Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik
und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Exam Spring 2010

Embedded Systems

Prof. L. Thiele

NOTE:

- Die beigefügte Lösung ist ein Vorschlag. Fur̈ Korrektheit, Vollständigkeit und Verständlichkeit wird keine Verantwortung ubernommen.

Aufgabe 1 : Echtzeitplanung (maximal 38 Punkte)

1.1: RM-Zeitplanung (maximal 5 Punkte)

Gegeben sei die folgende Menge periodischer Tasks, die mittels ratenmonotoner Ablaufplanung (RM) zeitlich eingeplant wurde:

	C_i	T_i	Φ_i
τ_1	1	5	0
τ_2	2.5	5	0.5

Φ_i ist der initiale Offset von Task τ_i. Task τ_1 hat eine höhere Priorität als τ_2.

Bestimmen Sie die exakte ungünstigste Antwortzeit von τ_2.
Hinweis: Die Aufgabe kann graphisch gelöst werden.

Lösungsvorschlag:
$R_2 = 3$.

□

1.2: Zeitplanung mit unveränderlichen Prioritäten (maximal 7 Punkte)

(a) (2 Punkte) Was ist der mittlere Arbeitsausnutzungsfaktor U (utilization) für die folgende mittels RM zeitlich eingeplante Task-Menge?

	C_i	T_i
τ_1	5	10
τ_2	1	10
τ_3	2	100

Lösungsvorschlag:
$U = 5/10 + 1/10 + 2/100 = 0.62$.

□

(b) (2 Punkte) Gegeben seien n periodische Tasks, die mittels RM zeitlich eingeplant wurden. Der Arbeitsausnutzungsfaktor dieser Task-Menge ist U_n. Es ist bekannt, dass $U_n > n(2^{1/n} - 1)$. Geben Sie an, ob die folgende Aussage wahr oder falsch ist: Die Task-Menge ist mittels RM auf jeden Fall *nicht* zeitlich einplanbar.

Lösungsvorschlag:
False

□

(c) (3 Punkte) Unterbrechbare Zeitplanung mit unveränderlichen Prioritäten (Fixed Priority Preemptive Scheduling – FPPS) ist eine Verallgemeinerung von RM, bei dem einem Task eine beliebige Priorität zugewiesen werden kann. Geben Sie an, ob die folgenden Aussagen wahr oder falsch sind:

(i) Angenommen, eine Task-Menge ist mittels FPPS zeitlich einplanbar, wobei $D_i = T_i$ gilt. Ist diese Task-Menge mittels RM zeitlich einplanbar?

Lösungsvorschlag:
Yes

□

(ii) Angenommen, eine Task-Menge ist mittels RM zeitlich einplanbar, wobei $D_i = T_i$ gilt. Ist diese Task-Menge bei einer beliebigen Prioritätszuordnung mittels FPPS zeitlich einplanbar?

Lösungsvorschlag:
No

□

1.3: Total Bandwidth Server (TBS) (maximal 8 Punkte)

Gegeben sei die folgende Menge periodischer Tasks:

	C_i	T_i
τ_1	1	6
τ_2	1	3
τ_3	3	10

Gegeben ist ein Total Bandwidth Server (TBS).

(a) (2 Punkte) Wie gross kann der Arbeitsausnutzungsfaktor U_s des TBS maximal sein, damit die obige Menge von Tasks zusammen mit dem TBS mittels EDF zeitlich einplanbar ist?

Lösungsvorschlag:
Maximum $U_s = 1 - 1/6 - 1/3 - 3/10 = 0.2$

□

(b) (6 Punkte) Nehmen Sie $U_s = 0.2$ an. Entwerfen Sie den EDF-Ablaufplan für den Fall, dass der TBS die folgenden drei *aperiodischen* Anfragen $J_4(a_4 = 2, C_4 = 1)$, $J_5(a_5 = 6, C_5 = 2)$ und $J_6(a_6 = 1, C_6 = 1)$ bedienen soll (a_4, a_5 und a_6 bezeichnen die Ankunftszeiten der drei Tasks). Benutzen Sie dazu Abbildung 1. Gehen Sie davon aus, dass die Ankunftszeit aller periodischen Tasks 0 ist.

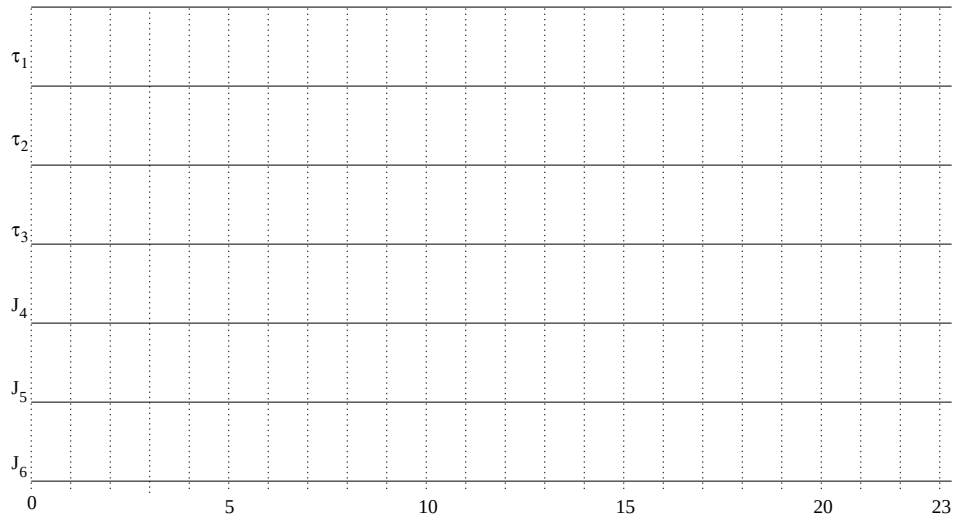


Abbildung 1: Ablaufplan für Aufgabe 1.3 (b)

Lösungsvorschlag:

Calculate deadlines first:

$d_{J_6} = \max 1, 0 + 1/0.2 = 6$ $d_{J_4} = \max 2, 6 + 1/0.2 = 11$ $d_{J_5} = \max 6, 11 + 2/0.2 = 21$

See Figure 2.

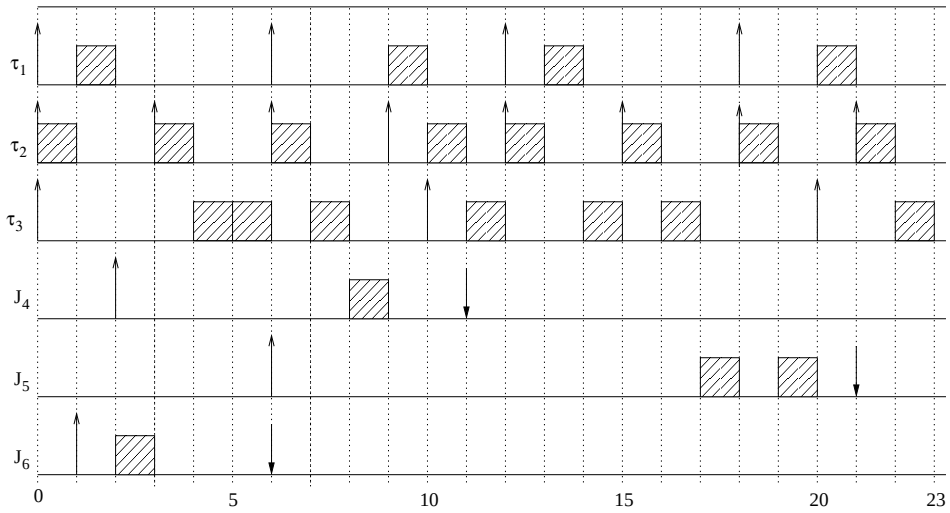


Abbildung 2: Schedule

□

1.4: Abfrageserver (Polling Server) (maximal 8 Punkte)

Gegeben sei die folgende Menge periodischer Tasks:

	C_i	T_i	Φ_i
τ_1	1	5	2
τ_2	1	5	2

Φ_i ist der initiale Offset von Task τ_i. Gehen Sie davon aus, dass ein Abfrageserver (Polling Server) mit den Parametern $T_s = 6$, $C_s = 2$ und $\Phi_s = 0$ zusammen mit dieser Menge von periodischen Tasks läuft.

Es werden zwei *aperiodische* Tasks τ_x ($C_x = 2$) und τ_y ($C_y = 1$) von dem Abfrageserver bedient. Die aperiodischen Tasks sind unterbrechbar, und die Priorität von τ_y ist höher als die von τ_x. Die Ankunftszeiten der Tasks τ_x und τ_y liegen beide zwischen 0 und 5, und sind nicht zwingend ganzzahlig: $a_x \in [0, 5]$, $a_y \in [0, 5]$, $a_x \in \mathbb{R}$, $a_y \in \mathbb{R}$.

Geben Sie den Ablaufplan mit der ungünstigsten Antwortzeit für τ_x an, und tragen Sie ihn in Abbildung 3 ein. Überlegen Sie sich dazu die ungünstigste Anordnung der Startzeitpunkte der beiden aperiodischen Tasks.

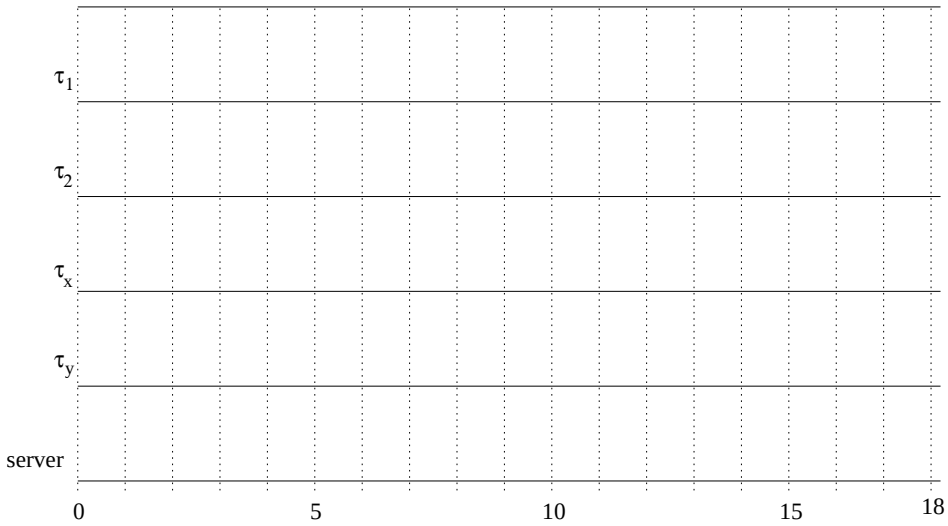


Abbildung 3: Ablaufplan für Aufgabe 1.4

Lösungsvorschlag:

See Figure 4.

