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

Klausur
Hardware-Software-Co-Design

7. Oktober 2009

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	Σ
max. Punkte	15	30	25	20	90
erreichte Punkte					
Note					

Aufgabe 1 (Kurzfragen) (15 Punkte)

a) Geben Sie das aus der Vorlesung bekannte Doppeldachmodell an und beschriften Sie es. (2 Punkte)

b) Was sind die drei Grundprobleme bei der Synthese? (1 Punkt)

c) Nennen Sie drei mögliche Hardware-/Software-Schnittstellen. (1 Punkt)

d) Nennen Sie zwei Sprachen für den Entwurf digitaler Schaltungen. (1 Punkt)

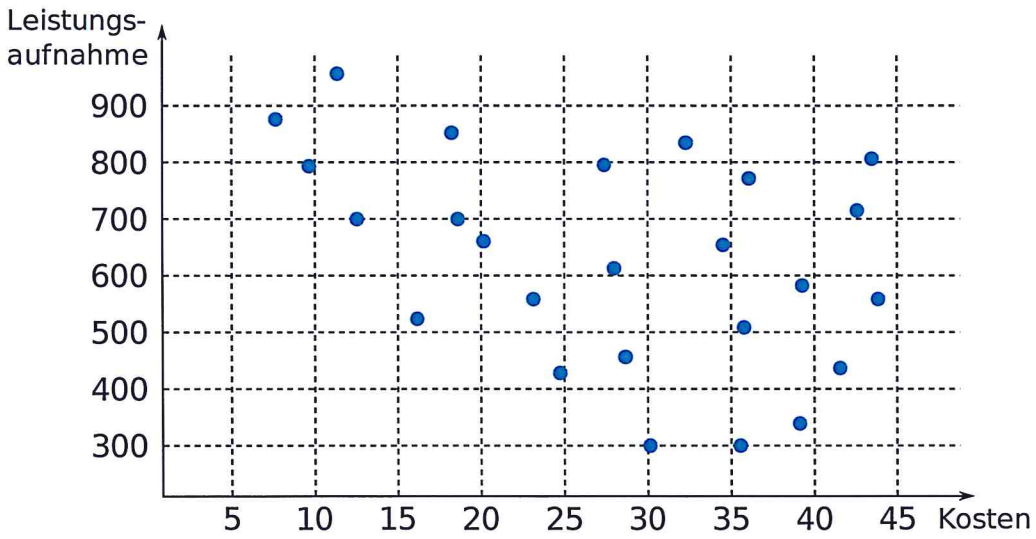
e) Nennen Sie vier verschiedene Optimierungsalgorithmen, die bei der Partitionierung Anwendung finden! (2 Punkte)

f) Durch welche Methoden verbessern evolutionäre Algorithmen ihre Lösungsmengen und wie funktionieren diese? (1 Punkt)

g) Was sind die Unterschiede zwischen einer Partitionierung mittels ILPs und EAs? (2 Punkte)

h) Wie kann man ein Mehrzieloptimierungsproblem auf ein Einzieloptimierungsproblem abbilden? (1 Punkt)

i) Markieren Sie alle Paretopunkte in dem folgenden Diagramm. Beide Zielfunktionen sollen minimiert werden. (2 Punkte)



j) Wie viele Möglichkeiten gibt es, 100 Objekte in 3 Partitionsblocke (A,B,C) zu partitionieren, wenn 10 dieser Objekte nur in Partition A oder C liegen können? (2 Punkte)

Aufgabe 2 (Compiler und Codegenerierung) (30 Punkte)

a) Welche sechs Phasen werden während der Compilierung durchlaufen? (3 Punkte)

b) Gegeben ist der folgende Grundblock: (6 Punkte)

```
(1) t = x * y;
(2) t = t * c;
(3) a[i] = t * c + x * y;
```

Konstruieren Sie den DAG (gerichteten azyklischen Graphen) für diesen Grundblock. Die Variable `t` sei nur innerhalb dieses Grundblocks aktiv ist.

c) Gegeben sei folgender Algorithmus

```
x := 1;
for i:=1 to m do
  while (F2[i]>0) do
    F[x]:=i;
    F2[i]:=F2[i] - x*x*i;
    x:=x+1;
  endwhile;
endfor;
```

Wandeln Sie den Code in Drei-Adress-Code um. (5 Punkte)

d) Gegeben ist der Bubble-Sort-Algorithmus als Drei-Adress-Code. Identifizieren Sie alle Grundblöcke und markieren Sie diese in dem gegebenen Drei-Adress-Code: (6 Punkte)

