

Universität Erlangen-Nürnberg  
Technische Fakultät  
Lehrstuhl für Hardware-Software-Co-Design  
Prof. Dr.-Ing. Jürgen Teich

# Klausur Hardware-Software-Co-Design

7. Oktober 2008

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	$\Sigma$
max. Punkte	10	20	22	8	60
erreichte Punkte					
<b>Note</b>					

## Aufgabe 1 (Kurzfragen)

(10 Punkte)

- a) Geben Sie das aus der Vorlesung bekannte Doppeldachmodell an. (1 Punkt)
- b) Was sind die drei Grundprobleme bei der Synthese? (1 Punkt)
- c) Nennen Sie Ihnen bekannte Zielplattformen. (1 Punkt)
- d) Nennen Sie die Kriterien, die an eine Zielarchitektur zur Realisierung eines Systems zur Bildverarbeitung gestellt werden. (1 Punkt)
- e) Was sind die charakteristischen Architektureigenschaften von digitalen Signalprozessoren (DSPs)? (1 Punkt)
- f) Welche Verfahren zur Registervergabe kennen Sie? (1 Punkt)
- g) Welche Peephole- und lokalen Optimierungsverfahren zur Codeerzeugung kennen Sie aus der Vorlesung? (1 Punkt)

h) Welche globalen Optimierungsverfahren zur Codeerzeugung kennen Sie? (1 Punkt)

i) In folgender Tabelle sind für vier Entwurfspunkte Metriken der Entwurfsqualität dargestellt. Für diese Metriken gibt es geschätzte Werte  $E(D)$  sowie gemessene Werte  $M(D)$ . Bestimmen Sie die Treue des Schätzverfahrens? (1 Punkt)

Entwurfspunkt	$M(D)$	$E(D)$
W	117	104
X	133	132
Y	144	116
Z	127	127

j) Wie viele Möglichkeiten gibt es, 100 Objekte in 3 Partitionsblöcke (A,B,C) zu partitionieren, wenn 10 dieser Objekte nur in Partition A oder C liegen können? (1 Punkt)

## Aufgabe 2 (Compiler und Codegenerierung)

(20 Punkte)

a) Welche Phasen werden während der Compilierung durchlaufen?

(2 Punkte)

b) Gegeben ist der folgende Grundblock:

(2 Punkte)

(1)  $a = i * f$

(2)  $b = g - h$

(3)  $c = a - g$

(4)  $d = (c * b + f * e) - (c * b + b * g * g)$

Konstruieren Sie den DAG (gerichteten azyklischen Graphen) für diesen Grundblock.

c) Gegeben ist der Bubble-Sort-Algorithmus als Drei-Adress-Code: (2 Punkte)

1. Identifizieren Sie alle Grundblöcke und markieren Sie diese in dem gegebenen Drei-Adress-Code!

```
( 10) i := n - 1
( 20) if i < 0 goto (140)
( 30)  j := 1
( 40)  if j > i goto (120)
( 50)    k := j - 1
( 60)    if a[k] >= a[j] goto (100)
( 70)      t := a[k]
( 80)      a[k] := a[j]
( 90)      a[j] := t
(100)      j := j + 1
(110)    goto (40)
(120)  i := i - 1
(130)  goto (20)
(140) i := 0
```

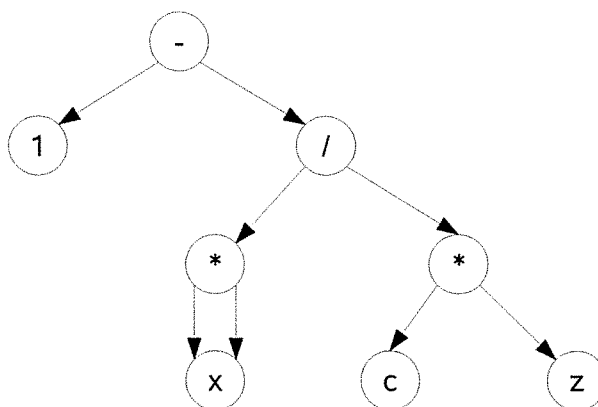
2. Stellen Sie den Kontrollfluss des Programmausschnittes als Kontrollflussgraphen (CFG) dar!

d) Gegeben sei folgende Spezifikation

$$(1 - (x * x)) / (c * z)$$

als DAG:

(6 Punkte)



