

# Rechnerstrukturen

Wintersemester 2002/2003

## 7. Übungsblatt

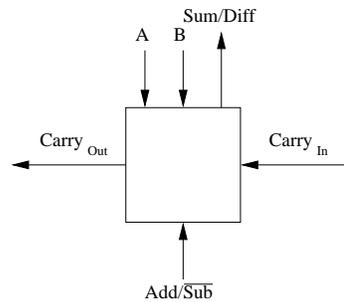
Abgabetermin: 07.01.03 in der Übung

### Aufgabe 1:

20 Punkte

In der sogenannten *sign and magnitude* Zahlendarstellung der Breite  $n$ -Bit wird das am weitesten links stehende Bit zur Repräsentation des Vorzeichens verwendet (0 = positiv, 1 = negativ), während die verbleibenden  $n - 1$  Bits den Absolutbetrag des Wertes enthalten. Folglich gibt es zwei Darstellungen des Wertes 0, nämlich mit positivem und negativem Vorzeichen. Die Addition zweier Werte mit gleichem Vorzeichen gestaltet sich vergleichsweise einfach. Die beiden vorzeichenlosen Beträge werden addiert und das Ergebnis erhält das gleiche Vorzeichen wie die beiden Operanden.

- Beschreiben Sie, wie die Addition zweier Zahlen mit unterschiedlichem Vorzeichen ablaufen muss.
- Entwerfen Sie einen 4-Bit-sign-and-magnitude-Addierer (d.h. ein Vorzeichen- und drei Daten-Bits). Verwenden Sie dabei einen kaskadierbaren 1-Bit-Addierer/ Subtrahierer-Baustein mit folgendem Anschlußschema:



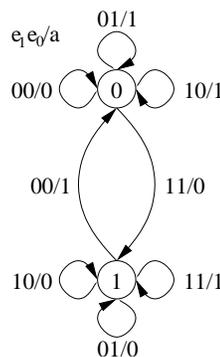
Ferner können Sie weitere aus der Vorlesung bekannte Bausteine verwenden.

- Wie erkennt man an den Ausgängen einen Überlauf? In welchen Fällen kann potentiell ein Überlauf entstehen?

### Aufgabe 2:

22 Punkte

Gegeben ist folgendes Zustandsdiagramm eines Mealy-Automaten mit den Eingängen  $e_1$  und  $e_0$  sowie dem Ausgang  $a$ :



- Entwickeln Sie ein äquivalentes Zustandsdiagramm für einen Moore-Automaten und vergleichen Sie die Anzahl der benötigten Zustände.
- Entwerfen Sie für die beiden Automaten jeweils eine Schaltung auf der Grundlage von JK-Flip-Flops.