TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Technische Informatik

Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

Praktikum Programmierbare Schaltkreise WS 2009/10
Komplex: 7-Segment-Anzeige

Ansprechpartner
Martin Zabel
martin.zabel@tu-dresden.de
INF/1095

20. November 2009

Aufgabe 1

Entwickeln Sie einen Dekoder für die Umwandlung einer 4-Bit-Binärzahl in die 7-Segment-Darstellung einer Hexadezimalziffer. Die Position der Segmente a bis g können Sie der Beschreibung der Versuchsboards entnehmen. Beachten Sie, dass die Ansteuerung der Segmente low-aktiv erfolgt.

Für die Implementierung ist es zweckmäßig statt 7 Einzelsignalen (a bis g) einen 7-Bit-Signalvektor als Ausgangssignal vorzusehen.

a) Implementieren Sie den Dekoder mittels VHDL sowohl als `case`-Statement als auch als `with-select`-Statement. Überprüfen Sie die korrekte Funktion im Simulator. Implementieren Sie dazu eine Testbench, die alle möglichen Eingangskombinationen überprüft.

b) Erstellen Sie eine passende UCF-Datei für das Praktikumsboard. Für die Eingabe der Binärzahl sind die Umschalter SW1 bis SW4 des Erweiterungsboards zu nutzen. Steuern Sie nur das rechte Segment der 4-stelligen Anzeige an, indem Sie die Anodenauswahl AN1 bis AN4 entsprechend belegen (low-aktiv).

Synthetisieren Sie ihr VHDL-Modell und beheben Sie aufgetretene Fehler und Warnungen. Überprüfen Sie die korrekte Funktion des Dekoders auf dem Praktikumsboard.

Vergleichen Sie die beiden Implementierungen, indem Sie im Praktikumsprotokoll die benötigten Ressourcen aus dem *Fit Report* auswerten.

Aufgabe 2

Entwickeln Sie einen Dekoder für die Umwandlung einer 3-Bit-Binärzahl (vorzeichenlos) in einen *n*-bittigen One-Hot-Code.

a) Wie viele Stellen hat der One-Hot-Code, wenn genau alle Eingangsbelegungen abgedeckt werden sollen.

b) Implementieren Sie den Dekoder mittels VHDL. Überprüfen Sie die korrekte Funktion im Simulator. Implementieren Sie dazu eine Testbench, die alle möglichen Eingangskombinationen überprüft.

c) Erstellen Sie eine passende UCF-Datei für das Praktikumsboard. Für die Eingabe der Binärzahl sind die Umschalter SW1 bis SW3 des Erweiterungsboards zu nutzen. Die Ausgabe erfolgt auf den LEDs beginnend bei LD1 für die niedrigste Binärzahl.

Synthetisieren Sie ihr VHDL-Modell und beheben Sie aufgetretene Fehler und Warnungen. Überprüfen Sie die korrekte Funktion des Dekoders auf dem Praktikumsboard.

Werten Sie im Praktikumsprotokoll die benötigten Ressourcen aus dem *Fit Report* aus.

Aufgabe 3

Entwickeln Sie einen 21-stelligen Taktteiler mit synchronem Reset und Clock-Enable. Die obersten 8 Bits sind auf den LEDs des Erweiterungsboards darzustellen, wobei das MSB auf LD8 abgebildet werden soll. Getaktet wird das Design durch den Boardtakt von 1,842 MHz.

a) Mit welcher Frequenz blinken die LEDs LD1 bis LD8?

b) Implementieren Sie den Taktteiler in VHDL und überprüfen Sie die Funktion mittels Simulation.

c) Erstellen Sie eine passende UCF-Datei für das Praktikumsboard. Als Reset ist der Umschalter SW1 zu verwenden. Gezählt werden soll, wenn SW2 = '1' ist.

Synthetisieren Sie ihr VHDL-Modell und beheben Sie aufgetretene Fehler und Warnungen. Überprüfen Sie die korrekte Funktion des Dekoders auf dem Praktikumsboard.

Werten Sie im Praktikumsprotokoll die benötigten Ressourcen aus dem *Fit Report* aus.

d) Ermitteln Sie die maximale Taktfrequenz für den Zähler aus dem *Timing Report.*

Aufgabe 4

Entwickeln Sie eine Multiplex-Ansteuerung für die 4-stellige 7-Segment-Anzeige des Erweiterungsboards.

- Nutzen Sie die höchstwertigsten 2 Bit eines *n*-stelligen Taktteilers für die Auswahl des Segments. Leiten Sie daraus die One-Hot-Kodierung für die Ansteuerung der Anoden ab. Dimensionieren Sie den Teiler so, dass die Umschaltfrequenz über 25 Hz und unter 200 kHz liegt.
- Stellen Sie zunächst die Hexadezimalzahl $12EF_{16}$ dar.

Getaktet wird das Design durch den Boardtakt von 1,842 MHz. Ein Reset ist nicht vorzusehen. Stattdessen ist der Zähler mit einem geeigneten Power-Up-Wert zu initialisieren.

a) Wie viele Bit werden für den Zähler benötigt?

b) Implementieren Sie die Ansteuerung mittels VHDL. Überprüfen Sie die korrekte Funktion mittels Simulation.

c) Erstellen Sie eine passende UCF-Datei für das Praktikumsboard. Synthetisieren Sie ihr VHDL-Modell und beheben Sie aufgetretene Fehler und Warnungen. Überprüfen Sie die korrekte Funktion auf dem Praktikumsboard.

Ermitteln Sie die benötigten Ressourcen als auch die maximale Taktfrequenz aus den *Reports*.

Aufgabe 5

Zwecks Wiederverwendung in anderen Projekten ist nun die Multiplex-Ansteuerung aus Aufgabe 4 als ein separates VHDL-Modul mit Namen `seg4x7` zu realisieren. Für die Ein- und Ausgänge des Moduls sind folgende Namen zu verwenden:

Signal	Richtung	Typ	Bedeutung
clk	Eingang	`std_logic`	Board-Takt 1.842 MHz
value	Eingang	`std_logic_vector(15 downto 0)`	Binärwert
seg7_sg	Ausgang	`std_logic_vector(6 downto 0)`	Segment/Kathoden, low-aktiv
seg7_an	Ausgang	`std_logic_vector(3 downto 0)`	Anoden, low-aktiv

a) Erstellen Sie das VHDL-Modul `seg4x7` auf Basis des Quellcodes von Aufgabe 4.

b) Erstellen Sie ein weiteres Top-Level-Modul, welches das Modul `seg4x7` instanziiert. Führen Sie folgende Tests mit den angegebenen Eingaben auf dem Praktikumsboard durch:

i) Wert $12EF_{16}$.

ii) Einem 16-Bit-Wert zusammengesetzt aus zwei Bytes. Das niederwertiges Byte wird über die Umschalter eingestellt. Das höherwertige Byte entspricht der bitweisen Negation des niederwertigen Bytes.

iii) Die höchstwertigsten 16 Bit eines 32-Bit-Taktteilers. Der Teiler soll über einen Reset (SW1) und ein Clock-Enable (SW2) verfügen.