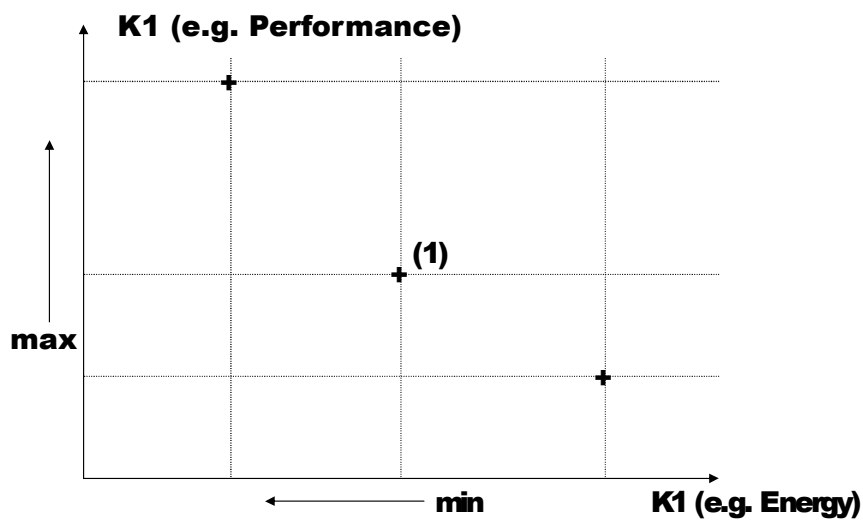


Theoretisches Aufgabenblatt 6

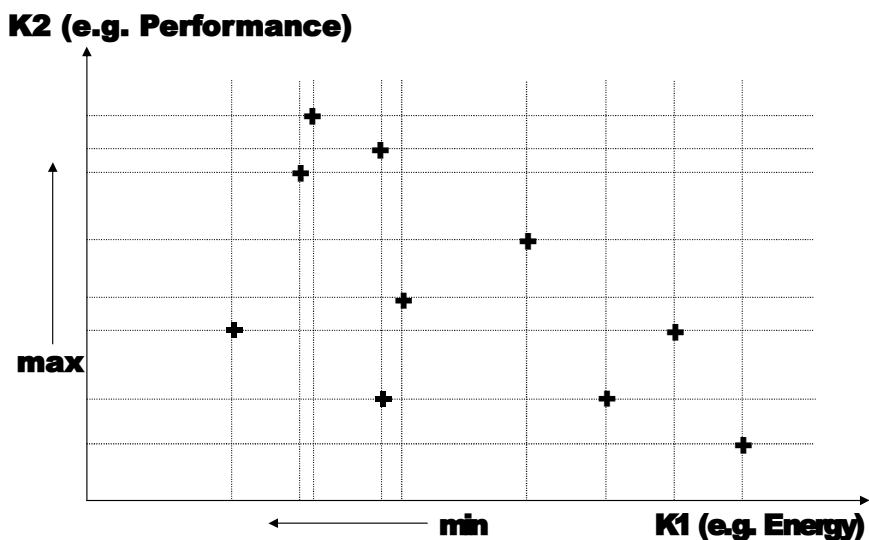
Hints: Dieses Aufgabenblatt wird am 16.01.2018 von 10.15 Uhr bis 11.45 Uhr in Raum OH14/E23 besprochen. Sie sind nicht verpflichtet, Ihre Lösungen abzugeben

1 Pareto-Optimierung

- Das nachfolgende Diagramm bildet die Evaluation von Designs hinsichtlich mehrerer Kriterien ab. Wir nehmen an, dass wir ein Kriterium maximieren und ein anderes minimieren möchten. Bitte kennzeichnen Sie die Region, die von Design (1) dominiert ist (die Region, in der Designs in Bezug auf Design (1) "inferior" sind). Kennzeichnen Sie ebenfalls die Region, in der Designs Design (1) dominieren (die Region, in der Designs in Bezug auf Design (1) "superior" sind).



- Die im folgenden Diagramm mit "+" gekennzeichneten Designpunkte seien gegeben. Fügen Sie die Pareto-Front hinzu!



2 Hardware-Software-Partitionierung

Nehmen Sie an, dass ein Task entweder komplett auf einem FPGA oder komplett auf dem Mikroprozessor ausgeführt werden kann. Nehmen Sie außerdem an, dass es keine Abhängigkeit zwischen den Tasks gibt. Wenn ein Task τ_i auf dem FPGA ausgeführt wird, benötigt er dafür B_i konfigurierbare logische Blöcke (configurable logical blocks (CLB)) und hat eine Ausführungszeit von F_i . Wenn der Task auf dem Mikroprozessor ausgeführt wird, ist seine Ausführungszeit C_i . Die Tasks, die dem Mikroprozessor zugeordnet werden, können nur sequentiell ausgeführt werden, während die dem FPGA zugeordneten Tasks vollständig parallel ausgeführt werden können.