□

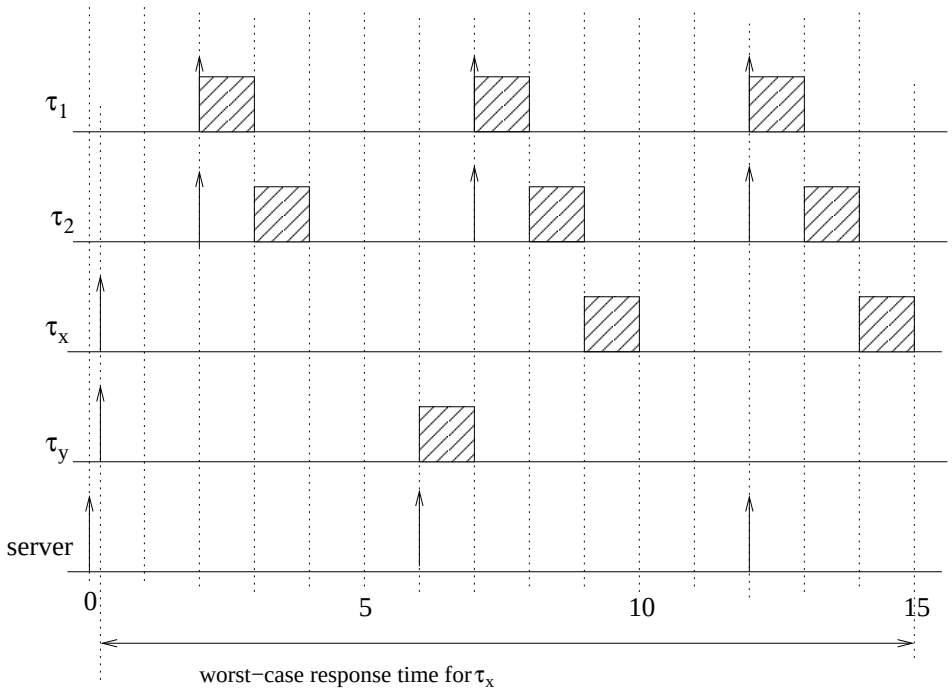


Abbildung 4: Schedule giving worst-case response time for τ_x

1.5: RM-Zeitplanung (maximal 10 Punkte)

Sind die folgenden vier periodischen Tasks mittels RM-Zeitplanung ausführbar?

	C_i	T_i
τ_1	1	5
τ_2	3	7
τ_3	2	11
τ_4	3	25

Lösungsvorschlag:

Utilization factor $U = 0.2 + 0.43 + 0.182 + 0.12 = 0.932 > (U_{lub} = 0.757)$. So, we need to apply the sufficient and necessary test.

$R_1 = 1$

$R_2 = 3 + \lceil 4/5 \rceil * 1 = 3 + 1 = 4$

$R_3 = 2 + \lceil 6/5 \rceil * 1 + \lceil 6/7 \rceil * 3 = 2 + 2 + 3 = 7$

$R_3 = 2 + \lceil 7/5 \rceil * 1 + \lceil 7/7 \rceil * 3 = 2 + 2 + 3 = 7$

$R_4 = 3 + \lceil 9/5 \rceil * 1 + \lceil 9/7 \rceil * 3 + \lceil 9/11 \rceil * 2 = 3 + 1 + 6 + 2 = 12$

$R_4 = 3 + \lceil 12/5 \rceil * 1 + \lceil 12/7 \rceil * 3 + \lceil 12/11 \rceil * 2 = 3 + 3 + 6 + 4 = 16$

$R_4 = 3 + \lceil 16/5 \rceil * 1 + \lceil 16/7 \rceil * 3 + \lceil 16/11 \rceil * 2 = 3 + 4 + 9 + 4 = 20$

$R_4 = 3 + \lceil 20/5 \rceil * 1 + \lceil 20/7 \rceil * 3 + \lceil 20/11 \rceil * 2 = 3 + 4 + 9 + 4 = 20$

□

Aufgabe 2 : Kommunikation und gemeinsame Nutzung von Ressourcen (maximal 32 Punkte)

2.1: Unterteilte und nicht-unterteilte Kommunikation (maximal 12 Punkte)

Stationen A, B und C benutzen das gleiche Kommunikationsmedium um Nachrichten auszutauschen. Eine Nachricht kann nur dann fehlerfrei ausgetauscht werden (d.h. ohne Kollision), wenn die sendende Station das Kommunikationsmedium für die gesamte Dauer des Sendevorgangs exklusiv zum Übertragen von Nachrichten benutzt.

Stationen A, B und C übertragen Nachrichten mit einer Länge von 2 Zeiteinheiten. Die Nachrichten werden von den Stationen periodisch zum Senden bereitgestellt. Die jeweiligen Perioden T, sowie der initiale Offset Φ sind wie folgt:

Station A: $T_A = 5$, $\Phi_A = 5$
d.h. Nachrichten werden zu den Zeitpunkten 5, 10, 15, ... bereitgestellt
Station B: $T_B = 6$, $\Phi_B = 6$
d.h. Nachrichten werden zu den Zeitpunkten 6, 12, 18, ... bereitgestellt
Station C: $T_C = 7$, $\Phi_C = 7$
d.h. Nachrichten werden zu den Zeitpunkten 7, 14, 21, ... bereitgestellt

Stationen benutzen das Kommunikationsmedium ohne Zugriffskontrolle. Im Falle einer Kollision werden Nachrichten *nicht* erneut übertragen. Es werden zwei Zugriffsstrategien benutzt:

Nicht-unterteilt Eine Station überträgt eine Nachricht sobald diese verfügbar wird.

Unterteilt Ab dem Zeitpunkt 0 ist die Zeit in gleich lange Zeitscheiben (slots) unterteilt, wobei die Länge einer Zeitscheibe einer einzelnen Nachricht entspricht (d.h. 2 Zeiteinheiten). Ist eine Nachricht verfügbar, wird diese von der Station zu Beginn der nächsten verfügbaren Zeitscheibe übertragen (d.h. zu den Zeitpunkten 0, 2, 4, 6, ...).

(a) (1 Punkt) Was ist die wesentliche Voraussetzung (hier implizit erfüllt) für die Verwendung von unterteilten Kommunikationsprotokollen?

Lösungsvorschlag:
In order to exploit a Slotted communication scheme Stations need to be synchronized, i.e., they need to have some sort of knowledge about a common time.

(b) (3 Punkte) Füllen Sie die Tabellen 1 und 2 aus. Markieren Sie, unter Verwendung von fortlaufenden Nachrichtennummern (z.B. 1, 2, 3, ...), für jede der drei Stationen die Zeiteinheiten, in denen die jeweilige Station Nachrichten versendet.

(c) (2 Punkte) Markieren Sie in den Tabellen 1 und 2 ebenfalls, in welchen Zeiteinheiten mehr als eine Station gleichzeitig sendet und somit eine Kollision hervorgerufen wird.

(d) (2 Punkte) Berechnen Sie f_A, f_B und f_C, d.h. die jeweils prozentualen Anteile an bereitgestellten Nachrichten, die während der ersten 38 Zeiteinheiten fehlerfrei durch die drei Stationen versendet werden.

Lösungsvorschlag:

Nicht-unterteilt

$$f_A = \frac{1}{7} \approx 14.29\%$$
$$f_B = \frac{2}{6} \approx 33.33\%$$
$$f_C = \frac{1}{5} = 20\%$$

Unterteilt

$$f_A = \frac{4}{7} \approx 57.14\%$$
$$f_B = \frac{3}{6} = 50\%$$
$$f_C = \frac{4}{5} = 80\%$$

□

(e) (2 Punkte) Was ist die Hauptursache für die unterschiedlichen Anteile der fehlerfrei versendeten Nachrichten zwischen den beiden Zugriffsstrategien?

Lösungsvorschlag:
In the Slotted version, collisions either occur for an entire message duration or they do not occur at all. In the Non-Slotted version, many collisions occur only partially, leading to a higher fraction of time spent with unsuccessful transmissions.

(f) (2 Punkte) Nehmen Sie an, dass das Zugriffsprotokoll weiss, wann die drei Stationen Nachrichten bereitstellen. Berechnen Sie den maximalen Anteil an bereitgestellten Nachrichten $f_{\max}$, der dann durch die Stationen A, B und C während der ersten 38 Zeiteinheiten fehlerfrei versendet werden kann.

Lösungsvorschlag:

$$f_{\max} = \frac{16}{18} \approx 88.89\%$$

□

A						1	**1**				2	2				**3**	3				4	**4**				**5**	5				**6**	**6**				**7**	**7**		...
B							**1**	**1**					2	2					3	3					4	**4**					**5**	**5**					**6**	6	...
C								**1**	1						2	**2**						**3**	3						4	4						**5**	**5**		...
t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38

Tabelle 1: Nicht-unterteilte Zugriffsstrategie für Aufgaben 2.1 (b) und (c)

A							**1**	**1**			2	2					3	3			4	4					5	5			**6**	**6**					**7**	**7**	...
B							**1**	**1**					2	2					3	3					4	4					**5**	**5**					**6**	**6**	...
C									1	1					2	2							3	3					4	4							**5**	**5**	...
t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38

Tabelle 2: Unterteilte Zugriffsstrategie für Aufgaben 2.1 (b) und (c)

Lösungsvorschlag:

□

2.2: CSMA/CD Protokolle (maximal 8 Punkte)

Zwei Stationen A und B benutzen ein unterteiltes (slotted) Kommunikationsprotokoll mit CSMA/CD. Wenn eine neue Nachricht versendet werden soll, wird diese sofort ohne Verzögerung versendet. Bei einer Kollision wird die Nachricht wiederholt erneut versendet bis die Nachricht erfolgreich übertragen wurde. Dabei wird folgendes Schema angewandt: Wenn für eine Nachricht i aufeinanderfolgende Kollisionen bemerkt wurden, wartet die jeweilige Station k Zeitscheiben, bevor sie die Nachricht erneut sendet. Der Parameter k ist eine positive Ganzzahl und wird gleichverteilt aus dem Interval $[0, 2^i - 1]$ gewählt, d.h.

- Beim ersten Versuch: $k = 0$
- Nach einer Kollision: $k \in [0, 1]$
- Nach zwei aufeinanderfolgenden Kollisionen: $k \in [0, 3]$
- ...
- Nach i aufeinanderfolgenden Kollisionen: $k \in [0, 2^i - 1]$

(a) (1 Punkt) Worin besteht der Vorteil einer zufälligen Wartezeit (random backoff) nach einer Kollision bevor erneut gesendet wird?

