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:

- The given solution is only a proposal. For correctness, completeness, or understandability no responsibility is taken.

Aufgabe 1 : Real-time scheduling (maximal 38 Punkte)

1.1: RM-Scheduling (maximal 5 Punkte)

Consider the following periodic task set scheduled according to the rate monotonic (RM) scheduling policy:

	C_i	T_i	Φ_i
τ_1	1	5	0
τ_2	2.5	5	0.5

Φ_i is the initial offset of task τ_i. Task τ_1 has higher priority than task τ_2.

Determine the exact worst-case response time of τ_2.
Hint: The question can be solved graphically.

Lösungsvorschlag:
$R_2 = 3$.

□

1.2: Scheduling with fixed priorities (maximal 7 Punkte)

(a) (2 Punkte) What is the utilization U of the following task set scheduled with the RM policy?

	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) Consider n periodic tasks scheduled according to RM. The utilization factor for this task set is U_n. We know that $U_n > n.(2^{1/n} - 1)$. State whether the following statement is true or false: This task set is definitely unschedulable by RM.

Lösungsvorschlag:
False

□

(c) (3 Punkte) Fixed priority premptive scheduling (FPPS) is a generalization of the RM policy in the sense that a task may be assigned any arbitrary priority. State whether the following statements are true or false:

(i) Given that a task set is schedulable under FPPS (assuming $D_i = T_i$). Is this task set schedulable by RM?

Lösungsvorschlag:
Yes

□

(ii) Given that a task set is schedulable under RM (assuming $D_i = T_i$). Is such a task set always schedulable for any priority order under FPPS?

Lösungsvorschlag:
No

□

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

Consider the following set of periodic tasks:

	C_i	T_i
τ_1	1	6
τ_2	1	3
τ_3	3	10

Given is a single Total Bandwidth Server (TBS).

(a) (2 Punkte) What can be the maximum utilization of the TBS (U_s), such that the above task set is schedulable together with the TBS under the EDF policy?

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

□

(b) (6 Punkte) Consider that $U_s = 0.2$. Construct the EDF schedule in the case that the TBS serves the following three aperiodic requests: $J_4(a_4 = 2, C_4 = 1)$, $J_5(a_5 = 6, C_5 = 2)$ and $J_6(a_6 = 1, C_6 = 1)$ where a_4, a_5, and a_6 denote the arrival times of the requests. Use Figure 1 for your answer. Assume that the arrival time of all periodic tasks is 0.

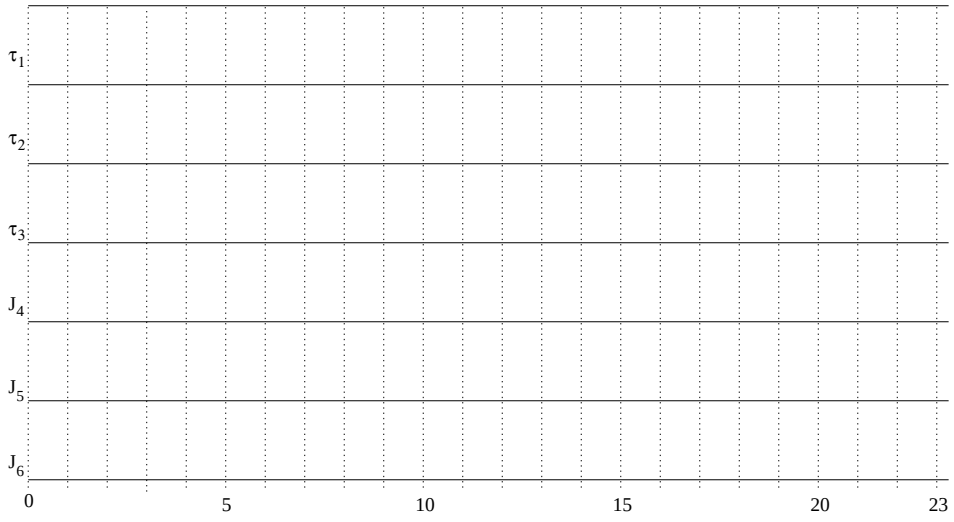


Abbildung 1: Schedule for Aufgabe 1.3 (b)

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.

□

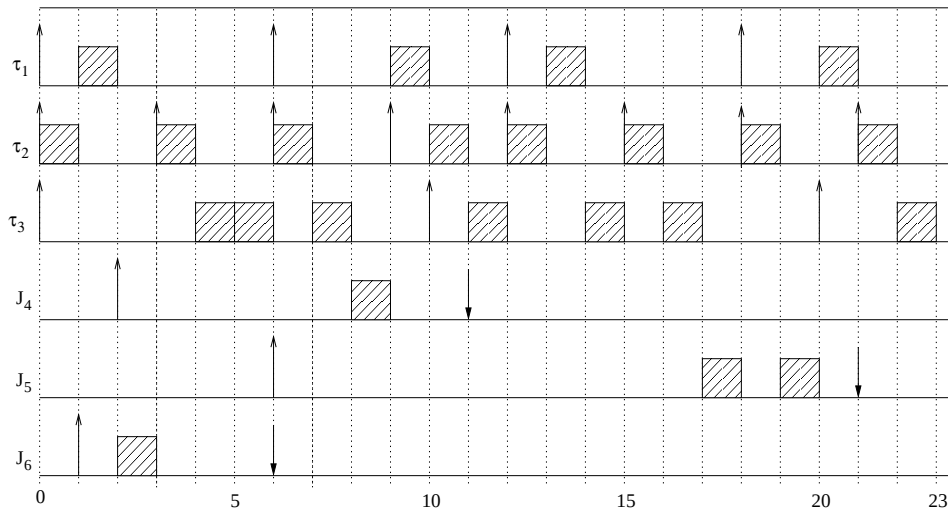


Abbildung 2: Schedule

1.4: Polling Server

(maximal 8 Punkte)

Given is the following periodic task set:

	C_i	T_i	Φ_i
τ_1	1	5	2
τ_2	1	5	2

Φ_i is the initial offset of task τ_i. Consider a single polling server with parameters $T_s = 6$, $C_s = 2$ and $\Phi_s = 0$ running together with the above periodic task set.

There are two aperiodic tasks τ_x ($C_x = 2$) and τ_y ($C_y = 1$) served by the polling server. The aperiodic tasks are preemptible and the priority of task τ_y is higher than the one of task τ_x. The arrival times of tasks τ_x and τ_y are between 0 and 5, and they are not necessarily integers, i.e., $a_x \in [0, 5]$, $a_y \in [0, 5]$, $a_x \in \mathbb{R}$, $a_y \in \mathbb{R}$.

Draw in Figure 3 the schedule which gives the worst-case response time for task τ_x. Hint: think about the worst possible arrangement of the arrival times of the two aperiodic tasks.

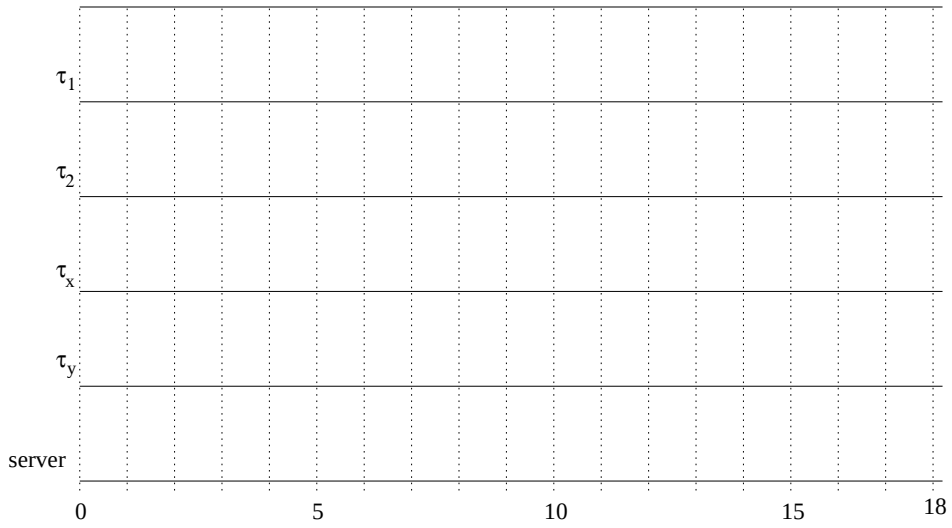


Abbildung 3: Schedule for Aufgabe 1.4

Lösungsvorschlag:

See Figure 4.

