



# FPGA-Entwurf mit VHDL

Sommersemester 2011

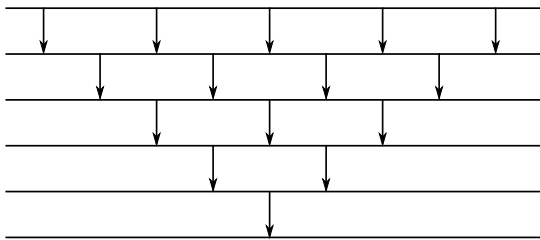
## Serie 8

Abgabe: bis Fr. 24.06. um 10 Uhr im Schrein oder per Mail an lwi@informatik.uni-kiel.de.

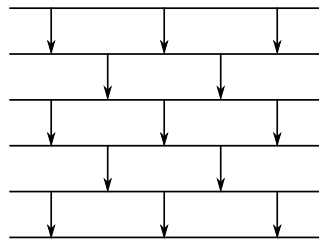
### Aufgabe 1

Welche der folgenden Vergleichernetze sortieren? Begründen bzw. widerlegen Sie. Korrigieren Sie nicht sortierende Vergleichernetze auf minimalem Wege.

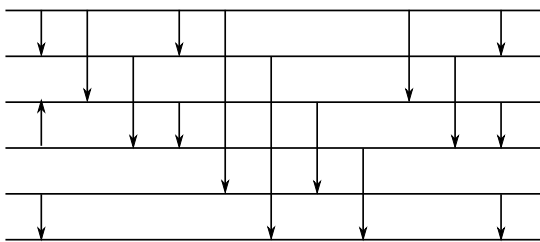
a)



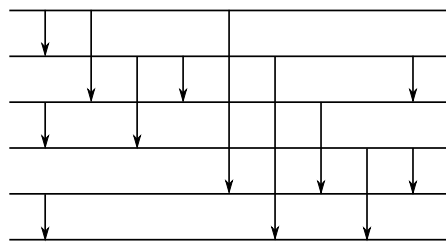
b)



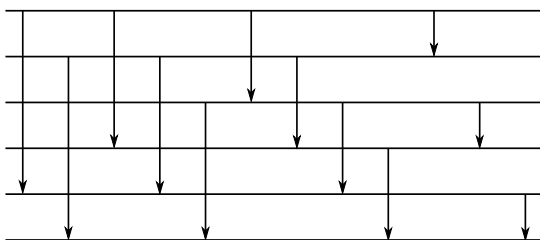
c)



d)



e)

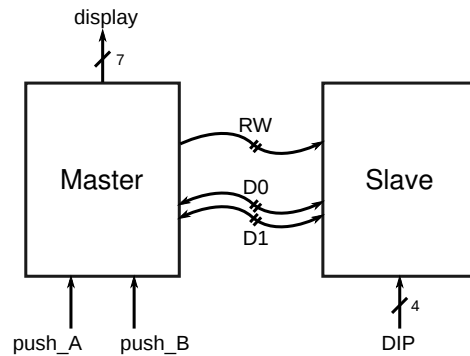


10 Punkte

## Aufgabe 2 – Programmieraufgabe

Implementieren Sie ein simples Bussystem mit nur zwei Teilnehmern, die folgende Funktion erfüllen.

Der Busmaster ist an die 7-Segment-Anzeige des Spartan3-FPGA-Boards angeschlossen. Außerdem kann er die beiden Push-Buttons auslesen. Die vier DIP-Schalter sind am Slave angeschlossen. Durch Drücken der Push-Buttons soll nun der Master über den Bus den Zustand der DIP-Schalter vom Slave anfordern. Nach Drücken des Buttons A wird der Wert direkt angefordert. Drücken des Buttons B fordert dagegen den gespiegelten Wert an, d.h. folgende Bits sind vertauscht:  $3 \leftrightarrow 0$  und  $2 \leftrightarrow 1$ . Nach Erhalt der Daten soll der entsprechende Wert hexadezimal auf der 7-Segment-Anzeige etwa 1 Sekunde lang angezeigt werden.



Auf der Homepage steht Ihnen ein Framework zur Verfügung, das lediglich fertiggestellt werden muss. Es stehen Ihnen drei Busleitungen zur Verfügung: Eine unidirektionale Leitung RW, die anzeigen soll, ob der Master auf den Bus schreibt ( $RW = 0$ ) oder liest ( $RW = 1$ ) und zwei Datenleitungen D0 und D1 die bidirektional betrieben werden sollen.

Der Bus wird physikalisch auf dem FPGA-Board realisiert indem entsprechende User-IOs miteinander verbunden werden. Aus diesem Grund haben Sie keine Information über die Verzögerung. Sie dürfen aber davon ausgehen, dass alle Busleitungen die gleiche Verzögerung besitzen. Außerdem brauchen Sie sich keine Gedanken über Übertragungsfehler machen.

Um zu vermeiden, dass zwei Pins am FPGA gegeneinander treiben, ändern Sie bitte die entsprechende Beschaltung im Framework nicht! Auch welche IO-Pins verwendet werden ist anhand der beigegeführten UCF-Datei festgelegt. Ändern Sie bitte auch diese nicht!

Um weiter zu verdeutlichen, dass es sich bei den Busteilnehmern um getrennte Komponenten handelt, sollen Sie sich nicht einmal den Systemtakt teilen. Hierzu gibt der Master seinen Systemtakt über einen IO-Pin nach außen. Dieser wird mit einem weiteren IO-Pin verbunden, welcher dann als Takteingang für den Slave dient. Diese Funktion ist bereits im Framework implementiert.

Für eine serielle Datenübertragung sind bereits die bekannten Einheiten `Serializer` und `Deserializer` im Framework eingebunden, ebenso wie die Komponente `conv_7seg` zur Darstellung einer 4-bit-Zahl auf dem 7-Segment-Display.

- Implementieren Sie den Bus-Slave.
- Implementieren Sie den Bus-Master.
- Testen Sie das Gesamtsystem mit Hilfe der im Framework vorhandenen Testbench und generieren Sie eine Konfigurationsdatei.

40 Punkte

## Zusatzaufgabe 3

Leider treten in der Realität des Öfteren doch Übertragungsfehler auf.

Erläutern Sie: Was ist ein Wiederholungscode der Länge  $n$  und welche Eigenschaften müssen für  $n$  gelten, damit ein Übertragungsfehler sicher korrigiert werden kann?

Wie und an welchen Stellen müssten Sie Ihr Design aus Aufgabe 2 anpassen um einen Wiederholungscode generischer Länge für die Übertragung der Busdaten zu implementieren?

**Hinweis:** Sie müssen die Implementierung nicht durchführen. Eine Beschreibung der auftretenden Probleme und ein grober Lösungsansatz in Worten genügt!

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](mailto:lwi@informatik.uni-kiel.de)