

Aufgabe 1: Carry-Lookahead-Addierer

Bearbeiten Sie die folgenden Aufgaben den Vorlesungsfolien entsprechend:

1. Lösen Sie die rekursive Berechnung vom Übertrag \bar{u}_4 in Und- und Oder-Verknüpfungen der Generate-Carry g_i und Propagate-Carry Bits p_i mit $i \leq 3$ auf.
2. Berechnen Sie nun g_i und p_i mit $i \leq 3$ für die konkreten Werte $a_{3..0} = 1010$ und $b_{3..0} = 0110$.
3. Berechnen Sie abschließend die ersten vier Ziffern s_i mit $i \leq 3$ der Summe von a und b aus 2 nach der Carry-Lookahead-Addierer-Methode.

Aufgabe 2: Integer-Vector Rechner

Implementieren Sie eine Funktion in NASM, die zwei Integer-Vektoren miteinander verrechnet. Nutzen Sie dazu die Vektorbefehle Ihrer SSE-Unit um die Arrays aus dem Speicher direkt in jeweils ein xmm-Register zu laden und dann die jwg. Operationen auszuführen.

Die Arrays sind jeweils 128-Bit groß, das wird bei intel als „double quad-word“ bezeichnet. Die einzelnen Operationen sollen dann auf sogenannten „packed words“ arbeiten, sprich auf acht 16-Bit Zahlen gleichzeitig.

```
void vcalcw(uint16_t a[8], uint16_t b[8],  
            uint16_t c[8], char op);
```

Parameter:

- a – Array mit den ersten Operanden
- b – Array mit den zweiten Operanden
- c – Array für die Ergebnisse
- op – Operation die ausgeführt werden soll, wobei $op \in \{+, -, *, =\}$.