□

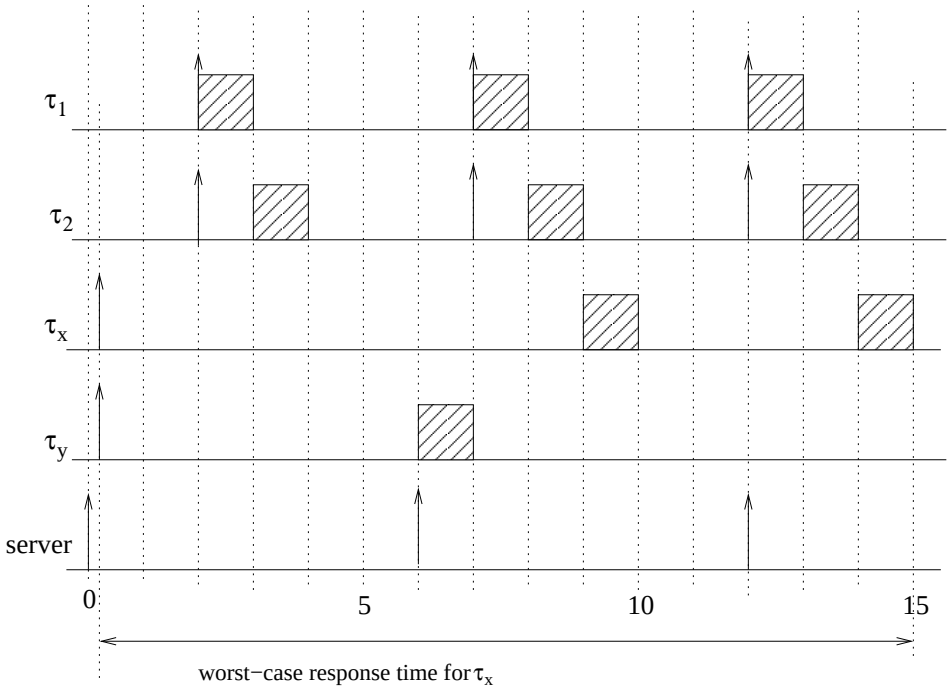


Abbildung 4: Schedule giving worst-case response time for τ_x

1.5: RM-Scheduling (maximal 10 Punkte)

Are the following four periodic tasks schedulable under the RM policy? Motivate your answer.

	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 : Communication and Resource Sharing (maximal 32 Punkte)

2.1: Slotted and Non-slotted Communication (maximal 12 Punkte)

Three stations A, B, and C share the same communication medium to send and receive messages. A message is transmitted free of errors (i.e. without collisions) when the sending station uses the communication medium exclusively for the entire duration of the message transmission.

Stations A, B, and C transmit messages with the same duration of 2 time units. Messages are sent periodically. The respective periods T and initial offsets Φ are as follows:

Station A: $T_A = 5$, $\Phi_A = 5$
i.e. messages are sent at time instances: 5, 10, 15, ...
Station B: $T_B = 6$, $\Phi_B = 6$
i.e. messages are sent at time instances: 6, 12, 18, ...
Station C: $T_C = 7$, $\Phi_C = 7$
i.e. messages are sent at time instances: 7, 14, 21, ...

Stations access the communication medium without performing any access control. In the case of collisions, no retransmissions are performed. Two medium access policies are used:

Non-slotted A station transmits a message as soon as it becomes available.

Slotted Starting from time instant 0, time is divided into equal time slots with the length of a single message (i.e. 2 time units). When a message becomes available, the station transmits it at the beginning of the next available time slot (i.e. only at time instants 0, 2, 4, 6, ...).

(a) (1 Punkt) What is the essential condition (here implicitly satisfied) that allows using slotted communication protocols?

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) Please use Tables 1 and 2. For each station A, B, and C, mark the time units where the station is transmitting messages using consecutive message numbering (i.e. 1, 2, 3, ...).

(c) (2 Punkte) In both Tables 1 and 2, highlight the time intervals where more than one station are transmitting a message and thus a collision occurs.

(d) (2 Punkte) Calculate f_A, f_B and f_C, i.e., the respective percentage of messages that are transmitted without errors during the first 38 time units by the three stations.

Lösungsvorschlag:

Non-slotted

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

Slotted

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

□

(e) (2 Punkte) What is the main reason for the differences in the percentages of sent error-free messages for the two access policies?

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) Assume that the access protocol knows in advance when the three stations are ready to transmit. Calculate the maximum percentage of messages $f_{\max}$ that can be transmitted without errors by the three stations during the first 38 time units.

Lösungsvorschlag:

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

□

Lösungsvorschlag:

□

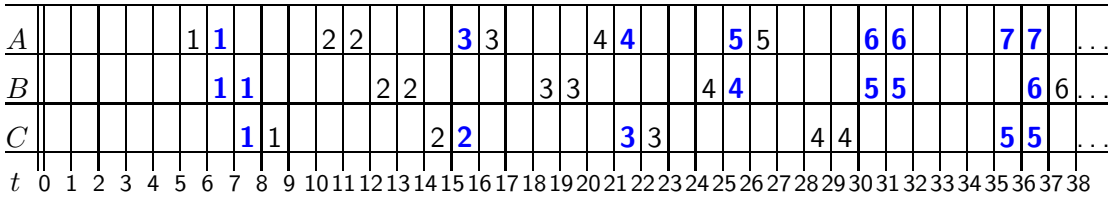


Tabelle 1: Non-slotted access policy for Aufgaben 2.1 (b) und (c)

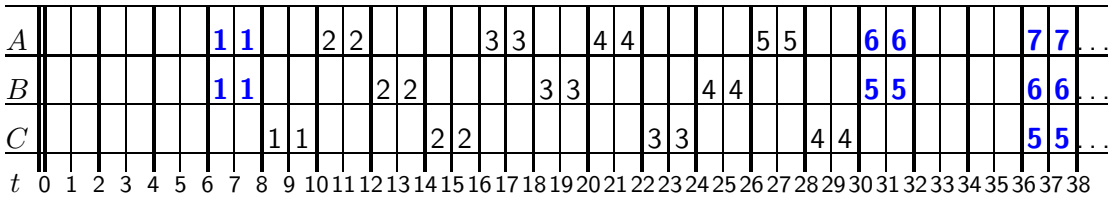


Tabelle 2: Slotted access policy for Aufgaben 2.1 (b) und (c)

2.2: CSMA/CD Protocols (maximal 8 Punkte)

Two Stations A and B make use of a slotted communication protocol with CSMA/CD. When a new message is to be sent, this is done without any delay. In case of a collision, the message is repeatedly sent until it is successfully transmitted. The following policy is used: If for a message, i consecutive collisions are observed, a station waits for k time slots before sending the message again. Parameter k is a positive integer and it is chosen from a uniform distribution with the interval $[0, 2^i - 1]$.

- At the first transmission: $k = 0$
- After one collision: $k \in [0, 1]$
- After two consecutive collisions: $k \in [0, 3]$
- ...
- After i consecutive collisions: $k \in [0, 2^i - 1]$

(a) (1 Punkt) What is the benefit of waiting a random backoff time after a collision has been detected?

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) What is the benefit of having a maximum backoff time increasing with the number of consecutive collisions?

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) Consider the case in which stations A and B start transmitting one message each at time slot 0; the transmission of the message fits within a single slot. What is the probability of not having received any message from any Station within 5 transmission attempts?

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) Assume that the first five transmission attempts failed, and the transmission of messages from A and B succeeded with the sixth attempt. At which slot number is the last of the two messages received, at latest?

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

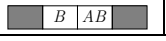
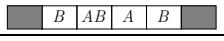
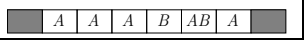
- $T(0) = 0$
- $T(1) = T(0) + 2^1 = 2$
- $T(2) = T(1) + 2^2 = 6$
- $T(3) = T(2) + 2^3 = 14$
- $T(4) = T(3) + 2^4 = 30$
- $T(5) = T(4) + 2^5 = 62$

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

2.3: Resource Sharing (maximal 12 Punkte)

A processor has to execute four mutually independent tasks with priorities $P_1 > P_2 > P_3 > P_4$. The processor schedules the tasks based on their active priorities p_1, p_2, p_3 and p_4. There are two exclusive access resources, A and B.

