	<p>Fachbereich: Informatik & Ingenieurwissenschaften</p> <p>Studiengang: Informatik</p> <p>Vorlesung: 06-3016/2030 Realzeitsysteme</p> <p>Klausur - Lösungen</p>
Doz.: Döben-Henisch	doeben_at_fb2.fh-frankfurt.de
	1//20

AUFGABE 1: Max. 16 Pkt; Erreichte Pkt: _____

Gegeben seien die folgenden drei periodischen Tasks $\{t_1, t_2, t_3\}$:

- t_1 ($r_0 = 0, C = 5, D = 25, T = 30$)

- t_2 ($r_0 = 0, C = 10, D = 40, T = 50$)

- t_3 ($r_0 = 0, C = 20, D = 55, T = 75$)

A1.1)(8 Pkt) Stellen Sie fest, ob die einzelnen Ereignisse planbar sind.

A1.2)(4 Pkt) Stellen sie einen Plan mit Hilfe des EDF-Algorithmus auf.

A1.3)(1 Pkt) Geben Sie die Zeiten an, während deren der Prozessor untätig ('dle') ist.

A1.4)(3 Pkt) Klären sie für die folgenden aperiodischen Tasks, ob Sie unter Voraussetzung der verfügbaren freien Zeiten des Prozessors garantiert ausgeführt werden können; Zeichnen Sie die aperiodischen Tasks in ihren Plan ein.

- t_4 ($r = 40, C = 10, D = 25, T = 15$)

- t_5 ($r = 70, C = 15, D = 40, T = 35$)

- t_6 ($r = 100, C = 20, D = 55, T = 40$)

- t_7 ($r = 105, C = 5, D = 25, T = 25$)

- t_8 ($r = 120, C = 5, D = 40, T = 15$)

Lösung Aufg.1

A1.1)(8 Pkt) Stellen Sie fest, ob die einzelnen Ereignisse planbar sind.

ANHANG 5: Planbarkeit eines einzelnen Ereignisses

Berechne e1:

1. Identifiziere eine Menge (H) von Ereignissen, deren Priorität höher oder gleich der Priorität P_i von e_i sind. Teile dann die Menge (H) auf in zwei Mengen (H1) und (Hn). (H1) enthält solche Ereignisse, deren Perioden T grösser oder gleich der Deadline D_i von e_i sind. (Hn) soll solche Ereignisse enthalten, deren Perioden T kleiner als die Deadline D_i von e_i sind.

Gegeben sind die Tasks:

- t1 ($r_0 = 0$, $C = 5$, $D = 25$, $T = 30$)

- t2 ($r_0 = 0$, $C = 10$, $D = 40$, $T = 50$)

- t3 ($r_0 = 0$, $C = 20$, $D = 55$, $T = 75$)

Festlegung von Prioritäten, hier nach Deadlines:

- t1 ($r_0 = 0$, $C = 5$, $D = 25$, $T = 30$) -> 3

- t2 ($r_0 = 0$, $C = 10$, $D = 40$, $T = 50$) -> 2

- t3 ($r_0 = 0$, $C = 20$, $D = 55$, $T = 75$) -> 1

Berechnung von e1:

$H = \{\}$, also $H1 = \{\} = Hn$

2. Berechne die totale effektive Nutzungszeit f_i von Ereignis e_i . Diese totale Nutzungszeit setzt sich aus zwei Bestandteilen zusammen: (i) die totale Nutzungszeit der Ereignisse in (H_n) , sowie (ii) die Ausführungszeit von e_i plus Verspätungszeiten durch Blockaden plus Präemptionszeiten verursacht durch Ereignisse aus (H_1) , alles dividiert durch die Periode von e_i :

$$f_i = \text{SUM}(j \text{ in } (H_n)) C_j / T_j + 1/T_i * (C_i + B_i + \text{SUM}(k \text{ in } (H_1)) C_k)$$

Berechne f_1 von e_1 :

$$f_1 = 0 + 1/T_1 * (C_1 + B_1 + 0) = C_1/T_1 = 5/30 = 1/6 = 0.17$$

3. Berechne die Nutzbarkeitsgrenze $U(n, Dt_i)$ mit $n = \text{Anzahl Elemente in } (H_n) + 1$ sowie $Dt_i = D_i/T_i$. Es gilt:

$$U(n, Dt_i) =$$

$$(i) n * ((2 * Dt_i)^{1/n} - 1) + 1 - Dt_i \text{ falls } 0.5 < Dt_i \leq 1.0$$

$$(ii) Dt_i \text{ falls } 0 \leq Dt_i \leq 0.5$$

$$n = 0 + 1 = 1$$

$$Dt_1 = D_1/T_1 = 25/30 = 5/6 = 0.83$$

wegen $Dt_1 > 0.5$ gilt:

$$U(1, Dt_1) = 1 * ((2 * 0.83)^{(1/1)} - 1) + 1 - 0.83 = 1 * (1.67)^1 - 1 + 1 - 0.83 = 0.83$$

4. Vergleiche die effektive totale Nutzungszeit von e_i mit der oberen Schranke:

$$f_i \leq U(n, Dt_i)$$

Falls f_i kleiner oder gleich der Grenze ist, dann kann dieses Ereignis seine Deadline einhalten.

$$f_1 \leq U(1, Dt_1)$$

$$0.17 \leq 0.83$$

Berechne e2:

$$H = \{t1\}, H1 = \{\}, Hn = \{t1\}$$

$$f2 = \text{SUM}(j \text{ in } (Hn))C_j/T_j + 1/T_2 * (C_2 + B_2 + \text{SUM}(k \text{ in } (H1))C_k) \\ = C1/T1 + C2/T2 = 5/30 + 10/50 = 55/150 = 11/30 = 0.37$$

$$n = 1 + 1 = 2$$

$$Dt2 = C2/T2 = 10/50 = 0.2$$

$$U(2, Dt2) = n * ((2 * Dt2)^{1/n} - 1) + 1 - Dt2$$

$$U(2, Dt2) = 2 * ((2 * 0.2)^{1/2} - 1) + 1 - 0.2 = 2 * (1.4142 - 1) + 1 - 0.2 = 2 * 0.4142 + 0.8 = 0.8284 + 0.8 = 1.6284$$

$$f2 \leq U(n, Dt2)$$

$$0.37 \leq 1.6284$$

Berechne e3:

$$H = \{t2, t1\}, H1 = \{\}, Hn = \{t2, t1\}$$

$$f3 = \text{SUM}(j \text{ in } (Hn))C_j/T_j + 1/T_3 * (C_3 + B_3 + \text{SUM}(k \text{ in } (H1))C_k) \\ = C1/T1 + C2/T2 + C3/T3 = 5/30 + 10/50 + 20/75 = 19/30 = 0.63$$

