Digitale Schaltungstechnik, WS 2022/2023

1. Übungsblatt — Abgabe: keine

1 Inbetriebnahme der FPGA-Platine

Für den praktischen Teil der Übungen werden Sie die *Nexys2*-Platine von Digilent benutzen. Die Platine enthält einen programmierbaren Logikbaustein – ein sogenanntes *FPGA (Field Programmable Gate Array)* – und Peripherie wie z. B. Taster, Schalter, LEDs und Siebensegmentanzeigen. Zur Kommunikation mit dem Rechner ist eine USB-Schnittstelle vorhanden, die auch zur Stromversorgung benutzt wird. Eine Übersicht aller Funktionen finden Sie im Handbuch der Platine (→ https://digilent.com/reference/_media/reference/programmable-logic/nexys-2/nexys2_rm.pdf).

a) Die Schaltungsinformation für das FPGA, das sogenannte *Bitfile*, wird mit einem speziellen Programm auf die Platine geladen. Das Transferprogramm namens *Adept 2* wird von Digilent zur Verfügung gestellt (→ https://digilent.com/reference/software/adept/start).

   Laden Sie für das von Ihnen bevorzugte Betriebssystem die beiden Programmbestandteile *System* bzw. *Runtime* und *Utilies* herunter und installieren Sie sie.

b) Bevor Sie die Platine mit dem beigelegten USB-Kabel mit Ihrem Rechner verbinden, setzen Sie den Jumper `POWER SELECT` (oben links) auf die Position `USB`, den Jumper `MODE` (oben rechts) auf `JTAG`, sowie den Schalter `POWER SWITCH` (oben links) auf „aus“, d. h. in die Position in Richtung der Mitte der Platine.

   Schließen Sie die Platine nun an Ihren Rechner an und schalten Sie sie mit dem `POWER SWITCH` ein. Die rote LED mit der Beschriftung `POWER` sollte daraufhin leuchten.

c) Der Hersteller Digilent bietet auf der Nexys2-Homepage ein Demo-Projekt an, mit dem Sie überprüfen können, ob die Platine funktioniert (→ https://digilent.com/reference/_media/nexys/nexys2/nexys_2_500k_bist.zip).

   Laden Sie das Archiv herunter und übertragen Sie mit Adept das darin enthaltene Bitfile `demowithmemcfg.bit` auf das FPGA: Unter Windows können Sie dafür die grafische Benutzeroberfläche verwenden, unter Linux führen Sie den folgenden Befehl aus:

   ```
   djtgcfg prog -d Nexys2 -i 0 -f demowithmemcfg.bit
   ```

   Wenn die Übertragung geklappt hat, leuchtet die gelbe LED mit der Beschriftung `DONE` auf. Wenn die Platine keinen Schaden hat, sollte nach kurzer Zeit auf der Siebensegmentanzeige der Schriftzug `PASS` erscheinen. Probieren Sie die verschiedenen Schiebeschalter und Knöpfe auf der Platine aus. Wenn Sie den `RESET`-Knopf (oben rechts) drücken, wird die Konfiguration wieder aus dem FPGA gelöscht und Sie müssen sie erneut übertragen.

2 Installation der Entwicklungsumgebung

Um eigene Bitfiles zu erstellen, wird die Entwicklungsumgebung *ISE* vom Hersteller Xilinx verwendet. In der (eingeschränkten, aber ausreichenden) Variante *WebPack* kann sie nach Registrierung für Linux und Windows kostenlos heruntergeladen werden:
(→ `https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html`).

a) Laden Sie das für Ihren Rechner passende Paket herunter und installieren Sie das ISE WebPack. Planen Sie dafür genug Zeit und Speicherplatz ein – der Download ist etwa 6 GB groß, das installierte Programmpaket belegt um die 17 GB!

b) Als erstes Beispiel sollen Sie das FPGA so konfigurieren, dass Sie mit einem Schiebeschalter eine LED ein- und ausschalten können. Starten Sie dazu also die ISE. Unter Linux geht das mit dem folgenden Befehl (der genaue Pfad zu der `.sh`-Datei kann abweichen):

```
source /opt/Xilinx/14.7/ISE_DS/settings32.sh; ise
```

Legen Sie ein neues Projekt an (*File → New Project...*) und geben Sie einen beliebigen Namen dafür ein. Es wird automatisch ein gleichlautendes Verzeichnis erstellt, in dem alle zugehörigen Dateien angelegt werden. Wählen Sie als „Top-level source type“ *Schematic* aus.

