

# Klausur SS 2018

## Aufgabe 1: Regelung

### Teilaufgabe 1: Zeitkontinuierliche Regelung

- a)  $R(s) = K_p(1 + 1/T_N s)$
- b) Keine Sensoren. Es soll nur eine Steuerung implementiert werden. Der Mensch fungiert quasi als Sensor und ändert die Einstellung der Heizung, wenn er unzufrieden ist.
- c) Die Vorsteuerung soll ungeachtet der Störgrößen sehr schnell die Zielgröße einrichten. Der Regler soll dann nachjustieren um die Zielgröße möglichst genau zu erreichen.
- d) Der Realteil aller Nullstellen ist kleiner als Null ( $-1$  und  $-1/2$ ). das System ist stabil
- e) Gütekriterien sind: Stabilität, Schnelligkeit und Genauigkeit

### Teilaufgabe 2: Zeitdiskrete Regelung

- a) Rechner arbeiten digital und digitale Regler sind präziser

Digital ist flexibler - muss keinen Schaltkreis umbauen um den Regler zu ändern, nur Software

- b) Mit diesem Messgerät können Frequenzen bis maximal 100 Hz abgetastet werden, sodass sie verlustfrei rekonstruiert werden können
- c) Verwende Tiefpassfilter mit cutoff bei 100Hz
- d)  $3x'(t) + x(t) = w(t)$

nutze die Gesetze der Linearisierung und Differenzierung

$$3sX(s) + X(s) = W(s)$$

forme um

$$G(s) = X(s)/W(s) = 1/(3s + 1)$$

umformen sodass:  $G(s) = b * 1/(s + a)$

$$G(s) = 1/3 * 1/(s + 1/3)$$

=> Tabelle

$$G(z) = 1/3 * z/(z - e^{-1/3 * T})$$

*[glaube das ist falsch, hätte gesagt*

$$3(x(t) - x(t-1))/T + x(t) = w(t)$$

$$\Rightarrow w(s) = 3/T * (x(z) - x(z)/z) + x(z) = w(z)$$

$$\Rightarrow G(z) = x(z)/w(z) = 1/(1 + 3/T * (1 - 1/z))]$$

- e)  $G(s) = 4 * 1/(s + 1)^2 \Rightarrow$  Tabelle

$$G(z) = 4 * (T * z * e^{-T}) / (z - e^{-T})^2$$

- f) Nein, das System ist nicht stabil, da einer der Pole einen Betrag größer als 1 hat

## Aufgabe 2: Rechnerarchitekturen

### Teilaufgabe 1

- a) Die Beschleunigung durch die Pipeline ist nicht deterministisch (zB. bei jumps muss die pipeline möglicherweise geleert werden), dh der worst case muss angenommen werden
- b)  $1\text{M/s} * 1\text{ms} = 1000$ . Dh Nach 1000 Takten war der Zähler 0 => Am anfang war er 1000 = 512 + 256 + 128 + 64 + 32 + 8 = 0b1111101000 = 0011 1110 1000 = 0x03e8

$$1\text{MHz} = 1/1.000.000 \text{ sek}, 1\text{ms} = 1/1000\text{sek}; 1\text{ms}/1\text{MHz} = 1/10000 * 1000000 = 1000$$

- c) Nein, da hier die Slave-Komponente selbstständig die Bereitsheit der Daten per DTACK signalisiert. Übung 3: "Nein, beim asynchronen Bus gibt es keine Wartezeiten. Grund dafür: Die Datenübergabe wird über die Handshake-Signale AS und DTACK gesteuert. Deswegen muss nicht wie beim synchronen Bus auf vordefinierte Takte gewartet werden, um Daten zu übernehmen!"
- d) Jeder Master muss Zugriffserlaubnis jedes mal bei einem zentraler Bus-Arbitr anfragen (?), der Arbitr vergibt nach Prioritäten der Masters
- e)  $33\text{MHz} * 300\text{ns} = 33\text{M/s} * 0.3\mu\text{s} = 9.9 \text{ Zyklen} = 9 \text{ ganze Zyklen}$

$$33\text{MHz} = 1/33.000.000 \text{ s}, 300\text{ns} = 300/1.000.000.000 = 3/10.000.000$$