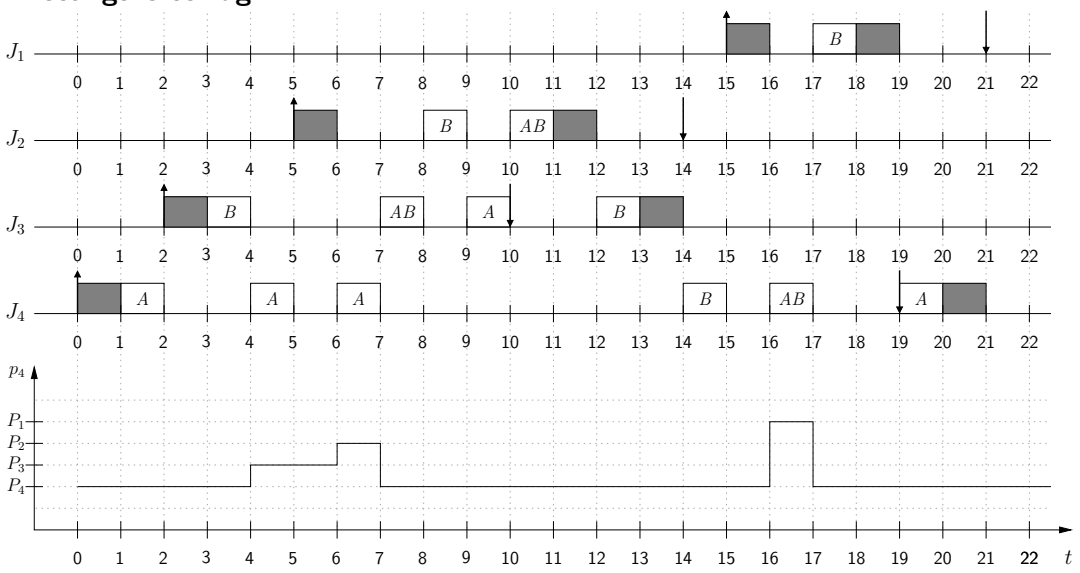
The following table contains information on the arrival time, deadline, and execution time for each task. The rectangular boxes in the last row show the processing structure of each task. Each task block consists of several blocks representing the task sections. Each section has a unit time length. Non-critical sections are shaded. Critical sections are marked with the identifier(s) of the blocked exclusive shared resource(s). If an exclusive resource is needed and blocked for a time interval lasting several time units, it is not released during this time interval. For example, task J_2 takes 4 time units to execute and has two critical sections. In the first critical section, task J_2 blocks only resource B, and in the second it blocks both resources A and B. Resource B is not released between the two critical sections.

	J_1	J_2	J_3	J_4
Arrival time	15	5	2	0
Deadline	21	14	10	19
Execution time	3	4	6	8
Task structure				

(a) (6 Punkte)

Using the Priority Inheritance Protocol (PIP), draw the schedule for the tasks. Use the provided diagram. Mark the critical sections of each task using A and B as done in the last row of the given table. Show the active priorities p_4 of task J_4.

Lösungsvorschlag:



(b) (1 Punkt) Are all deadlines satisfied? If not, what is the maximum lateness in time units?

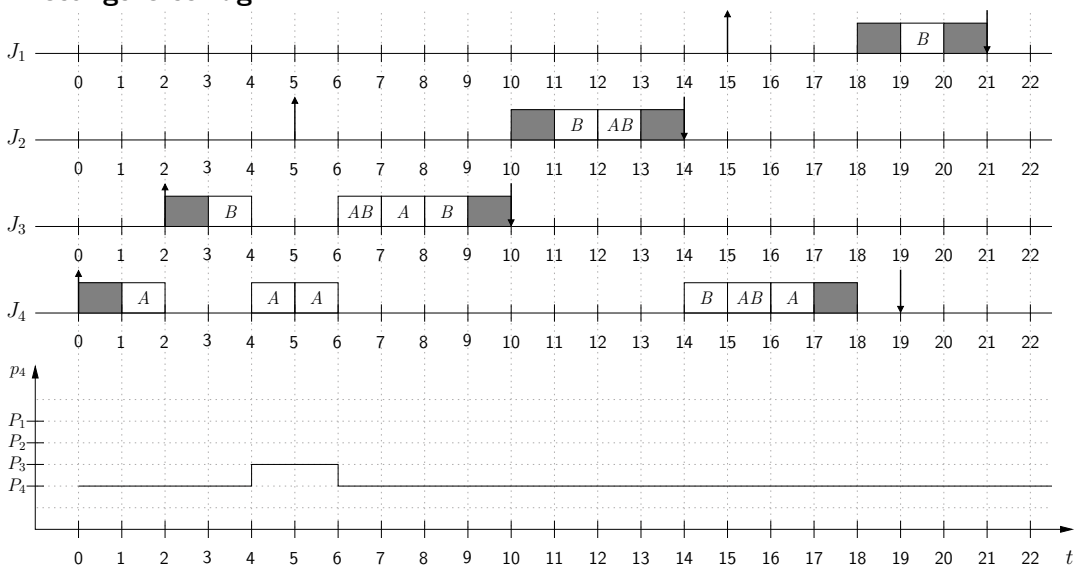
Lösungsvorschlag:
No, jobs J_3 and J_4 miss their deadlines. The maximum lateness is 4 time units (job J_3).

(c) (3 Punkte) Propose a possible change in the task priorities that would allow all deadlines to be met. Give reasons to your answer.

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) Apply the Priority Inheritance Protocol (PIP) for the tasks with the changed priorities and draw the respective schedule. Use the provided diagram. Show the active priorities p_4 of task J_4.

Lösungsvorschlag:



Aufgabe 3 : Low Power Design (maximal 47 Punkte)

3.1: Dynamic Voltage Scaling (maximal 24 Punkte)

Let the power consumption of a processor, running at frequency f Hz, be given as $P = \left(\frac{f}{10^6}\right)^3$ mW. This processor runs by harversting energy from the environment, which is available at a constant rate of 4 mW. Excess energy is buffered in a battery which has unlimited storage capacity. The battery is assumed to be lossless. At the beginning of the execution of the schedule, the battery is almost completely discharged, the initial amount of energy in it is 0.15mJ. A schedule is considered feasible, if all deadlines are met and the available energy never falls below 0.

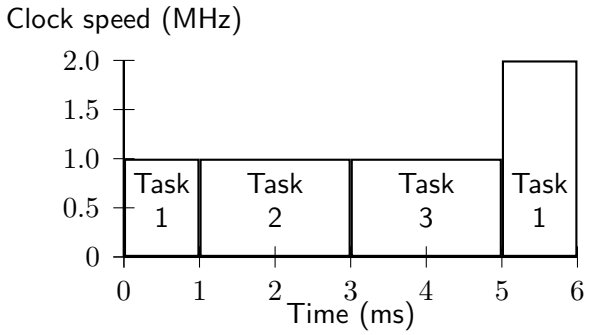
This processor serves the following set of tasks:

Task ID	1	2	3
Arrival time (ms)	0	1	3
Absolute Deadline (ms)	6	2	4
Cycles (x10^3)	3	2	2

Note: Perform all calculations with an accuracy of 2 decimal places using the μJ unit.

(a) (5 Punkte)

The schedule followed is shown below:



Plot the energy remaining in the battery versus time for this schedule using Figure 5. How much energy is stored in the battery after the execution? Is this a valid schedule?

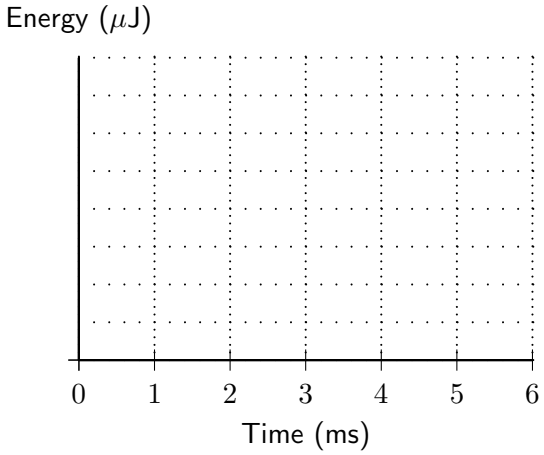


Abbildung 5: Stored energy vs. time for Aufgabe 3.1 (a)

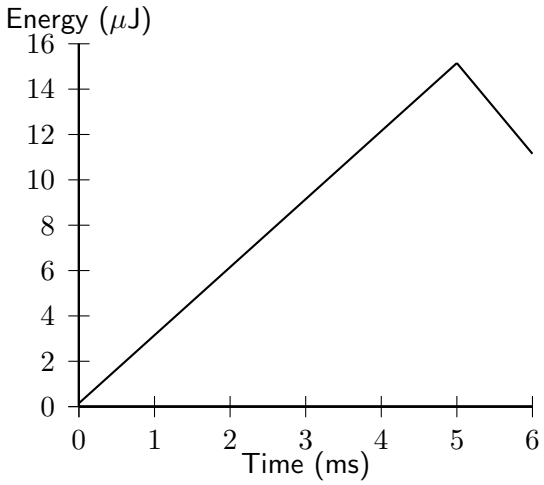


Abbildung 6: Energy vs Time for online given schedule

Lösungsvorschlag: Not valid as deadlines are missed.

