

Beispielhafte Prüfungsaufgaben zur Vorlesung
Embedded Systems
Gestellt im Herbst 2005

Die beigefügte Lösung ist ein Vorschlag. Für Korrektheit, Vollständigkeit und Verständlichkeit wird keine Verantwortung übernommen.

Aufgabe 1 : Ablaufplanung periodischer Tasks (maximal 45 Punkte)

Eine Einprozessormaschine soll ein Set S von vier periodischen Tasks τ_i abarbeiten. Die Perioden T_i , die relativen Deadlines D_i sowie die Ausführungszeiten C_i können der folgenden Tabelle entnommen werden:

	τ_1	τ_2	τ_3	τ_4
$T_i = D_i$	5	6	10	15
C_i	1	2	3	1

Alle vier Tasks starten zum Zeitpunkt 0. Im weiteren sind alle Tasks unabhängig voneinander, und zu jedem Zeitpunkt unterbrechbar.

1.1: Rate Monotonic Scheduler (maximal 20 Punkte)

- (a) (3 Punkte) Überprüfen Sie mit der hinreichenden Bedingung für die Rate Monotonic (RM) Ablaufplanung, ob das Taskset S mit einem RM Scheduler in Echtzeit abgearbeitet werden kann.

Lösungsvorschlag:

$$U = \sum_{i=1}^4 \frac{C_i}{D_i} = 0.9 \not\leq 0.757 = 4(2^{1/4} - 1)$$

□

- (b) (10 Punkte) Überprüfen Sie mit der notwendigen und hinreichenden Bedingung für die RM Ablaufplanung, ob das Taskset S mit einem RM Scheduler in Echtzeit abgearbeitet werden kann.

Lösungsvorschlag:

$$\begin{aligned} R_4^0 &= C_4 = 1 & I_4^0 &= \left\lceil \frac{1}{5} \right\rceil 1 + \left\lceil \frac{1}{6} \right\rceil 2 + \left\lceil \frac{1}{10} \right\rceil 3 = 6 \\ R_4^1 &= 1 + 6 = 7 & I_4^1 &= \left\lceil \frac{7}{5} \right\rceil 1 + \left\lceil \frac{7}{6} \right\rceil 2 + \left\lceil \frac{7}{10} \right\rceil 3 = 9 \\ R_4^2 &= 1 + 9 = 10 & I_4^2 &= \left\lceil \frac{10}{5} \right\rceil 1 + \left\lceil \frac{10}{6} \right\rceil 2 + \left\lceil \frac{10}{10} \right\rceil 3 = 9 \\ R_4^3 &= R_4^2 = 10 \leq T_4 \end{aligned}$$

$$\begin{aligned} R_3^0 &= C_3 = 3 & I_3^0 &= \left\lceil \frac{3}{5} \right\rceil 1 + \left\lceil \frac{3}{6} \right\rceil 2 = 3 \\ R_3^1 &= 3 + 3 = 6 & I_3^1 &= \left\lceil \frac{6}{5} \right\rceil 1 + \left\lceil \frac{6}{6} \right\rceil 2 = 4 \\ R_3^2 &= 3 + 4 = 7 & I_3^2 &= \left\lceil \frac{7}{5} \right\rceil 1 + \left\lceil \frac{7}{6} \right\rceil 2 = 6 \\ R_3^3 &= 3 + 6 = 9 & I_3^3 &= \left\lceil \frac{9}{5} \right\rceil 1 + \left\lceil \frac{9}{6} \right\rceil 2 = 6 \\ R_3^4 &= R_3^3 = 9 \leq T_3 \end{aligned}$$

$$\begin{aligned}
 R_2^0 &= C_2 = 2 & I_2^0 &= \left\lceil \frac{2}{5} \right\rceil 1 = 1 \\
 R_2^1 &= 2 + 1 = 3 & I_2^1 &= \left\lceil \frac{3}{5} \right\rceil 1 = 1 \\
 R_2^2 &= R_2^1 = 3 \leq T_2
 \end{aligned}$$

$$R_1^0 = C_1 = 1 \leq T_1$$

□

- (c) (2 Punkte) Kann das Taskset S , aufgrund der Resultate der obigen Überprüfungen, mit einem RM Scheduler in Echtzeit abgearbeitet werden?

Lösungsvorschlag:

Ja!

□

- (d) (2 Punkte) Warum berechnet man oft zuerst die hinreichende Bedingung für die Rate Monotonic Ablaufplanung, obwohl die Berechnung der notwendigen und hinreichenden Bedingung alleine reichen würde?

Lösungsvorschlag:

Die Überprüfung der hinreichenden Bedingung ist viel einfacher, und rechnerisch weniger intensiv. Dieser kleinere Rechenaufwand ist oft auch in Systemen mit Online-Admittance-Tests von Bedeutung.

□

- (e) (3 Punkte) Zeichnen Sie den Ablaufplan des Tasksets S mit RM-Scheduling für das Zeitintervall $t \in [0..40]$ in die vorbereitete Abbildung 4 ein.

Lösungsvorschlag:

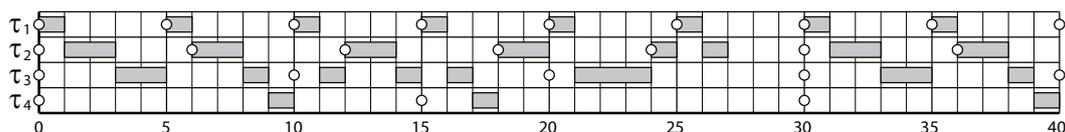


Abbildung 1: RM Ablaufplan.

□

- (a) (2 Punkte) Überprüfen Sie mit der notwendigen und hinreichenden Bedingung für die Earliest Deadline First (EDF) Ablaufplanung, ob das Taskset S mit einem EDF Scheduler in Echtzeit abgearbeitet werden kann.

Lösungsvorschlag:

$$U = \sum_{i=1}^4 \frac{C_i}{D_i} = 0.9 \leq 1$$

□

- (b) (8 Punkte) Zeichnen Sie den Ablaufplan des Tasksets S mit EDF-Scheduling für das Zeitintervall $t \in [0..40]$ in die vorbereitete Abbildung 4 ein. Bei Tasks mit gleicher Deadline wird zuerst der Task mit der kleinsten Periode abgearbeitet, unter dem Vorbehalt dass keine unnötigen Kontext-Switches stattfinden.

Lösungsvorschlag:

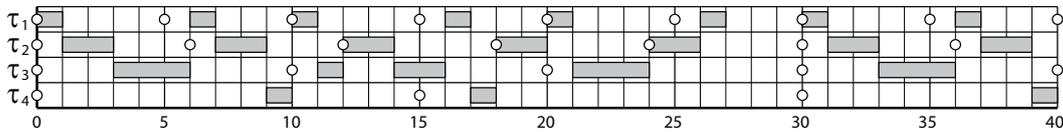


Abbildung 2: EDF Ablaufplan.

□

1.3: Time Triggered Cyclic Executive Scheduler

(maximal 15 Punkte)

- (a) (5 Punkte) Berechnen Sie eine mögliche Framelänge f für einen Time Triggered Cyclic Executive Scheduler (TT) der benutzt werden kann um das Taskset S abzuarbeiten. Berücksichtigen Sie dabei alle in der Vorlesung vorgestellten Bedingungen an die Framelänge f , und zeigen Sie, dass die von Ihnen gewählte Framelänge alle diese Bedingungen erfüllt.

Lösungsvorschlag:

$$f \leq T_i \quad \forall i \quad \Rightarrow f \leq 5$$

$$\exists i : T_i \bmod f = 0 \quad \Rightarrow f \in \{2, 3, 5\}$$

$$f \geq C_i \quad \forall i \quad \Rightarrow f \in \{3, 5\}$$

$$2f - \gcd(T_i, f) \leq D_i \quad \forall i \quad \Rightarrow f \in \{3\}$$

$$\Rightarrow f = 3$$

□

- (b) (1 Punkte) Berechnen Sie die minimal mögliche Periode P des TT-Schedulers für das Taskset S mit der von Ihnen berechneten Framelänge f .