Auf der nächsten Seite müssen Sie die Informationen über das verwendete FPGA angeben:

Family:	Spartan3E
Device:	XC3S500E
Package:	FG320
Speedgrade:	-4

Fügen Sie dem neuen, leeren, Projekt ein Schematic hinzu (*Project → New Source...*). Es öffnet sich eine Bearbeitungsfläche in der rechten Hälfte des Programmfensters. Links erscheint eine Liste der verfügbaren Schaltsymbole (falls nicht, klicken Sie unten den Reiter *Symbols* an). Wählen Sie aus der Kategorie *Buffer* das Symbol *buf* aus und platzieren Sie eine Instanz im Schematic (*Add → Symbol*). Verbinden Sie die beiden Enden des Buffers jeweils mit einem I/O Marker (*Add → I/O Marker*). Durch Rechtsklick auf die neu erstellten I/O Marker können Sie unter *Object Properties → Category → Nets* dem Eingangs- und Ausgangsnetz des Buffers jeweils einen Namen ihrer Wahl geben (z. B. „Schalter“ und „LED“). Damit ist das Schematic fertig (es sollte ähnlich zu dem in Abbildung 1 gezeigten sein) und Sie sollten es abspeichern.

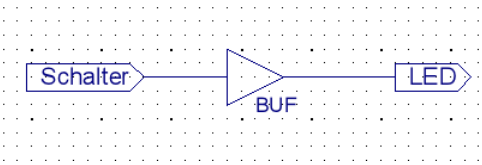

Abbildung 1: Schematic des Beispielprojekts

Bevor Sie das Schematic in ein Bitfile übersetzen lassen können, muss die ISE wissen, welchen *Pins* (Anschlussbeinchen) des FPGAs die Netznamen zugeordnet werden sollen. Dies wird durch ein *User Constraint File (UCF)* festgelegt, das Sie durch *Project → New Source... → Implementation Constraints File* anlegen (geben Sie der Datei die Endung `.ucf`). Die Datei sollte im Reiter *Design* in der hierarchischen Ansicht des Projekts unter dem Schematic erscheinen (siehe Abbildung 2). Durch Doppelklick auf den Eintrag öffnet sich in der rechten Programmhälfte ein Bearbeitungsfenster. Sie müssen für das Eingangs- und das Ausgangsnetz jeweils eine Zeile mit dem folgenden Format eintippen:

```
NET "<Name des Netzes>" LOC = "<Name des Pins>";
```

Als Netznamen müssen Sie die zuvor im Menü der I/O Marker gewählten Bezeichnungen verwenden. Die Namen der Pins können Sie grundsätzlich im Handbuch oder im Schaltplan (→ https://digilent.com/reference/_media/reference/programmable-logic/nexys-2/nexys2_sch.pdf) der Nexys2-Platine nachschlagen. In diesem Fall sehen Sie beispielsweise auf Seite 5 des Handbuchs, dass der Schiebeschalter „SW0“ an den Pin „G18“ und die LED „LD0“ an den Pin „J14“ angeschlossen ist. Der Inhalt der UCF-Datei könnte also wie folgt lauten:

```
NET "Schalter" LOC = "G18";
NET "LED"      LOC = "J14";
```

Um das Erstellen der UCF-Datei und das Herausfinden der richtigen Pins zu erleichtern, gibt es von Digilent eine Vorlage, die Sie für spätere Projekte verwenden können (→ https://digilent.com/reference/_media/reference/programmable-logic/nexys-2/nexys2_500general_ucf.zip).