□

(b) (8 Punkte) For the same set of tasks, derive the *online* YDS schedule. Plot the schedule in the timeline provided. Plot also the energy left in the battery vs. time in the graph provided. Please use Figures 7 and 8. Is this a valid schedule?

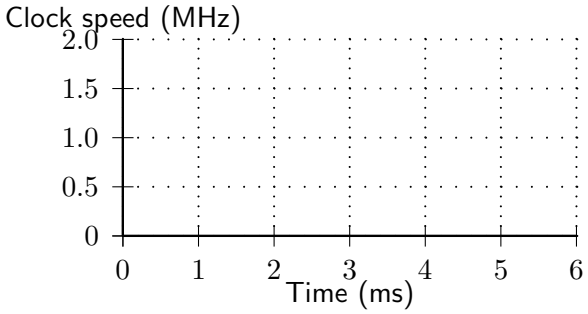


Abbildung 7: Online YDS schedule for Aufgabe 3.1 (b)

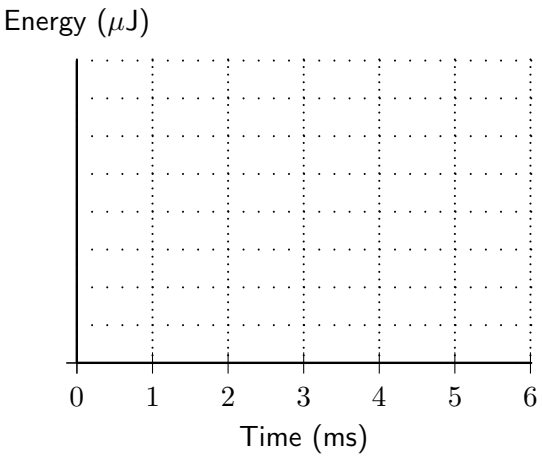


Abbildung 8: Stored energy vs. time for the online YDS schedule

Lösungsvorschlag:

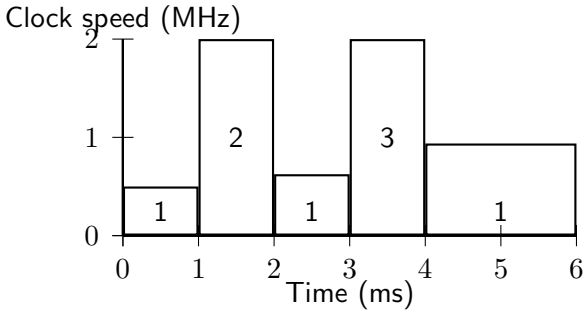


Abbildung 9: Online YDS schedule

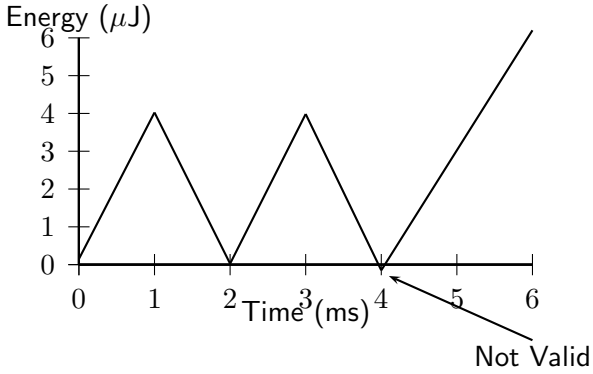


Abbildung 10: Energy vs Time for online YDS schedule

□

(c) (6 Punkte) For the same set of tasks, derive the *offline* YDS schedule. Plot the schedule in the timeline provided. Plot also the energy left in the battery vs. time in the graph provided. Please use Figures 11 and 12. Is this a valid schedule?

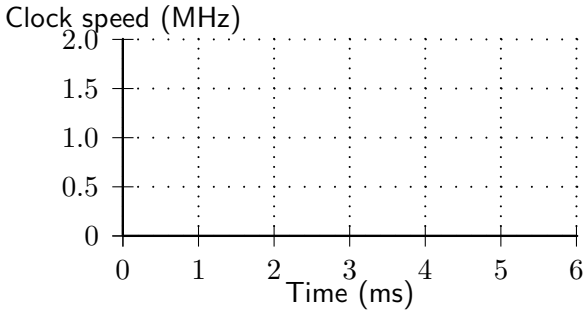


Abbildung 11: Offline YDS schedule for Aufgabe 3.1 (c)

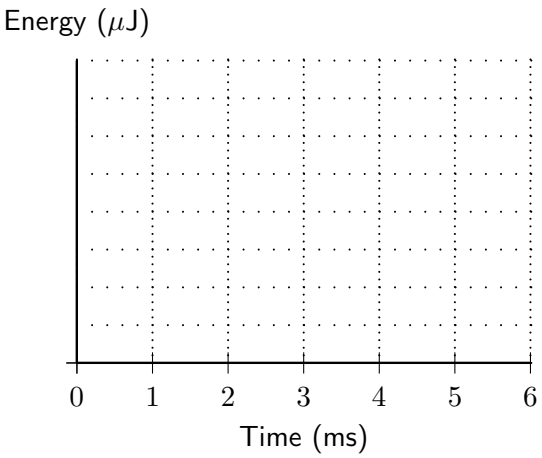


Abbildung 12: Stored energy vs. time for the offline YDS schedule

Lösungsvorschlag:

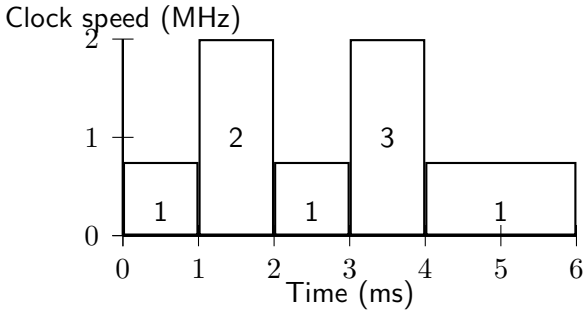


Abbildung 13: Offline YDS schedule

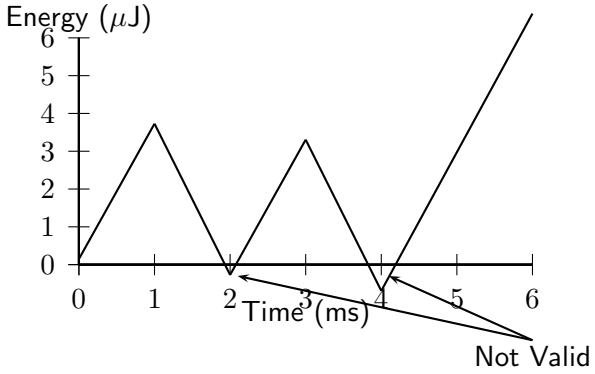


Abbildung 14: Energy vs Time for offline YDS schedule

□

(d) (5 Punkte) For the same task set, derive a feasible schedule which maximizes the remaining energy at the end of the schedule. Plot the schedule in the timeline provided. Plot also the energy left in the battery vs. time in the graph provided. Please use Figures 15 and 16.