Lösungsvorschlag:

$$P_{min} = \text{lcm}(T_i) = \text{lcm}(5, 6, 10, 15) = 30$$

□

- (c) (9 Punkte) Der zyklische Ablaufplan des Time Triggered Cyclic Executive Scheduler soll nun so erstellt werden, dass die durchschnittliche Antwortzeit für die einzelnen Tasks minimiert wird. Dabei soll das Minimieren der Antwortzeit eines Tasks mit kleinerer Periode absolut Priorität haben vor dem Minimieren der Antwortzeit eines Tasks mit grösserer Periode. Beachten Sie, dass die Tasks hier nicht unterbrochen werden sollen. Zeichnen Sie den resultierenden Ablaufplan des Tasksets S mit TT-Scheduling für das Zeitintervall $t \in [0..40]$ in die vorbereitete Abbildung 4 ein.

Lösungsvorschlag:

□

Task 1.3c and its solution was constructed with an earlier formulation of the cyclic executive scheduler. The question and its solution is no longer valid with the current formulation. Note inserted on 14.08.2017

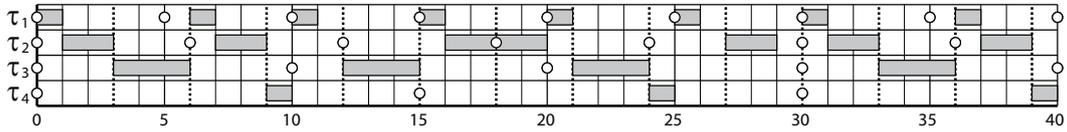


Abbildung 3: TT Ablaufplan.

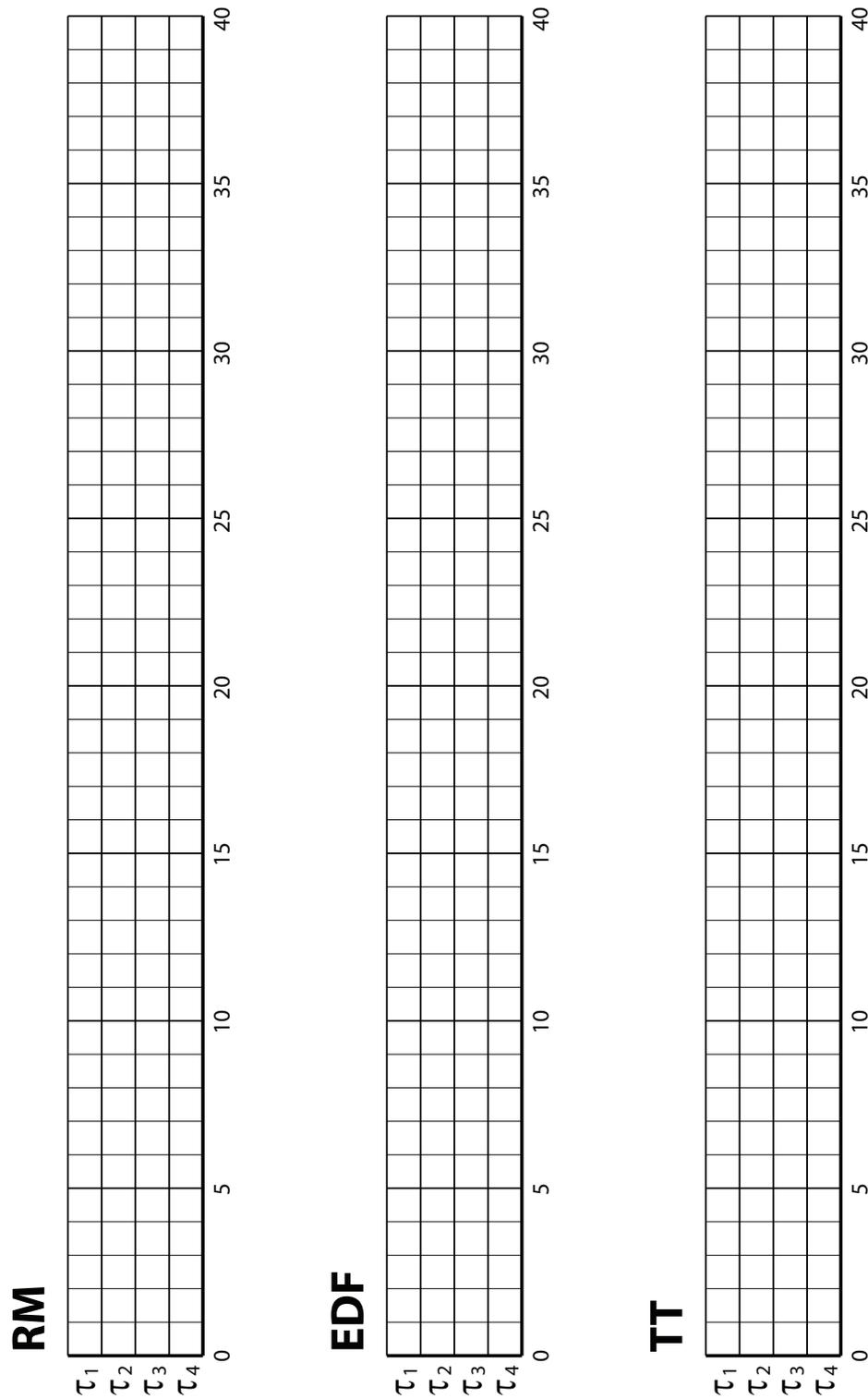


Abbildung 4: Ablaufpläne.

Aufgabe 2 : Resource sharing

(maximal 30 Punkte)

2.1: Allgemeine Fragen

(maximal 8 Punkte)

- (a) (3 Punkte) Erklären Sie den Unterschied zwischen privaten, geteilten und exklusiven Ressourcen und beschreiben Sie an einem Beispiel die auftretenden Probleme bei exklusiven Ressourcen.

Lösungsvorschlag:

Private Ressourcen werden ausschliesslich von einem Prozess/Thread benutzt, wohingegen geteilte Ressourcen von mindestens zwei u. U. konkurrierenden Prozessen/Threads benutzt werden können. Bei gleichzeitiger Nutzung einer Ressource durch mehrere Prozesse/Threads ohne weitere Gegenmassnahmen können z. B. Dateninkonsistenzen auftreten, oder unleserliche Ausgaben auf Drucker oder Monitor entstehen. Solche Ressourcen werden auch exklusive Ressourcen genannt, da eigentlich immer nur ein Prozess/Thread exklusiv darauf zugreifen sollte.

□

- (b) (3 Punkte) Nennen Sie drei Möglichkeiten, wie das beim Resource Sharing auftretende Problem des gegenseitigen Ausschlusses (mutual exclusion) gelöst werden kann.

Lösungsvorschlag: Non-preemptive Tasks, disabled Interrupts, statisches Scheduling, Einsatz von Semaphoren. (je 1 Punkt, maximal 3 Punkte)

□

- (c) (2 Punkte) Erläutern Sie die Anwendung einer Lösungsmöglichkeit aus Aufgabenteil c) genauer (3-4 Sätze).

Lösungsvorschlag: Ein (binärer) Semaphor s ist eine Datenstruktur, auf die nur über die atomaren Aufrufe $\text{wait}(s)$, bzw. $P(s)$, und $\text{signal}(s)$, bzw. $V(s)$, zugegriffen werden kann. Möchte ein Prozess A einen kritischen Abschnitt betreten, führt er ein $\text{wait}(s)$ auf dem Semaphor aus. Damit meldet er den Anspruch auf Eintritt in den kritischen Abschnitt an. Ist der Semaphor frei, erhält der Prozess diesen, belegt ihn, betritt den kritischen Abschnitt und gibt mit $\text{signal}(s)$ den Semaphor wieder frei, nachdem er seinen kritischen Abschnitt verlassen hat. Ist beim Aufruf von $\text{wait}(s)$ der Semaphor durch einen anderen Prozess B belegt, so wartet A bis B den Semaphor wieder freigibt.