Lösungsvorschlag:
To decrease the probability of new collisions. If the same deterministic backoff time is used by all sending Stations, new collisions occur at every retransmission attempt.

(b) (1 Punkt) Worin besteht der Vorteil einer maximalen Wartezeit, die mit der Anzahl an aufeinanderfolgenden Kollisionen länger wird?

Lösungsvorschlag:
To have high maximum backoff times only when it is really necessary: high backoff times imply high latencies. If many repeated collisions are detected, it is likely that many Stations are trying to send and thus high backoff times decrease the probability that two or more Stations will transmit again at the same time.

(c) (4 Punkte) Betrachten Sie den Fall, dass die Stationen A und B in Zeitscheibe 0 je eine Nachricht zu übertragen beginnen, wobei die Übertragung der Nachricht der Länge einer Zeitscheibe entspricht. Wie gross ist die Wahrscheinlichkeit, dass innerhalb von fünf Übertragungsversuchen immer noch keine Nachricht empfangen wurde?

Lösungsvorschlag:
The probability of not having received any message after i retransmission, $p(i)$, is $p(i) = p(i-1) \cdot \frac{2^i}{2^{2i}} p(i-1) \cdot \frac{1}{2^i}$, with $p(0) = 1$:

- $p(0) = 1$ (a collision certainly occurs at the first transmission attempt)
- $p(1) = p(0) \cdot \frac{1}{2}$
- $p(2) = p(1) \cdot \frac{1}{4} = 2^{-3}$
- $p(3) = p(2) \cdot \frac{1}{8} = 2^{-6}$
- $p(4) = p(3) \cdot \frac{1}{16} = 2^{-10}$

Therefore, after 4 retransmission the probability of not having received any message from any Station is 2^{-10}. The key point is that a new retransmission occurs only when all the previous transmission attempts fail, and thus $p(i)$ is always lower than $p(i-1)$.

(d) (2 Punkte) Nehmen Sie zudem an, dass die ersten 5 Übertragungsversuche fehlgeschlagen sind, und die Übertragung der Nachrichten von A und B nach dem sechsten Versuch gelingt. In welcher Zeitscheibe wird die letzte der beiden Nachrichten spätestens empfangen?

Lösungsvorschlag:
The i-th retransmission attempt ends at slots $T(i) = T(i-1) + 2^i$, with $T(0) = 0$:

- $T(0) = 0$
- $T(1) = T(0) + 2^1 = 2$
- $T(2) = T(1) + 2^2 = 6$
- $T(3) = T(2) + 2^3 = 14$
- $T(4) = T(3) + 2^4 = 30$
- $T(5) = T(4) + 2^5 = 62$

Therefore, the last message at the fifth retransmission attempt is received at slot 62, at latest.

2.3: Gemeinsame Nutzung von Ressourcen (maximal 12 Punkte)

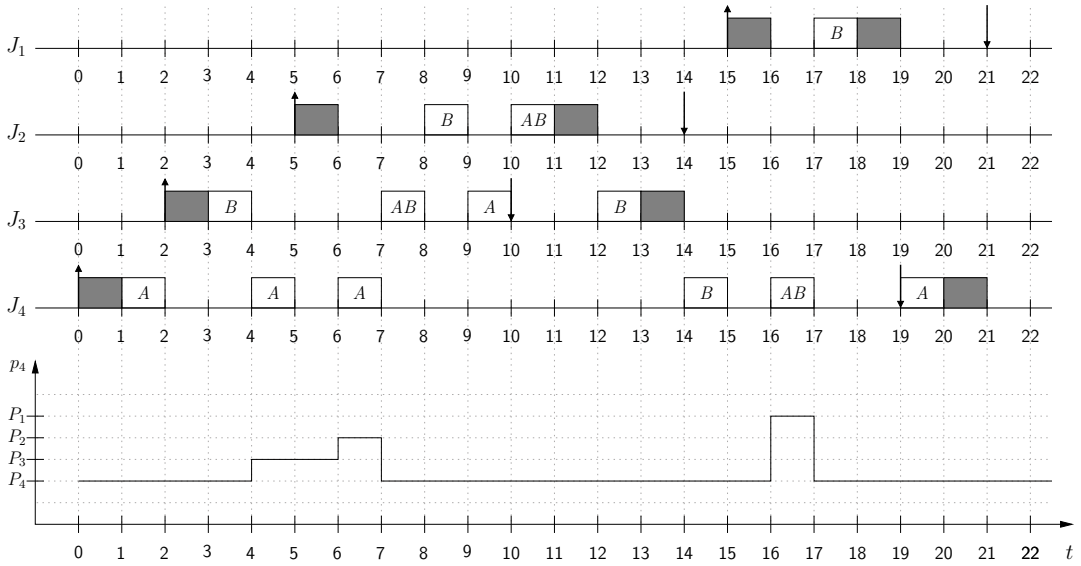
Ein Prozessor soll vier voneinander unabhängige Tasks verarbeiten (nominale Prioritäten: $P_1 > P_2 > P_3 > P_4$). Der Prozessor plant diese Tasks basierend auf ihren aktiven Prioritäten p_1, p_2, p_3 und p_4 zeitlich ein. Dabei gibt es zwei Ressourcen, bezeichnet durch A und B, auf die nur ein exklusiver Zugriff möglich ist.

Die nachfolgende Tabelle zeigt für jeden Task die Ankunftszeit, die benötigte Verarbeitungszeit und die jeweilige Deadline; die rechteckigen Boxen in der letzten Zeile repräsentieren Informationen für die Verarbeitung der einzelnen Tasks. Jede Box besteht aus mehreren Blöcken, welche die Taskabschnitte repräsentieren, wobei jeder Abschnitt eine Zeiteinheit lang ist. Nicht-kritische Abschnitte sind schattiert dargestellt. Kritische Abschnitte hingegen sind mit den Kennung(en) der blockierten exklusiven Ressource(n) versehen. Wird eine exklusive Resource für mehrere aufeinander folgende Zeitschritte benötigt, kann sie zwischenzeitlich nicht freigeben werden. Zum Beispiel ist Task J_2 vier Zeiteinheiten lang und hat zwei kritische Abschnitte. Im ersten kritischen Abschnitt blockiert J_2 nur die Ressource B, im zweiten die Ressourcen A und B. Die Resource B kann zwischen den beiden kritischen Abschnitten nicht freigegeben werden.

	J_1	J_2	J_3	J_4
Ankunftsszeit	15	5	2	0
Deadline	21	14	10	19
Verarbeitungsszeit	3	4	6	8
Taskstruktur	B	B AB	B AB A B	A A A B AB A

(a) (6 Punkte) Geben Sie einen Ablaufplan für die Tasks unter Verwendung des Priority Inheritance Protocol (PIP) an. Verwenden Sie das untenstehende Schema. Markieren Sie die kritischen Abschnitte der Tasks mit A und/oder B wie oben beschrieben. Geben Sie darüber hinaus die aktiven Prioritäten p_4 von Task J_4 an.

Lösungsvorschlag:



(b) (1 Punkt) Werden alle Deadlines eingehalten? Falls nicht, geben Sie die maximale Verspätung in Zeiteinheiten an.

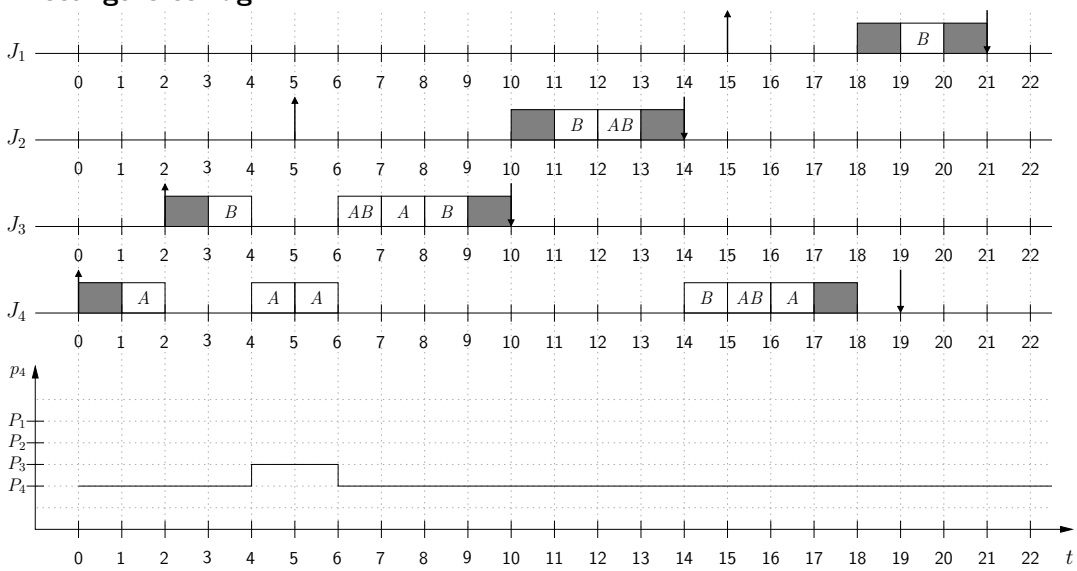
Lösungsvorschlag:
No, jobs J_3 and J_4 miss their deadlines. The maximum lateness is 4 time units (job J_3).

(c) (3 Punkte) Schlagen Sie eine mögliche Veränderung der Taskprioritäten vor, die es erlaubt, alle Deadlines einzuhalten. Begründen Sie Ihren Vorschlag.

Lösungsvorschlag:
Jobs missing their deadlines (J_3 and J_4) are the two with the smallest priorities, but J_3 has a deadline earlier than J_2, and J_4 earlier than J_1. Thus, a natural solution is to apply the EDF scheduling policy: $P_3 > P_2 > P_4 > P_1$.

(d) (2 Punkte) Wenden Sie nun erneut das Priority Inheritance Protocol (PIP) auf die veränderten Taskprioritäten an, und erstellen Sie einen Ablaufplan. Verwenden Sie das nachfolgende Schema. Geben Sie auch hier die aktiven Prioritäten p_4 von Task J_4 an.

Lösungsvorschlag:



Aufgabe 3 : Low Power Design (maximal 47 Punkte)

3.1: Dynamic Voltage Scaling (maximal 24 Punkte)

Gegeben sei ein Prozessor mit der Taktfrequenz f Hz und der Leistung $P = \left(\frac{f}{10^6}\right)^3$ mW. Der Prozessor bezieht seine Energie ausschliesslich von der Umgebung, die ihm mit einer konstanten Rate von 4 mW zur Verfügung gestellt wird. Überschüssige Energie wird in einer Batterie mit einer unbegrenzten Speicherkapazität zwischengespeichert. Die Batterie wird als verlustfrei angenommen. Zu Beginn der Ausführung eines Ablaufplanes ist die Batterie nahezu komplett entladen; die gespeicherte Anfangsenergie beträgt 0.15 μJ. Ein Ablaufplan ist *zulässig*, wenn alle Deadlines eingehalten werden und die verfügbare Energie nie unter 0 fällt.