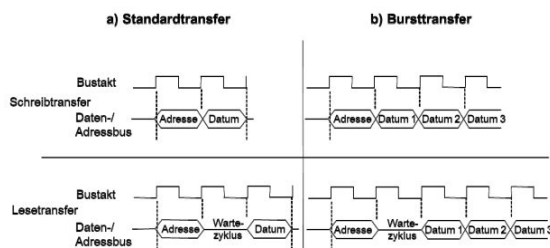
$$300\text{ns}/33\text{MHz} = 3/10.000.000 * 33.000.000 = 3/10 * 33 = 9,9 \text{ Zyklen}$$

(VL07 Folie 25: "PCI-Bus Transferarten")

Bulk-Lesezugriff mit einem Warte-Zyklus: 2 taktzyklen overhead (adresse schreiben + 1 takt warten)

-> 9-2) Zyklen an Daten => 7 Datenworte

Bulk-Schreibzugriff: 1 Taktzyklus Overhead => 8 Datenworte



### Teilaufgabe 2

- a) Verstärkungsfaktor  $V = \text{Infinity}$  (minimale Spannungsdifferenz wird maximal verstärkt),  $R_{in} = \text{infinity}$  (kein Stromfluss in den OP rein),  $R_{out} = 0$  (kein Spannungsabfall am Ausgang)
- b) Hat etwas Verzögerung gegen das ideale Signal und nähert sich langsam (kurvig) an, außerdem Wert gekappt bei  $V+$

#### c) Gleichungen: (gleiche aufgabe wie WS1617:Aufgabe2.2)

$I_{12}R_{12} = 5V$  (gesamte 5V müssen über  $R_{12}$  abfallen da der OP eine virtuelle Erde bei (-) erzwingt (+Maschengleichung basically))

$I_{11}R_{11} = U_e$  (gleiche Begründung)

$$U_a = I_2 R_2$$

$I_2 = -(I_{11} + I_{12})$  (Knotengleichung links vom (-), Kein Strom fließt in den OP da  $R_{in} = \text{inf}$ )

**Damit ergibt sich:**

$$U_a = -(U_e/R_{11} + 5V/R_{12}) * R_2$$

Gewollte Transformation damit  $-0.2 \rightarrow 0V$  und  $-0.4 \rightarrow 5V$ :

$$U_a = -(U_e + 0.2V) * 5V/(0.4 - 0.2)V = -(U_e + 0.2V) * 25$$

dh zum Beispiel:  $R_{11}=1k\Omega$ ,  $R_{12}=25k\Omega$ ,  $R_2=25k\Omega$

d) Puh KA

## Aufgabe 3

### Teilaufgabe 1

- a) 1011101100
- b) (e?) Im Gegensatz zu NRZ und RZ muss der Takt nicht auf einer extra Leitung gesendet werden, da das Manchester Signal in jedem Takt genau einen oder zwei Hi/Lo Übergänge hat, mit dem sich der Empfänger durchgehend weiter synchronisieren kann (bei RZ z.B. wenn eine lange Sequenz von 0 ist kann der Empfänger den Takt verlieren) (Manchester = NRZ XOR Takt; beinhaltet also Takt)
- c) (f?) Vollvermascht vielleicht? Jeder ist mit jedem direkt verbunden, ein Ausfall beeinflusst nichts außer der betroffenen Stelle. Damit eine Komponente vom Netz getrennt wird müssen min  $(n-1)$  Verbindungen ausfallen [ $n$  = Anzahl der Knoten], was schon sehr viel wäre.
- d) (g?) I1 hat höchste Priorität

I1	
I2	5
I3	4
I4	6
I5	5
I6	7
I7	4

### Teilaufgabe 2

- a) Prozessorauslastung = Summe über alle Tasks von Tausführungszeit / Tperiode, Wenn  $> 100\%$  dann existiert kein Schedulingverfahren so dass alle Tasks ihre Periode (=maximal delay) einhalten
- b) Synchron ist unflexibel, da Perioden fix sind und damit auch rechenaufwändiger (wenn viele Zyklen das Programm sinnlos aufgerufen wird) bzw Latenz ist höher (auf ein Ereignis kann nur per "Polling" reagiert werden nicht per Interrupt)
- c) LLF: Executed ticks, Remaining ticks, Laxity at start of each time slot

