Universität des Saarlandes
Fachrichtung 6.2 – Informatik
Dr.-Ing. D. Kästner

17. Januar 2002

Übungsblatt 8: Codeerzeugung für eingebettete Prozessoren (WS 2001/2002)

Abgabetermin: 24. Januar 2002

Aufgabe 1 (20 Punkte)

Gegeben sei das folgende SHARC-Assemblerprogramm für einen FIR-Filter.

```
fir:
dm(i0,m0)=r0;
r0=dm(i0,m0);
r3=pm(i8,m8);
r2=r0*r3;
r0=dm(i0,m0);
r3=pm(i8,m8);
r1=r0*r3;
r0=dm(i0,m0);
r3=pm(i8,m8);
lcntr=1000, do macs until lce;
   r1=r0*r3;
   r0=dm(i0,m0);
   r3=pm(i8,m8);
macs:   r2=r2+r1;
r2=r2+r1;
r1=r0*r3;
r0=r2+r1;
rts;
```

a) Bestimmen Sie den Datenabhängigkeitsgraphen dieses Programmes und berechnen Sie für jede Operation eine Priorität nach der *highest-level-first* Heuristik. Iterationsübergreifende Abhängigkeiten (*loop-carried dependences*) werden bei der Prioritätsbestimmung vernachlässigt. Wenden Sie den List Scheduling Algorithmus auf das Programm an und bestimmen Sie den resultierenden Schedule.

b) Welche Verbesserungen können bezüglich Codegröße und Ausführungszeit des generierten Codes durch eine geschicktere Registerzuteilung (*register assignment*) erzielt werden? Ermitteln Sie dazu eine geschickte Registerumbenennung, und geben Sie einen verbesserten Schedule, sowie die prozentualen Verbesserungen gegenüber dem Resultat von Teil a) an. Verwenden Sie die Anzahl der (VLIW-) Instruktionen als ein Maß für die Codegröße und nehmen Sie zur Bestimmung der Ausführungszeit an, daß jede Instruktion in einem Taktzyklus ausgeführt werden kann. Es sollen hierbei keine Operationen entfernt oder hinzugefügt werden; lediglich Veränderungen durch *register renaming* sind zu beachten.