Rechnerarchitektur 1
DI Alwin Zulehner

14. April 2016
Johannes Kepler Universität

Übung 3

Abgabe bis **Donnerstag, 21. April 10:00** via EPIIC: `http://ep.iic.jku.at`.

1. Huffman-Code (8 Punkte)

Im Folgenden sollen Zeichen effizient kodiert werden, um eine möglichst hohe Kompressionsrate zu erzielen. Verwendet wird dazu der Huffman-Code. Eine (repräsentative) Stichprobe der zu kodierenden Daten dient der Ableitung des Codes.

(a) Als Stichprobe dient der folgende Satz:

UM REKURSION ZU VERSTEHEN MUSS MAN ZUNAECHST REKURSION VERSTEHEN

Generiere für diesen Text den Huffman-Code, wobei Leerzeichen ignoriert werden. Beachte folgende Konventionen:

1. Sortiere die Zeichen von links nach rechts absteigend nach ihren Häufigkeiten und bei gleicher Häufigkeit in lexikographischer Reihenfolge.
2. Beim Zusammenfassen zweier "Knoten" sollen immer die am weitesten links stehenden Knoten gewählt werden, die für eine Verschmelzung in Frage kommen.
3. Bei der Zuweisung der Codebits bekommt immer der linke Nachfolger eine '1', der rechte eine '0' zugewiesen.
4. Ein neuer Knoten wird immer oberhalb seines linken Kindes angelegt.

Gib in deiner Lösung sowohl den Code für jeden einzelnen Buchstaben als auch den kodierten Text an. Trenne dabei zur Übersichtlichkeit die einzelnen kodierten Worte durch Leerzeichen.

(b) Dekodiere mit Hilfe des in Teilaufgabe a) generierten Codes folgende Nachricht: 1111011000110100001000011001010010111100000000

(c) Gib die Kompressionsrate ((Länge der unkodierten Nachricht)/(Länge der kodierten Nachricht)) für die Nachricht aus b) an, wenn ein Zeichen im unkodierten Fall durch 8 Bit repräsentiert wird.

2. Zahlendarstellung (2 Punkte)

Stelle die positive Hexadezimalzahl 0xDEADBEEF als Oktalzahl dar, ohne vorher ihren Dezimalwert zu berechnen. Gib Zwischenschritte und nötige Überlegungen an.

3. Zahlencodierung (6 Punkte)

Die folgende Tabelle enthält Dezimalzahlen und 5 Bit Binärzahlen in Betrag/Vorzeichen Darstellung, 1er Komplement Darstellung, 2er Komplement Darstellung und Offset Darstellung. Fülle die Tabelle entsprechend aus, sodass jede Zeile die gegebene Zahl in jeder Darstellung enthält.

Dezimal	Betrag/Vorzeichen	1er Komplement	2er Komplement	Offset
-13_{10}				
9_{10}				
	00101_2			
		01100_2		
			11111_2	
				10110_2

4. Addieren mit Binärzahlen (8 Punkte)

Es soll ein Addierer für Binärzahlen in Betrag/Vorzeichen Darstellung realisiert werden der $S = A + B$ berechnet. Dazu stehen ein Addierer und ein Subtrahierer für positive Operanden als Bauteile zur Verfügung. Verwende so wenig 2:1-Multiplexer mit beliebiger Breite wie möglich um das skizzierte Design zu vervollständigen. Die Eingänge der Multiplexer dürfen negiert werden. Andere Bauteile sind nicht erlaubt.