

## Rechnerstrukturen

Wintersemester 2002/2003

### 10. Übungsblatt

Abgabetermin: 28.01.03 in der Übung

#### Aufgabe 1:

12 Punkte

Gegeben sei wieder die Rekurrenzrelation der letzten Aufgabe:

$$\begin{aligned}x_0 &= 0 \\x_1 &= 1 \\x_i &= 4x_{i-2} + (-1)^i * x_{i-1} \text{ für } i \geq 2\end{aligned}$$

Entwerfen Sie diesmal ein Makroprogramm zur Berechnung von  $x_n$  für die aus der Vorlesung bekannte einfache 16 Bit-CPU. Gehen Sie hierbei davon aus, daß  $n$  zu Beginn der Berechnung an Adresse 42 im Speicher steht. Das Ergebnis soll an Adresse 43 abgelegt werden. Dokumentieren Sie Ihren Makrocode, so daß der zugrundeliegende Algorithmus klar zu erkennen ist. Ein eventuell auftretender Überlauf kann ignoriert werden.

#### Aufgabe 2:

18 Punkte

In der gegenwärtigen Realisierung der einfachen 16 Bit-CPU aus der Vorlesung erzeugt die ALU die beiden Kontrollbits N und Z. Diese Bits können derzeit nur von dem in der Ausführung befindlichen Mikrobefehl ausgewertet werden. Reale CPUs erlauben demgegenüber häufig einen Zugriff auf die Kontrollbits auf Makrobefehlsebene. Hierzu soll ein – auf Makrobefehlsebene sichtbares – *Programmstatusregister* (PSR) in die CPU-Architektur aufgenommen werden.

1. Erweitern Sie die einfache 16 Bit-CPU um ein PSR und die benötigte Steuerlogik. Gehen Sie davon aus, daß die ALU folgende Kontrollbits erzeugt:

**N** negativ  
**Z** zero  
**C** carry  
**V** overflow

2. Durch die Einführung des PSR wird es möglich, Sprungbefehle nicht nur auf den Inhalt des Akkumulators, sondern auch auf die im PSR gespeicherten Kontrollbits zu beziehen. Entwerfen Sie exemplarisch den Mikrocode für einen Makrocode-Sprungbefehl *JOV jump if overflow*. Inwiefern beeinflusst diese Befehlsart (*Sprung entsprechend der Kontrollbits im PSR*) den Aufbau der Mikroinstruktionen?
3. Welche der aus der Vorlesung bekannten Makrobefehle sollten – um eine sinnvolle Verwendung der Kontrollbits im PSR zu erlauben – eine Änderung des PSR bewirken und welche nicht?