□

2.2: Priority Inversion

(maximal 22 Punkte)

- (a) (2 Punkte) In der Vorlesung haben Sie das Priority Inheritance Protokoll (PIP) kennengelernt. Löst das PIP das Problem von Deadlocks? Geben Sie eine kurze Begründung für ihre Antwort.

Lösungsvorschlag:

Nein, das PIP kann Deadlocks nicht verhindern, wie zum Beispiel die ES-Folie 5-16 vom 27.04.2005 zeigt. (1 Punkt für die Antwort, 1 Punkt für das Beispiel/die Begründung; andere korrekte Beispiele bekommen natürlich ebenfalls die volle Punktzahl)

□

- (b) (14 Punkte) Gegeben sind die vier Tasks J_1, J_2, J_3 und J_4 . Die unten stehende Tabelle enthält Angaben zu ihren Ankunftszeiten, Deadlines, ihrer Ausführungszeit sowie den Prioritäten. Die Tasks mit festen Prioritäten sollen von einem Prozessor mittels prioritätsbasiertem Scheduling möglichst innerhalb ihrer Deadlines abgearbeitet werden (fixed priority scheduling).

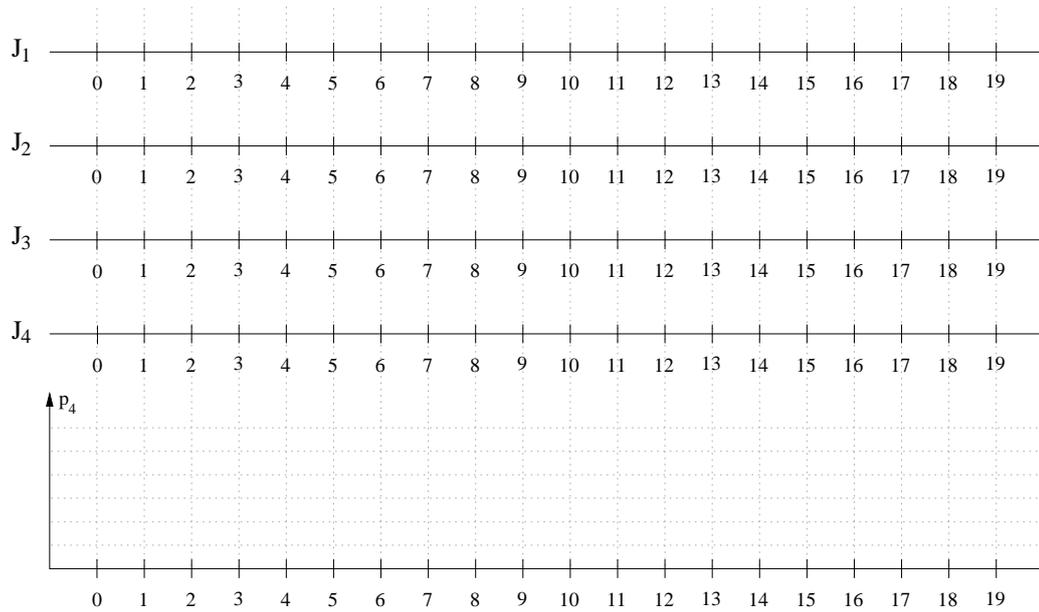
Die letzte Zeile der Tabelle bedarf zuvor einer Erklärung. Die Zeichnungen in der letzten Zeile der Tabelle stellen Informationen zum Ablauf der einzelnen Tasks dar. Jeder Task beinhaltet einen oder mehrere kritische Abschnitte (critical sections), in denen der Task auf die beiden exklusiven Ressourcen A bzw. B zugreift. In den Zeichnungen hat jeder Block die Länge Eins, nicht-kritische Abschnitte sind schattiert dargestellt. Die kritischen Abschnitte sind mit den entsprechenden Buchstaben A und B markiert. Zum Beispiel beinhaltet der sieben Zeiteinheiten lange Task J_4 zwei kritische Abschnitte. Zum einen den fünf Zeitabschnitte langen Abschnitt A und zum anderen den drei Zeiteinheiten langen Abschnitt B . A und B sind hier geschachtelt, d. h. während der Zeitabschnitte 3 bis 5 muss J_4 sowohl auf Ressource A als auch auf Ressource B zugreifen können.

	J_1	J_2	J_3	J_4
Ankunftszeit	6	5	3	0
Deadline	12	13	16	16
Ausführungszeit	3	3	3	7
nominale Priorität	höchste	mittlere	mittlere	niedrigste
Struktur der Tasks	■ A ■	■ A ■	■ B ■	■ A AB AB AB A ■

Aufgabenstellung: Erstellen Sie nun eine Ablaufplanung mit dem aus der Vorlesung bekannten Priority Inheritance Protokoll.

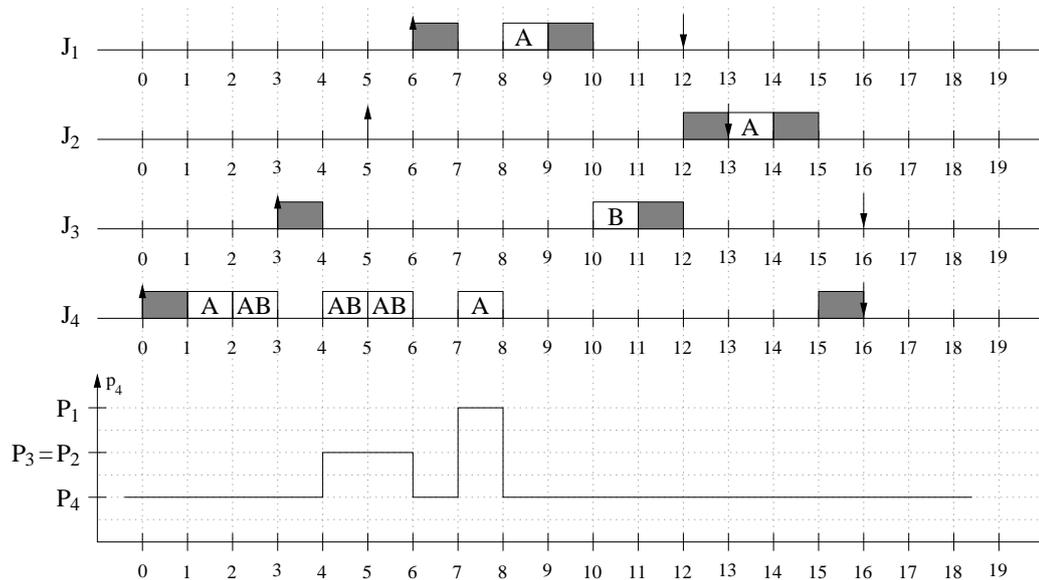
Füllen Sie dazu das unten vorbereitete Diagramm aus. Markieren Sie die kritischen Abschnitte der Tasks entsprechend mit den Buchstaben A und B wie in der letzten Zeile der Tabelle vorgegeben. Bitte zeichnen Sie zudem den Verlauf der aktiven Priorität p_4 von Task J_4 in das vorgegebene Diagramm.

Hinweis: Achten Sie genau auf die Prioritäten der einzelnen Tasks!



Lösungsvorschlag:

Die Schwierigkeit bei der Lösung der Aufgabe wird wohl darin liegen, dass in dem obigen Beispiel die Tasks J_2 und J_3 dieselbe Priorität haben. Aus diesem Grund ist extra das zusätzliche Ausfüllen des p_4 -Verlaufs gefordert. Tasks gleicher Priorität werden nach Folie 5-12 vom 27.4.2005 nach dem FCFS-Prinzip bearbeitet (4x3 Punkte für die ersten vier Diagramme, 2 Punkte für das p_4 -Diagramm).



□

- (c) (1 Punkt) Werden durch die Anwendung des PIP alle Deadlines eingehalten? Wenn nicht, wie gross ist die maximale Verspätung (maximum lateness) in Zeiteinheiten?