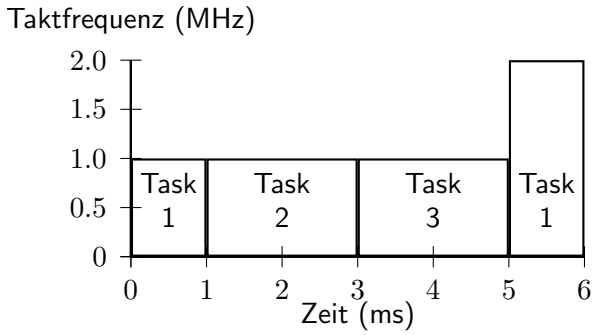
Der Prozessor bedient die folgenden Tasks:

Task ID	1	2	3
Ankunftszeit (ms)	0	1	3
Absolute Deadline (ms)	6	2	4
Anzahl Zyklen (x10^3)	3	2	2

<u>Hinweis:</u> Führen Sie alle Berechnungen mit einer Genauigkeit von zwei Nachkommastellen und in der Einheit μJ aus.

(a) (5 Punkte)

Der vorgesehene Ablaufplan sieht folgendermassen aus:



Stellen Sie für diesen Ablaufplan die in der Batterie gespeicherte Energie als Funktion der Zeit graphisch dar. Benutzen Sie dazu Abbildung 5. Wieviel Energie ist nach der Ausführung in der Batterie gespeichert? Ist dies ein zulässiger Ablaufplan?

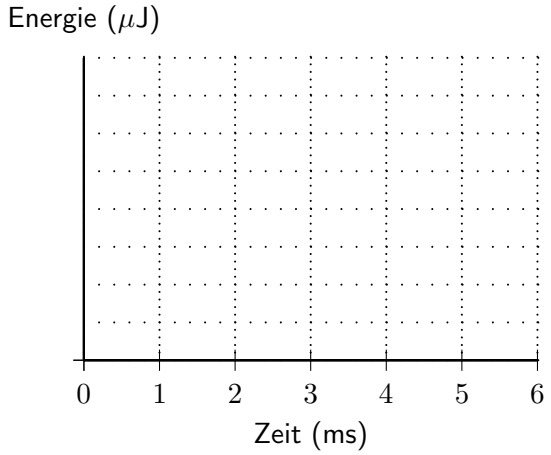


Abbildung 5: Gespeicherte Energie über die Zeit für Aufgabe 3.1 (a)

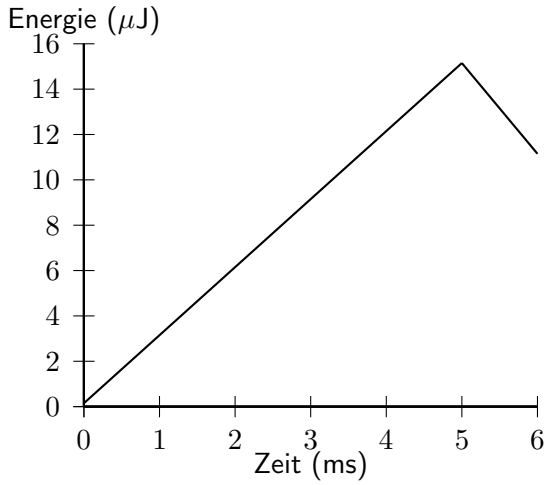


Abbildung 6: Energy vs Time for online given schedule

Lösungsvorschlag: Not valid as deadlines are missed.

□

(b) (8 Punkte) Geben Sie für die gleiche Task-Menge den *online* YDS-Ablaufplan an. Benutzen Sie hierfür die angegebene Zeitachse. Stellen Sie ebenfalls die in der Batterie gespeicherte Energie über die Zeit im bereitgestellten Graphen dar. Benutzen Sie dazu Abbildungen 7 und 8. Ist dies ein zulässiger Ablaufplan?

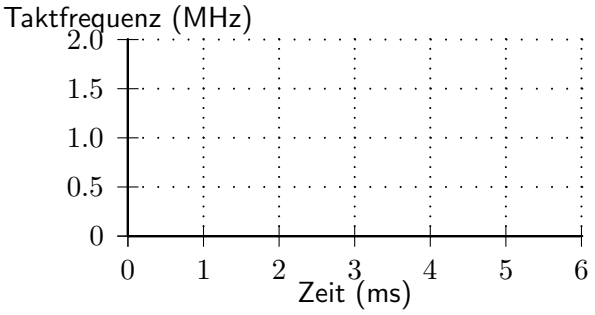


Abbildung 7: Online YDS-Ablaufplan für Aufgabe 3.1 (b)

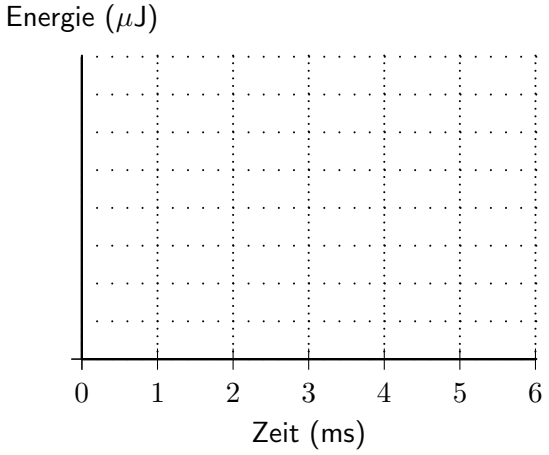


Abbildung 8: Gespeicherte Energie über die Zeit für online YDS-Ablaufplan

Lösungsvorschlag:

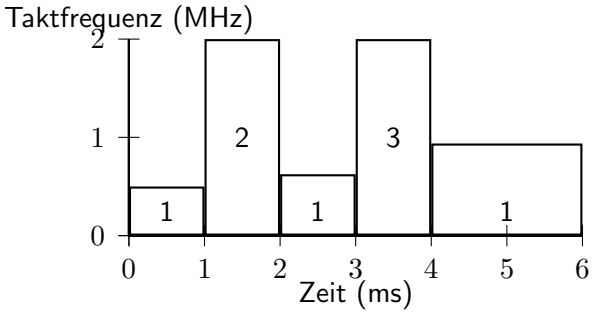


Abbildung 9: Online YDS schedule

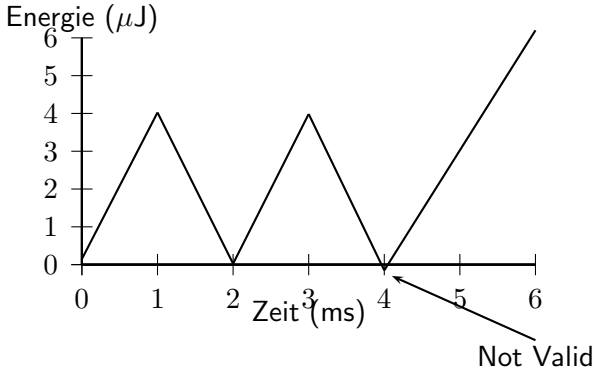


Abbildung 10: Energy vs Time for online YDS schedule

□

(c) (6 Punkte) Geben Sie nun für die Task-Menge den *offline* YDS-Ablaufplan an. Benutzen Sie hierfür die angegebene Zeitachse. Stellen Sie ebenfalls die in der Batterie gespeicherte Energie über die Zeit im bereitgestellten Graphen dar. Benutzen Sie dazu Abbildungen 11 und 12. Ist dies ein zulässiger Ablaufplan?

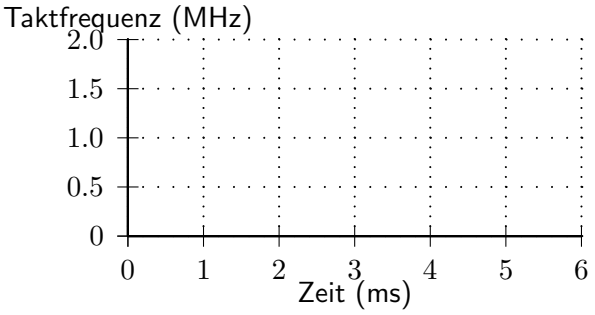


Abbildung 11: Offline YDS-Ablaufplan für Aufgabe 3.1 (c)

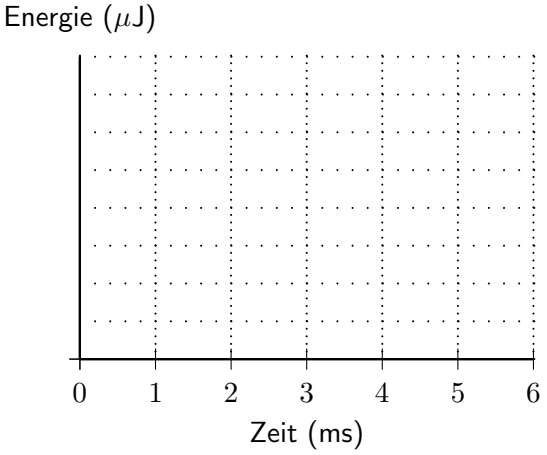


Abbildung 12: Gespeicherte Energie über die Zeit für offline YDS-Ablaufplan

Lösungsvorschlag:

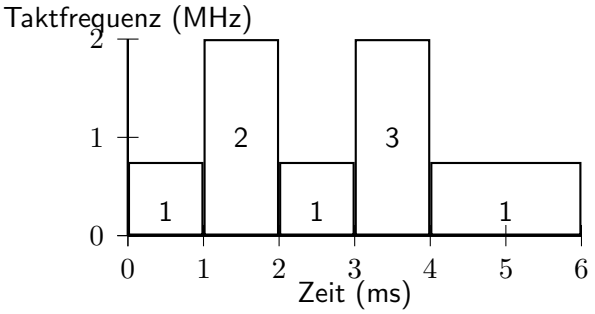


Abbildung 13: Offline YDS schedule

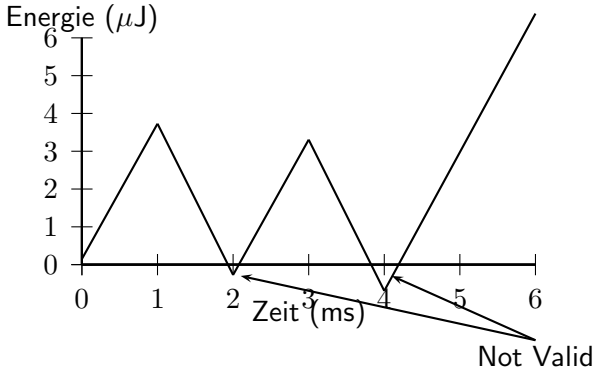


Abbildung 14: Energy vs Time for offline YDS schedule

□

(d) (5 Punkte) Geben Sie nun für die Task-Menge einen zulässigen Ablaufplan an, welcher die verfügbare Energie am Ende der Ausführung maximiert. Stellen Sie ebenfalls die in der Batterie gespeicherte Energie über die Zeit im bereitgestellten Graphen dar. Benutzen Sie dazu Abbildungen 15 und 16.

