



## FPGA-Entwurf mit VHDL

Sommersemester 2011

### Serie 9

Abgabe: bis Fr. 01.07. um 10 Uhr im Schrein oder per Mail an lwi@informatik.uni-kiel.de.

#### Aufgabe 1

Zeichnen Sie die Sortiernetze für Odd-Even-Transposition-Sort und Bitonic-Sort für 8 Elemente auf. Geben Sie jeweils an, wieviele Vergleiche durchgeführt werden müssen und in wievielen Zeitschritten dies am schnellsten möglich ist.

Wieviele Vergleiche und Zeitschritte wären bei den jeweiligen Sortierverfahren nötig, wenn 32 Elemente sortiert werden sollten?

10 Punkte

#### Aufgabe 2 – Programmieraufgabe

```
use work.data_type.all;

entity oets8 is
port (
  clk : in std_logic;
  DI  : in data(0 to 7);
  DO  : out data(0 to 7)
);
end oets;
```

a) Implementieren Sie ein Package `data_type.vhd` in welchem der Datentyp `data` definiert wird. Er soll ein Feld frei wählbarer Größe des Typs `std_logic_vector` der Länge `DATA_WIDTH` darstellen. Die Konstante `DATA_WIDTH` müssen Sie also ebenfalls in diesem Package deklarieren.

Ferner wird im Folgenden ein Clock-Constraint für obiges Signal `clk` benötigt. Erstellen Sie ein solches Constraint in einer `.ucf`-Datei, welches angibt, dass der Takt `100MHz` beträgt. Geben Sie **keinen** IO-Standard oder Location-Constraint an.

5 Punkte

b) Erstellen Sie das Sortiernetz für Odd-Even-Transposition-Sort für 8 Elemente der Länge 4 in VHDL. Nach jedem Schritt sollen dabei die Zwischenwerte in Register gespeichert werden. Ihr Design soll also eine Pipeline mit der Tiefe gleich der Anzahl der Zeitschritte, die für das Sortieren benötigt werden, darstellen.

- Testen Sie Ihr Design.
- Synthetisieren und implementieren Sie Ihr Design für einen Spartan xc3s5000-4fg676 mit der in a) erstellten .ucf-Datei.

15 Punkte

c) Erstellen Sie analog zu b) das Sortiernetz für Bitonic-Sort, ebenfalls für 8 Elemente der Länge 4, in VHDL. Testen, synthetisieren und implementieren Sie Ihr Design ebenfalls analog zu b).

15 Punkte

d) Vergleichen und interpretieren Sie die Syntheseeergebnisse, sowie die Ergebnisse nach dem Place-and-Route.

5 Punkte

### Zusatzaufgabe 3 – Programmieraufgabe

```
use work.data_type.all;

entity bsort is
generic (
  ELEMENTS : positive := 32 -- power of two
);
port (
  clk : in std_logic;
  DI : in data(0 to ELEMENTS-1);
  DO : out data(0 to ELEMENTS-1)
);
end oets;
```

Lösen Sie Aufgabe 2, allerdings sei nun die Anzahl der Elemente bei 2b) und 2c) generisch. Sie dürfen annehmen, dass es sich bei Bitonic-Sort hier um eine Zweierpotenz handelt. Bei Odd-Even-Transposition-Sort sei die Anzahl gerade. Für den Vergleich beider Verfahren synthetisieren und implementieren Sie bitte für Sortiernetze mit je 32 Elementen der Länge 4.

**Hinweis:** Aufgabe 3 ersetzt Aufgabe 2. Natürlich werden die Punkte aus Aufgabe 2 aber hier entsprechend angerechnet.

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)