Lösungsvorschlag:

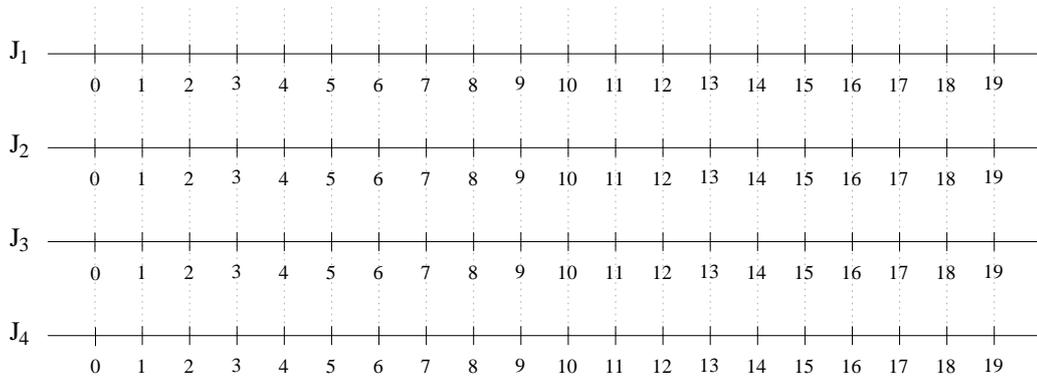
Nein, J_2 verletzt seine Deadline um zwei Zeiteinheiten.

(0 Punkte für geratene Antworten, wenn also die Antwort nicht aus b) hervorgeht; ist nur die Lateness durch b) falsch, aber die Antwort „nein“, so gibt es 0.5 Punkte)

□

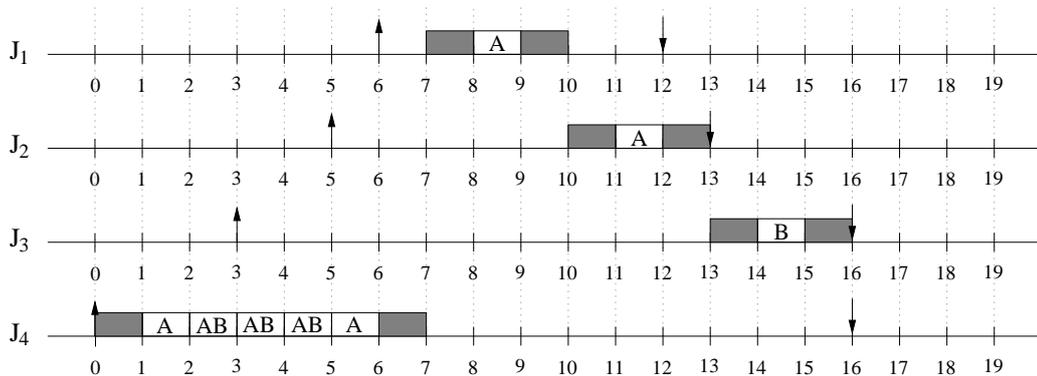
- (d) (5 Punkte) Gibt es einen Ablaufplan für die angegebenen Tasks J_1 bis J_4 , der alle Deadlines einhält? Welche Prioritätsänderungen müssten dazu vorgenommen werden?

Falls es einen korrekten Ablaufplan gibt, tragen Sie ihn bitte in die unten vorbereitete graphische Darstellung ein. Begründen Sie ansonsten, warum es keinen korrekten Ablaufplan geben kann.



Lösungsvorschlag:

Ja, es gibt einen Ablaufplan, der alle Deadlines einhält (1 Punkt). Allerdings muss man dafür die Priorität mindestens eines Tasks ändern. Wenn PIP alle Deadlines einhalten soll, wäre z. B. die Prioritätsreihenfolge $J_4 > J_1 > J_2 > J_3$ eine gute Wahl (1 Punkt):



(korrekter Ablaufplan: 3 Punkte)

□

Aufgabe 3 : Low Power Design

(maximal 55 Punkte)

Es gelten die Formeln aus der Vorlesung für Leistungsverbrauch P und Verzögerung τ einer CMOS Schaltung. Die Schwellspannung V_t sei stets vernachlässigbar klein gegenüber der Spannung V_{dd} .

3.1: Dynamic Voltage Scaling (DVS)

(maximal 10 Punkte)

- (a) (2 Punkte) Beschreiben Sie kurz, welche Arten von Verlustleistung bei CMOS-Schaltungen allgemein auftreten. Welche Verlustleistungsart soll mit DVS minimiert werden?

Lösungsvorschlag:

- statische und dynamische Verlustleistung
- mit DVS soll der dynamische Teil der Verlustleistung minimiert werden

□

- (b) (2 Punkte) Zur Abarbeitung einer Aufgabe werden N Zyklen benötigt. Würden Sie die Taktfrequenz f unter sonst gleichen Bedingungen erhöhen oder absenken, um den Energieverbrauch zu minimieren?

Lösungsvorschlag:

- weder eine Erhöhung noch eine Erniedrigung der Taktfrequenz f beeinflusst den Energieverbrauch der Task
- der Energieverbrauch ist unabhängig von f

□

- (c) (4 Punkte) Ein Prozessor wird mit einer Spannung $V_{dd} = 1.5$ V und der maximal möglichen Frequenz $f_{max,1.5V} = 600$ MHz betrieben. Nun soll V_{dd} auf 1.1 V abgesenkt und der Prozessor mit der nun maximal möglichen Frequenz $f_{max,1.1V}$ betrieben werden. Auf welchen Betrag P' wird durch diese Massnahme der ursprüngliche Leistungsverbrauch P reduziert?

Lösungsvorschlag:

$$\frac{V'_{dd}}{V_{dd}} = 0,7\bar{3}$$

$$f_{max} \propto V_{dd}$$

$$P \propto V_{dd}^3, \quad \frac{P'}{P} = 0,7\bar{3}^3 \approx 0,394 \quad \text{und} \quad P' \approx 0,394P$$

□

- (d) (2 Punkte) Nennen Sie 2 Vorteile, welche sich durch das Absenken der Spannung V_{dd} ergeben.

Lösungsvorschlag:

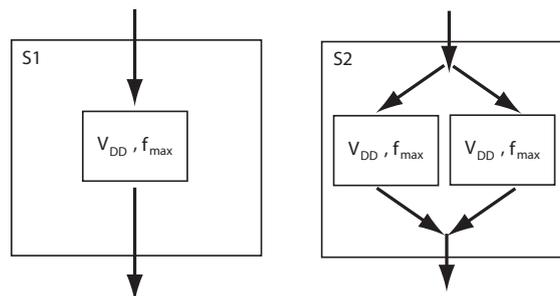
- reduzierter Energieverbrauch
- verlängerte Lebenszeit (z.B. batteriebetriebene, mobile Systeme)
- niedrigere Temperaturen
- reduzierter Kühlungsbedarf. . .

□

3.2: Parallelität

(maximal 20 Punkte)

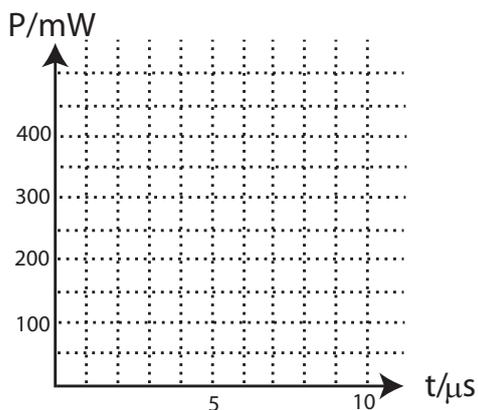
Es sollen zwei Prozessorsysteme S1 und S2 betrachtet werden (siehe Abbildung). System S1 besteht aus einem Prozessor; System S2 ist eine Parallelschaltung zweier Prozessoren vom Typ S1.



