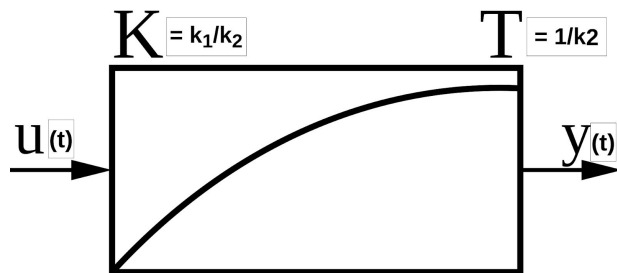


# Klausur SS 2019

## Aufgabe 1: Regelung

### Teilaufgabe 1: Zeitkontinuierliche Regelung

- a) Im Gegensatz zur Steuerung misst die Regelung die Reaktion auf die Steuerung und verbessert nach, falls das gewünschte Ergebnis nicht erzielt wurde (Rückführung des Ist-Wertes)
- b)  $G(s) = X(s) / U(s) = k_1 / (s + k_2) = (k_1/k_2) / (1 + s * (1/k_2))$   
Der letzte Umformungsschritt versucht die Formel auf ein PT1 Glied zu bringen:  $G(s) = K / (1 + sT)$   
Dann wären  $K = k_1/k_2$  und  $T = 1/k_2$



- c) 1. Unterdeterminante:  $1 > 0$   
2. Unterdeterminante:  $8 > 0$   
3. Unterdeterminante:  $-40 < 0 \Rightarrow$  System ist nicht stabil
- d) Das System ist stabil, da es keine Polstellen mit positivem Realteil gibt.

### Teilaufgabe 2: Zeitdiskrete Regelung

- a) Da digitale Rechner digital und somit zeitdiskret arbeiten
- b) Sinuswelle mit geringerer Frequenz als angegebenes Signal zeichnen

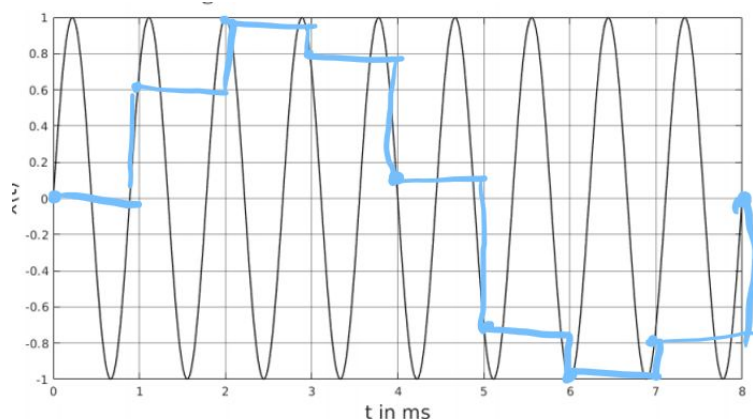


Abbildung 2: Abzutastendes Signal  $x(t)$

- c) Der effekt heißt Aliasing. Lösung: höhere Abtastfrequenz
- d)  $y(k) = e(k)K_p(1 + T_V/T_A) - K_p(T_V/T_A)e(k-1)$

---

Ausführlicher:

Rechenweg 1:

1. gegebene Diskretisierung einsetzen:  $R(z) = K_p(1 + (T_v/T_A)(z - 1)/z) = K_p(1 + (T_v/T_A)(1 - 1/z))$
2. Übertragungsfunktion auf Regeldifferenz anwenden:  
 $Y(z) = R(z)E(z) = K_p(E(z) + (T_v/T_A)(E(z) - E(z)/z))$
3. Z-Transformation rückgängig machen, wobei k der Index ist und  $T = T_A$  das Abtastungsintervall per Linearität + Rechtsverschiebungsregel ( $Z\{f(kT - T)\} = (1/z) F(z)$ ):  
 $y(k) = K_p(e(k) + (T_v/T_A)(e(k) - e(k - 1)))$

Rechenweg 2:

Die Formel in den kontinuierlichen Zeitbereich also  $y(t) = K_p(e(t) + T_v \dot{e}(t))$  und dann die Diskretisierung von

$\dot{e}(t) = (e(k) - e(k - 1))/T_A$  womit sich dann insgesamt ergibt zeitdiskret

$y(k) = K_p(e(k) + (T_v/T_A)(e(k) - e(k - 1)))$ . Das ist dann genauso wie in VL15 Seite 63 "Digitaler PID Regler"

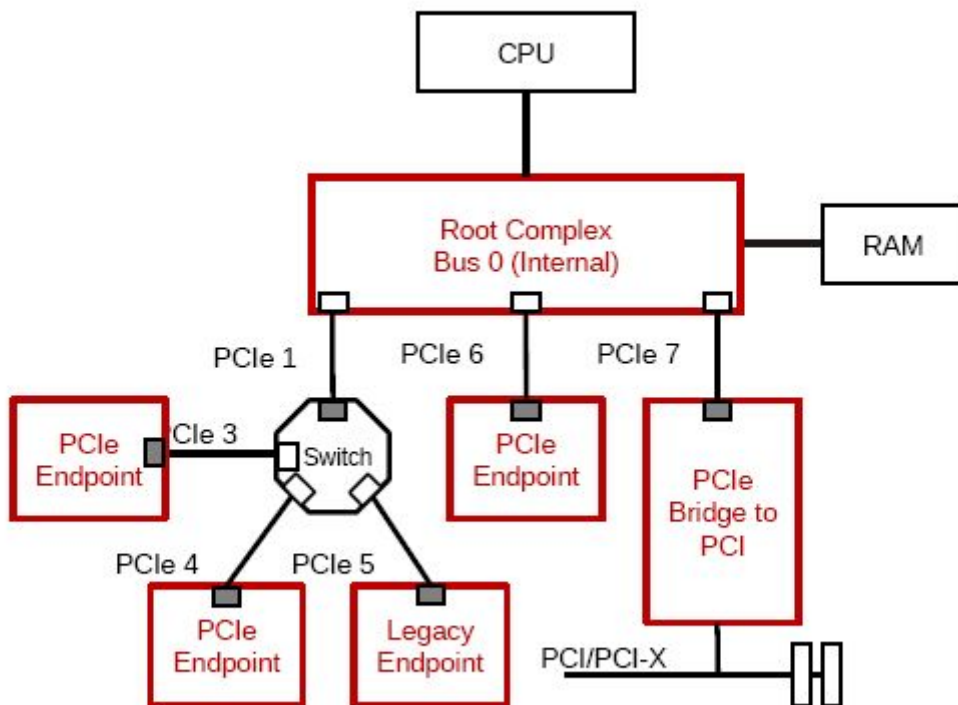
- e) pq-Formel auf  $z^2 + 2z + 5$  anwenden. Daraus folgt:  $z_1 = -1 + 2i$  und  $z_2 = -1 - 2i$ .  
Da mindestens eine dieser Nullstellen im Betrag größer als 1 ist und somit außerhalb des Einheitskreises liegt ist das System instabil.

## Aufgabe 2: Rechnerarchitekturen, Busse, Operationsverstärker, Analog-/Digitaltechnik

### Teilaufgabe 1: Rechnerarchitekturen und Busse

- a) Es handelt sich um einen Mikrocontroller da ein Prozessor bspw. keine AD/DA Wandler aufweist.
- b) Parallel, da mehrere Datenleitungen
- c) Jitter sind ungewollte Schwankungen im Signal die auf die unterschiedlichen Zugriffszeiten in der Speicherhierarchie zurückzuführen sind (Register, L1-L3 Cache...). Dies kann behoben werden, indem man die Ausgabewerte zwischenspuffert.
- d) Asynchron, da eine DTACK Leitung (Data Transfer Acknowledge) vorhanden ist, die bei einem synchronen Bus nicht notwendig/vorhanden wäre, da sie einen asynchronen Transfer bestätigt.  
AS: Gültige Adresse (Lesezugriff) bzw. gültige Adresse und Datum (Schreibzugriff) liegen am Bus an.  
DTACK: Slave hat Daten vom Bus übernommen (Schreibzugriff) bzw. auf den Bus gelegt (Lesezugriff)  
R/W: Gibt an ob es sich um einen Lese- oder Schreibzugriff handelt.

e) Ein Root Complex verbindet die CPU und RAM mit der PCIe Hierarchie.