1. Ihre Zielmaschine verfügt über zwei Universalregister und folgenden Befehlssatz:

$R_i := R_i \text{ op } R_j$

$R_i := R_i \text{ op } M_j$

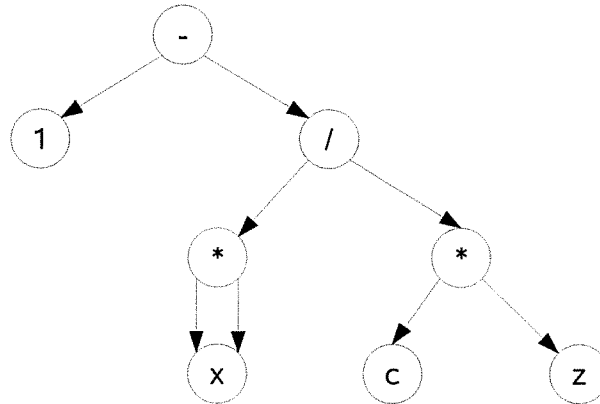
$R_i := R_j$

$R_i := M$

$M := R_i$

Die Kosten für die Subtraktion seien  $c_- = 1$  und für Multiplikation als auch Division  $c_* = c_/ = 2$ .

Wenden Sie das Verfahren der dynamischen Programmierung zur Codeerzeugung an, und generieren Sie den Zielcode. Die Konstante '1' sei dabei im Speicher abgelegt.





3. Bestimmen Sie eine Registerbindung mit Hilfe von Graphfärbung für vier Register! Geben Sie Ihren Lösungsweg an.

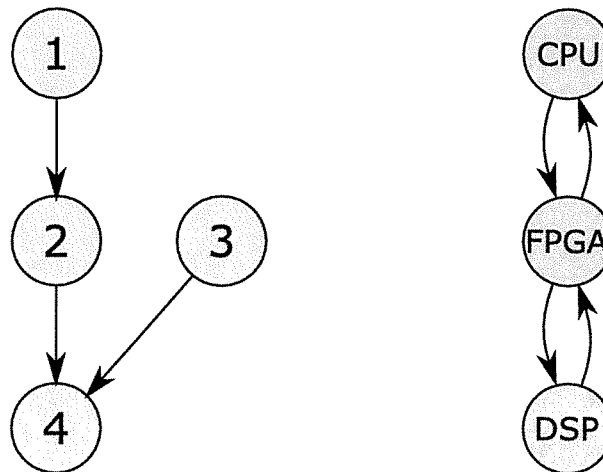
4. Nun haben Sie einen Rechner mit drei Registern. Da der Graph nicht 3-färbbar ist, müssen Variablen im Speicher gehalten werden. Bestimmen Sie, welche Variablen bevorzugt in Registern gehalten werden sollen sowie eine Registerbindung mit einer Heuristik für 3 Register!



## Aufgabe 3 (Entwurfsraumexploration)

(22 Punkte)

Gegeben sei ein Datenflussgraph mit vier Tasks  $t_1, \dots, t_4$  (im Bild links) und ein Architekturgraph mit drei Knoten  $r_{cpu}, r_{fpga}, r_{dsp}$  (rechts). Im Folgenden soll ein ILP (ganzzahliges lineares Programm) erstellt werden, das zur Entwurfsraumexploration genutzt werden kann.



- a) Task 1 und 2 können auf der CPU oder dem FPGA laufen. Task 3 kann auf dem FPGA oder dem DSP laufen und Task 4 kann nur auf dem FPGA ausgeführt werden. Zeichnen Sie die Bindungskanten in obiges Bild ein um eine vollständige Spezifikation zu erzeugen. (1 Punkt)
- b) Geben Sie die Nebenbedingungen an, die fordern, dass jeder Task genau einmal entsprechend seiner möglichen Bindungen aktiviert wird. (3 Punkte)

- c) Geben Sie für jede Ressource eine Nebenbedingung an, die fordert, dass sie stets aktiviert wird, wenn ein Task auf sie gebunden wurde. Verhindert das bereits die Allokation leerer Ressourcen? Was wäre hierfür nötig? (4 Punkte)

- d) Geben Sie für Task 2 die Nebenbedingungen an, die gewährleisten, dass die Kommunikation entsprechend dem Datenflussgraph möglich ist. (4 Punkte)

- e)  $c_{i,k}$  gibt die Kosten an, die die Implementierung von Task  $i$  auf Ressource  $k$  verursachen würden. Geben Sie die Nebenbedingung an, die gewährleistet, dass auf dem FPGA maximal Kosten von 9 entstehen. (4 Punkte)

$$c_{1,fpga} = 2$$

$$c_{2,fpga} = 2$$

$$c_{3,fpga} = 5$$

$$c_{4,fpga} = 1$$

- f) Geben Sie die Zielfunktion an, um die Gesamtkosten zu minimieren. Die noch fehlenden Kosten sind wie folgt gegeben: (4 Punkte)