Was ist die minimale Größe des FPGAs hinsichtlich der Anzahl der konfigurierbaren logischen Blöcke (CLBs), um alle Tasks $\{\tau_1, \tau_2, \dots, \tau_n\}$ vor einer Deadline D_i für τ_i mit $x_i F_i \leq D_i$ zu beenden, wenn angenommen wird, dass alle Tasks zum Zeitpunkt $t = 0$ ankommen und ausführbar sind?

Formulieren Sie dieses Hardware-Software-Partitionierungsproblem als ganzzahliges lineares Programm (integer linear program (ILP)) und erklären Sie es. Es wird nicht von Ihnen erwartet, dass Sie das ILP lösen.

3 Multiprocessor Partitioned Scheduling

Gegeben seien M identische (homogene) Prozessoren. Jeder Task τ_i sei ein periodischer Task mit Utilization U_i und Periode T_i . Die Utilization von Task τ_i ist unabhängig von dem Prozessor, auf dem τ_i ausgeführt wird. Bitte formulieren Sie ein ganzzahliges lineares Programm (ILP), um das gegebene Taskset $\{\tau_1, \tau_2, \dots, \tau_n\}$ so zu partitionieren, dass τ_i einem bestimmten, dedizierten Prozessor zugeordnet wird und zudem alle Tasks unter einem *Rate Monotonic* Schedule ihre Deadlines einhalten. Gehen Sie davon aus, dass die gegebenen Task voneinander unabhängig sind.

Hint: Sie können den Liu and Layland Bound verwenden.

4 Compositional WCET

Mr. Smart hat die WCET $4ms$ für ein Programm berechnet, das auf einer (deterministischen) Plattform mit einer CPU-Frequenz von 1GHz ausgeführt wird. Das System ist ein wenig zu leistungsstark und kann auf eine CPU-Frequenz von 500MHz verlangsamt werden (der Rest der Plattform bleibt unverändert), um die Leistungsaufnahme zu verringern. Mr. Smart folgert, dass die WCET für das betrachtete Programm unter der neuen Konfiguration $8ms$ beträgt. Ist diese Abschätzung sicher? Ist die Abschätzung zu pessimistisch? Falls sie pessimistisch ist, wie können wir sie verbessern? Falls sie nicht pessimistisch ist, warum nicht?

5 May Cache und Must Cache

Gegeben sei ein LRU-Cache mit Assoziativität 4.

- Ist es möglich, weniger als 4 (verschiedene) Einträge im Must Cache zu haben (abstrakte Interpretation)?
- Ist es möglich, mehr als 4 (verschiedene) Einträge im Must Cache zu haben (abstrakte Interpretation)?
- Ist es möglich, weniger als 4 (verschiedene) Einträge im May Cache zu haben (abstrakte Interpretation)?
- Ist es möglich, mehr als 4 (verschiedene) Einträge im May Cache zu haben (abstrakte Interpretation)?

Erklären Sie Ihre Antwort.

6 Cache-Analyse

Betrachten Sie das folgende Programm:

```
read a;  
read b;  
read a;  
if (a>b) {  
    read c;  
    read d;  
} else {  
    read e;  
    read f;  
}  
read x;  
read a;
```

- (a) Führen Sie für das gegebene Programm Analysen hinsichtlich des May Cache und Must Cache durch, wobei Sie annehmen, dass die LRU-Ersetzungsstrategie (last-recently used) mit Assoziativität 4 angewendet wird und dass der Cache zu Programmstart leer ist. Ist es möglich, zu bestimmen, ob der letzte Zugriff in einem Cache Hit oder Cache Miss resultiert?
- (b) Nehmen Sie nun an, dass der Cache eine FIFO-Ersetzungsstrategie (first-in-first-out) verwendet. Könnte eine May-Cache- oder Must-Cache-Analyse bestimmen, ob der letzte Zugriff auf "a" in einem Cache Hit oder Cache Miss resultiert, wenn der Cache zu Programmstart leer ist?

7 WCET für Systeme mit Scratchpad-Memory

Erklären Sie die Unterschiede hinsichtlich der WCET-Analyse unter Verwendung von Scratchpad-Memory und Cache-Memory.