7. Übung
Vorlesung Rechnerstrukturen WS 99/2000

1. Gegeben sei die Rekurrenzrelation 10 P

   $x_0 = 0$

   $x_1 = 1$

   $x_i = 4x_{i-2} + (-1)^i * x_{i-1}$ für i >= 2

   Entwerfen Sie ein Mikroprogramm 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 Mikrocode, so daß der zugrundeliegende Algorithmus klar zu erkennen ist.

   Bemerkungen:
   - ein eventuell auftretender Überlauf kann ignoriert werden
   - Speicherzugriffe benötigen zwei Mikroinstruktionszyklen, d.h. zum Lesen einer Speicherzelle muß in zwei aufeinanderfolgenden Mikroinstruktionen das RD-Bit gesetzt sein. Nach Ablauf der beiden Zyklen sind die Daten in MBR gültig. Schreibzugriffe verlaufen analog.

2. 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.

   2.a. Erweitern Sie die einfache 16 Bit-CPU um ein PSR und die benötigte Steuerlogik. Gehen Sie davon aus, daß die ALU folgende Kontrollbits erzeugt: 8 P

N	negativ
Z	zero
C	carry
V	overflow

   2.b. 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 P

   2.c. 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? 4 P

AG Systemsoftware und Verteilte Systeme
Prof. Dr. P. Sturm, Steffen Rothkugel
V-119, (0651) 201-3311

2.d. Übersetzen Sie folgendes Hochsprachen-Codefragment in entsprechende Makrobefehle des *ursprünglichen* Befehlssatzes aus der Vorlesung 6 P

```
if x == mem42 then x = x + mem7
```

Gehen Sie davon aus, daß der Inhalt der Variablen `x` gegenwärtig *ausschließlich* im Akkumulator zur Verfügung steht. `mem42` bzw. `mem7` seien Variablen, die an Adresse 42 bzw. 7 im Speicher stehen.

Schlagen Sie eine Erweiterung des Makrobefehlssatzes vor, der durch Verwendung des PSR effizienteren Code erlaubt. Entwerfen Sie den entsprechenden Mikrocode.

2.e. In der ursprünglichen CPU können nur 16 Bit-Werte addiert werden. Welche Möglichkeiten sehen Sie, eine korrekte 32 Bit-Addition zu realisieren? 3 P

Ausgegeben: 17.12.1999

Abgabe: bis spätestens Freitag 14.1.2000 *zu Beginn* der Vorlesung