$$n = 2 + 1 = 3$$

$$Dt3 = C3/T3 = 20/75 = 0.27$$

$$U(3, Dt3) = n * ((2 * Dt3)^{1/n} - 1) + 1 - Dt3$$

$$U(3, Dt3) = 3 * ((2 * 0.27)^{1/3} - 1) + 1 - 0.27 = 3 * (1.1344 - 1) + 1 - 0.27 = 3 * 0.1344 + 0.73 = 0.4032 + 0.73 = 1.1332$$

$$f3 \leq U(3, Dt3)$$

$$0.63 \leq 1.1332$$

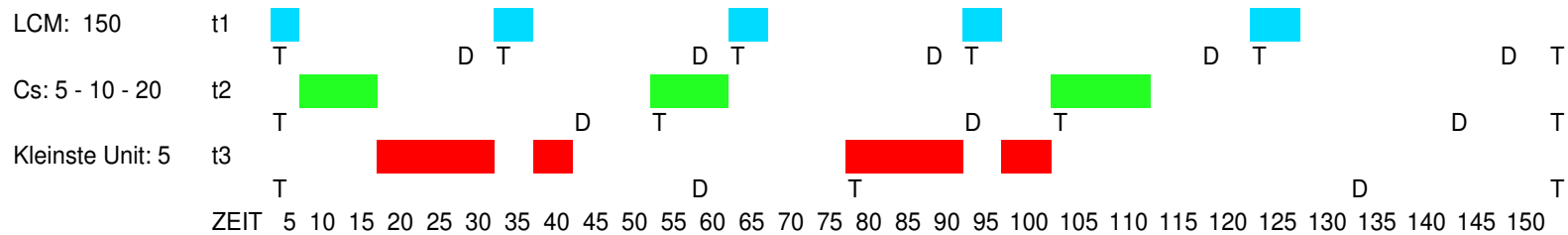
Lösung Aufg.1

A1.2)(4 Pkt) Stellen sie einen Plan mit Hilfe des EDF-Algorithmus auf.

Der *Earliest Deadline First Algorithmus (EDF)*: dieser ordnet einem Task dynamisch eine Priorität P zu entsprechend der aktuellen absoluten Deadline d; je kürzer die absolute Deadline d umso grösser die Priorität.

Im Falle von Mehrdeutigkeiten kann nach einer beliebigen regel verfahren werden!

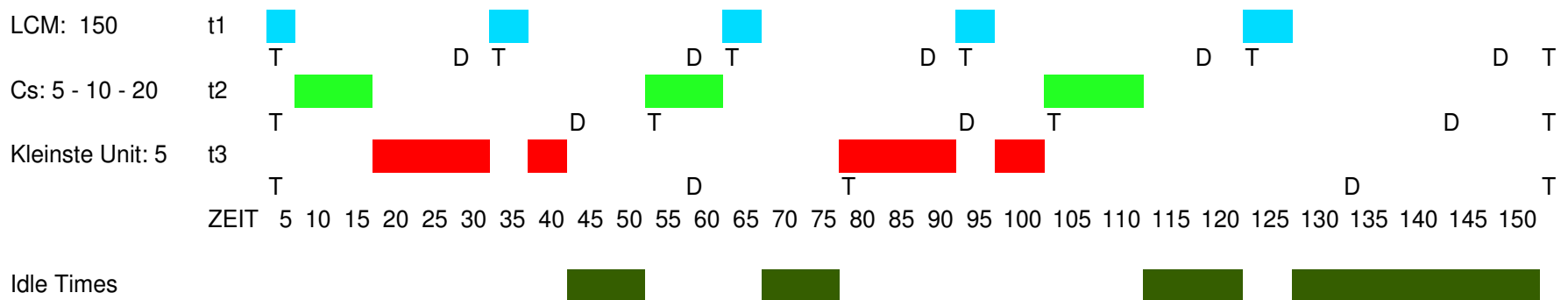
Perioden: 30 - 50 -75



Lösung Aufg.1

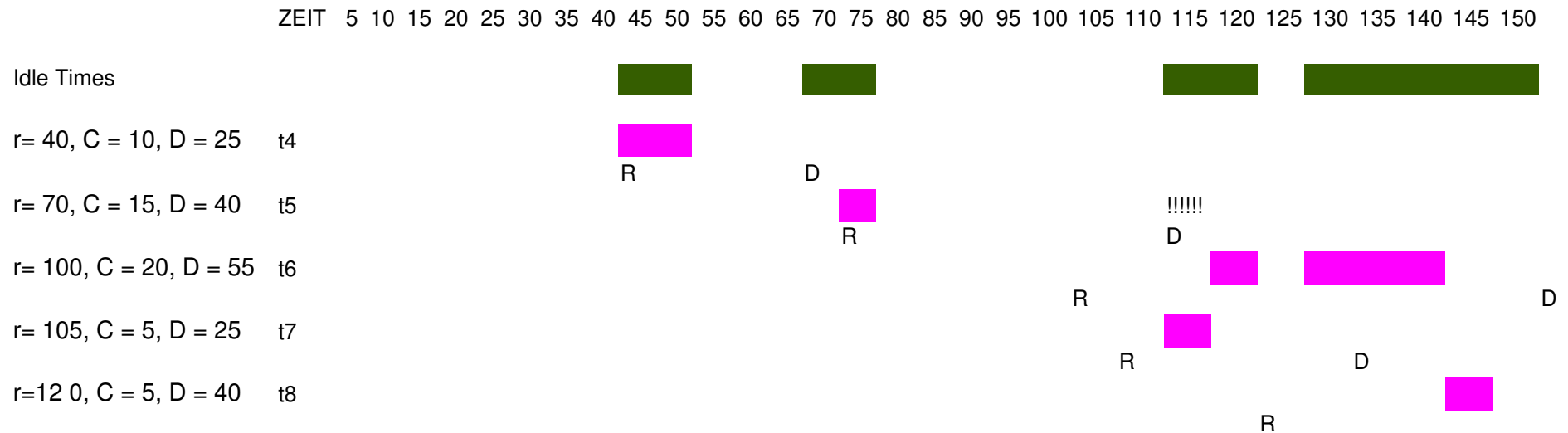
A1.3)(1 Pkt) Geben Sie die Zeiten an, während deren der Prozessor untätig ('dle')st.

