ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

INSTITUT FÜR INFORMATIK

Prof. Dr. Bernd Becker
Dr.-Ing. Christoph Scholl
Dipl. Inf. Marc Herbstritt

Freiburg, 31. Oktober 2000

2. Übungsblatt zur Vorlesung

Technische Informatik I

Aufgabe 1

Beim Entwurf des Instruktionssatzes eines neuen Rechners wird die Befehlsbreite auf 32 Bit festgelegt.

a) Wieviele Befehle können durch diese 32 Bit kodiert werden, wenn man keine Einschränkungen bezüglich der Abhängigkeiten zwischen den Leitungen annimmt?

b) Zur Realisierung eines `MOVE`-Befehls, der den Inhalt des Registers `S` in das Register `D` schreibt, werden bei der Befehlskodierung zwei Bit-Blöcke mit je 3 Bit für die Angabe von `S` und `D` reserviert, wie nachfolgende Abbildung beispielhaft veranschaulicht.

Desweiteren kann man annehmen, dass `S` und `D` immer unterschiedlich sind (d.h. die Befehle „`MOVE S S`“ bzw. „`MOVE D D`“ kommen nie vor). Ausserdem sind die restlichen Bits (31-28 und 21-0 in der Abbildung) frei wählbar. Wieviele gültige Möglichkeiten gibt es dann für den `MOVE`-Befehl?

Aufgabe 2

Sei S ein System, welches eine Aufgabe A bestehend aus n Teilaufgaben $A_1, \ldots, A_n$ ausführt, wobei S für alle Teilaufgaben die gleiche Zeit benötigt. T sei eine verbesserte Version von S, die 40% der Teilaufgaben von A 10-mal schneller ausführen kann als S. Die Ausführungszeit der restlichen Aufgaben sei bei S und T gleich.

Wie groß ist die Beschleunigung (**speedup**), d.h. der Quotient aus der alten und der neuen Ausführungszeit, die man bei der Ausführung von A durch den Einsatz von T gegenüber S erhält. Begründe deine Überlegungen.

Aufgabe 3

(Halloween-Aufgabe)
Betrachten sie folgendes Maschinensprachenprogramm, welches ab Zeile 3110 im Hauptspeicher steht:

```
3110:  LOAD   ACC, PC      ; ACC := PC
3111:  ADDI   -1           ; ACC := ACC-1
3112:  STORE  ACC, PC      ; PC := ACC
```

`ACC` bezeichnet dabei den Akkumulator, d.h. das Arbeitsregister, und `PC` den Programmzähler (engl.: *program counter*), d.h. die Adresse des nächsten Befehls im Hauptspeicher. Der Programmzähler PC wird nach der Ausführung einer Instruktion um 1 erhöht.
Was tut bzw. berechnet dieses Programm, wenn es ausgeführt wird?

Aufgabe 4

Unter der Annahme, daß in einem sequentiellen Programm die Anweisung S_1 vor der Anweisung S_2 steht, kann man *Datenabhängigkeiten* folgendermaßen klassifizieren:

- *True Dependence*(read after write): S_2 liest eine Variable, die in S_1 beschrieben wird.
- *Anti Dependence*(write after read): S_2 schreibt auf eine Variable, die in S_1 gelesen wird.
- *Output Dependence*(write after write):S_2 schreibt auf eine Variable, die auch in S_1 beschrieben wird.

Betrachte das folgende sequentielle Programmstück:

```
S1:  JMP    S3          ; springe nach S3
S2:  LOAD   a,R4        ; R4:=[a]
S3:  LOAD   b,R1        ; R1:=[b]
S4:  LOAD   c,R2        ; R2:=[c]
S5:  MUL    R1,R2,R1    ; R1:= R1·R2
S6:  ADD    R1,R2,R2    ; R2:= R1+R2
```

Bestimme alle Datenabhängigkeiten in diesem Programmstück. Unterscheide dabei zwischen True Dependencies, Anti Dependencies und Output Dependencies.

Abgabetermin: Dienstag, 07.11.2000, 14:00Uhr