

CT Praktikum 8

Kontrollstrukturen - Taschenrechner

1 Einleitung

In diesem Praktikum implementieren wir einen einfachen Taschenrechner auf dem CT Board. Die Operanden werden dabei über die Schiebeschalter eingegeben und die auszuführende Funktion wird mit dem Hex-Codier-Schalter eingestellt.

2 Lernziele

- Sie können eine Fallunterscheidung (auch Mehrfachauswahl) mit einer Sprungtabelle (Jump Table) in Assembler realisieren.

Hinweis: Ein C-Compiler wendet bei der Übersetzung von switch-case statements nach Assembler in der Regel genau diese Technik an.

- Sie vertiefen Ihre Kenntnisse in der Anwendung von arithmetischen und logischen Operationen.

3 Aufgabe

Schreiben Sie ein Programm, welches auf den an den Schiebeschaltern eingestellten Werten verschiedene Operationen durchführt und das Resultat an den LEDs anzeigt. Verwenden Sie dazu den vorgegebenen Programmrahmen *calc.a86*.

Das Programm soll die folgenden Anforderungen erfüllen:

- Der erste Operand (op1) soll über die 8 Schiebeschalter an Port 705h eingelesen werden.
- Der zweite Operand (op2) soll über die 8 Schiebeschalter an Port 704h eingelesen werden.
- Die gewünschte Operation soll über den Hex-Codier-Schalter gemäss der Tabelle 1 eingestellt werden.
- Die Stellung des Hex-Codier-Schalters soll auf Port 708h angezeigt werden.
- Falls in Tabelle 1 nicht anders beschrieben, soll ein 8-bit breites Resultat über die 8 LEDs an Port 700h angezeigt werden. Allfällige Über- oder Unterläufe werden dabei nicht angezeigt. Die LEDs an Port 701h sollen dunkel sein.
- Bei der Darstellung einer Eins soll die entsprechende LED leuchten.
- Das Programm soll eine Sprungtabelle verwenden.

Hex-Codier-Schalter	LED-Anzeige	Hex-Codier-Schalter	LED-Anzeige
0h	alle 16 dunkel	8h	!op1 → not
1h	op1 + op2	9h	!(op1 & op2) → nand
2h	op1 – op2	Ah	!(op1 # op2) → nor
3h	op1 * op2 (vorzeichenlos) 16-bit Resultat auf 701h und 700h anzeigen	Bh	!(op1 \$ op2) → xnor
4h	op1 / op2 (vorzeichenlos) Ergebnis: LED 700h Rest: LED 701h	Ch	alle 16 leuchten
5h	op1 & op2 → and	Dh	alle 16 leuchten
6h	op1 # op2 → or	Eh	alle 16 leuchten
7h	op1 \$ op2 → xor	Fh	alle 16 leuchten

Tabelle 1: Übersicht der Operationen

4 Zusatzaufgabe

Was passiert, wenn bei der Division (Hex-Codier-Schalter auf 4h) op2 = 0 gewählt wird?
Ergänzen Sie Ihr Programm, damit in diesem Fall kein Programmabsturz erfolgt.