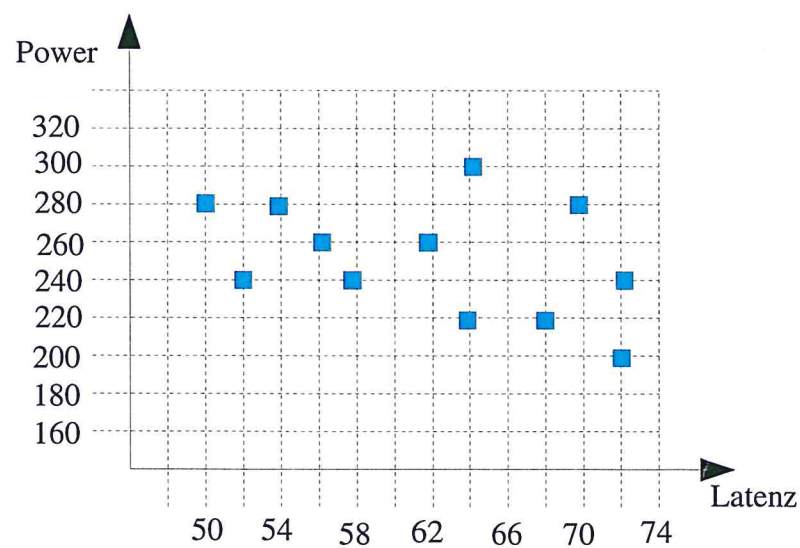
$$c_{1,cpu} = 5$$

$$c_{2,cpu} = 7$$

$$c_{3,dsp} = 4$$

- g) Durch welche Methoden verbessern Evolutionäre Algorithmen ihre Lösungsmengen?(1 Punkt)

- h) Markieren Sie in folgendem Diagramm alle Pareto-Punkte. Beide Zielfunktionen sind zu minimieren. (1 Punkt)

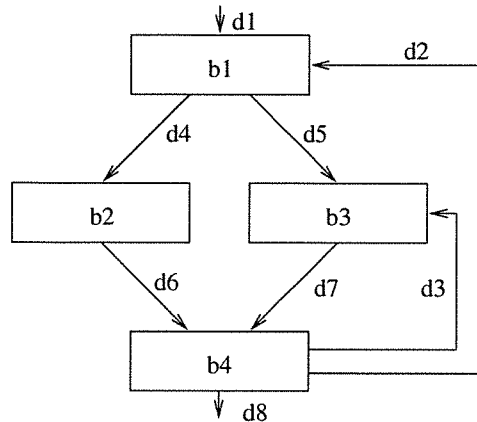


## Aufgabe 4 (Schätzung)

(8 Punkte)

## a) WCET-Abschätzung

In der folgenden Abbildung ist ein Grundblockgraph mit vier Blöcken  $b_1$ ,  $b_2$ ,  $b_3$  und  $b_4$  gegeben, die alle die gleiche Ausführungszeit  $t_{exec}$  haben. Der Rücksprung von  $b_4$  nach  $b_1$  wird 9 mal und der Rücksprung von  $b_4$  nach  $b_3$  zweimal ausgeführt. Außerdem wissen Sie, dass der Block  $b_2$  doppelt so oft ausgeführt wird wie der Block  $b_3$ .

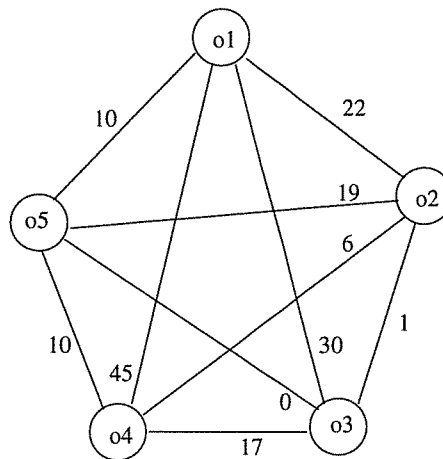


- Stellen Sie ein lineares Gleichungssystem zur Bestimmung der Worst-Case-Execution-Time (WCET) auf. (2 Punkte)

- Lösen Sie das lineare Gleichungssystem und geben Sie eine möglichst gute Abschätzung der WCET an. Der Rechenweg und Begründungen sind anzugeben. (2 Punkte)

**b) Partitionierung**

Gegeben ist der folgende Graph mit fünf Knoten und ihren Closeness-Werten:



Führen Sie für diesen Graphen das hierarchische Clusteringverfahren durch. Verwenden Sie Mittelwertbildung. (4 Punkte)