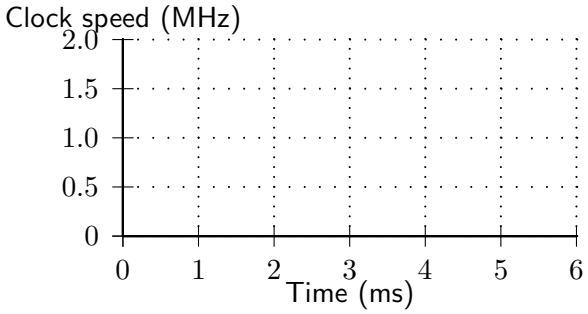


Abbildung 15: Schedule for Aufgabe 3.1 (d)

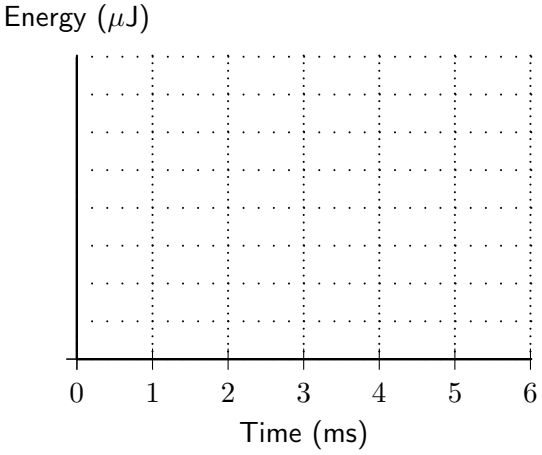


Abbildung 16: Stored energy vs. time for the schedule

Lösungsvorschlag:

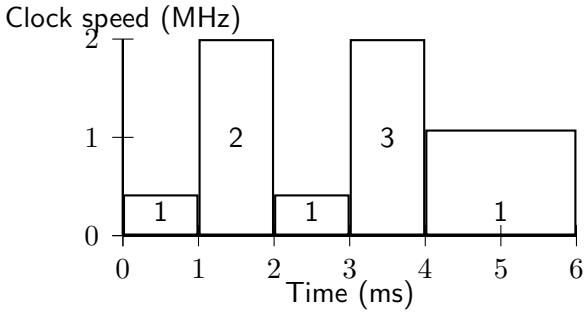


Abbildung 17: Modified schedule

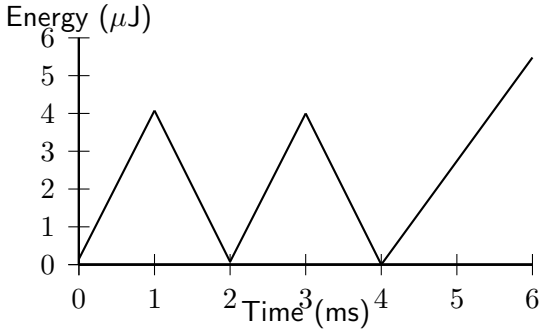


Abbildung 18: Energy vs Time for modified schedule

□

3.2: Dynamic Power Management (maximal 12 Punkte)

Consider a processor such that its power consumption when executing tasks is $P_{active} = 210$mW and when not executing tasks is $P_{idle} = 10$mW. The following set of periodic tasks is executed:

Task ID	1	2
Arrival time (ms)	0	5
Absolute Deadline (ms)	8	10
Execution time (ms)	3	2
Period (ms)	10	10

Definition: $E_{av} = \lim_{N\to\infty} \frac{\text{Energy consumed in } N \text{ Periods}}{N}$

Definition: In a work-conserving schedule, the processor is never idle if there are tasks ready to be processed.

(a) (3 Punkte) Plot the power consumption as a function of time when the above tasks are executed under a work-conserving schedule. Please use Figure 19. What is the value of E_{av}?

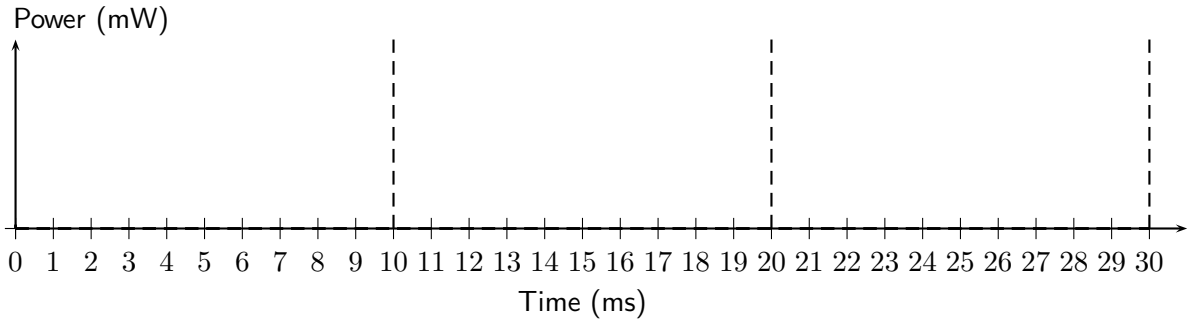


Abbildung 19: Diagram for Aufgabe 3.2 (a)

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

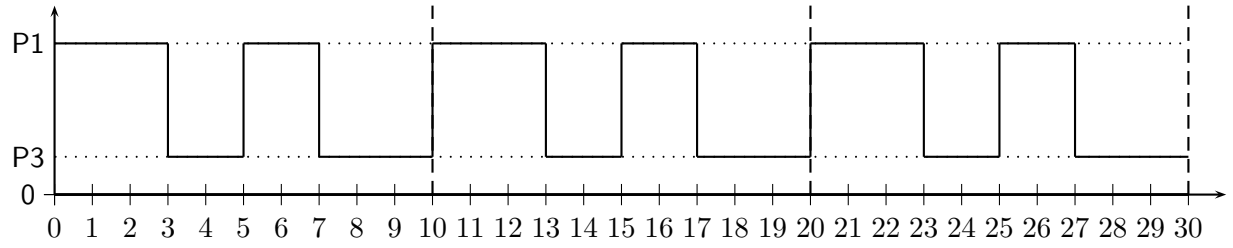


Abbildung 20: Power consumption work-conserving schedule

$E_{av} = 1.1mJ$

□

For the following questions, assume that the above processor can switch to a standby mode in which no energy is consumed. A switch from the normal operating mode to the standby mode, takes 1 ms and incurs an energy overhead of 25μJ which is evenly distributed during this time. A switch from standby mode to the normal operating mode is instantaneous and incurs no energy overhead.

(b) (3 Punkte) Plot the power consumption as a function of time when the above tasks are executed under work-conserving schedule with switches to standby mode whenever there are no tasks waiting to be executed. Please use Figure 21. What is the value of E_{av}?

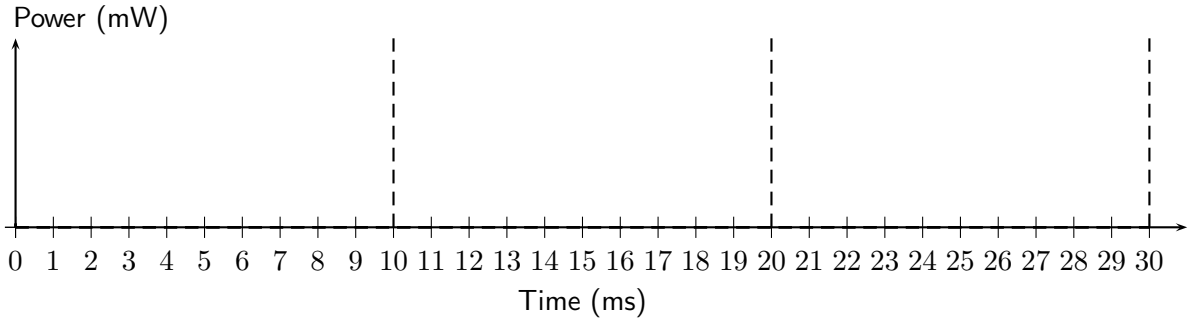


Abbildung 21: Diagram for Aufgabe 3.2 (b)

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

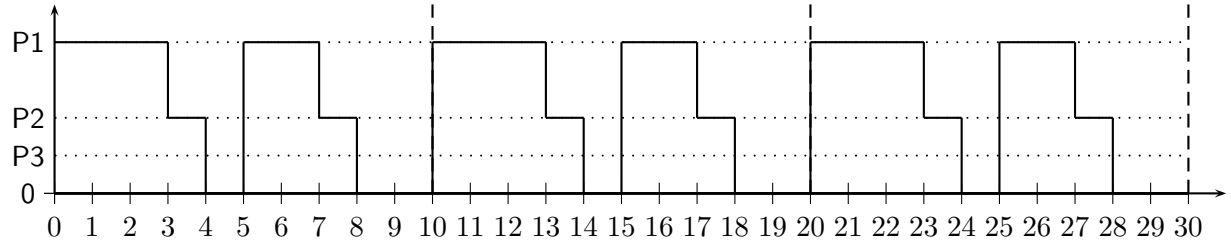


Abbildung 22: Power consumption work-conserving schedule with greedy standby switch

E_{av} = 1.1mJ

□

(c) (3 Punkte) Now assume that the system does NOT necessarily switch to standby mode when there are no tasks to be executed. Plot the power consumption of the work-conserving schedule which minimizes the energy consumed. Please use Figure 23. What is the value of E_{av}?