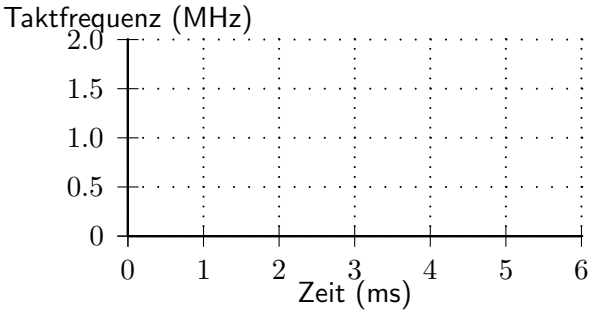


Abbildung 15: Ablaufplan für Aufgabe 3.1 (d)

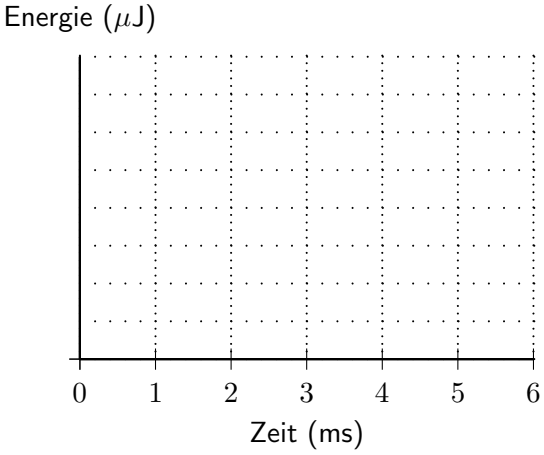


Abbildung 16: Gespeicherte Energie über die Zeit für Ablaufplan

Lösungsvorschlag:

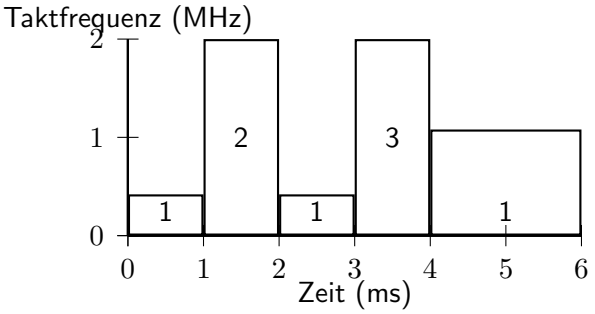


Abbildung 17: Modified schedule

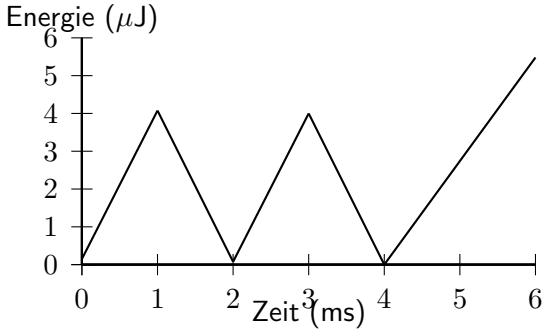


Abbildung 18: Energy vs Time for modified schedule

□

3.2: Dynamic Power Management (maximal 12 Punkte)

Gegeben ist ein Prozessor mit einem Leistungsbedarf von $P_{active} = 210$ mW wenn ein Task ausgeführt wird und $P_{idle} = 10$ mW wenn kein Task ausgeführt wird. Folgende Menge von periodischen Tasks wird abgearbeitet:

Task ID	1	2
Ankunftszeit (ms)	0	5
Absolute Deadline (ms)	8	10
Verarbeitungszeit (ms)	3	2
Periode (ms)	10	10

Definition: $E_{av} = \lim_{N\to\infty} \frac{\text{Energieverbrauch innerhalb von } N \text{ Perioden}}{N}$

Definition: Bei einem *arbeitserhaltenden Ablaufplan* ist der Prozessor nicht inaktiv, wenn Tasks zur Bearbeitung anstehen.

(a) (3 Punkte) Stellen Sie die Leistung als Funktion der Zeit graphisch dar, wenn die obigen Tasks nach einem arbeitserhaltenden Ablaufplan verarbeitet werden. Benutzen Sie dazu Abbildung 19. Welchen Wert hat E_{av}?

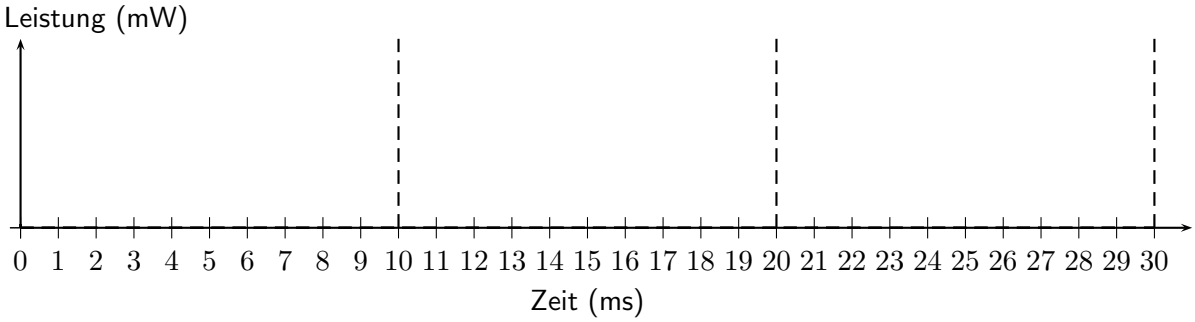


Abbildung 19: Diagramm zu Aufgabe 3.2 (a)

Lösungsvorschlag: Here P1 = P_{active} = 210mW and P3 = P_{idle} = 10 mW.

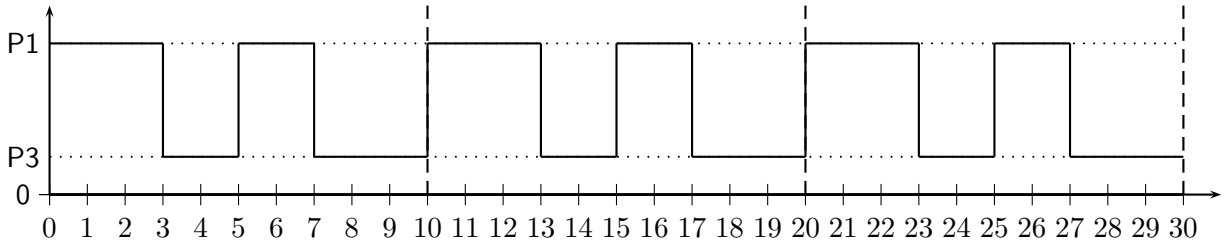


Abbildung 20: Power consumption work-conserving schedule

$E_{av} = 1.1mJ$

□

Nehmen Sie für die nachfolgenden Aufgaben an, dass der obige Prozessor in einen Standby-Modus wechseln kann, in dem keine Energie verbraucht wird. Ein Wechsel vom normalen Betriebsmodus in den Standby-Modus dauert 1 ms und benötigt 25μJ Energie, welche gleichverteilt während dieser Zeit verbraucht wird. Ein Zurückwechseln in den normalen Betriebsmodus passiert augenblicklich und verbraucht keine Energie.

(b) (3 Punkte) Stellen Sie die Leistung als eine Funktion der Zeit graphisch dar, wenn die obige Task-Menge nach einem arbeitserhaltenden Ablaufplan verarbeitet wird, wobei immer dann in den Standby-Modus gewechselt wird, wenn keine Tasks zur Bearbeitung anstehen. Benutzen Sie dazu Abbildung 21. Wie gross ist E_{av}?

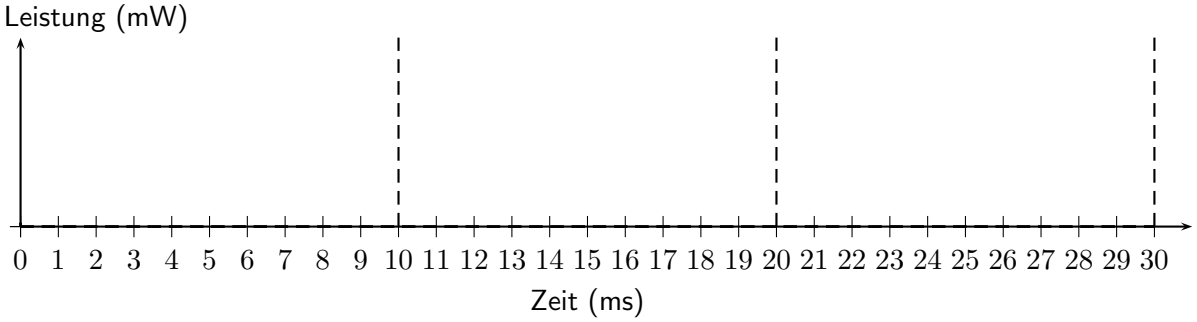


Abbildung 21: Diagramm zu Aufgabe 3.2 (b)

Lösungsvorschlag: Here P2 = E_{switch} / t_{switch} = 25mW

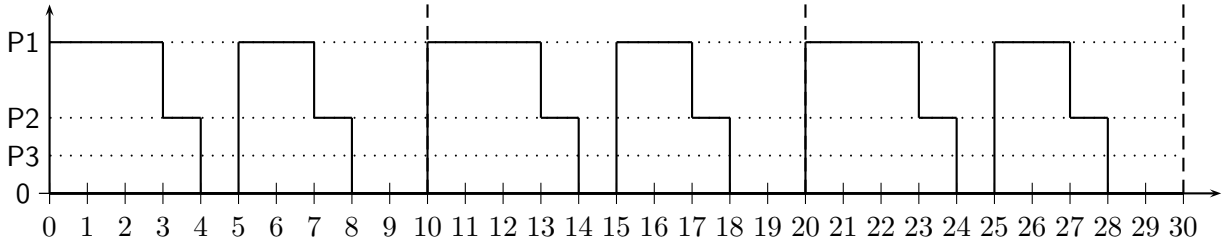


Abbildung 22: Power consumption work-conserving schedule and greedy standby switch

E_{av} = 1.1mJ

□

(c) (3 Punkte) Nehmen Sie nun an, dass das System *nicht* unbedingt in den Standby-Modus wechselt, wenn keine Tasks zur Verarbeitung anstehen. Stellen Sie die Leistung des arbeitserhaltenden Ablaufplans mit dem minimalen Energieverbrauch graphisch dar. Benutzen Sie dazu Abbildung 23. Wie gross ist E_{av}?