## Teilaufgabe 2: Operationsverstärker und Analog-/Digitaltechnik

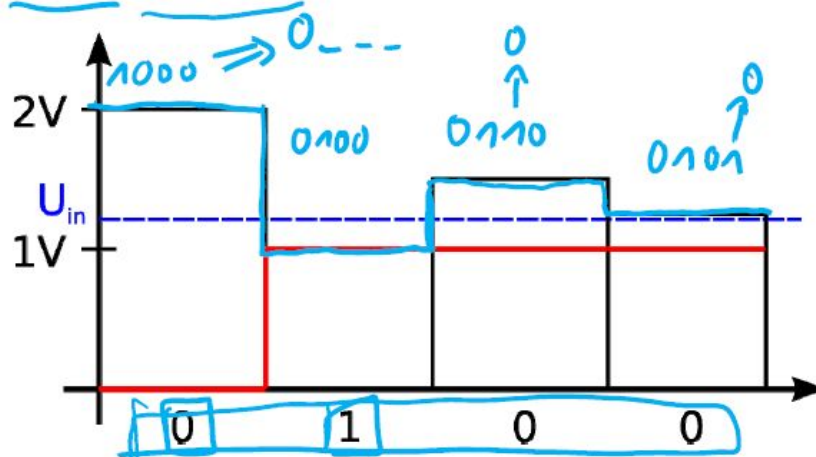
- a) Die Summe aller Ströme in einer Masche ergibt immer 0.
- b) Digitalwert: 1100 ~ 3·3.25V

Beispiel Übung:

### ■ b) SAR (Successive Approximation Register)

Auflösung, Samplerate, Komplexität der Schaltung, kritischste Komponente für hohe Genauigkeit

Beispiel:  $U_{in} = 1,23V$ ;  $U_{ref} = 4V$ ,  $U_{LSB} = 0,25V$ , 4bit  $\rightarrow 4 (0100_2)$



Sukzessiver Vergleich der Eingangsspannung mit Bruchteilen der Referenzspannung führt zum Wandlungsergebnis.

- c) Eine Messbrücke gilt als abgeglichen, wenn kein Strom mehr durch  $I_M$  fließt. Dies ist für  $R_v=R_x$  der Fall. Dient also dazu einen variablen Widerstand an einen anderen anzupassen. Oder umgekehrt: Mit bekannten Widerständen  $R$  und  $R_v$  den unbekannten Widerstand  $R_x$  zu messen.
- d) Subtrahierer mit  $R_0 = R_1 = 2 \text{ k}\Omega$ ,  $R_2=R_3=1 \text{ k}\Omega$  und  $U_1=U_e$ ,  $U_2=5V$ .

## Aufgabe 3: Echtzeitkommunikation und Echtzeitprogrammierung

### Teilaufgabe 1: Echtzeitkommunikation

- a) I6 darf senden. Hört auf zu senden nach Bit:

Knoten	Bit
I1	4
I2	1
I3	1
I4	2
I5	5
I6	-
I7	1

- b) siehe Übung. Basically immer absteigende Flanke in der Mitte des Taktes falls 1 ansonsten immer low.
- c) (USB ist kein Feldbus, da nicht echtzeitfähig)  
Anforderungen an Feldbusse: Echtzeitfähigkeit, Effiziente Übermittlung kleiner Datenmengen, Geringe Latenz, Robustheit, Netzausdehnung bis 1000m, Knotenanzahl bis 254, Interoperabilität  
**Beispiele:** Profibus, CAN-Bus, Safety Bus, INTERBUS, ASI-Bus
- d) Carrier Sense Multiple Access / Collision Detection. Bei diesem Verfahren hören Sender beim senden auf dem Bus zu und erkennen (detection) daran eine Kollision. Kommt es zur Kollision wird diese durch den Back-Off Algorithmus geregelt (in der Regel erstmal aufhören zu senden und nach zufällig langem Zeitfenster (exp wachsend) erneut versuchen)
- e)  $n*(n-1)/2$