Lösungsvorschlag: E_{av} = 1.095mJ

□

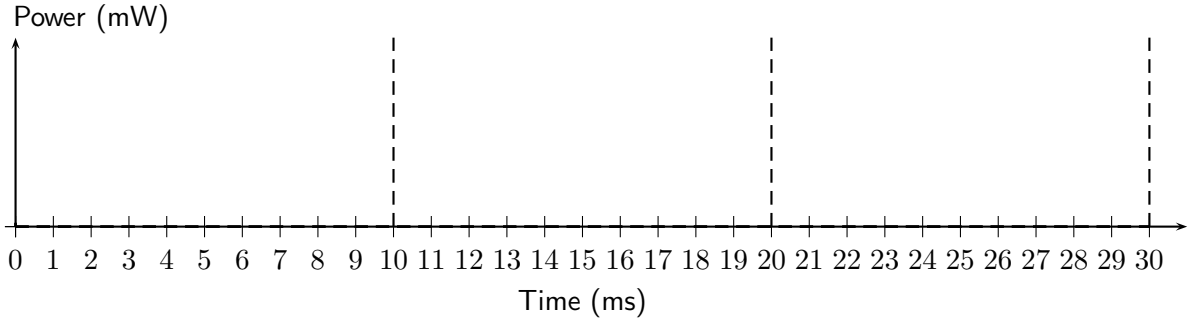


Abbildung 23: Diagram for Aufgabe 3.2 (c)

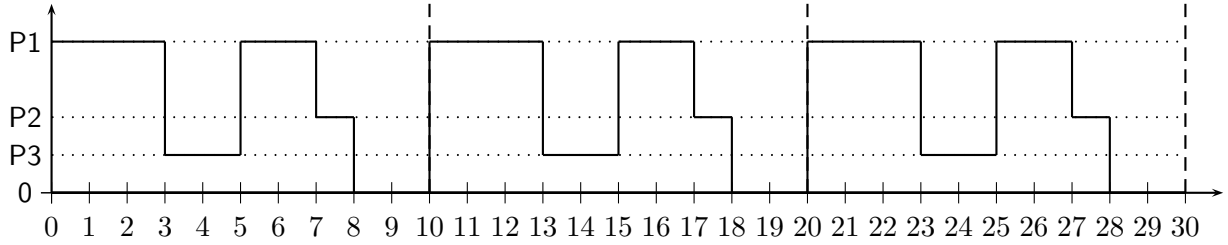


Abbildung 24: Power consumption work-conserving schedule with NO greedy standby switch

(d) (3 Punkte) Now assume that the schedule is NOT necessarily work-conserving, i.e., the only condition is that deadlines are met. Plot the power consumption of the schedule which minimizes the energy consumed. Please use Figure 25. What is the value of E_{av}?

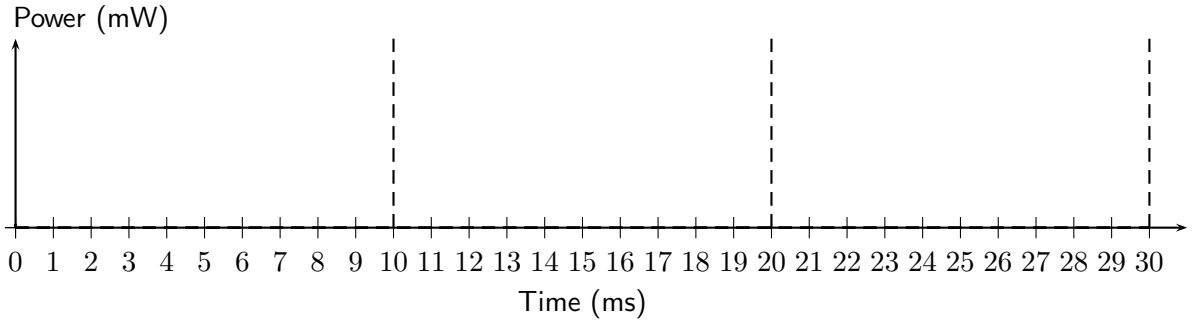


Abbildung 25: Diagram for Aufgabe 3.2 (d)

Lösungsvorschlag: $E_{av} = 1.075mJ$

□

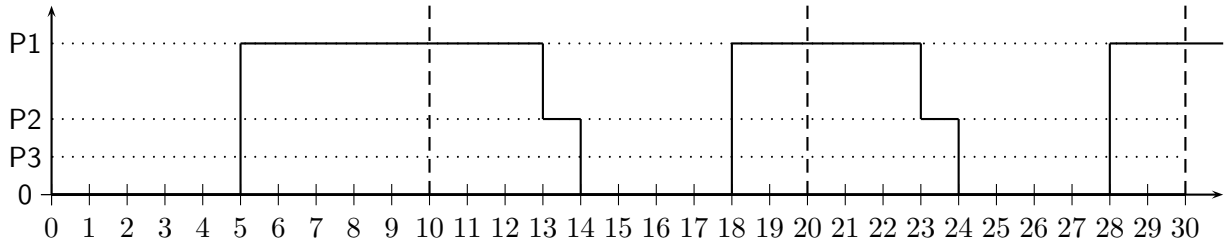
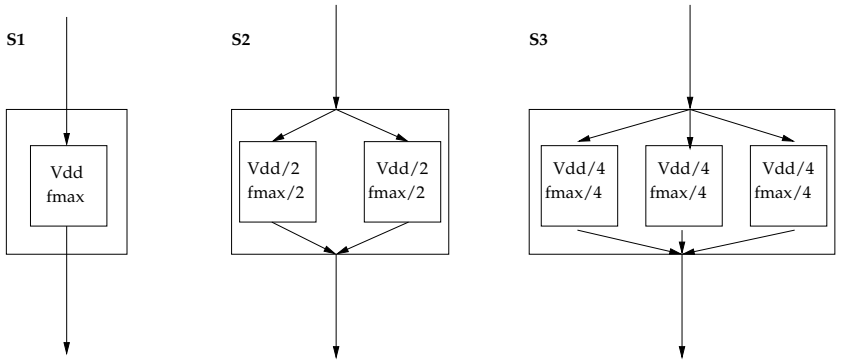


Abbildung 26: Power consumption with no work-conserving restriction and no greedy standby switch

3.3: Multi-Core Systems (maximal 11 Punkte)

Consider the three systems as shown below:



All three systems are based on the same processing core (S1 has one core, S2 two cores, and S3 three cores). The core has a leakage current with a constant leakage power of $P_{leak} = 0.4$ mW, which is always consumed. During execution of a task, additionally speed dependent power P_{dyn} is consumed. It scales cubically with the operating voltage (V_{dd}): $P_{dyn} \sim V_{dd}^3$.

System S1 is configured with the following parameters: $V_{dd} = 1.5$ V and $f_{max} = 100$ MHz. The speed dependent power of system S1 is $P_{dyn} = 100$ mW. The voltage and frequency values of systems S2 and S3 are shown above in the figure. The execution of task τ takes 300 cycles on S1, 300 cycles on S2, and 300 cycles on S3.

Definition: The energy-delay product is the product of: the energy required for the execution of the task and the execution time of the task.

(a) (5 Punkte) Plot the power consumption of the three systems when executing task τ. Please use Figure 27. Calculate the energy-delay products for the three systems when executing task τ.