Lösungsvorschlag: E_{av} = 1.095mJ

□

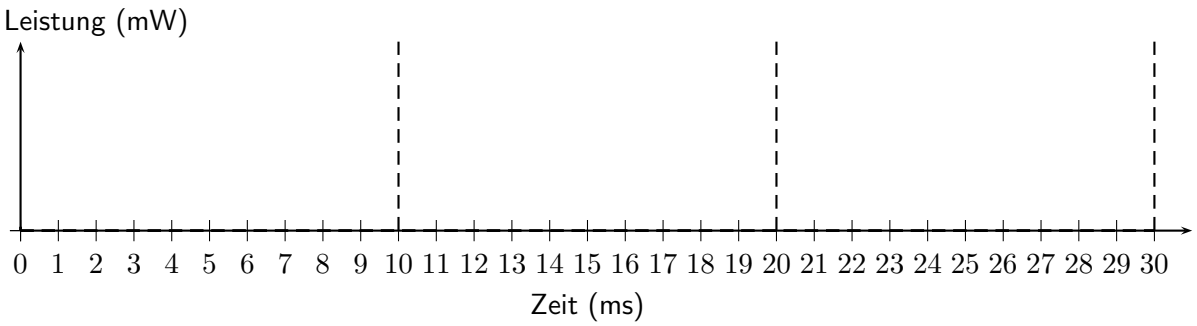


Abbildung 23: Diagramm zu Aufgabe 3.2 (c)

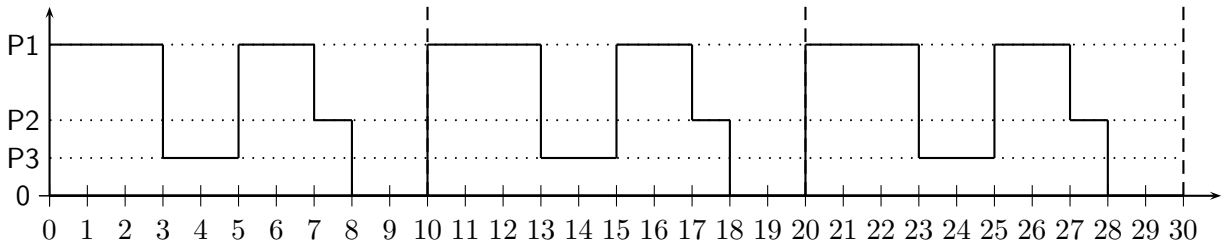


Abbildung 24: Power consumption work-conserving schedule and no greedy standby switch

(d) (3 Punkte) Gehen Sie nun ebenfalls davon aus, dass der Ablaufplan *nicht* arbeitserhaltend sein muss, d.h. die einzige Bedingung ist, dass alle Deadlines eingehalten werden. Stellen Sie die Leistung des Ablaufplans mit dem minimalen Energieverbrauch graphisch dar. Benutzen Sie dazu Abbildung 25. Wie gross ist E_{av}?

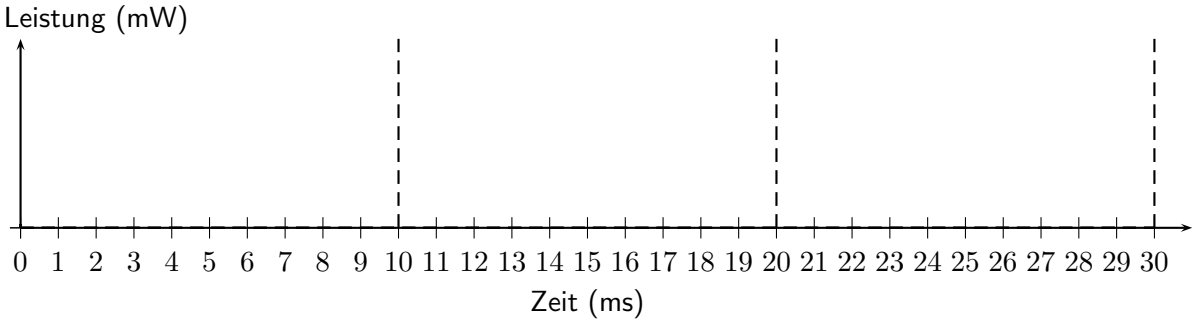


Abbildung 25: Diagramm zu Aufgabe 3.2 (d)

Lösungsvorschlag: $E_{av} = 1.075$mJ

□

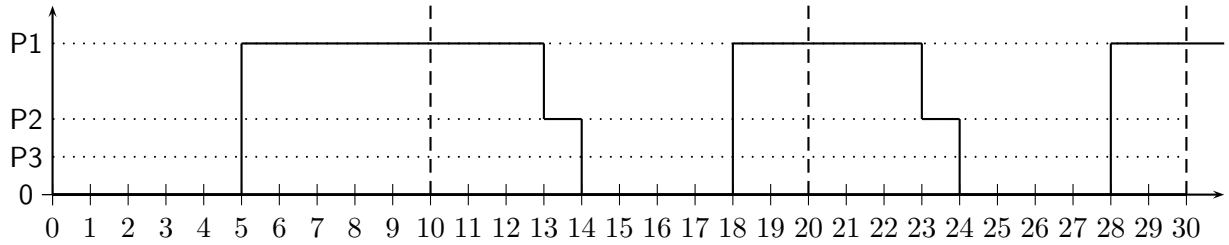
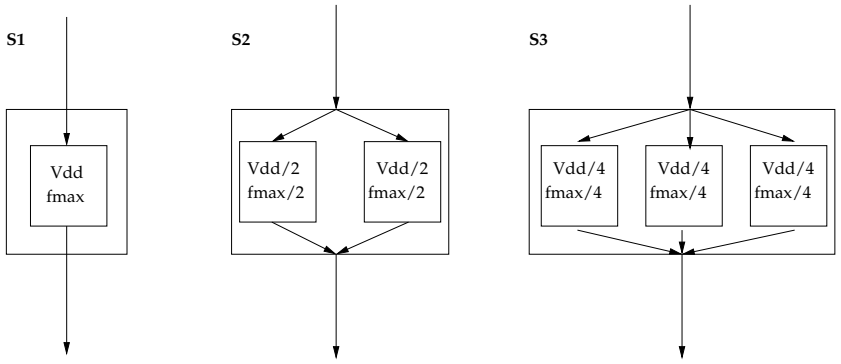


Abbildung 26: Power consumption with no work-conserving restriction and no greedy standby switch

3.3: Multi-Core Systeme (maximal 11 Punkte)

Gegeben seien die folgenden drei Systeme:



Alle drei Systeme basieren auf dem gleichen Prozessorkern (S1 basiert auf einem, S2 auf zwei, und S3 auf drei Kernen). Dieser hat einen Leckstrom mit einer konstanten Verlustleistung von $P_{leak} = 0.4$ mW, welche immer verbraucht wird. Bei der Ausführung eines Tasks wird zusätzlich eine frequenzabhängige Leistung P_{dyn} verbraucht. Diese skaliert mit der dritten Potenz der Betriebsspannung (V_{dd}): $P_{dyn} \sim V_{dd}^3$.

Das System S1 ist folgendermassen konfiguriert: $V_{dd} = 1.5$ V und $f_{max} = 100$ MHz. Die frequenzabhängige Leistung von System S1 ist $P_{dyn} = 100$ mW. Die Spannungs- und Frequenzwerte von S2 und S3 sind wie im obigen Schema angegeben. Die Verarbeitung des Tasks τ dauert 300 Zyklen auf S1, 300 Zyklen auf S2 und 300 Zyklen auf S3.

Definition: Das *Energie-Verzögerungszeit Produkt* ist das Produkt aus der Energie, die für die Ausführung eines Tasks benötigt wird, multipliziert mit der Ausführungszeit des Tasks.

(a) (5 Punkte) Stellen Sie die Leistung der drei Systeme graphisch dar, wenn Task τ verarbeitet wird. Benutzen Sie dazu Abbildung 27. Berechnen Sie ausserdem die Energie-Verzögerungszeit Produkte aller drei Systeme, wenn Task τ verarbeitet wird.

Lösungsvorschlag:

□

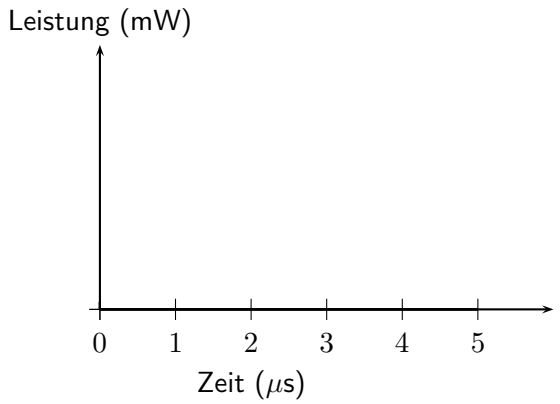


Abbildung 27: Diagramm zu Aufgabe 3.3 (a)

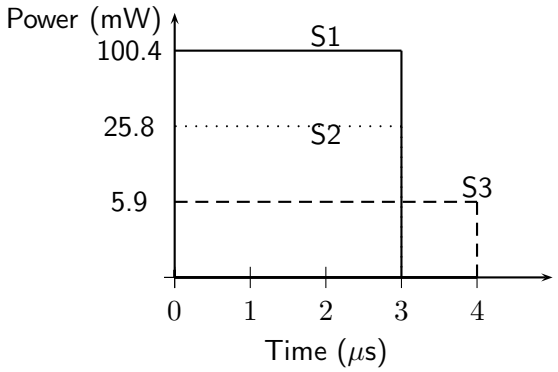


Abbildung 28: Power consumption of different systems

(b) (4 Punkte) Gegeben sei ein System mit N Kernen, wobei jeder Kern mit einer Taktfrequenz von f_{max}/N bei einer Spannung von V_{dd}/N betrieben wird. Das System kann Task τ innerhalb von $300/N$ Zyklen verarbeiten. Berechnen Sie die optimale Anzahl von Kernen, um das Energie-Verzögerungszeit Produkt zu minimieren.