Perioden: 30 - 50 -75



Lösung Aufg.1

A1.4)(3 Pkt) Klären sie für die folgenden aperiodischen Tasks, ob Sie unter Voraussetzung der verfügbaren freien Zeiten des Prozessors garantiert ausgeführt werden können; Zeichnen Sie die aperiodischen Tasks in ihren Plan ein.



AUFGABE 2: Max. 16 Pkt; Erreichte Pkt: _____

Berücksichtigen sie die folgenden drei präemptiven Planungsalgorithmen:

- Der *Rate Monotonic Algorithmus (RM)*: dieser ordnet feste Prioritäten zu Tasks entsprechend ihrer Periode zu; je kürzer die Periode T , umso höher die Priorität P .
- Der *Earliest Deadline First Algorithmus (EDF)*: dieser ordnet einem Task dynamisch eine Priorität P zu entsprechend der aktuellen absoluten Deadline d ; je kürzer die absolute Deadline d umso grösser die Priorität.
- Der *Least Laxity first Algorithmus (LLF)*: ordnet den Tasks dynamisch Prioritäten entsprechend ihrer relativen Laxity zu. Die relative Laxity soll

immer bei Eintreffen eines neuen Tasks berechnet werden.

Gegeben seien die folgenden drei periodischen Tasks $\{t_1, t_2, t_3\}$:

- t_1 ($r_0 = 0, C = 1, D = 3, T = 3$)

- t_2 ($r_0 = 0, C = 1, D = 4, T = 4$)

- t_3 ($r_0 = 0, C = 2, D = 3, T = 6$)

A2.1)(2 Pkt) Berechnen Sie die *totale Auslastung* ('*utilization*') $U(S,3)$ des Systems S mit einem Prozessor und den drei genannten Tasks.

A2.2)(4 Pkt) Berechne einen Plan mit dem RM Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

A2.3)(4 Pkt) Berechne einen Plan mit dem EDF-Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

A2.4)(6Pkt) Berechne einen Plan mit dem LLF-Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

Lösung Aufg.2

A2.1)(2 Pkt) Berechnen Sie die *totale Auslastung* ('*utilization*') $U(S,3)$ des Systems S mit einem Prozessor und den drei genannten Tasks.

Die *totale Auslastung* ('*utilization*') $U(S,n)$ eines Systems S erhält man dann durch den Ausdruck

$$U(S,n) = \sum_{i=1}^n c_i/p_i$$

mit

$n :=$ Anzahl der verschiedenen Tasks (bzw. Aktionen). Es muss gelten $U(S,n) \leq 1$

- t_1 ($r_0 = 0, C = 1, D = 3, T = 3$)

- t_2 ($r_0 = 0, C = 1, D = 4, T = 4$)

- t_3 ($r_0 = 0, C = 2, D = 3, T = 6$)

$$U(S,3) = 1/3 + 1/4 + 2/6 = 11/12 = 0.92$$

Lösung Aufg.2

A2.2)(4 Pkt) Berechne einen Plan mit dem RM Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

Der *Rate Monotonic Algorithmus (RM)*: dieser ordnet feste Prioritäten zu Tasks entsprechend ihrer Periode zu; je kürzer die Periode T, umso höher die Priorität P.

- t1 ($r_0 = 0, C = 1, D = 3, T = 3$) -> 3

- t2 ($r_0 = 0, C = 1, D = 4, T = 4$) -> 2

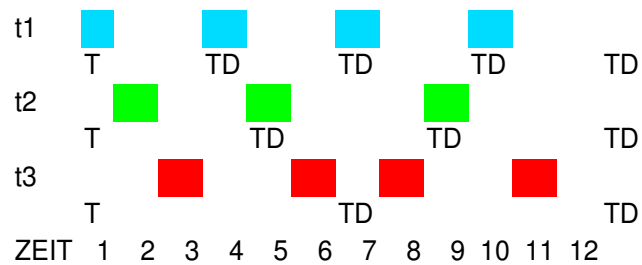
- t3 ($r_0 = 0, C = 2, D = 3, T = 6$) -> 1

Perioden: 3 - 4 - 6

LCM: 12

Cs: 1 - 1 - 2

Kleinste Unit: 1

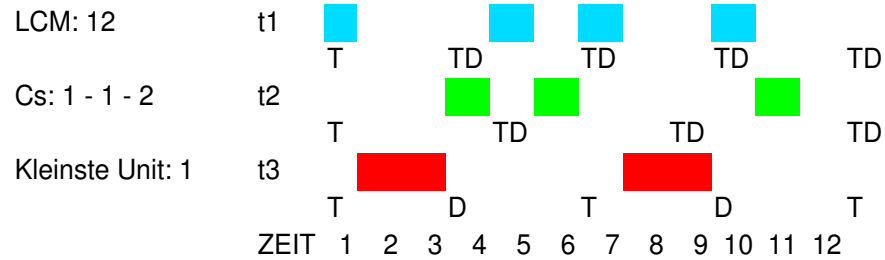


Lösung Aufg.2

A2.3)(4 Pkt) Berechne einen Plan mit dem EDF- Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

Der *Earliest Deadline First Algorithmus (EDF)*: dieser ordnet einem Task dynamisch eine Priorität P zu entsprechend der aktuellen absoluten Deadline d; je kürzer die absolute Deadline d umso grösser die Priorität.

Perioden: 3 - 4 - 6



Lösung Aufg.2

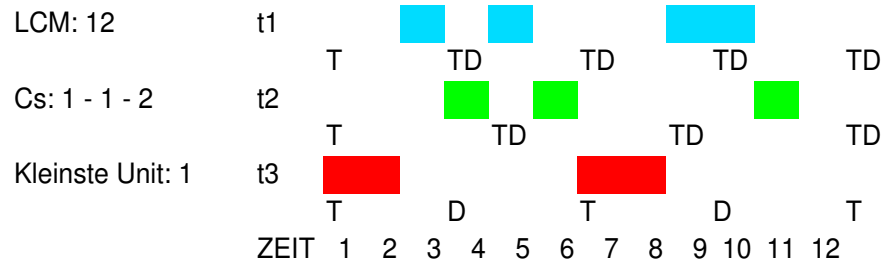
A2.4)(6 Pkt) Berechne einen Plan mit dem LLF- Algorithmus (Auftreten eines Ereignisses: Pfeil nach oben; deadline: Pfeil nach unten)

Der *Least Laxity first Algorithmus (LLF)*: ordnet den Tasks dynamisch Prioritäten entsprechend ihrer relativen Laxity zu. Die relative Laxity soll immer bei Eintreffen eines neuen Tasks berechnet werden.