Lösungsvorschlag:

□

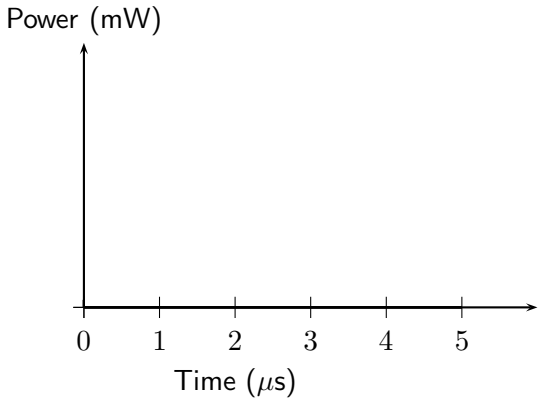


Abbildung 27: Diagram for Aufgabe 3.3 (a)

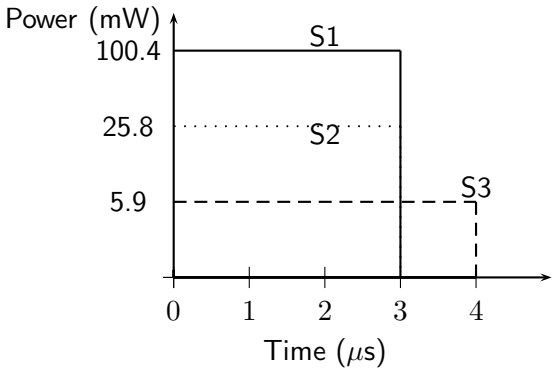


Abbildung 28: Power consumption of the different systems

(b) (4 Punkte) Consider a system with N cores, where each core has a clock speed of f_{max}/N and an operating voltage of V_{dd}/N. The system can execute task τ in $300/N$ cycles. Calculate the optimal number of cores to be used which minimizes the energy-delay product.

Lösungsvorschlag:

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

□

(c) (2 Punkte) Give two reasons why the minimum energy-delay product in a practical system can be achieved for small values of N.

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 : Synthesis (maximal 43 Punkte)

4.1: LIST-Scheduling (maximal 16 Punkte)

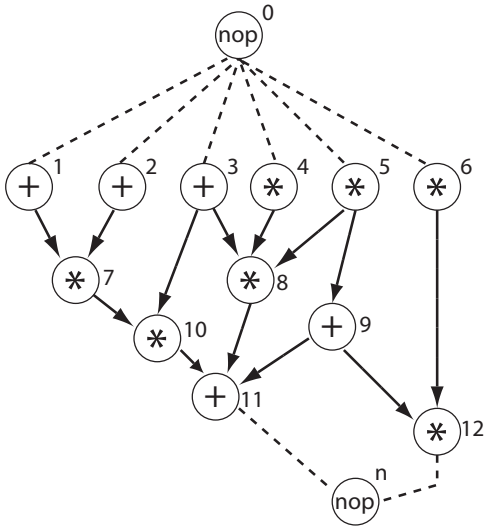


Abbildung 29: Sequence graph

Consider the sequence graph in Figure 29.

(a) (10 Punkte) We have allocated two units of resource type ALU (r_1), and two units of resource type Multiplicator (r_2). Determine a schedule for the given sequence graph with the LIST algorithm and the following assumptions. Addition and subtraction operations can be executed on the two allocated units of resource r_1, and multiplication operations can be executed on the two allocated units of resource r_2. It takes one time unit to execute an addition or subtraction operation, and two time units for a multiplication operation. Consider that the first operation starts at time $t = 0$ and that the *nop* hierarchy nodes take zero execution time. For the priority of an operation, consider the number of successor nodes up to the 'n' nop node. Please use Table 3 for your solution. For each time step t, $U_{t,k}$ denotes the set of operations that are ready to be scheduled on resource k, $S_{t,k}$ denotes the set of operations that start at time t on resource k, and $T_{t,k}$ denotes the set of operations in execution at time t on resource k.

(b) (1 Punkt) What is the calculated latency?

Lösungsvorschlag:
$L = 8$.

□

(c) (1 Punkt) What would be the latency if there were infinite number of resources allocated, and addition/subtraction and multiplication take 1 time unit to execute each.

Lösungsvorschlag:
$L = 4$.

□

(d) (1 Punkt) Does the LIST algorithm calculate the minimum latency for this type of problems? Explain briefly your answer. Name another method that you have learnt in the lecture which is guaranteed to calculate the minimum latency for this type of problems.

Lösungsvorschlag:
No. The LIST algorithm is a heuristic algorithm. ILP will find the optimum solution to the problem (e.g. the minimum latency).

□

Answer the following question in general without considering a specific Sequence graph:

(e) (3 Punkte) Assume that you have to solve the ILP formulation of the problem (without actually solving it). Remember from lecture notes that you have to determine the mobility (i.e. the possible start times) of each task by using the ASAP and ALAP schedules. How would you choose the maximum latency L_{max} for the ALAP schedule? Give reasons.

Lösungsvorschlag:
L_{max} is calculated with the LIST algorithm because this 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: Table for Aufgabe 4.1 (a)

4.2: Timing Constraints (maximal 8 Punkte)

Given are: a sequence graph in Figure 30 and the execution times of the functions from the graph in Table 4.

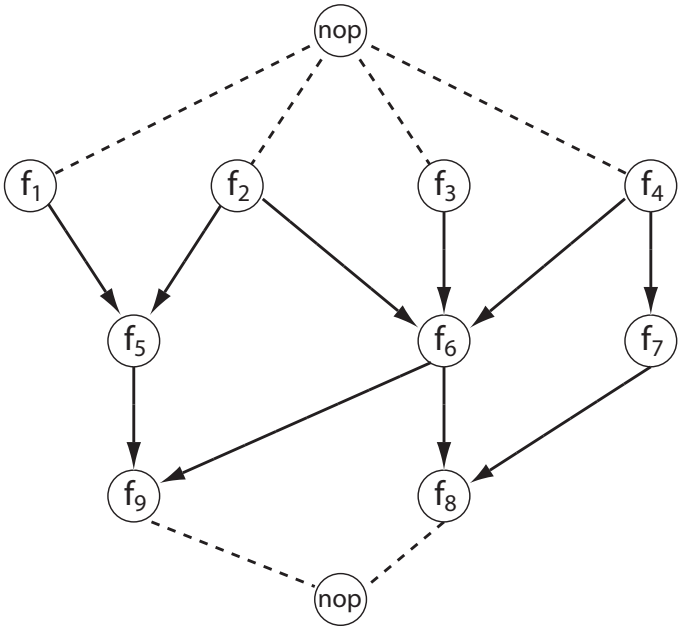


Abbildung 30: Sequence graph

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: Execution times of functions

(a) (2 Punkte) Draw the weighted constraint graph $G_C = (V_C, E_C, d)$. You can use the graph in Figure 30.

Lösungsvorschlag:
See Figure 31.

□

(b) (3 Punkte) Draw the following additional timing constraints in the weighted constraint graph G_C (again you can add them to the graph shown in Figure 30.

Lösungsvorschlag:
An edge $(\nu_i, \nu_j) \in G_C$ with weight $d(\nu_i, \nu_j)$ means: $\tau(\nu_j) - \tau(\nu_i) \geq d(\nu_i, \nu_j)$. See Figure 32.

□

i) f_6 may start at the earliest 6 time units after the start of f_5.

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

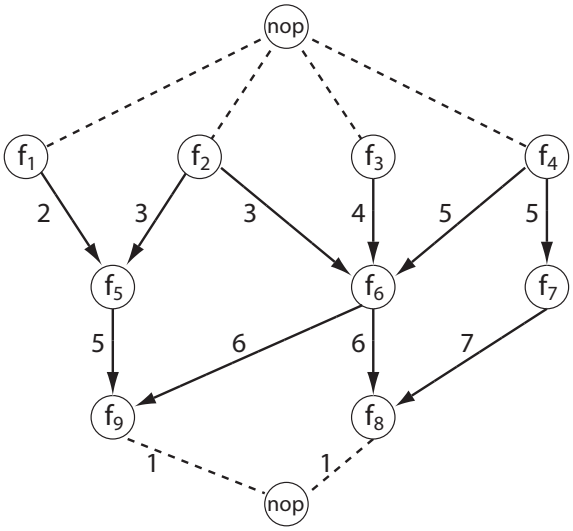


Abbildung 31: *Possible solution:* Weighted constraint graph for Aufgabe 4.2.a)

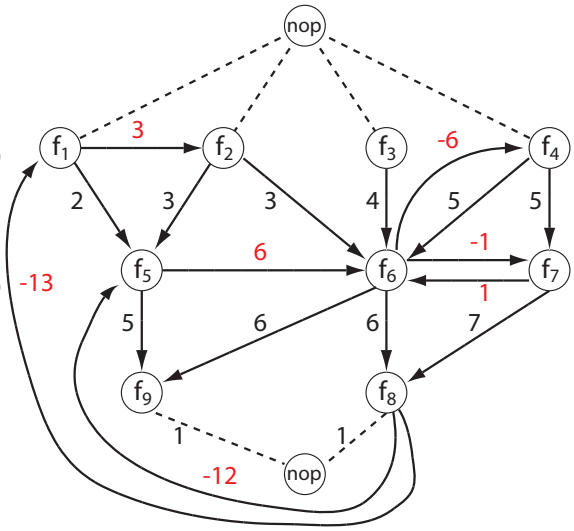


Abbildung 32: *Possible solution:* Weighted constraint graph for Aufgabe 4.2.b)

□

ii) f_6 must start exactly 1 time unit after the start of f_7.