Time	A	B	C
0	-	<b>0,4,8</b>	-
1	-	<b>1,3,8</b>	-
2	<b>0,3,3</b>	2,2,8	-
3	<b>1,2,3</b>	2,2,7	-
4	2,1,3	2,2,6	<b>0,3,2</b>
5	2,1,2	2,2,5	<b>1,2,2</b>
6	<b>2,1,1</b>	2,2,4	2,1,2

7	3,0,1	2,2,3	<b>2,1,1</b>
8	-	<b>2,2,2</b>	2,1,1
9		<b>3,1,2</b>	-
10		4,0,2	

	0-1	1-2	2	3	4	5	6	7	8	9	10
LLF	B	B	A	A	C	C	A	C	B	B	-
EDF	B	B	A	A	A	C	C	C	B	B	

- d) LLF hat dauernd Kontextswiches da sich die least laxity nach jedem Takt ändern kann. Vorteil ist besseres Verhalten bei Überlast
- e) (Wenn Prozessorauslastung  $\leq n \cdot (2^{1/n} - 1)$  ist Prioritätenverteilung per RMS ein ausführbares Scheduling.)  
 Voraussetzungen: Preemptive, konstante Periodendauern, Periodendauer = Zeitschranke, Tasks müssen voneinander unabhängig sein  
 RMS verteilt Prioritäten nach  $1/\text{Periodendauer}$ , dh häufigere Prozesse haben höhere Priorität
- f) Auslastung =  $3/10 + 4/40 + 1/12.5 + 3/20 = 30\% + 25\% + 8\% + 15\% = 78\% > 4 \cdot (2^{1/4} - 1) = 75\%$   
 Nein, es kann nicht garantiert werden, dass ein Schedule gefunden wird, aber es kann einer existieren. Alle möglichen Schedules müssen ausprobiert werden.

## Aufgabe 4

### Teilaufgabe 1

- a) Garantie von Rechtzeitigkeit, Gleichzeitigkeit, Zuverlässigkeit für IPC, Busse, I/O, Scheduling, etc?
- b) Ein niedrigpriorirer Prozess hält ein Lock auf eine Ressource, ein höherpriorer muss auf diesen warten. Ein mittelpriorir fängt nun an zu laufen und verhindert dass der niedrigpriorie Prozesszeit bekommt. Damit blockiert er effektiv auch den höchstprioren. Vermeiden indem wenn jemand auf ein Lock warten muss er seine Priorität dem Prozess "leiht", auf den er wartet. Damit würde der mittelpriore den niedrigprioren gar nicht preempten können.
- c) Dauernde Abfrage ob/was die neue Werte sind. Selten verwendet, da verwendet sinnlos 100% CPU-Zeit während keine neuen Werte verfügbar sind
- d) Offset:  $\log_2(1000) = 10$  bit, Seitenindex:  $\log_2(1\text{G}/1\text{k}) = \log_2(1\text{M}) = \log_2(1000 \cdot 1000) = 20$  bit  
 Logische Adresse: 22bit + 10bit (volle Breite 32 bit auch wenn erste Bits unbenutzt)  
 Physikalische Adresse: 20bit + 10bit (kachelgröße = seitengröße)

### Teilaufgabe 2

- a) SFF - Single Failure Fraction , PFH Probability of Failure per hour. Relativ heißt anteil bestimmter failures an allen failures, absolut heißt absolute wsk von bestimmter klasse fehler (?)
- b) Geräte lesen/schreiben direkt über den (zB PCI) bus Daten in den RAM, ohne mit der CPU kommunizieren zu müssen. Voraussetzung ist dass die Zugriffsrechte und genaue Bereiche vorher definiert werden, und dass der Bus es technisch erlaubt (muss Multi-Master sein)
- c) Gefahr (Hazard) = etwas was passieren kann was schlecht ist, Risiko (risk) = ein Hazard der analysiert ist mit bestimmter Wahrscheinlichkeit / Schlimmheit

- d) zB computer vision zum erkennen wenn Personen in der Nähe sind, oder besseres / genaueres interpretieren von Sensordaten allgemein. Problem ist die Erklärbarkeit, es ist schwierig zu analysieren und nachzuvollziehen ob / wie wahrscheinlich Fehler sind gegen ein explizit programmiertes System
- e) ka, vllt medizinisch oder so, seh ich nicht in den folien