Ausgehend von der absoluten Deadline d_i und der Ausführungszeit c_i eines Tasks kann man die *laxity* l_i des Tasks berechnen bezogen auf den aktuellen Zeitpunkt: $l_i = d_i - c_i$

Die *restliche nominale* --oder auch *relative*-- Laxity $L(t)$ berechnet sich mit $L(t) = D(t) - C(t)$ oder $L(t) = r+D-t-C(t)$

Perioden: 3 - 4 - 6



AUFGABE 3: Max. 20 Pkt; Erreichte Pkt: _____

Im Anhang 1 dieser Klausuraufgabe ist eine Schaltung zur Erzeugung bzw. zum Messen von Impulsen abgebildet (mit kurzer Beschreibung). Beschreiben Sie, wie Sie unter Benutzung dieser Schaltung die folgenden Aufgaben lösen würden:

A3.1)(5 Pkt) Messen sie den *Abstand* zwischen zwei Impulsen; messen Sie die *Frequenz* von Impulsen

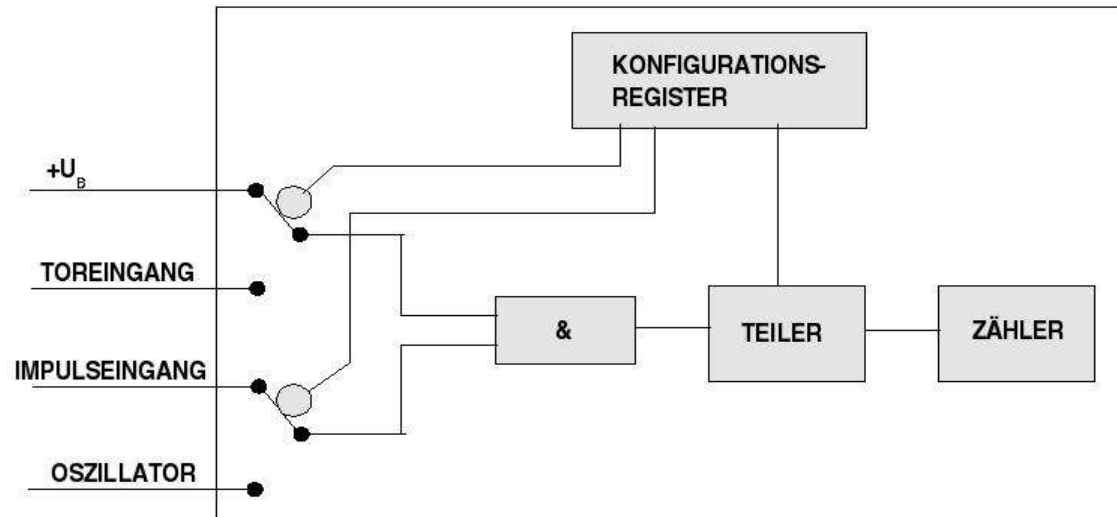
A3.2)(5 Pkt) Wie kann man einen *zyklischen Interrupt* erzeugen, der dadurch gekennzeichnet ist, dass nur nach *Pulsweiten mit einer festen Länge* ein Interrupt erzeugt wird?

A3.3)(5 Pkt) Sie sollen mit der Periode T bei einem bestimmten Gerät die Leistung für ein definiertes Zeitintervall L ausschalten. Wie können sie dies tun?

A3.4)(5 Pkt) Mechanische Bauteile können *Encoder* enthalten, die bei Bewegung pro mm oder Winkelgrad eine bestimmte Menge von Impulsen abgeben. Wie können Sie dann feststellen, welche Strecke zwischen zwei Zeitpunkten zurückgelegt wurde? Wie könnten Sie die Geschwindigkeit (Wegstrecke/Zeit) dieser Bewegung ermitteln?