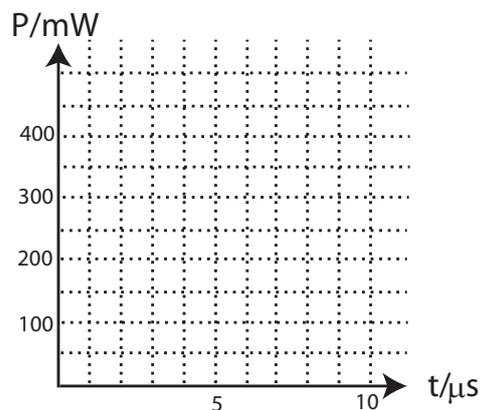
Beide Systeme beginnen zum Zeitpunkt $t = 0$ einen Task A bestehend aus $N = 1000$ Zyklen abuarbeiten, wobei bei System S2 jeder Prozessor exakt $N/2 = 500$ Zyklen zugeteilt bekommt.

- (a) (10 Punkte) Zunächst arbeiten beide Prozessorsysteme bei $V_{dd,1} = 1,5 \text{ V}$ und der bei dieser Spannung maximalen Frequenz $f_{max} = 200 \text{ MHz}$. Der Leistungsverbrauch eines Prozessors vom Typ S1 beträgt unter diesen Bedingungen 200 mW . Skizzieren Sie die Gesamtleistung P der Systeme S1 und S2 bei der Bearbeitung von Task A in den Diagrammen (a) und (c).

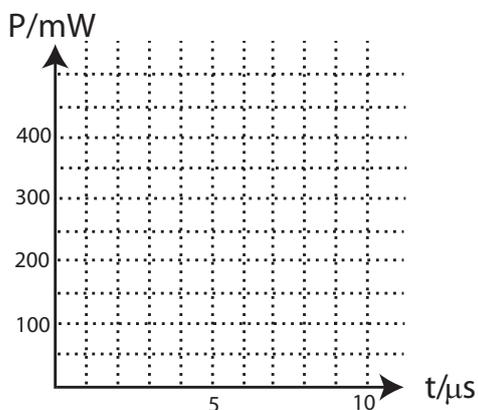
Nun wird die Spannung auf $V_{dd,2} = 0,75 \text{ V}$ reduziert. Vervollständigen Sie die Diagramme (b) und (d) unter der Annahme, dass beide Systeme nach der Spannungsreduktion erneut mit der maximal möglichen Frequenz betrieben werden.



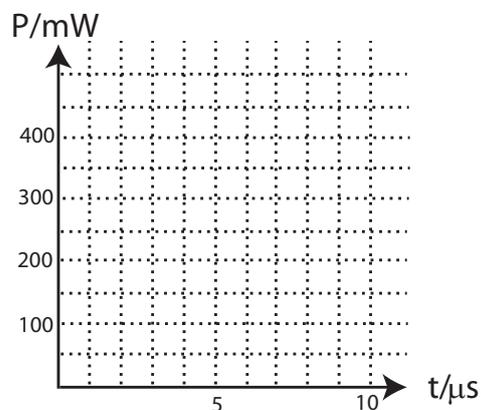
(a) Konfiguration : S1 und $V_{dd,1} = 1,5 \text{ V}$



(b) Konfiguration : S1 und $V_{dd,2} = 0,75 \text{ V}$

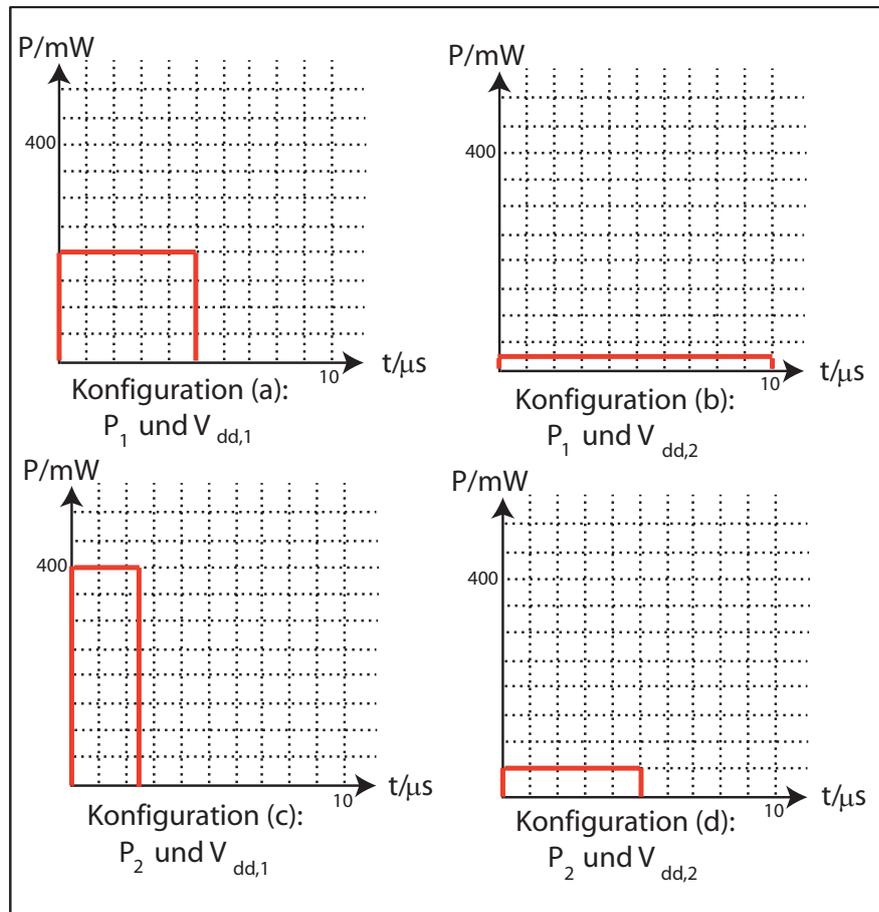


(c) Konfiguration : S2 und $V_{dd,1} = 1,5 \text{ V}$



(d) Konfiguration : S2 und $V_{dd,2} = 0,75 \text{ V}$

Lösungsvorschlag:



□

- (b) (8 Punkte) Als w wird die Zeitdauer bezeichnet, die vom Start bis zur Beendigung einer Task verstreicht. Zwei weitere wichtige Metriken sind das

Energy-Delay Produkt EDP, d.h. das Produkt aus verbrauchter Energie E und Bearbeitungsdauer w und das

Power-Delay Produkt PDP, d.h. das Produkt aus Leistung P und Bearbeitungsdauer w .

Vervollständigen Sie folgende Tabelle mit +, – und = Zeichen, indem Sie die jeweiligen Metriken der Konfigurationen (b)(c)(d) mit der Ausgangskonfiguration (a) vergleichen.

Verwenden Sie das Zeichen ...

+ , falls sich der Wert der Metrik gegenüber Konfiguration (a) vergrößert hat.

– , falls sich der Wert der Metrik gegenüber Konfiguration (a) verkleinert hat.

= , falls die jeweilige Konfiguration den gleichen Metrikwert wie Konfiguration (a) besitzt.

	<i>w</i>	PDP	EDP
Konfiguration (b)			
Konfiguration (c)			
Konfiguration (d)			

Lösungsvorschlag:

	<i>w</i>	PDP	EDP
Konfiguration (b)	+	–	–
Konfiguration (c)	–	=	–
Konfiguration (d)	=	–	–

□

(c) (2 Punkte) Welche Konfiguration ist Ihrer Meinung nach optimal bezüglich oben genannter Metriken? Begründen Sie Ihre Meinung. Nennen Sie 3 mögliche Nachteile dieser optimalen Konfiguration gegenüber Konfiguration (a).

Lösungsvorschlag:

Die optimale Konfiguration ist (d). Sie ist

- (b) vorzuziehen, weil eine kürzere Bearbeitungszeit mit gleicher Energie möglich ist.
- (c) vorzuziehen aufgrund des geringeren EDP-Wertes.

Nachteile der optimalen Konfiguration:

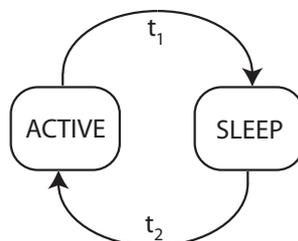
- erhöhte Komplexität
- erhöhte Kosten
- erhöhte Flächenbedarf (bzw. Grösse)
- erhöhter Energieverbrauch durch Parallelisierung. . .