Lösungsvorschlag:

$$\mathrm{EDP} = \left(\mathrm{P_{leak}} + \frac{\mathrm{P_{dyn}}}{\mathrm{N^3}}\right) \times \mathrm{N} \times \mathrm{t^2} \Rightarrow \mathrm{N_{EDP}^{opt}} = 8$$

□

(c) (2 Punkte) Geben Sie zwei Gründe an, warum das minimale Energie-Verzögerungszeit Produkt in einem realen System mit einem kleineren Wert N erreicht werden würde.

Lösungsvorschlag:

(a) It is difficult to completely parallelize a task into N parallel parts with no sequential overhead.

(b) Frequencies available are discrete and conservative quantizations are required.

(c) The cost of the system increases super-linearly with N (larger die area, more defects, costly also for design optimizations)

□

Aufgabe 4 : Synthese (maximal 43 Punkte)

4.1: LIST-Scheduling (maximal 16 Punkte)

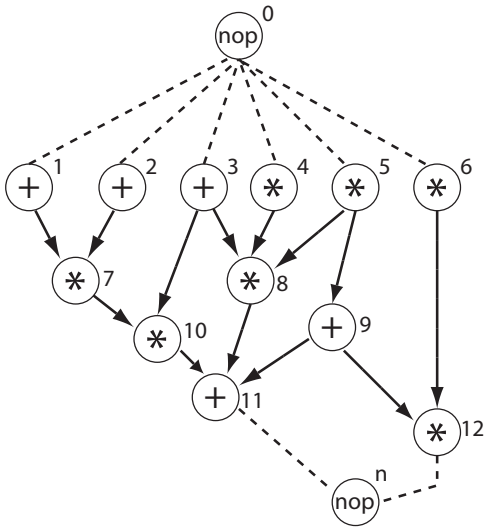


Abbildung 29: Sequenzgraph

Gegeben ist der Sequenzgraph aus Abbildung 29.

(a) (10 Punkte) Für eine Architektur, die aus zwei ALUs und zwei Multiplizierern besteht, soll für diesen Sequenzgraphen ein Ablaufplan mit dem LIST-Algorithmus ermittelt werden. Additionen und Subtraktionen werden auf **zwei Einheiten vom Typ** r_1 und Multiplikationen auf **zwei Einheiten vom Typ** r_2 ausgeführt. Dabei benötigen Addition und Subtraktion **eine Zeiteinheit** und die Multiplikation **zwei Zeiteinheiten**. Nehmen Sie an, dass die erste Operation zum Zeitpunkt $t = 0$ startet und dass der *nop* Hierarchieknoten keine Zeit beansprucht. Verwenden Sie für die Priorität einer Operation die maximale Länge des Pfades der Folgeknoten bis zum 'n' nop-Knoten. Benutzen Sie für die Ausführung des Algorithmus Tabelle 3. Tragen Sie für den Zeitschritt t und den Ressourcentyp k jeweils die Kandidatenmenge $U_{t,k}$, die Menge der laufenden Operationen $T_{t,k}$, sowie die zum Starten selektierten Operationen $S_{t,k}$ ein.

(b) (1 Punkt) Wie gross ist die resultierende Latenz?

Lösungsvorschlag:
$L = 8$.

□

(c) (1 Punkt) Wie gross ist die Latenz, wenn beliebig viele Ressourcen zur Verfügung stehen und Additions- und Multiplikationsoperationen jeweils **eine Zeiteinheit** beanspruchen?

Lösungsvorschlag:
$L = 4$.

□

(d) (1 Punkt) 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, es ist eine Heuristik. ILP würde eine optimale Lösung liefern.

□

Beantworten Sie die folgende Frage allgemein ohne den obigen Sequenzgraphen zu beachten:

(e) (3 Punkte) Angenommen, Sie müssten die Relationen des zugehörigen ILP aufstellen (ohne das ILP tatsächlich zu lösen). Wie in der Vorlesung beschrieben, müssten Sie die möglichen Startzeitpunkte jedes Tasks mit Hilfe von ASAP und ALAP Ablaufplänen bestimmen. Wie würden Sie die maximale Latenz L_{max} für den ALAP Ablaufplan auswählen? Begründen sie ihre Antwort.

Lösungsvorschlag:
L_{max} is taken from the LIST Schedule because it guarantees that a feasible schedule exists for this latency.

□

t	k	$U_{t,k}$	$T_{t,k}$	$S_{t,k}$
0	r_1	ν_1, ν_2, ν_3	—	ν_1, ν_2
	r_2	ν_4, ν_5, ν_6	—	ν_4, ν_5
1	r_1	ν_3	—	ν_3
	r_2	ν_6, ν_7	ν_4, ν_5	—
2	r_1	ν_9	—	ν_9
	r_2	ν_6, ν_7, ν_8	—	ν_7, ν_6
3	r_1	—	—	—
	r_2	ν_8	ν_6, ν_7	—
4	r_1	—	—	—
	r_2	$\nu_8, \nu_{10}, \nu_{12}$	—	ν_8, ν_{10}
5	r_1	—	—	—
	r_2	ν_{12}	ν_8, ν_{10}	—
6	r_1	ν_{11}	—	ν_{11}
	r_2	ν_{12}	—	ν_{12}
7	r_1	—	—	—
	r_2	—	ν_{12}	—
8	r_1	—	—	—
	r_2	—	—	—
9	r_1	—	—	—
	r_2	—	—	—
10	r_1	—	—	—
	r_2	—	—	—
11	r_1	—	—	—
	r_2	—	—	—

Tabelle 3: Tabelle für Aufgabe 4.1 (a)

4.2: Zeitbeschränkungen (maximal 8 Punkte)

Gegeben ist der Sequenzgraph in Abbildung 30 und die Ausführungszeiten der Funktionen in Tabelle 4.

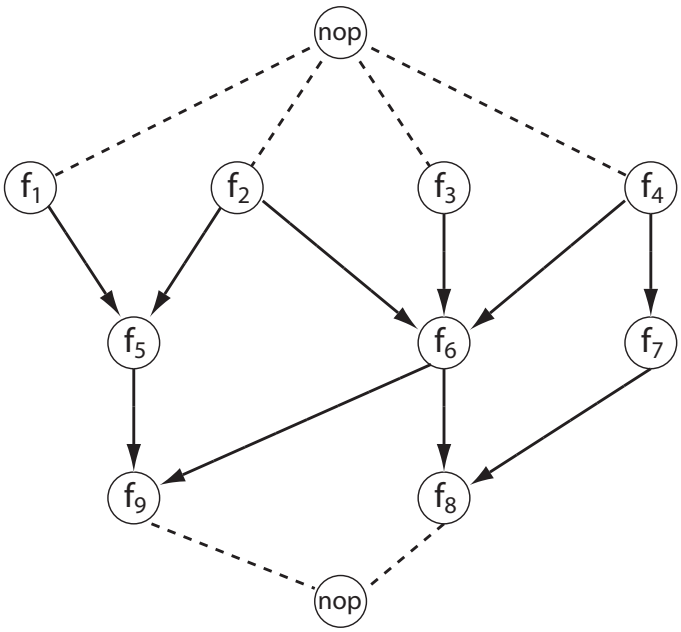


Abbildung 30: Sequenzgraph

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
2	3	4	5	5	6	7	1	1

Tabelle 4: Ausführungszeiten der Funktionen

(a) (2 Punkte) Zeichnen Sie den Beschränkungsgraphen (Weighted Constraint Graph) $G_C = (V_C, E_C, d)$. Sie können dazu den Graphen in Abbildung 30 ergänzen.

Lösungsvorschlag:
Siehe Abbildung 31.

□

(b) (3 Punkte) Fügen Sie folgende zusätzlichen Zeitbeschränkungen in den Beschränkungsgraphen G_C ein (Sie können dazu wiederum den Graphen in Abbildung 30 ergänzen):

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 32.

□

i) f_6 darf frühestens 6 Zeitschritte nach dem Start von f_5 starten.

Lösungsvorschlag:
$\tau(\nu_6) \geq \tau(\nu_5) + 6 \;\Rightarrow\; \tau(\nu_6) - \tau(\nu_5) \geq 6$

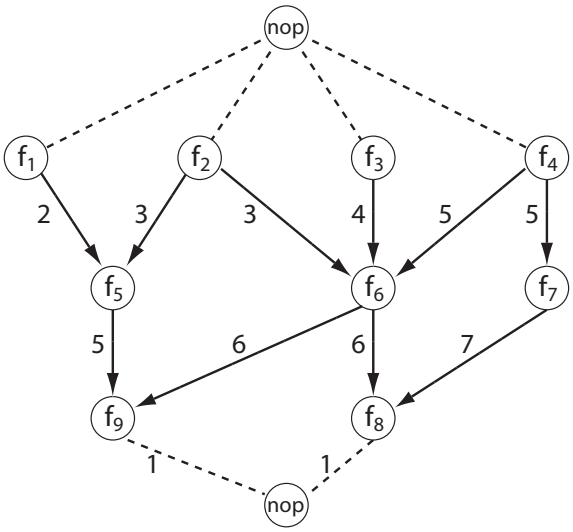


Abbildung 31: *Lösungsvorschlag:* Beschränkungsgraphen für Aufgabe 4.2.a)

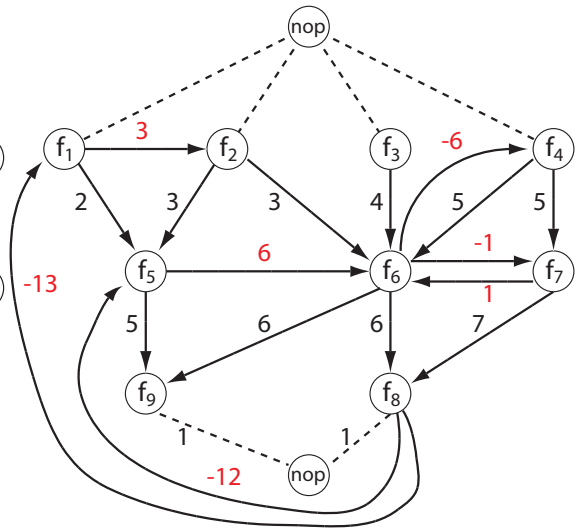


Abbildung 32: *Lösungsvorschlag:* Beschränkungsgraphen für Aufgabe 4.2.b)

□

ii) f_6 muss genau 1 Zeitschritt nach dem Start von f_7 starten.

Lösungsvorschlag:
$\tau(\nu_6) = \tau(\nu_7) + 1 \;\Rightarrow\; \tau(\nu_6) - \tau(\nu_7) \geq 1 \;\wedge\; \tau(\nu_7) - \tau(\nu_6) \geq -1$

□

iii) f_6 muss spätestens 6 Zeitschritte nach dem Start von f_4 starten.

Lösungsvorschlag:
$\tau(\nu_6) \leq \tau(\nu_4) + 6 \;\Rightarrow\; \tau(\nu_4) - \tau(\nu_6) \geq -6$

□

iv) f_2 darf frühestens 3 Zeitschritte nach dem Start von f_1 starten.