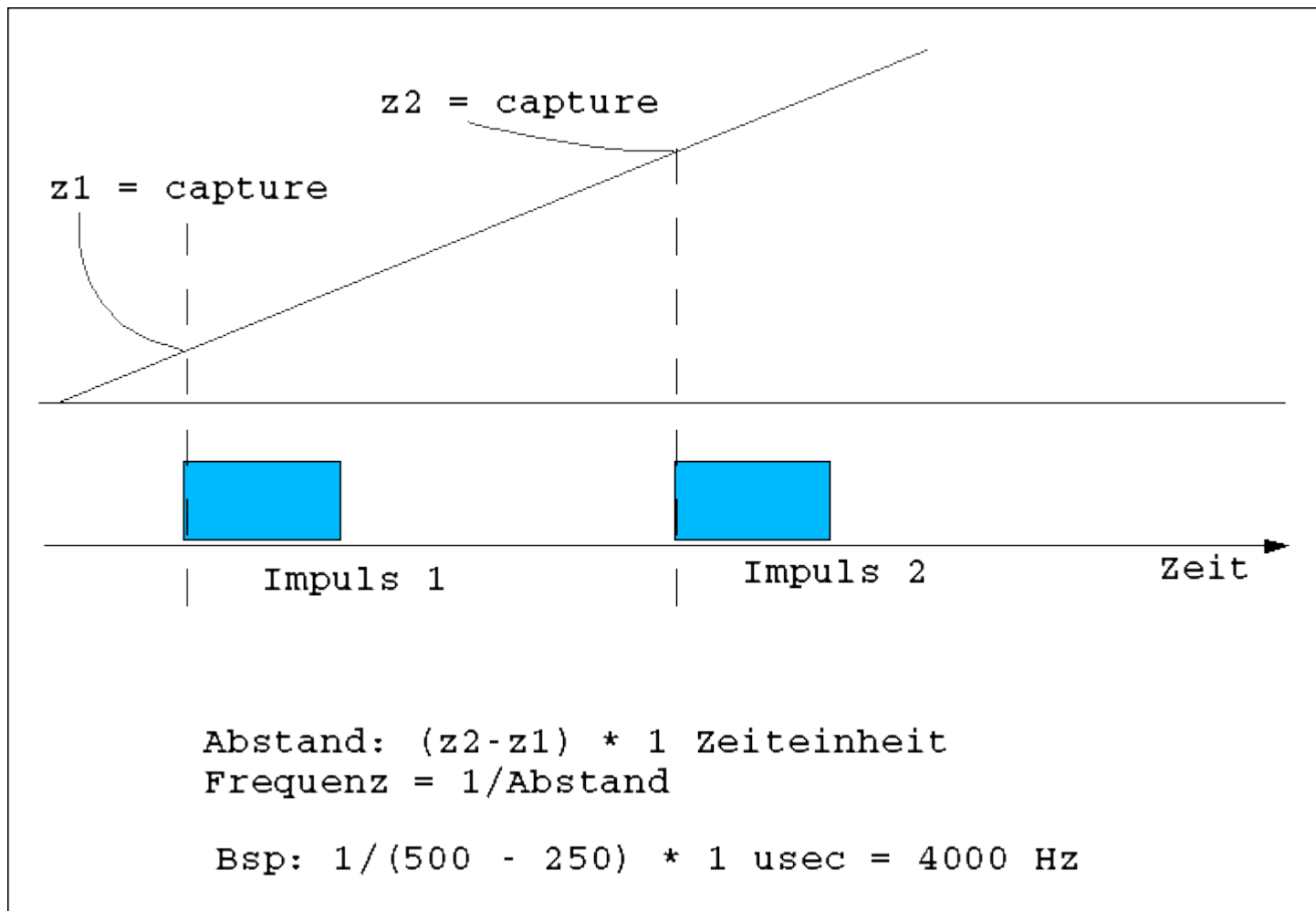
Lösung Aufg.3

A3.1)(5 Pkt) Messen sie den *Abstand* zwischen zwei Impulsen; messen Sie die *Frequenz* von Impulsen



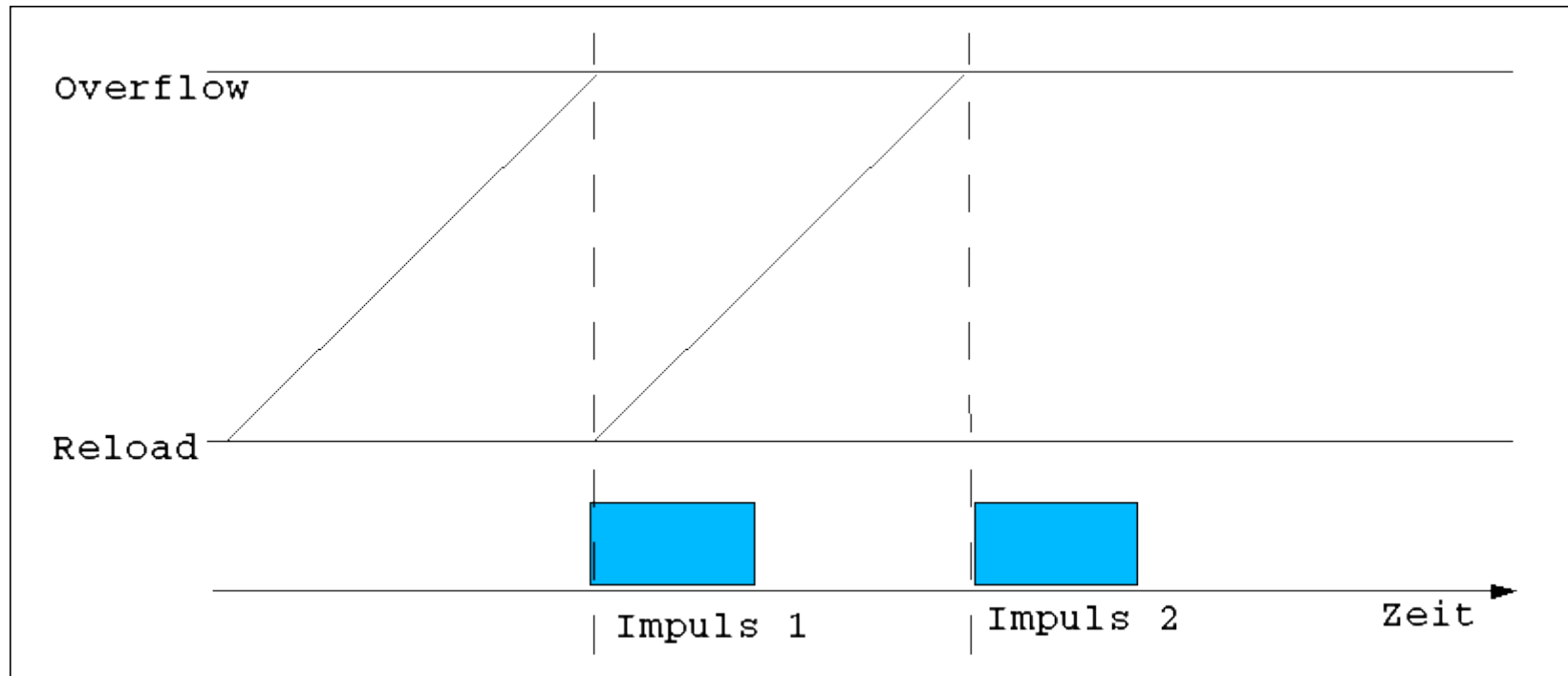
Die Grundeigenschaften dieser Schaltung werden wie folgt angenommen: Bei *konstanter Vorspannung* durch +U_B oder nur bei Vorliegen einer bestimmten *Torspannung* zählt ein Zähler einzelne *Impulse* oder die Impulse eines *Oszillators*.. Zusätzlich kann in allen vier Fällen der Zählvorgang durch einen *Teiler* modifiziert werden. Ein Zähler kann *aufwärts* oder *abwärts* laufen. Beim *Zählerüberlauf* kann ein Zähler aus einem Register mit einem *bestimmten Wert geladen* werden einen *Interrupt* auslösen und/oder einen *Pegelwechsel* an einem Anschluss verursachen. Die Abfrage eines Zählers kann mittels *capture* erfolgen oder mittels *compare*. Im Falle von *capture* wird der Inhalt des Zählerregisters gelesen, ohne dass der Zähler angehalten wird; im Falle von *compare* wird der Inhalt des Zählerregisters mit einem anderen Wert verglichen. Falls ungleich läuft der Zähler weiter, falls gleich wird ein Interrupt oder ein Pegelwechsel ausgelöst.

Bei einem Zähler mit N-Bit Breite, einem Startwert S und einer Inkrementzeit T_i beträgt die Zeitdauer T_O zwischen zwei Überläufen: $T_O = (2^N - S) * T_i$. Bei einem Systemtakt von z.B. 1 MHz würde ein Takt 1/1 MHz = 1*10⁻⁶sec = 1 Mikrosekunde dauern.