□

3.3: Dynamic Power Management (DPM)

(maximal 25 Punkte)

Gegeben ist ein Microcontroller vom Typ TI-MSP430@1MHz, der im aktiven Zustand $P_{active} = 1,2 \text{ mW}$ oder im schlafenden Zustand $P_{sleep} = 90,0 \text{ } \mu\text{W}$ verbraucht. Im SLEEP-Zustand findet eine ständige Überwachung auf mögliche Interrupts statt und es wird gegebenenfalls ein Wechsel in den ACTIVE-Zustand eingeleitet. Der Wechsel vom SLEEP-Zustand in den ACTIVE-Zustand dauert t_2 ; der Wechsel vom ACTIVE-Zustand in den SLEEP-Zustand dauert t_1 (siehe Abbildung). Während der Übergänge ist der Microcontroller nicht funktionsbereit. Nehmen Sie vereinfachend an, dass sich die Leistung während der Übergänge in stetiger, linearer Weise ändert. Für die Energieversorgung wird eine Energiequelle mit $E_{bat} = 25,0 \text{ kJ}$ eingesetzt.



- (a) (10 Punkte) Vernachlässigen Sie zunächst die Umschaltzeiten ($t_1=t_2=0$). Es bezeichnet t_{active} die Summe aller Zeitintervalle, in denen sich das System während seiner Lebensdauer im Zustand ACTIVE befindet. Um wieviel Prozent muss man den maximalen Wert von t_{active} mindestens reduzieren, um den Microcontroller mindestens ein Jahr lang (d.h. 365 Tage) betreiben zu können?

Lösungsvorschlag:Maximaler Wert von t_{active} :

$$t_{active,max} = \frac{E_{bat}}{P_{active}} \approx 241,1 \text{ Tage}$$

Für die Zeitdauer eines Jahres (t_{year}) muss für den prozentualen Zeitanteil a des Prozessors im ACTIVE-Zustand gelten:

$$t_{year} [P_{active}a + P_{sleep}(1 - a)] \leq E_{bat}$$

$$a \leq \frac{\frac{E_{bat}}{t_{year}} - P_{sleep}}{P_{active} - P_{sleep}} \approx 0,6331 \Rightarrow t_{active} \leq 0,6331 \cdot 365 \cdot 24 \cdot 60 \cdot 60 \text{ s} \approx 231,1 \text{ Tage}$$

\Rightarrow Reduktion von t_{active} um mindestens 4,17 %

□

- (b) (15 Punkte) Es gilt nun $t_1 = 90 \text{ } \mu\text{s}$ und $t_2 = 120 \text{ ms}$. Wie lange sollte die Zeitspanne Δt_{min} nach dem Verlassen des Zustands ACTIVE bis zum ersten Interrupt mindestens sein, damit sich das Umschalten in den SLEEP-Zustand lohnt?

Hinweis: Wenn die Energieeinsparung grösser ist als der Energieverlust durch den Zustandswechsel, lohnt sich das Umschalten!

Lösungsvorschlag:

$$E_{\text{Overhead}} = P_{\text{sleep}}t_2 + \frac{1}{2}(P_{\text{active}} - P_{\text{sleep}})t_2 \approx 10,8\mu\text{J} + 66,6\mu\text{J} = 77,4\mu\text{J}$$

$$E_{\text{Gewinn}} = \frac{1}{2}(P_{\text{active}} - P_{\text{sleep}})t_1 + (\Delta t_{\text{min}} - t_1)(P_{\text{active}} - P_{\text{sleep}})$$

$$E_{\text{Gewinn}} \geq E_{\text{Overhead}} \quad \Rightarrow \quad \Delta t_{\text{min}} \approx 69,8\text{ms}$$

□

Aufgabe 4 : Synthese

(maximal 50 Punkte)

4.1: LIST Scheduling

(maximal 25 Punkte)

In dieser Aufgabe soll für die folgende Funktion mit dem List-Algorithmus ein gültiger Ablaufplan gefunden werden.

```
func (var a, var b, var c, var d, var *x, var *y, var *z){
    x = (a +1 b) +2 a *3 (c +4 d) -5 (d -6 c);
    y = (a -7 b) +8 c *9 d;
    z = a +10 b *11 c;
}
```

(Notation: +₄ : Index dieser Addition ist 4)

Die Zielarchitektur besteht aus drei Recheneinheiten. Additionen und Subtraktionen werden auf zwei Einheiten vom Typ r_1 und Multiplikationen auf einer Einheit vom Typ r_2 ausgeführt. Dabei benötigen Addition und Subtraktion eine Zeiteinheit und die Multiplikation zwei Zeiteinheiten.

- (a) (8 Punkte) Zeichnen Sie den azyklischen Sequenzgraphen $G_S = (V_S, E_S)$ und den Ressourcengraphen G_R . Vertauschen Sie dabei nicht die Reihenfolge der Operationen und nummerieren Sie die Knoten gemäss Index (+₄ → v_4).

Lösungsvorschlag: Siehe Abbildung 1.

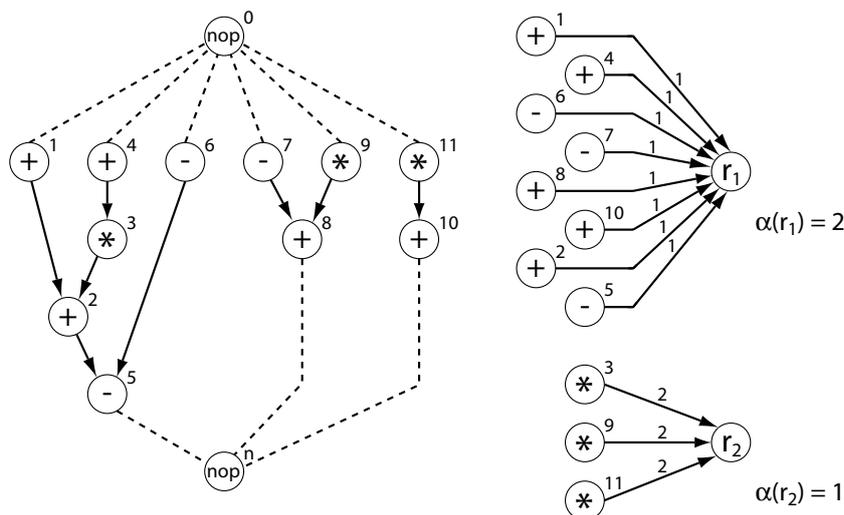


Abbildung 1: Lösungsvorschlag: Sequenz- und Ressourcengraph für Aufgabe 4.1.a)

- (b) (5 Punkte) Bestimmen Sie die Mobilität aller Operationen, indem Sie die ASAP und ALAP Ablaufpläne aufzeichnen. Benutzen sie die Latenz vom ASAP als Vorgabe für den ALAP Algorithmus. Nehmen Sie an, dass die erste Operation zum Zeitpunkt $t = 0$ startet und dass der *nop* Hierarchieknoten keine Zeit beansprucht.