### Teilaufgabe 2: Echtzeit Scheduling

- a) Bei präemptiven Scheduling-Verfahren darf der Scheduler eine Task beliebig unterbrechen und eine andere Scheduling, unabhängig von der verbliebenen Dauer des "Jobs", während bei nicht präemptiven Task der Scheduler auf das "freiwillige" Abgeben des jeweiligen Task wartet (bzw. bis diese fertig mit dem Job sind).
- b) A=hoch, B = Mittel, C = niedrig
- c)  $H = \frac{1}{3} + \frac{1}{4} + \frac{1}{6} = \frac{59}{60} = 98.33 \%$
- d)  $H \leq 100\%$
- e) Wir berechnen die tatsächliche Prozessorauslastung (z.B. bei  $c = 59/60$ ) und schauen ob diese unter der maximalen Prozessorauslastung für FPP liegt:

$$H_{max} = n(2^{1/n} - 1), \quad n = \text{Anzahl der Tasks}$$

- f) Da non-preemptive einfach nur ABC im Zyklus (in dieser Reihenfolge).

# Aufgabe 4: Echtzeitbetriebssysteme und Echtzeitsysteme

## Teilaufgabe 1: Echtzeitbetriebssysteme

- a)  $1\text{GB} = 2^{30}$ ,  $1\text{KB} = 2^{10} \rightarrow$  Logisch:  $22 + 10$ , Physisch:  $20 + 10$
- b) Task = Schwergewichtiger Prozess, Thread = Leichtgewichtiger Prozess  
Ein Thread existiert immer nur in einem Task. Eine Task hat immer mindestens einen Thread.
- c) Eine Semaphore ist eine Datenstruktur zur Synchronisation des Zugriffs auf geteilte Betriebsmittel. Sie besteht aus einem Zähler und zwei nicht unterbrechbaren Operationen P(S) und V(S) (Passieren & Verlassen). Reihenfolgensynchronisation, Sperrsynchrisation oder Verbrauchersynchronisation.
- d) Es ist keine Defragmentierung notwendig (würde Verfügbarkeit verletzen). Außerdem können einzelne Seiten gesperrt werden, statt ganze Blöcke (Hilft für Rechtzeitigkeit, da zu jederzeit mehr Speicher verwendbar).
- e) Bedeutet, dass selbst wenn ein hoch priorisierter Thread A auf einen niedrig priorisierten Thread B wartet, die effektive Priorität von A erhalten bleibt, also B nun eine hohe Priorität hat (Prioritätenvererbung). Dies verhindert, dass ein neuer Thread C mit mittlerer Priorität Thread B unterbricht (Prioritäteninversion).  
Dies ist wichtig um Erhaltung der Deadlines gewährleisten zu können.

## Teilaufgabe 2: Echtzeitsysteme

- a) Die Meldung zum Zurücksetzen des intelligenten Watchdog erfolgt nicht mehr durch Auslösen eines einfachen Befehls oder Triggersignals, sondern durch komplexe Befehle oder komplexe Antworten. Hierbei müssen dynamische Berechnungen ausgeführt und ggf. Fragen zur Authentisierung korrekt beantwortet werden. Damit können höhere **Sicherheitsanforderungsstufen** erreicht werden und das Risiko von unerkannten Fehlfunktionen reduziert werden. (QUELLE: Wikipedia, da **nicht in VL**)
- b) Von Links nach Rechts: fest, hart, weich
- c) A state in which safety (protection of the human from the machine) has been achieved and can be maintained.
- d) Gefahr: Potenziell gefährliche Situation, die durch ein Event getriggert wird.  
Risiko: Wahrscheinlichkeit und Schwere des Auftreten einer Gefahr.
- e) Permanente vs Temporäre Fehler und Systematische vs Zufällige Fehler.