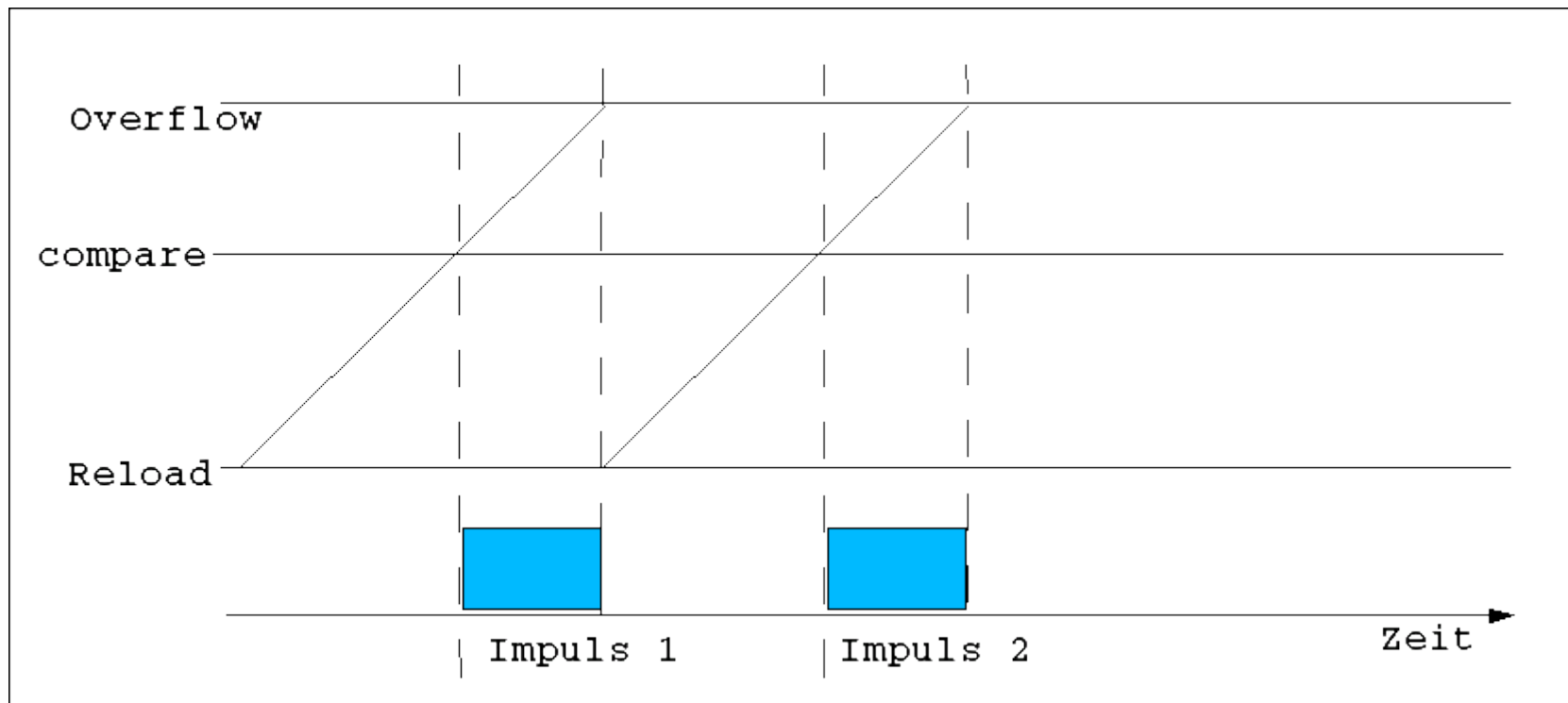
Lösung Aufg.3

A3.2)(5 Pkt) Wie kann man einen *zyklischen Interrupt* erzeugen, der dadurch gekennzeichnet ist, dass nur nach *Pulsweiten mit einer festen Länge* ein Interrupt erzeugt wird?



Lösung Aufg.3

A3.3)(5 Pkt) Sie sollen mit der Periode T bei einem bestimmten Gerät die Leistung für ein definiertes Zeitintervall L ausschalten. Wie können sie dies tun?



Lösung Aufg.3

A3.4)(5 Pkt) Mechanische Bauteile können *Encoder* enthalten, die bei Bewegung pro mm oder Winkelgrad eine bestimmte Menge von Impulsen abgeben. Wie können Sie dann feststellen, welche Strecke zwischen zwei Zeitpunkten zurückgelegt wurde? Wie könnten Sie die Geschwindigkeit (Wegstrecke/Zeit) dieser Bewegung ermitteln?

Grundidee: man benutzt einen Zähler Z1, um die Anzahl der Impulse zu zählen, die eine Wegstrecke kodieren. $Z1 * 1\text{Längeneinheit} = \text{Gesamtweg}$; man benutzt einen anderen Zähler Z2, um die ablaufende Zeit zu messen. $Z2 * 1\text{Zeiteinheit} = \text{Verbrauchte Zeit}$. $\text{Geschwindigkeit} = Z1/Z2$.

AUFGABE 4: Max. 12 Pkt; Erreichte Pkt: _____

Im Anhang 2 gibt es die Skizze eines S-Bahnhofs. Lösen Sie die folgenden Aufgaben:

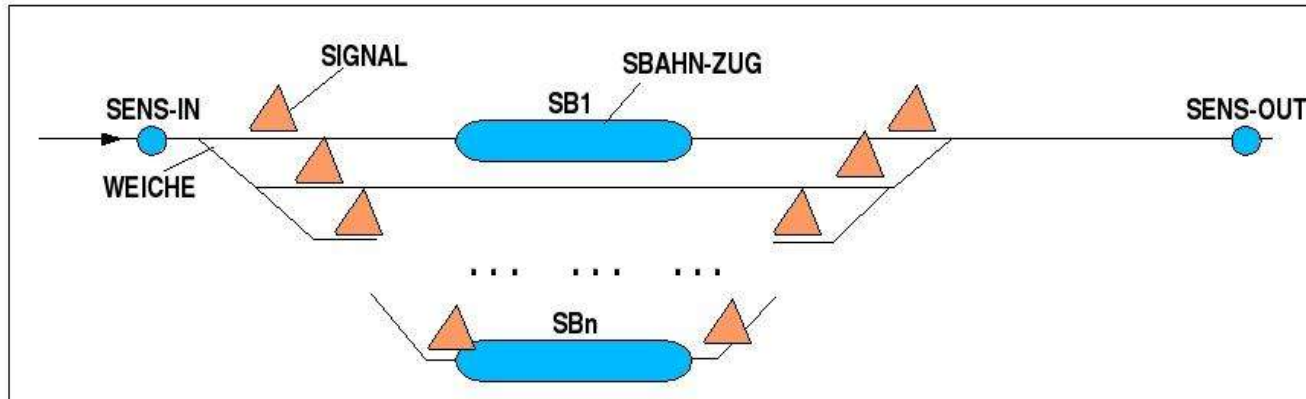
A4.1)(4 Pkt) Erstellen sie ein *Zeitdiagramm*, auf dem man die Aktionen der einzelnen Züge parallel erkennen kann. Ermitteln sie, *wieviele Gleise* der Bahnhof haben sollte, um eine maximale Auslastung zu ermöglichen.