Lösungsvorschlag: Aus der Differenz der Startzeitpunkte im ASAP und ALAP Ablaufplan (siehe Abbildung 2) ergeben sich folgende Mobilitäten:

	ν_1	ν_2	ν_3	ν_4	ν_5	ν_6	ν_7	ν_8	ν_9	ν_{10}	ν_{11}
μ	2	0	0	0	0	3	3	2	2	2	2

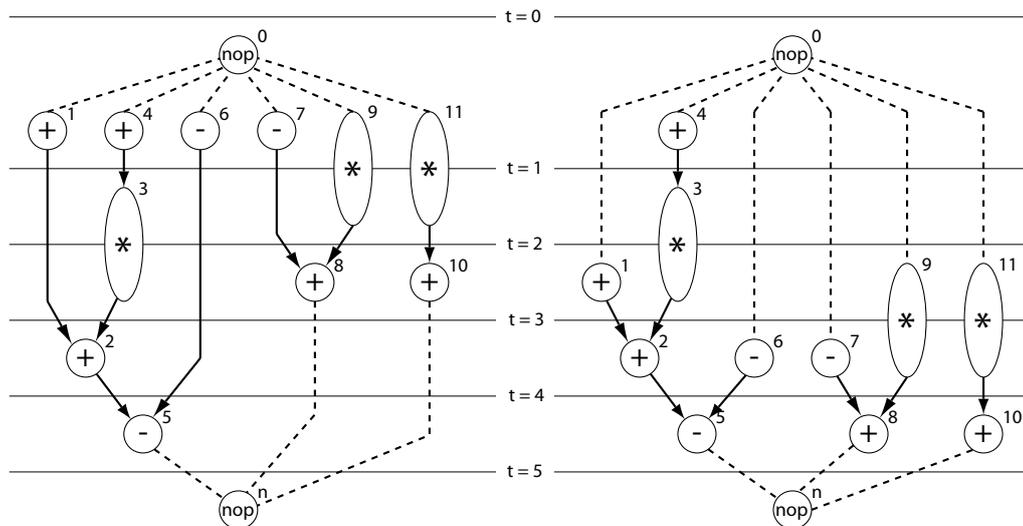


Abbildung 2: **Lösungsvorschlag:** ASAP und ALAP Ablaufpläne für Aufgabe 4.1.b)

- (c) (10 Punkte) Ermitteln Sie den Ablaufplan mit dem List-Algorithmus. Verwenden Sie für die Priorität der Operationen die vorher ermittelte Mobilität (kleinere Mobilität = grössere Priorität). Benutzen Sie dazu die Tabelle 1. Tragen Sie für den Zeitschritt t und den Ressourcentyp k jeweils die Kandidatenmenge $K_{t,k}$, die Menge der laufenden Operationen $G_{t,k}$ sowie die zum Starten selektierten Operationen $S_{t,k}$ ein. Wie gross ist die resultierende Latenz?

Lösungsvorschlag: Siehe Tabelle 1. Die Resultierende Latenz ist 7.

- (d) (2 Punkte) Liefert das List-Scheduling stets eine minimale Latenz für solche Probleme? Begründen Sie kurz Ihre Antwort. Nennen Sie eine andere Methode, die Sie in der Vorlesung kennengelernt haben, welche eine optimale Lösung für solche Probleme liefert.

Lösungsvorschlag: Nein, denn der List-Algorithmus ist eine Heuristik. Eine Transformation des Problems in ein ganzzahliges lineares Programm (ILP) würde eine optimale Lösung ergeben.

t	k	$K_{t,k}$	$G_{t,k}$	$S_{t,k}$
0	r_1	$\nu_1, \nu_4, \nu_6, \nu_7$	—	ν_1, ν_4
	r_2	ν_9, ν_{11}	—	ν_9
1	r_1	ν_6, ν_7	—	ν_6, ν_7
	r_2	ν_{11}, ν_3	ν_9	—
2	r_1	ν_8	—	ν_8
	r_2	ν_{11}, ν_3	—	ν_3
3	r_1	—	—	—
	r_2	ν_{11}	ν_3	—
4	r_1	ν_2	—	ν_2
	r_2	ν_{11}	—	ν_{11}
5	r_1	ν_5	—	ν_5
	r_2	—	ν_{11}	—
6	r_1	ν_{10}	—	ν_{10}
	r_2	—	—	—

Tabelle 1: **Lösungsvorschlag:** für Aufgabe 4.1.c).

4.2: Timing Constraints

(maximal 10 Punkte)

Gegeben ist der Sequenzgraph in Abbildung 3 und die Ausführungszeiten der Funktionen in Tabelle 2.

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
3	3	4	2	3	3	2	1	1	1

Tabelle 2: Ausführungszeiten der Funktionen

- (a) (1 Punkt) Zeichnen Sie den Beschränkungsgraphen (Weighted Constraint Graph) $G_C = (V_C, E_C, d)$.

Lösungsvorschlag: Siehe Abbildung 4.

- (b) (8 Punkte) Fügen Sie folgende zusätzliche Zeitbeschränkungen in G_C ein:

Lösungsvorschlag:

Eine Kante $(\nu_i, \nu_j) \in G_C$ mit Gewicht $d(\nu_i, \nu_j)$ bedeutet: $\tau(\nu_j) - \tau(\nu_i) \geq d(\nu_i, \nu_j)$.
Siehe Abbildung 5.

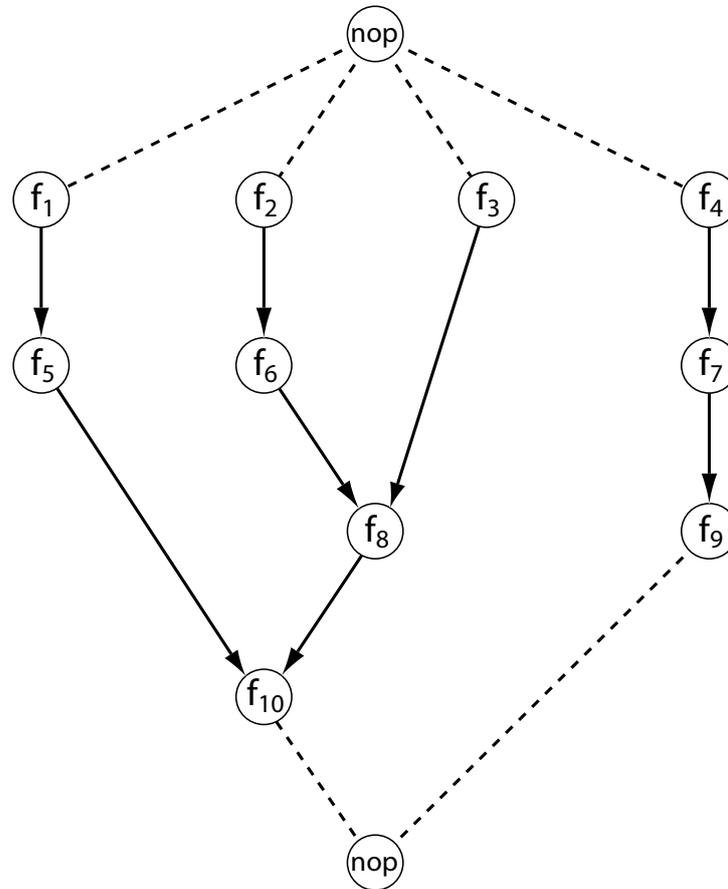


Abbildung 3: Sequenzgraph

- i) f_7 darf frühestens 3 Zeitschritte nach f_3 starten.

Lösungsvorschlag: $\tau(\nu_7) \geq \tau(\nu_3) + 3 \Rightarrow \tau(\nu_7) - \tau(\nu_3) \geq 3$

- ii) f_3 muss genau einen Zeitschritt nach f_2 starten.

Lösungsvorschlag:

$$\tau(\nu_3) = \tau(\nu_2) + 1 \Rightarrow \tau(\nu_3) - \tau(\nu_1) \geq 1 \wedge \tau(\nu_2) - \tau(\nu_3) \geq -1$$

- iii) Die Startzeitpunkte von f_5 und f_6 dürfen höchstens zwei Zeitschritte auseinanderliegen.

Lösungsvorschlag:

$$|\tau(\nu_5) - \tau(\nu_6)| \leq 2 \Rightarrow \tau(\nu_5) - \tau(\nu_6) \geq -2 \wedge \tau(\nu_6) - \tau(\nu_5) \geq -2$$

- iv) f_9 muss spätestens 4 Zeitschritte nach dem Beenden von f_2 starten.

Lösungsvorschlag: $\tau(\nu_9) \leq (\tau(\nu_2) + 3) + 4 \Rightarrow \tau(\nu_2) - \tau(\nu_9) \geq -7$

- (c) (1 Punkt) Existiert ein gültiger Ablaufplan unter den gegebenen Zeitbeschränkungen und unter der Annahme, dass beliebig viele Ressourcen zur Verfügung stehen? Begründen Sie Ihre Antwort.

Lösungsvorschlag: Ja, denn es existiert kein positiver Zyklus in G_C .