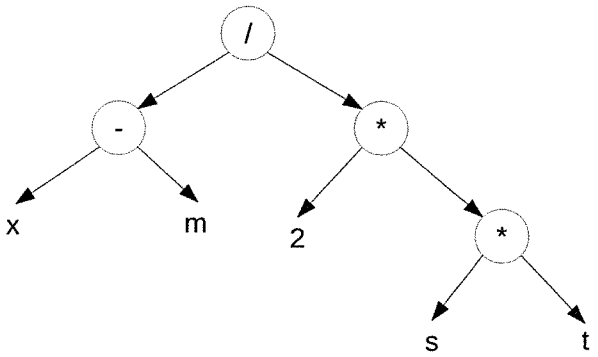
```
( 10) i := n - 1
( 20) if i < 0 goto (150)
( 30)   j := 1
( 40)   if j > i goto (130)
( 50)     k := j - 1
( 60)     t1 := a[k]
( 70)     t2 := a[j]
( 80)     if t1 >= t2 goto (110)
( 90)       a[k] := t2
(100)       a[j] := t1
(110)       j := j + 1
(120)     goto (40)
(130)     i := i - 1
(140)   goto (20)
(150) i := 0
```

e) Gegeben sei die Spezifikation $(x-m)/(2*s*t)$ als DAG. Ihre Zielmaschine verfügt über zwei Universalregister und folgenden Befehlssatz:

```
Ri := Ri op Rj
Ri := Ri op Mj
Ri := Rj
Ri := M
M := Ri
```

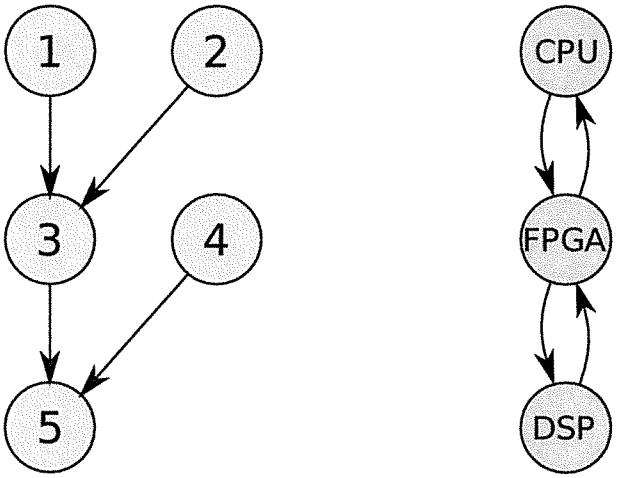
Die Kosten für die Subtraktion seien $c_- = 1$ und für Multiplikation als auch Division $c_* = c_/ = 2$.

Wenden Sie das Verfahren der dynamischen Programmierung zur Codeerzeugung an und generieren Sie den Zielcode. Die Konstante '2' sei dabei im Speicher abgelegt. (10 Punkte)



Aufgabe 3 (Entwurfsraumexploration) (25 Punkte)

Gegeben sei ein Datenflussgraph mit fünf Tasks $t_1, \ldots, t_5$ (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 t_1 und t_3 können auf der CPU oder dem FPGA laufen. Task t_2 und t_5 können auf dem FPGA oder dem DSP laufen und Task t_4 kann nur auf dem FPGA ausgeführt werden. Zeichnen Sie die entsprechenden Bindungskanten in obiges Bild ein, um somit eine vollständige Spezifikation zu erzeugen. (2 Punkte)

b) Geben Sie die Nebenbedingungen an, die fordern, dass jeder Task genau einmal entsprechend seiner möglichen Bindungen aktiviert wird. Definieren Sie sich hierzu Variablen zur Codierung der Bindungen sowie zur Codierung der Ressourcen. (5 Punkte)

c) Geben Sie für jede Ressource eine Nebenbedingungen 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? (5 Punkte)

d) Geben Sie für Task t_3 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 t_i auf Ressource r_k verursachen würden. Geben Sie die Nebenbedingung an, die gewährleistet, dass auf dem FPGA maximal Kosten von 9 entstehen. (5 Punkte)

$c_{1,fpga} = 2$
$c_{2,fpga} = 2$
$c_{3,fpga} = 5$
$c_{4,fpga} = 1$
$c_{5,fpga} = 3$

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

$c_{1,cpu} = 5$
$c_{2,dsp} = 4$
$c_{3,cpu} = 7$
$c_{5,dsp} = 3$

Aufgabe 4 (Schätzung) (20 Punkte)

a) In folgender Tabelle sind vier Entwurfspunkte und ihre geschätzte Entwurfsqualität $E(D)$ als auch ihre tatsächlich gemessenen Entwurfsqualität $M(D)$ dargestellt.

Entwurfspunkt	$E(D)$	$M(D)$
W	112	109
X	128	137
Y	139	121
Z	205	132

1. Bestimmen Sie die Exaktheit der Abschätzungen. (4 Punkte)

2. Bestimmen Sie die Treue des Schätzverfahrens. (4 Punkte)

b) Gegeben sei der Sequenzgraph aus Abbildung 1. Die Ausführungszeit einer Subtraktion betrage $del(v_-) = 30ns$ und die einer Multiplikation $del(v_*) = 80ns$. Im Folgenden soll die Ausführungszeit unter Verwendung zweier Multiplizierer und einer ALU (für die Subtraktion) bestimmt werden.

