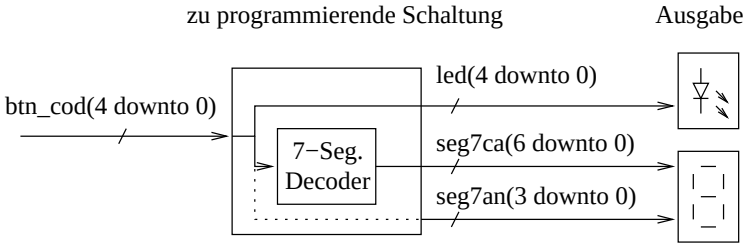
Aufgabe 2: Entwurf, Simulation und Test eines 7-Segmentdecoders

G. Kemnitz

08.11.2004

Die nachfolgende Abbildung zeigt den kompletten Versuchsaufbau:



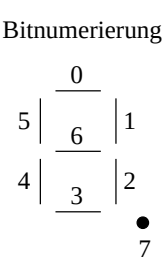
Die Daten von den 16 Tastern werden aus dem CoolRunner auf dem dio (Daughter IO-Board) in folgender Codierung an die Hauptplatine ausgegeben:

btn(15 downto 0)	btn_cod(4 downto 0)
0000000000000000	00000
xxxxxxxxxxxxxxx1	10000
xxxxxxxxxxxxxx10	10001
xxxxxxxxxxxxx100	10010
...	...
1000000000000000	11111

Das Signal btn_cod(4) ist Null, wenn keine Taste gedrückt ist und sonst Eins. Der Rest des Busses, die Signale btn_cod(3 downto 0), liefern die Nummer der gedrückten Taste als 4-Bit-Binärzahl.

Im zu programmierenden Logikschaltkreis werden die Signale btn_cod(4 downto 0) zum einen zu Kontrollzwecken direkt auf Leuchtdioden ausgegeben. Zum anderen sollen sie über einen zu entwerfenden Coder auf eine 7-Segmentanzeige ausgegeben und in folgender Weise angezeigt werden:

Eingabe	0000	0001	0010	0011	0100	0101	0110	0111
Ausgabe								
Eingabe	1000	1001	1010	1011	1100	1101	1110	1111
Ausgabe								



1 Projekt vorbereiten

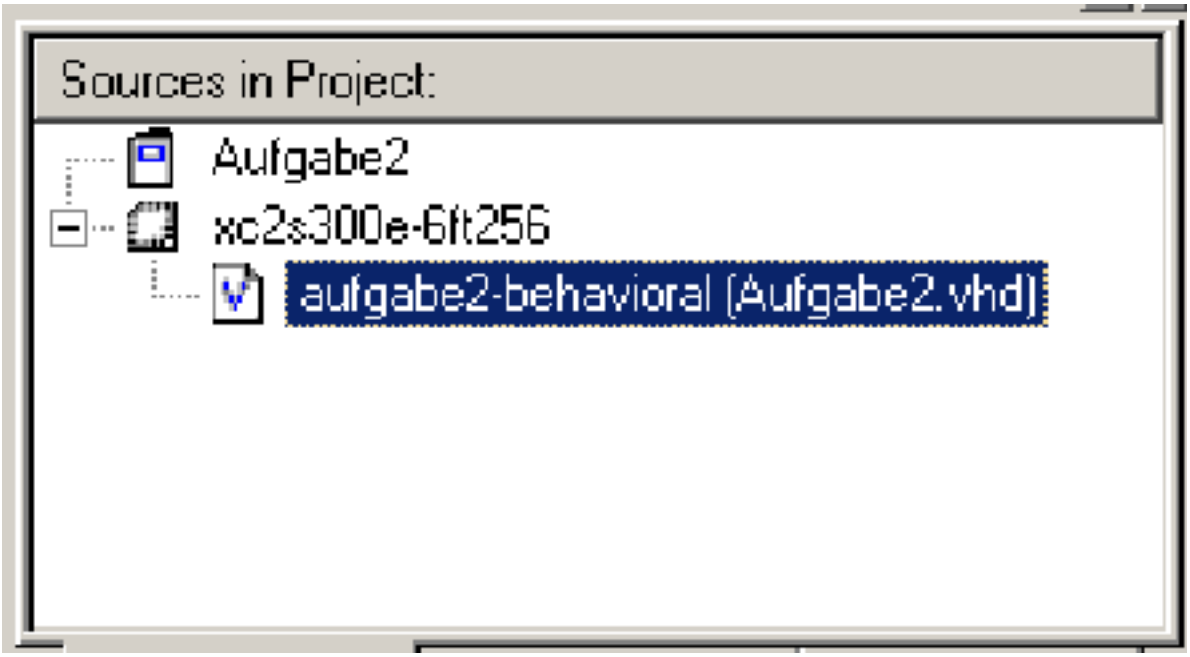
Legen Sie ein neues Verzeichnis

```
H:\TGP\Aufgabe2
```

an und kopieren Sie die Design-Files aus dem Netz in dieses Verzeichnis:

- Aufgabe2.npl: Projektdatei
- Aufgabe2.vhd: Entity für das Projekt und Gerüst für die Funktionsbeschreibung
- Praktikum.ucf: Constraints-Datei der Versuchsbaugruppe, identisch mit der ucf-Datei aus Aufgabe 1

Anschließend öffnen Sie das Projekt (”File”, ”Open Projekt”, "H:\TGP\Aufgabe2\", "Aufgabe2"). Ihr Source-Fenster wird, wenn alles geklappt hat, folgenden Baum angezeigt.



2 Quelldateien ansehen

Schauen Sie sich Aufgabe2.vhd an und füllen Sie folgende Wertetabelle für die hier beschriebene Funktion aus:

btn_cod(4 downto 0)	seg7ca(6 downto 0)	seg7an(3 downto 0)
00000 00001 00010 ... 11111		

3 Optionen einstellen

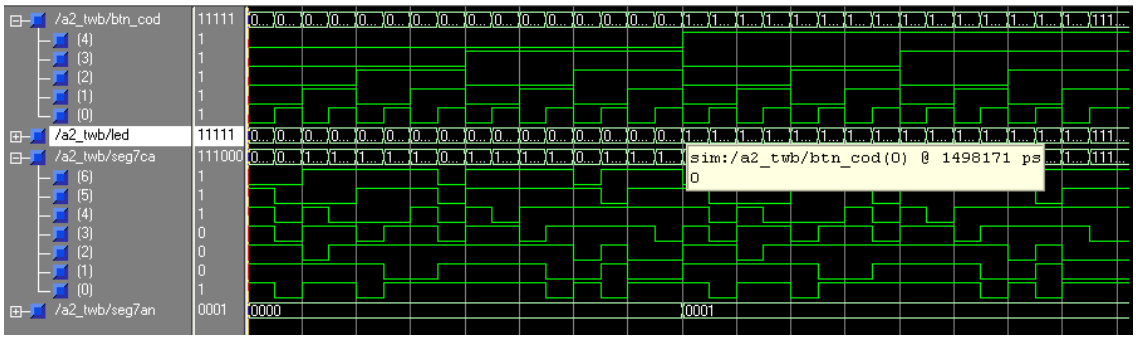
Damit die überflüssigen Einträge aus der ucf-Datei nicht auskommentiert werden müssen, ist wie im Tutorial unter "Implement Design, rechte Mouse-Taste, Properties, Translate Properties" das Flag "Allow Unmatched LOC Constraints" zu setzen. Gleichfalls ist unter "Generate Programming File, rechte Mouse-Taste, Properties, Startup Options" das Attribut "FPGA Start-Up Clock" in "JTAG Clock" umzuändern.

4 Übersetzen und Testen

- Doppel-Click auf "Generate Programming File" und hoffen, das keine Fehlermeldung auftritt.
- Doppel-Click auf Configure Device (iMPACT) und weiter wie im ersten Tutorial.
- Nach erfolgreichem Down-Load Kontrolle, dass die zuvor aufgestellte Wertetabelle mit dem tatsächlichen Schaltungsverhalten übereinstimmt.

5 Design ändern, simulieren und testen

Ändern Sie die Datei Aufgabe2.vhd so ab, dass richtige Ziffern ausgegeben werden. Legen Sie anschließend wie in Aufgabe 1 eine Testbench-Waveform an mit allen 32 Eingabevariationen. Simulieren Sie die Schaltung und kontrollieren Sie das Simulationsergeben ("Simulate Behavioral Model" reicht). Das Simulationsergebnis muss wie im nachfolgenden Bild aussehen:



Übersetzen Sie das Design und kontrollieren Sie, dass zu jeder Taste die richtige Ziffer angezeigt wird.

6 Zusatzaufgabe

Modifizieren sie die Beispielschaltung so, dass die Ziffern auf dem Kopf angezeigt werden.

7 Aufräumen

- Über Menuepunkt "Project, Cleanup Project Files" automatisch generierte Design-Files löschen.
- Netzteil zur Spannungsversorgung aus der Steckdose ziehen.
- Modelsim und Projektnavigator beenden.