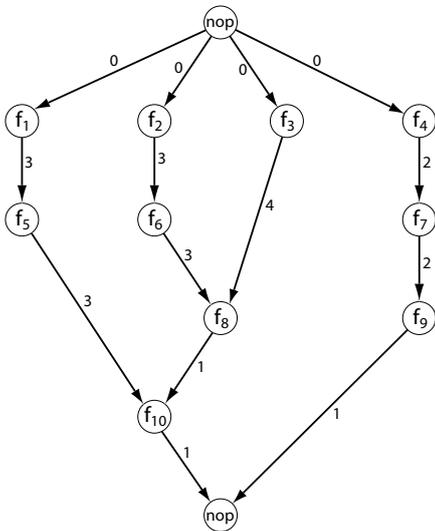


Abbildung 4: Lösungsvorschlag: Beschränkungsgraphen für Aufgabe 4.2.a)

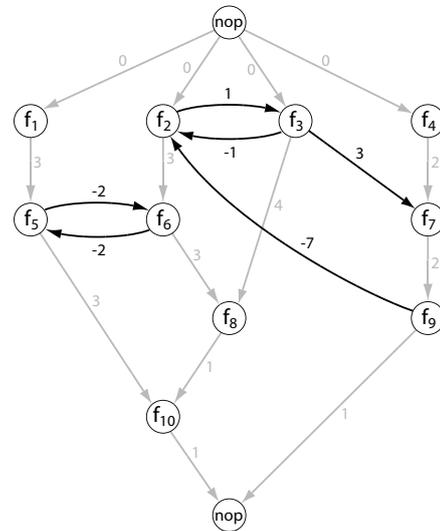


Abbildung 5: Lösungsvorschlag: Beschränkungsgraphen für Aufgabe 4.2.b)

4.3: Ganzzahlige Lineare Programme

(maximal 15 Punkte)

Gegeben sind die ASAP und ALAP Ablaufpläne (ohne Ressourcenbeschränkungen) für ein Problem (siehe Abb. 6).

Auf der Zielarchitektur gibt es zwei Recheneinheiten, eine für die Addition und eine für die Multiplikation. Formulieren Sie das Problem der Latenzminimierung unter Ressourcenbeschränkungen als ganzzahliges lineares Programm (ILP). Übernehmen Sie bei der Einführung der binären Variablen die Nummerierung aus Abb. 6. Nehmen Sie an, dass die erste Operation zum Zeitpunkt $t = 0$ startet.

Lösungsvorschlag:

- (1) Aus den gegebenen ASAP- und ALAP Ablaufpläne, lassen sich die frühesten (l_i) und spätesten (h_i) Startzeitpunkte der Tasks, sowie deren Ausführungszeit ($w(\nu_i)$) herauslesen:

i	l_i	h_i	$w(\nu_i)$
0	0	0	0
1	0	1	1
2	0	4	2
3	1	2	2
4	3	4	1
5	4	5	1
n	5	6	0

- (2) Einführung binärer Variablen:

$$x_{i,t} \in \{0, 1\}, \quad \forall \nu_i \in V_S, \quad \forall t : l_i \leq t \leq h_i$$

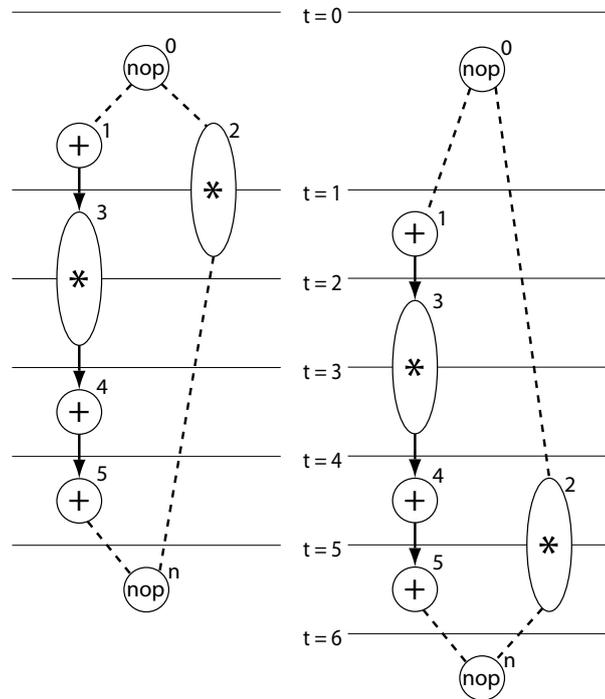


Abbildung 6: ASAP und ALAP Ablaufpläne

(3) Jeder Task startet nur einmal: $\sum_{t=l_i}^{h_i} x_{i,t} = 1, \quad \forall v_i \in V_S$

$$\begin{aligned} x_{1,0} + x_{1,1} &= 1 & x_{4,3} + x_{4,4} &= 1 \\ x_{2,0} + x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} &= 1 & x_{5,4} + x_{5,5} &= 1 \\ x_{3,1} + x_{3,2} &= 1 & x_{n,5} + x_{n,6} &= 1 \end{aligned}$$

(4) Berechnung der Startzeitpunkte aus $x_{i,t}$: $\sum_{t=l_i}^{h_i} t \cdot x_{i,t} = \tau(v_i)$

$$\begin{aligned} 0 \cdot x_{1,0} + 1 \cdot x_{1,1} &= \tau(v_1) & 3 \cdot x_{4,3} + 4 \cdot x_{4,4} &= \tau(v_4) \\ 0 \cdot x_{2,0} + 1 \cdot x_{2,1} + 2 \cdot x_{2,2} + 3 \cdot x_{2,3} + 4 \cdot x_{2,4} &= \tau(v_2) & 4 \cdot x_{5,4} + 5 \cdot x_{5,5} &= \tau(v_5) \\ 1 \cdot x_{3,1} + 2 \cdot x_{3,2} &= \tau(v_3) & 5 \cdot x_{n,5} + 6 \cdot x_{n,6} &= \tau(v_n) \end{aligned}$$

(5) Datenabhängigkeiten aus Sequenzgraph mit Berücksichtigung der Ausführungszeiten: $\tau(v_j) - \tau(v_i) \geq w(v_i) \quad \forall (v_i, v_j) \in E_S$

$$\begin{aligned} \tau(v_1) - \tau(v_0) &\geq 0 & \tau(v_5) - \tau(v_4) &\geq 1 \\ \tau(v_2) - \tau(v_0) &\geq 0 & \tau(v_n) - \tau(v_5) &\geq 1 \\ \tau(v_3) - \tau(v_1) &\geq 1 & \tau(v_n) - \tau(v_2) &\geq 2 \\ \tau(v_4) - \tau(v_3) &\geq 2 \end{aligned}$$

(6) Zu keinem Zeitpunkt werden mehr als die zur Verfügung stehenden Ressourcen benützt:

$$\sum_{\forall i: (v_i, r_k) \in E_R} \sum_{p=\max\{0, t-h_i\}}^{\min\{d_i-1, t-l_i\}} x_{i,t-p} \leq \alpha(r_k)$$

$$\forall r_k \in V_T, \quad \forall \min_{i=1}^{|V|} \{l_i\} \leq t \leq \max_{i=1}^{|V|} \{h_i\}$$

$$\begin{array}{llll} t = 0 : & x_{1,0} & \leq 1 \\ & x_{2,0} & \leq 1 \\ t = 1 : & x_{1,1} & \leq 1 \\ & x_{2,1} + x_{2,0} + x_{3,1} & \leq 1 \\ t = 2 : & & \\ & x_{2,2} + x_{2,1} + x_{3,2} + x_{3,1} & \leq 1 \\ t = 3 : & x_{4,3} & \leq 1 \\ & x_{2,3} + x_{2,2} + x_{3,2} & \leq 1 \\ t = 4 : & x_{4,4} + x_{5,4} & \leq 1 \\ & x_{2,4} + x_{2,3} & \leq 1 \\ t = 5 : & x_{5,5} & \leq 1 \\ & x_{2,4} & \leq 1 \end{array}$$

(7) Startbedingung: $\tau(\nu_0) = 0$

(8) Zielfunktion: minimize $\tau(\nu_n)$