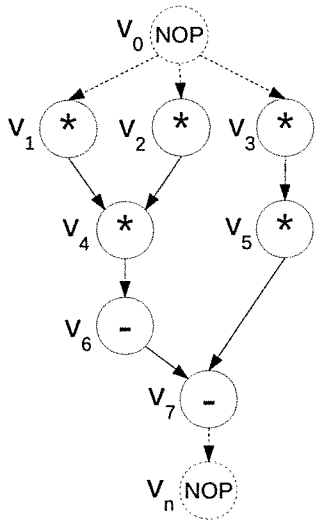
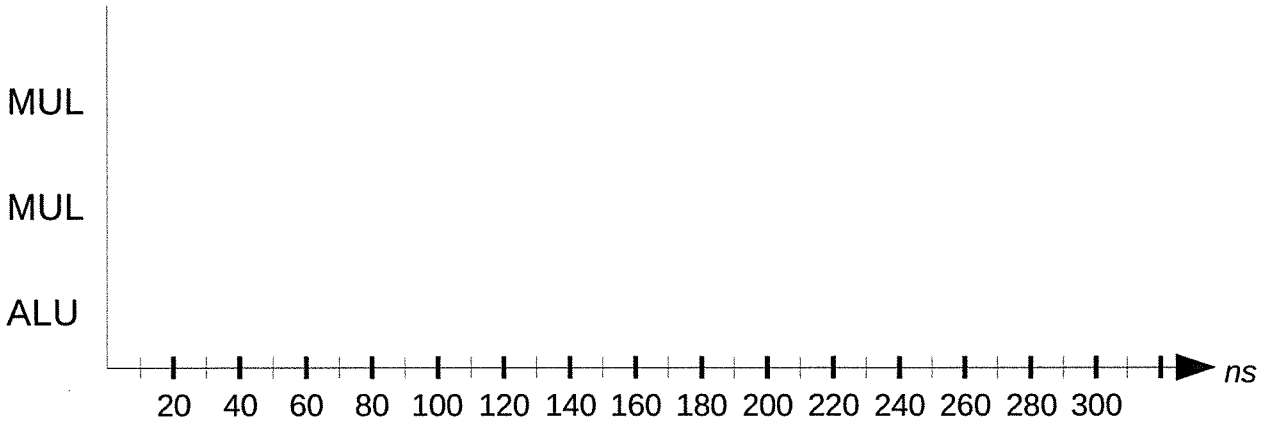
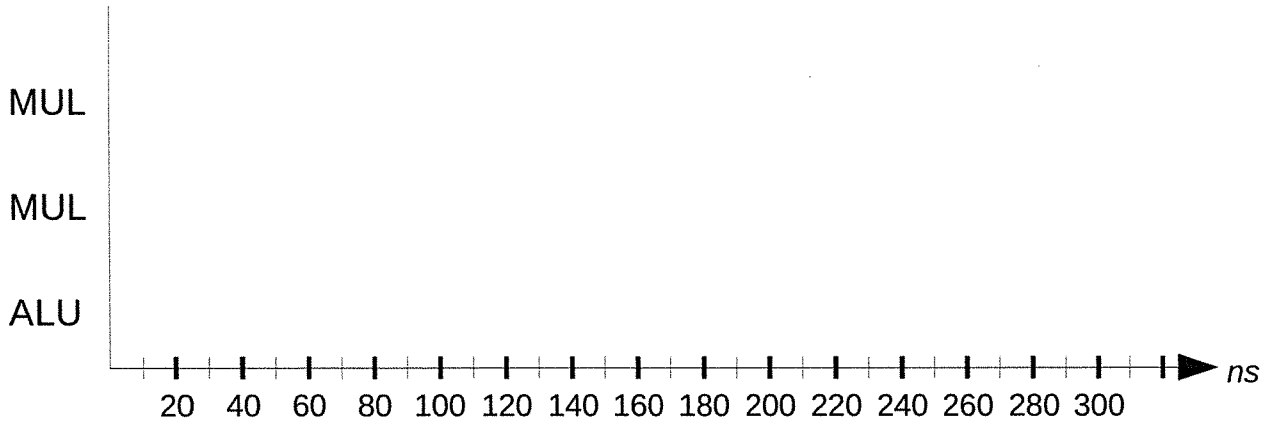


Abbildung 1: Sequenzgraph 4 a)



1. Die Taktrate sei durch die Operation mit maximaler Verzögerung gegeben. (3 Punkte)

2. Ermitteln Sie die Ausführungszeit für die Taktrate T, durch die der mittlere Taktschlupf minimiert wird. (8 Punkte)



c) Aufgabe 3b) hat gezeigt, dass man für Hardware-Schaltungen die Ausführungszeit exakt berechnen kann, sobald man sich auf eine Taktrate und Allokation festgelegt hat. Warum sind hingegen die Ausführungszeiten auf Prozessoren im Allgemeinen approximativ?

(1 Punkt)