Lösungsvorschlag:
$\tau(\nu_2) \geq \tau(\nu_1) + 3 \;\Rightarrow\; \tau(\nu_2) - \tau(\nu_1) \geq 3$

□

v) f_8 muss spätestens 7 Zeitschritte nach dem *Beenden* von f_5 starten.

Lösungsvorschlag:
$\tau(\nu_8) \leq (\tau(\nu_5) + 5) + 7 \;\Rightarrow\; \tau(\nu_5) - \tau(\nu_8) \geq -12$

□

vi) f_8 muss spätestens 11 Zeitschritte nach dem *Beenden* von f_1 starten.

Lösungsvorschlag:
$\tau(\nu_8) \leq (\tau(\nu_1) + 2) + 11 \;\Rightarrow\; \tau(\nu_1) - \tau(\nu_8) \geq -13$

□

(c) (3 Punkte) Existiert ein gültiger Ablaufplan unter den gegebenen Zeitbeschränkungen und unter der Annahme, dass beliebig viele Ressourcen zur Verfügung stehen? Begründen Sie Ihre Antwort.

Lösungsvorschlag:
No. One positive cycle in G_C : $f_1, f_5, f_6, f_4, f_7, f_8, f_1$. See Figure 32.

□

4.3: Iterativer Algorithmus (maximal 19 Punkte)

Gegeben ist ein iterativer Algorithmus, der mittels des markierten Graphen in Abbildung 33 spezifiziert ist. Die Ausführungszeiten der Operationen sind in Tabelle 5 gegeben.

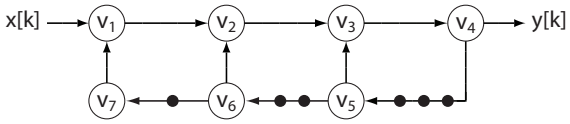


Abbildung 33: Markierter Graph

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$w(v_i)$	1	1	1	1	1	1	1

Tabelle 5: Ausführungszeiten

(a) (2 Punkte) Stellen Sie die Datenabhängigkeiten des markierten Graphen in einem erweiterten Sequenzgraphen $G_S = (V_S, E_S, d)$ dar.

Lösungsvorschlag:
Solution shown in Figure 34.

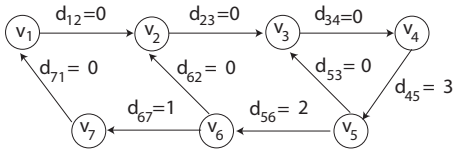


Abbildung 34: Erweiterten Sequenzgraph

□

(b) (3 Punkte) Drücken Sie alle bestehenden Datenabhängigkeiten in der folgenden Form aus:

$$\tau(v_j) - \tau(v_i) \geq w(v_i) - d_{ij} * P \quad \forall (v_i, v_j) \in E_S$$

Lösungsvorschlag:

$$\tau(v_2) - \tau(v_1) \geq 1$$
$$\tau(v_3) - \tau(v_2) \geq 1$$
$$\tau(v_4) - \tau(v_3) \geq 1$$
$$\tau(v_3) - \tau(v_5) \geq 1$$
$$\tau(v_2) - \tau(v_6) \geq 1$$
$$\tau(v_1) - \tau(v_7) \geq 1$$
$$\tau(v_7) - \tau(v_6) \geq 1 - 1 * P$$
$$\tau(v_6) - \tau(v_5) \geq 1 - 2 * P$$
$$\tau(v_5) - \tau(v_4) \geq 1 - 3 * P$$

□

(c) (6 Punkte) Nehmen Sie an, dass beliebig viele Ressourcen zur Verfügung stehen. Erstellen Sie einen Ablaufplan mit 4 Durchläufen (Iterationen), ohne funktionale Fliessbandverarbeitung (functional pipelining) und ohne Schleifenfaltung (loop folding). Geben Sie alle Abhängigkeiten an, einschliesslich derer zwischen den 4 Durchläufen. Benutzen Sie Abbildung 35.

Lösungsvorschlag:
See Figure 35.

□

(d) (8 Punkte) Nehmen Sie an, dass beliebig viele Ressourcen zur Verfügung stehen. Ermitteln Sie für den iterativen Algorithmus mit Schleifenfaltung (loop folding)

- das minimale Iterationsintervall P
  (z.B. durch Einsetzen in die Ergebnisse von Punkt b)
  und

  Lösungsvorschlag:
  Start with tasks which do not have predecessors in current iteration: $\tau(v_5) = 0$, $\tau(v_6) = 0$, and $\tau(v_7) = 0$.
  Then substitute in inequalities:

$$\begin{aligned} \tau(v_1) &\geq 1 \\ \tau(v_2) &\geq 2 \\ \tau(v_3) &\geq 3 \\ \tau(v_4) &\geq 4 \\ P &\geq 1 \\ P &\geq 1/2 \\ P &\geq 5/3 \end{aligned}$$

  Minimum feasible integer P is 2.

  □

- einen gültigen Ablaufplan der Operationen v_i über 4 Durchläufe in Form einer Skizze. Verwenden Sie dazu die vorbereitete Abbildung 36 und heben Sie die unterschiedlichen Durchläufe hervor (z.B. durch Beschriftung oder Markierung).

  Lösungsvorschlag:
  See Figure 36.

  □

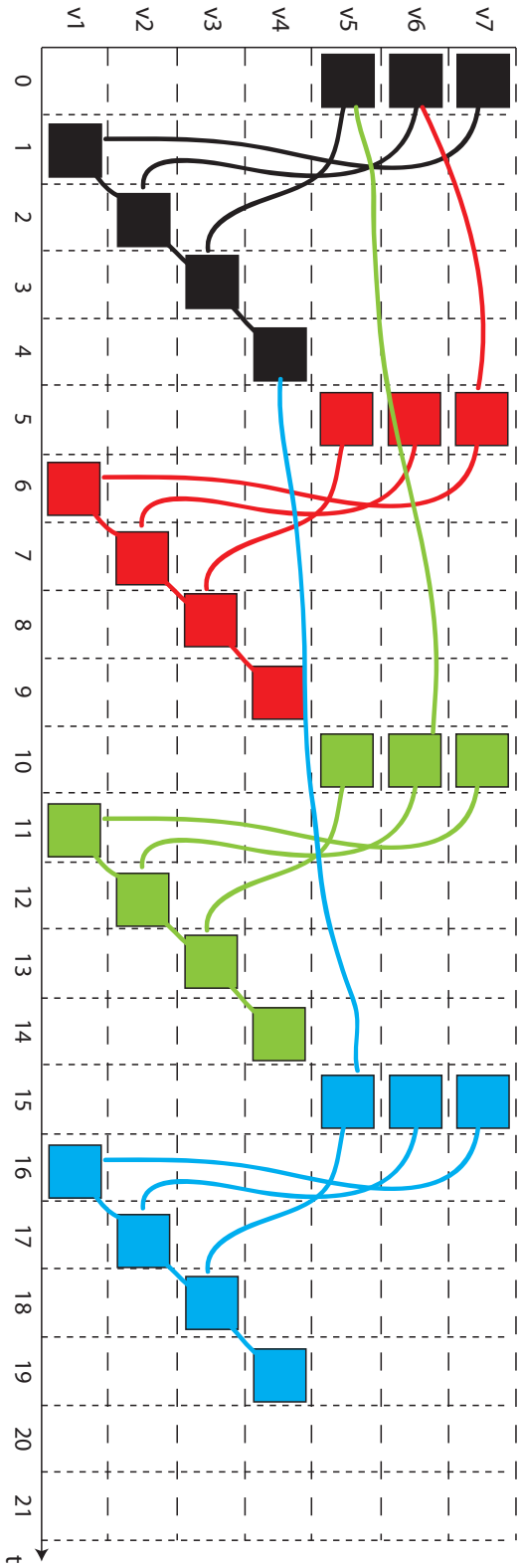


Abbildung 35: Ablaufplan ohne Schleifenfaltung

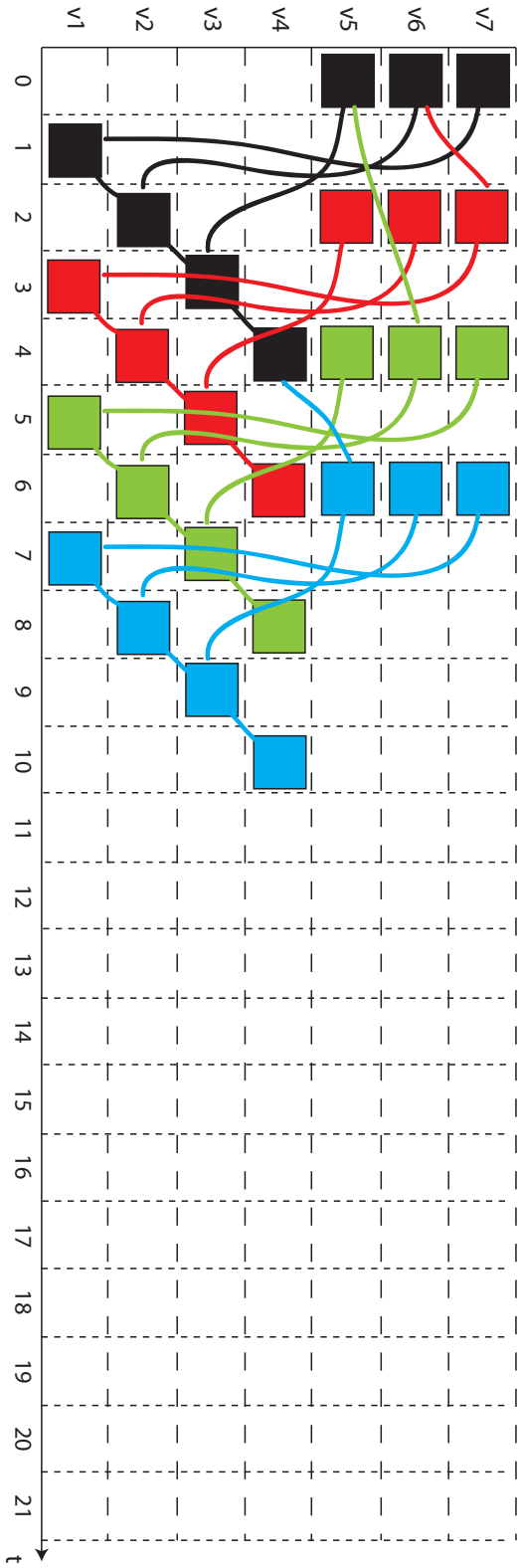


Abbildung 36: Ablaufplan mit Schleifenfaltung