

Überlauf-Erkennung beim Addierer

Wie bereits aus der Vorlesung bekannt, können Über- und Unterläufe bei der Addition anhand eines zusätzlichen Sicherungsbits erkannt werden, das eine Kopie des höchst-signifikanten Bits darstellt. Man benutzt nun für eine n -Bit-Addition einen $n+1$ -Bit Addierer, wobei die Sicherungsstelle ganz normal mitaddiert wird. Das Ergebnis der Addition ist also die Summe $s_{n-1}, s_{n-2}, \dots, s_1, s_0$, das ausgehende Carry-Bit (= Carry-Flag) c_n , sowie ein künstliches Summenbit s_n . Das künstliche Carry-Bit c_{n+1} hat keine Bedeutung und wird daher nicht verwendet. Ein Überlauf (Ergebnis ist größer als die größte darstellbare Zahl) ist nun aufgetreten, wenn s_{n-1} gleich 1 und s_n gleich 0 ist. Wenn s_n gleich 1 und s_{n-1} gleich 0 ist, hat ein Unterlauf (Ergebnis ist kleiner als die kleinste darstellbare Zahl) stattgefunden. Wenn s_{n-1} gleich s_n ist, ist weder Überlauf noch Unterlauf aufgetreten, also ist die Addition fehlerfrei verlaufen.

Das Schaltnetz auf der folgenden Folie zeigt die Ergänzung unseres Addierers, mit der wir Über- und Unterläufe erkennen und als Flags (ovf (overflow) und unf (underflow)) an die Flag-Einheit übermitteln können.

Berechnung von carry, underflow, overflow aus der Sicherungsstelle:

