



## FPGA-Entwurf mit VHDL

Sommersemester 2011

### Serie 5

Abgabe: bis Fr. 27.05. um 10 Uhr im Schrein oder per Mail an lwi@informatik.uni-kiel.de.

#### Aufgabe 1

Das Datenblatt `xapp462` beschreibt die Nutzung von digitalen Clock-Managern in Xilinx-FPGAs. Laden Sie sich dieses Datenblatt auf der Homepage von Xilinx herunter.

- a) Was sind die Hauptaufgaben eines DCMs?
- b) Der Eingabetakt habe eine Frequenz von  $50\text{MHz}$ . Wie erzeugt man daraus mit Hilfe eines DCMs  $100\text{MHz}$ ,  $150\text{MHz}$ ,  $25\text{MHz}$  und  $33.333\dots\text{MHz}$ ?

8 Punkte

#### Aufgabe 2

Machen Sie sich mit dem Pinout-Datenblatt des Memec Spartan-3 FPGA-Boards vertraut (s. Lehrstuhl-homepage). Das Board besitzt unter anderem vier DIP-Schalter, zwei Push-Buttons, vier User-LEDs und eine 7-Segment-Anzeige. Suchen Sie sich den Teil der Schaltpläne für die Verschaltung des Eingabetakts, der DIP-Schalter, der Push-Buttons, der LEDs sowie der 7-Segment-Anzeige heraus.

- a) Mit welcher Spannung arbeiten die Anschlüsse jeweils? Welchen Basistakt liefert das Board?
- b) Was wird am FPGA-Eingang bei welcher Schalterstellung der DIP-Schalter und Push-Buttons gelesen? Was ist ein Pullup- bzw. Pulldown-Widerstand?
- c) Wie werden die LEDs und die 7-Segment-Anzeige angesteuert?

12 Punkte

#### Aufgabe 3 – Programmieraufgabe

a)

Schreiben Sie eine Top-Level-Entity zu den Programmieraufgaben aus der letzten Aufgabenserie, die als Interface nur noch Signale des FPGA-Boards besitzt.

Benutzen Sie als Untereinheiten bitte unverändert die Komponenten aus der Musterlösung (s. Lehrstuhl-homepage) und passen Sie die Signale in Ihrer Top-Level-Entity entsprechend den Informationen aus dem Datenblatt an.

Benutzen Sie auch eine DCM. Ihr Design soll mit dem gleichen Takt, wie der Basistakt, getaktet werden.

Außerdem soll der Status später über die vier LEDs angezeigt werden und das synchrone Reset soll über einen DIP-Schalter erfolgen, wenn dieser auf Schalterstellung “on” gestellt ist.

Testen Sie Ihr Design mit einer Testbench. Generieren Sie eine Konfigurationsdatei.

**Hinweise:**

- Das `enable`-Signal dürfen Sie vorerst konstant mit 1 beschalten.
- Die 7-Segment-Anzeige besitzt keine Ansteuerung für den Punkt. Lassen Sie diesen Ausgang bitte offen.

```
entity simple_calc_top is
port (
  raw_clk : in std_logic;
  sync_reset : in std_logic; -- DIP1
  push_A_in : in std_logic; -- Push1
  push_B_in : in std_logic; -- Push2
  leds_out : out std_logic_vector(3 downto 0);
  display_out : out std_logic_vector(6 downto 0) -- no dot available!
);
end simple_calc_top;
```

20 Punkte

**b)**

Eine UCF-Datei beschreibt Bedingungen für die Implementierung eines Designs auf einem FPGA, u.a. welcher Anschluss des FPGAs mit welchem Port-Signal der Top-Level-Entity verdrahtet werden soll. Eine solche Datei können Sie in ISE unter “New Source” → “Implementation Constraints File” erstellen. Benutzen Sie als Anhaltspunkt allerdings bitte die auf der Lehrstuhlhomepage zur Verfügung gestellte, welche Sie auch mit “Add Source” einfach einbinden können.

Korrigieren und vervollständigen Sie die UCF-Datei zu obigem Top-Level-Design zur Verwendung auf dem Memec Spartan3-FPGA-Board.

10 Punkte

## Zusatzaufgabe 4 – Programmieraufgabe

Haben Sie eine Idee warum Ihr Design aus Aufgabe 3 trotz aller Bemühungen in der Realität wahrscheinlich sehr schwer zu bedienen sein wird? Finden Sie eine Lösung und implementieren Sie diese mit in Ihre Top-Level-Einheit. Generieren Sie eine Konfigurationsdatei.

**Hinweis:** Denken Sie über die Taktfrequenz des Designs nach und nutzen Sie das `enable`-Signal geeignet.

10 Zusatzpunkte

Die Abgabe der Programmieraufgaben soll bitte folgendermaßen erfolgen:

1. **Verschieben Sie oder benennen Sie eine evtl. generierte Konfigurationsdatei zunächst um, damit sie beim Cleanup nicht gelöscht wird!**
2. Dann führen Sie in ISE “Project → Cleanup Project Files” aus.
3. Verpacken Sie den Projektordner in ein gepacktes Archiv (.zip, .tar.gz o.ä.).
4. Senden Sie das Archiv per Mail an `lwi@informatik.uni-kiel.de`