A4.2)(5Pkt) Erstellen Sie einen Taskgraphen zu dieser Anwendungssituation. Benutzen Sie für den Taskgraphen die folgenden Vereinbarungen: es wird angenommen, dass *Ereignisse* durch *Pfeile* dargestellt werden, die Knoten verbinden, und die *Tasks* werden durch die *Knoten* repräsentiert. Ereignisse und Tasks bilden zusammen also einen *Graphen*. Zu jedem Pfeil, der ein Ereignis repräsentiert, gibt es ein Textlabel, das aussagenlogische Verknüpfungen enthalten kann (z.B. "SENS-IN=true UND SIGNAL-EIN-Nr.2=grün" zu lesen als: 'Einfahrtssensor SENSIN ist true UND Signal SIGNAL-EIN-Nr.2 ist grün), und zu jedem Knoten, der einen Task repräsentieren kann, gibt es in diesem Knoten ebenfalls Textlabel zur Beschreibung des Task.

A4.3)(3 Pkt) Erstellen sie eine *technische Tabelle* zu dieser Anwendungssituation.

Lösung Aufg.4

A4.1)(4 Pkt) Erstellen sie ein *Zeitdiagramm*, auf dem man die Aktionen der einzelnen Züge parallel erkennen kann. Ermitteln sie, *wieviele Gleise* der Bahnhof haben sollte, um eine maximale Auslastung zu ermöglichen.



Die vorausgehende Skizze eines S-Bahnhofs zeigt die Gleise, die für den Zugverkehr von 'Links' nach 'Rechts' zuständig sind. Es gibt einen Sensor SENS-IN, der anzeigt, wenn dieser Punkt überfahren wird sowie einen Sensor SENS-OUT; die Strecke vom Sensor SENS-OUT bis zur ersten Weiche ist grösser als der längste Zug. Signale stehen sowohl bei der Einfahrt in ein Gleis wie auch bei der Ausfahrt. Über Weichen wird die Zufahrt in die einzelnen Bahnhofsgleise geregelt. Für alle S-Bahnzüge wird die gleiche Länge angenommen. Die Periode T für jeden Zug ist 20 Minuten. Es wird angenommen, dass ein Zug von der Überfahung des Sensors SENS-IN bis zum Halt im Gleis 2,5 Min benötigt. Für das Aus- und Einsteigen stehen 10 Minuten zur Verfügung. Vom Start aus dem Gleis bis zum Erreichen von SENS-OUT benötigt der Zug 2,5 Min.

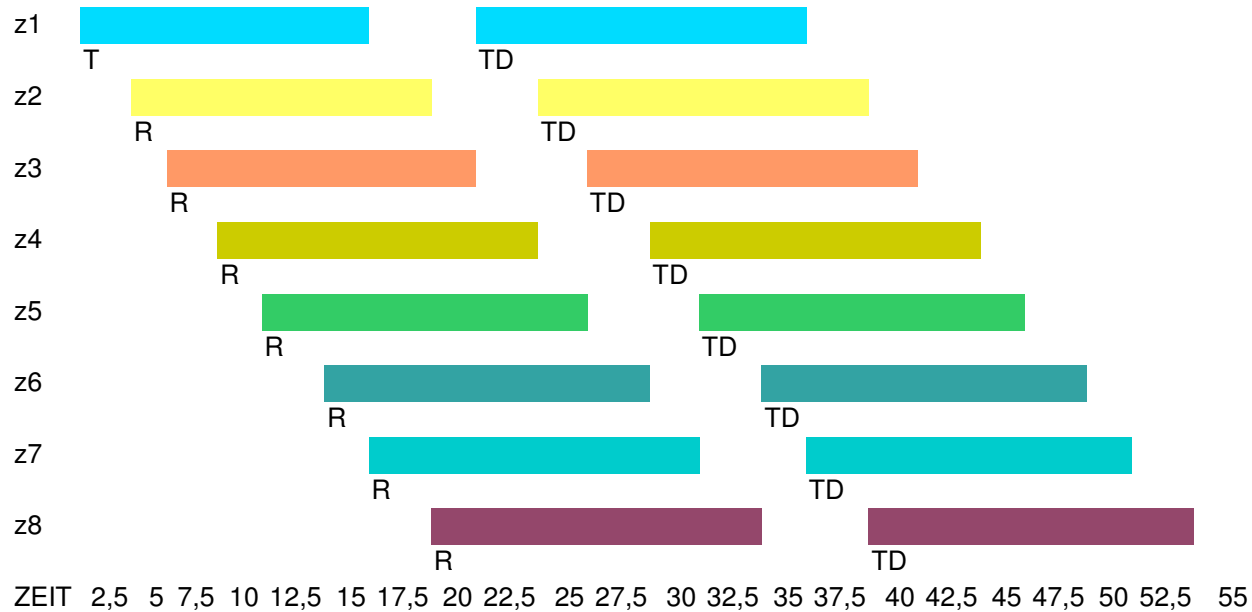
Zug_Nr.i($r=t.1$, $C=15$, $D=T$, $T=20$)

