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

6. Dezember 2001

Übungsblatt 6: Codeerzeugung für eingebettete Prozessoren (WS 2001/2002)

Abgabetermin: 10. Januar 2002

Aufgabe 1 (10 Punkte)

In der Vorlesung wurde das folgende einfache Maschinenmodell vorgestellt:

– r general purpose register $R_0, \ldots, R_{r-1}$,

– Zweiadreßinstruktionen:

R_i	$:=$	$M[V]$	Load
$M[V]$	$:=$	R_i	Store
R_i	$:=$	$R_i\ op\ M[V]$	Compute
R_i	$:=$	$R_i\ op\ R_j$	

Betrachte die Anweisungsfolge:

```
b=a*(c+(d-e));
a=a+b*(c+(d-e));
```

a) Wenden Sie das in der Vorlesung vorgestellte integrierte Codeerzeugungsverfahren für obiges Maschinenmodell mit $r = 2$ auf jeden der beiden Ausdrucksbäume an.

b) Ist die Instruktionsanzahl des resultierenden Gesamtcodes minimal? Falls nicht, geben Sie eine Instruktionssequenz minimaler Länge für obige Anweisungsfolge an.

Projekt (60 Punkte)

Ziel des Projektes ist die Implementierung eines List-Scheduling Algorithmus für den Analog Devices SHARC-Prozessor. Eingabe ist eine Repräsentation von Assemblerprogrammen des SHARC im `crl`-Format; Ausgabe soll die transformierte Datei, ebenfalls im `crl`-Format sein. Der Algorithmus soll auf Basisblöcken lauffähig sein. Zwischenschritte:

1. Berechnen Sie den Datenabhängigkeitsgraphen mit dem in Wilhelm/Maurer (Übersetzerbau, 2. Auflage) vorgestellten Verfahren.

2. Implementierung des List Scheduling Verfahrens mit highest-level-first Heuristik.

(Bewertungsverfahren: Der Scheduling-Algorithmus soll für die zur Verfügung gestellten Testprogramme korrekt funktionieren; zusätzlich werden zwei interne Dateien überprüft. Die Implementierung ist dann in der Übungsbesprechung zu erklären.)

Hilfsmittel und Materialien sind zu finden auf den cip-Rechnern unter `/home/stud/kaestner/CGEP`.

- `~kaestner/CGEP/doc` enthält alle relevanten Dokumentationen.
- `~kaestner/CGEP/input` enthalt die Eingabeprogramme.
- `~kaestner/CGEP/lib` und `~kaestner/CGEP/include` enthalten die erforderlichen Bibliotheken und Include-Dateien.
- `~kaestner/CGEP/bin` enthält das Visualisierungstool aiSee, sowie das Programm `crl2gdl`, das für eine `crl`-Datei eine Visualisierung im `gdl`-Format (Eingabe von aiSee) erzeugt.
- Die `crl`-Library (`libcrl.a`) stellt Funktionen zum Einlesen von `crl`-Dateien und zum Aufbau der internen Repräsentation bereit.
- Die `rary`-Library (`library.a`) stellt Hilfsfunktionen bereit, die von `libcrl` benötigt werden, wie z.B. Implementierungen von Listen und Hashtabellen.
- `~kaestner/CGEP/listsched` enthält ein Beispielprogramm `listsched.c`, das exemplarisch die interne Repräsentation des Kontrollflußgraphen durchläuft und für jede Operation die gelesenen und beschriebenen Ressourcen ausgibt. Ein Makefile steht ebenfalls bereit, das entsprechend angepaßt werden kann.