszuwählen. Die Einstellung der Schiebeschalter $sw_7..sw_4$ wird auf $DISP_3$ und $DISP_1$, und $sw_3..sw_0$ auf $DISP_2$ und $DISP_0$ angezeigt. Dieselbe Zählsequenz dient auch dazu, einen

zweiten 1-aus-4-Multiplexer anzusteuern, mit dessen Hilfe Dezimalpunkte in den Anzeigen $DISP_3..DISP_0$ aktiviert werden. Dazu sind die Taster btn_1 und btn_0 so zusammengeschaltet, daß beim Betätigen des Tasters btn_1 die Dezimalpunkte in den Anzeigen $DISP_3$ und $DISP_2$ leuchten, und beim Betätigen von btn_0 die Dezimalpunkte in den Anzeigen $DISP_1$ und $DISP_0$. Die letzte Komponente ist der 7-aus-4-Decoder, der entsprechend der Darstellung im Bild 1 eine Umcodierung eines 4-stelligen Bitmusters in einen 7-stelligen Bitmuster vornimmt.

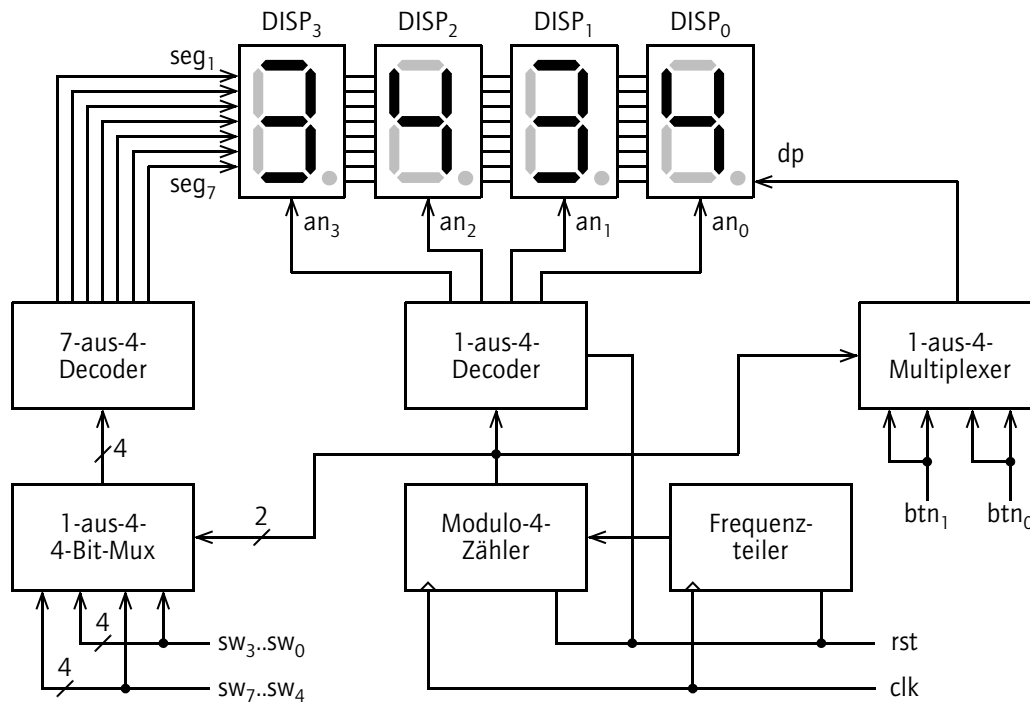


Bild 3: Blockschaltbild des BCD-Transcoders.

Resetphase. Während der Resetphase, d.h. wenn der Taster btn_3 betätigt ist, sollen alle LED-Segmente der 4-stelligen 7-Segmentanzeige ausgeschaltet werden. In diese Phase sollen auch der Modulo-4-Zähler und der Frequenzteiler zurückgesetzt werden.

Beispiel. Bei folgender Einstellung der Schiebeschalter $sw_7..sw_0 = (Lo, Lo, Hi, Hi, Lo, Hi, Lo, Lo)$ erscheinen in der 4-stelligen 7-Segmentanzeige die Ziffern (3, 4, 3, 4).