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 may start at the latest 6 time units after the start of f_4.

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

□

iv) f_2 may start at the earliest 3 time units after the start of f_1.

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

□

v) f_8 must start at the latest 7 time units after the finish of f_5.

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

□

vi) f_8 must start at the latest 11 time units after the finish of f_1.

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

□

(c) (3 Punkte) Is there a valid schedule under the given timing constraints assuming that you have unlimited resources? Give reasons for your answer.

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: Iterative Algorithms (maximal 19 Punkte)

Given is an iterative algorithm which is specified with the marked graph shown in Figure 33. The execution times of all operations are given in Table 5.

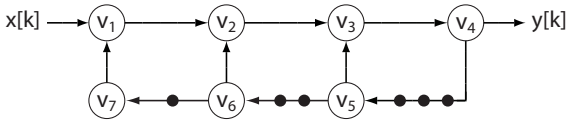


Abbildung 33: Marked Graph

	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: Execution times

(a) (2 Punkte) Represent the data dependencies from the marked graph with an extended sequence graph $G_S = (V_S, E_S, d)$.

Lösungsvorschlag:
Solution shown in Figure 34.

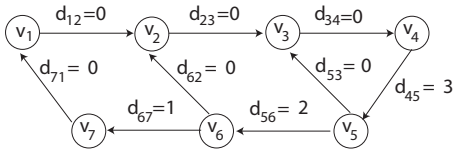


Abbildung 34: Extended sequence graph

□

(b) (3 Punkte) Express all data dependencies using inequations of the following type:

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

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) Assume that you have unlimited number of resources. Draw a schedule for 4 iterations without functional pipelining and without loop folding. Draw all dependencies on the schedule including the ones between iterations. Use Figure 35.

Lösungsvorschlag:
See Figure 35.

□

(d) (8 Punkte) Assume that you have unlimited number of resources. Determine for the iterative algorithm with loop folding:

- the minimum iteration interval P
  (e.g. by substituting in the results from point b)
  and

  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.

  □

- a valid schedule of operations v_i for 4 iterations in the form of a drawing. For the purpose, use Figure 36. Highlight on the drawing the different iterations and dependencies (e.g. by writing or marking).

  Lösungsvorschlag:
  See Figure 36.

  □

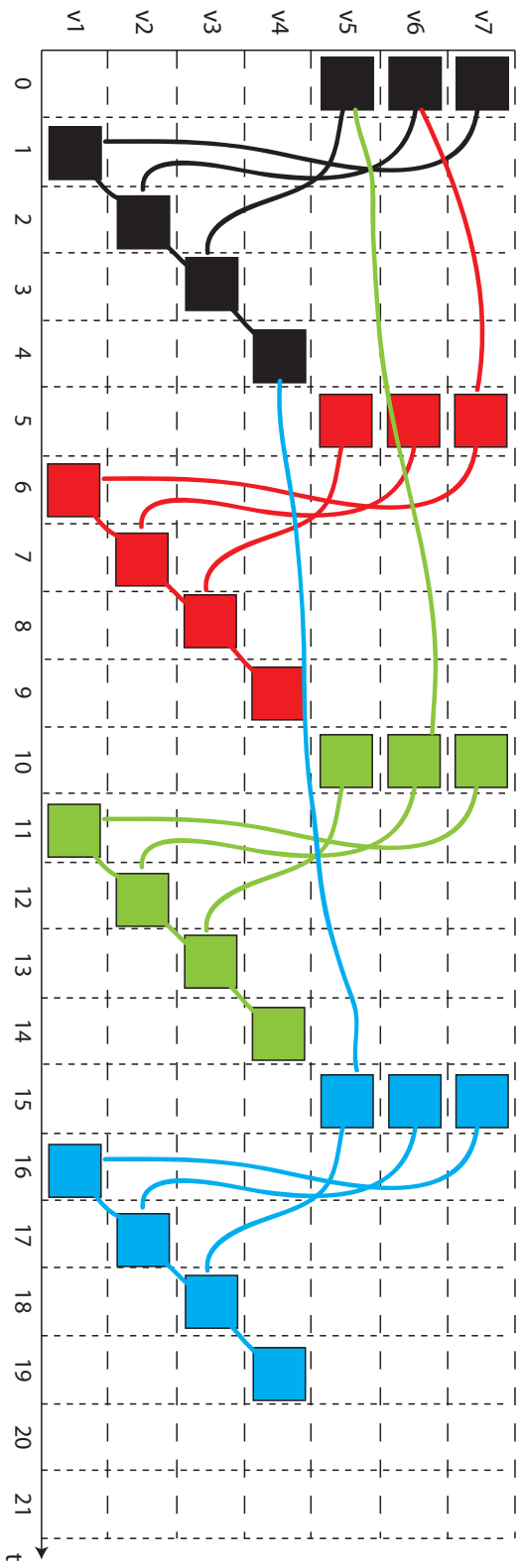


Abbildung 35: Schedule without loop folding

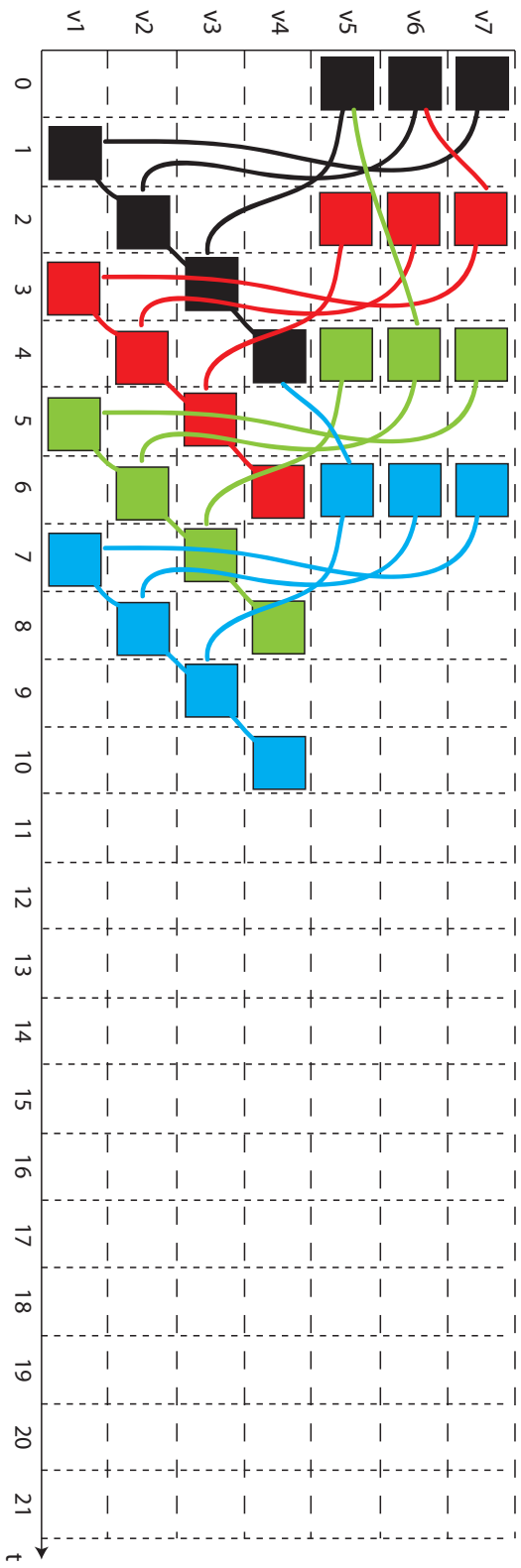


Abbildung 36: Scedule with loop folding