Perioden: 20

LCM: 20

Cs: 15

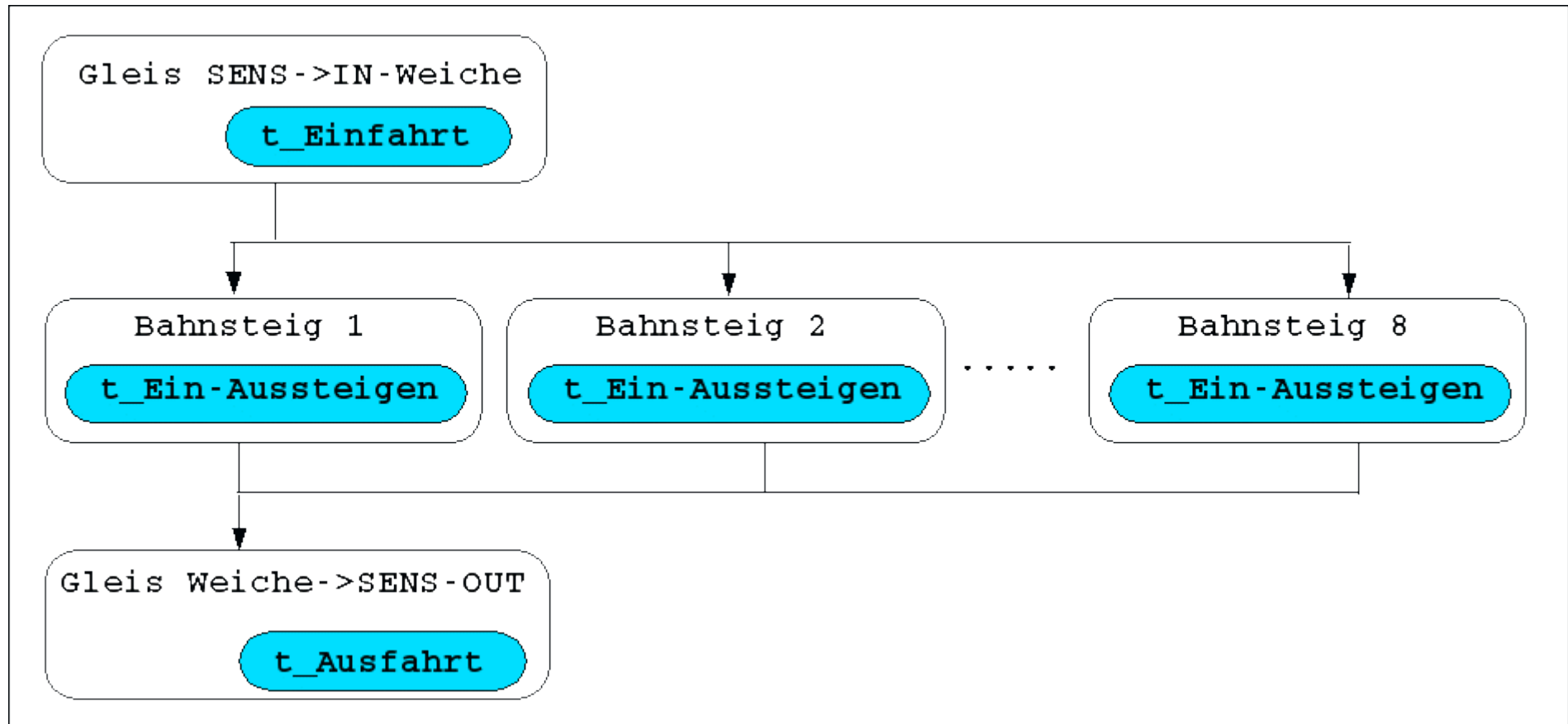
Kleinste Unit: 2,5



Lösung Aufg.4

A4.2)(5Pkt) Erstellen Sie einen Taskgraphen zu dieser Anwendungssituation. Benutzen Sie für den Taskgraphen die folgenden Vereinbarungen: es wird angenommen, dass *Ereignisse* durch *Pfeile* dargestellt werden, die *Knoten* verbinden, und die *Tasks* werden durch die *Knoten* repräsentiert. Ereignisse und Tasks bilden zusammen also einen *Graphen*. Zu jedem Pfeil, der ein Ereignis repräsentiert, gibt es ein Textlabel, das aussagenlogische Verknüpfungen enthalten kann (z.B. "SENS-IN=true UND SIGNAL-EIN-Nr.2=grün" zu lesen als: 'Eifahrtssensor SENS-IN ist true UND Signal SIGNAL-EIN-Nr.2 ist grün), und zu jedem Knoten, der einen Task repräsentieren kann, gibt es in diesem Knoten ebenfalls Textlabel zur Beschreibung des Task.

(Hinweis: Man erinnere sich daran, dass Ereignisse Tasks auslösen und dass Tasks *Ressourcen* benötigen. In dieser Aufgabe sind die Ressourcen verfügbare Gleisabschnitte. Die Tasks sind Züge, die Gleisabschnitte in Anspruch nehmen)



Lösung Aufg.4

A4.3)(3 Pkt) Erstellen sie eine *technische Tabelle* zu dieser Anwendungssituation.

Zug-Nr.	D = T		
	R	C	T
1	0	15	20
2	$R(z_1)+2.5$	15	20
3	$R(z_2)+2.5$	15	20
4	$R(z_3)+2.5$	15	20
5	$R(z_4)+2.5$	15	20
6	$R(z_5)+2.5$	15	20
7	$R(z_6)+2.5$	15	20

Die Antworten zu den folgenden Aufgaben kann man direkt dem Skript entnehmen:

AUFGABE 5: Max. 16 Pkt; Erreichte Pkt: _____

A5.1) Max. 4 Pkt; Erreichte Pkt: _____

-> Um Zeit messen zu können, benötigt man eine SI-Einheit; wie wird die SI-Einheit für Zeit benannt und wie wird sie definiert?

-> Zeitangaben benutzen nicht nur eine SI-Einheit, sondern nehmen auch Bezug auf *Zeitskalen*. Benennen Sie die drei wichtigsten Zeitskalen und beschreiben Sie, wie diese Skalen untereinander zusammenhängen.

-> Zu Charakterisierung des Verhaltens von Realzeitsystemen werden u.a. die Begriffe ' MTTF" ,MTR' sowie ' MBF' benutzt. Wie sind diese Begriffe definiert und was ist damit gemeint

A5.2) Max. 3 Pkt; Erreichte Pkt: _____

-> Was ist ein *System*? Wie unterscheiden sich *Realzeitsysteme* von normalen Systemen?

-> Es gibt unterschiedliche Charakterisierung von Realzeitsystemen. Die folgende Liste gibt einige oft vorkommenden Beschreibungen. Geben Sie zu jedem Begriffspaar eine kurze Beschreibung der wichtigsten Eigenschaften.

- Fail-safe/Fail-operational

- Resource-Adequate/ Resource Inadequate
- Event-triggered/ Time-triggered
- Hard Real-Time/ Soft Real-Time

A5.3) Max. 2 Pkt; Erreichte Pkt: _____

Was unterscheidet das *Engineering von Realzeitsystemen* vom normalen *Softwareengineering*?

A5.4) Max. 7 Pkt; Erreichte Pkt: _____

- Was ist der Unterschied zwischen einem *Ereignis* und einem *Task*?
- Was unterscheidet *periodische* Ereignisse von *spontanen* oder *aperiodischen* Ereignissen?
- Was unterscheidet *abhängige* von *unabhängigen* Tasks?
- Was bedeutet es, dass ein System *präemptiv* ist?
- Welches sind die *minimalen Parameter*, die man benötigt, um einen *Task* für eine Aufgabe zu *beschreiben*?
- Was versteht man unter einem *Jitter*, unter der *Jitter-Toleranz* und der relativen und der absoluten Jitter-Toleranz?
- Wie unterscheidet sich die *relative* von der *absoluten Deadline*? Inwiefern kann man mit einer Deadline eine Zeitfenster definieren?