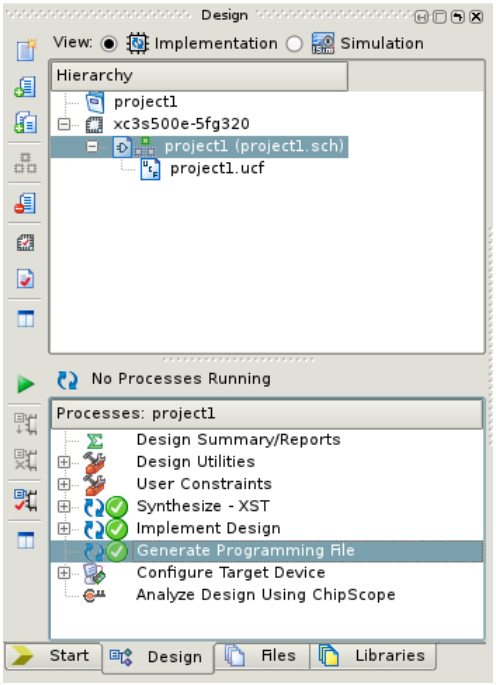

Abbildung 2: Hierarchische Ansicht des Projekts im linken Bereich des ISE-Programmfensters

Nachdem Sie das Constraint File gespeichert haben, können Sie das Bitfile erzeugen: Wählen Sie dazu in der hierarchischen Ansicht im Reiter *Design* Ihr Schematic aus. Mit einem Rechtsklick im unteren Bereich auf „Generate Programming File“, dann weiter auf „Process Properties“ und „Startup Options“, gelangen Sie zu den Einstellungen für den FPGA-Start. Wählen Sie hier unter „FPGA Start-Up Clock“ die „JTAG CLK“ aus, da wir den FPGA zunächst nur über JTAG programmieren. Schließen Sie das Fenster wieder und doppelklicken Sie auf „Generate Programming File“ (siehe Abbildung 2). Es werden verschiedene Schritte durchlaufen, die Sie mitverfolgen können. Nachdem der Vorgang erfolgreich abgeschlossen ist, können Sie mit Adept das Bitfile auf die Platine übertragen.

3 Taktteiler - Schematic

Am Pin `B8` des FPGA liegt ein 50 MHz Taktsignal an. Erzeugen Sie daraus mithilfe von Zählern ein Taktsignal mit niedriger Frequenz.

a) Für jedes Bit eines Zählers halbiert sich die Frequenz des Taktsignals. Geben Sie an, wieviel Bit der Zähler haben muss (bzw. welches Bit Sie abgreifen müssen), um ein Taktsignal zu erhalten, dass
   - eine Frequenz von ca. 100 kHz hat,
   - eine Frequenz von ca. 100 Hz hat,
   - eine Frequenz von ca. 3 Hz hat,
   - eine Periode von ca. 10 s hat.

b) Fügen Sie in ein neues Schematic zwei Instanzen des 16-Bit-Zählers *CB16CE* ein (Kategorie *Counter*). Indem Sie sie kaskadieren, erzeugen Sie effektiv einen 32-Bit-Zähler: Verbinden Sie den Eingang „CE“ *(Clock Enable)* des ersten Zählers mit einer konstanten „1“ *(vcc)* und seinen Ausgang „CEO“ *(Clock Enable Output)* direkt mit dem Eingang „CE“ des zweiten Zählers. Legen Sie an die Eingänge „CLR“ *(Clear)* der beiden Zähler eine „0“ *(gnd)* und an die Eingänge „C“ *(Clock)* das 50-MHz-Taktsignal an.

   An den Ausgang „Q“ (den Zählerstand) des zweiten Zählers verbinden Sie ein kurzes Stück Leitung. Da dieser Anschluss des Zählers aus 16 Bits besteht, erhalten Sie automatisch einen *Bus*, was Sie neben der dickeren Linie auch daran erkennen können, dass der Name der Leitung mit „(15:0)“ endet. Sie können auf einzelne Bits des Busses zugreifen, indem Sie ein normales Stück Leitung verlegen und ihm den Namen des Busses, gefolgt von der Nummer des Bits in Klammern, geben. Heißt der Bus bspw. „counter(15:0)“, bezeichnet „counter(0)“ das niederwertigste und „counter(15)“ das hochwertigste Bit.

   Die restlichen Anschlüsse („TC“ beider Zähler, das „Q“ des ersten Zählers und „CEO“ des zweiten) lassen Sie unverbunden.

c) Multiplexen Sie die 16 Ausgangsbits des zweiten Zählers auf eine LED, so dass Sie mithilfe von vier Schiebeschaltern die Blinkfrequenz einstellen können. Überprüfen Sie (für die langsamen Taktraten), ob Ihre Überlegungen vom ersten Aufgabenteil zutreffen. Verwenden Sie das Symbol *M16_1E* auf der Kategorie *Mux*. Verbinden Sie den enable Eingang „E“ mit *(vcc)*.