

Zentralübung Rechnerstrukturen im SS2007

Statistische Verfahren und Fehlertoleranz

Dr. Rainer Buchty

buchty@ira.uka.de

Universität Karlsruhe (TH) – Forschungsuniversität
Institut für Technische Informatik (ITEC)
Lehrstuhl für Rechnerarchitektur und Paralleles Programmieren

19.07.2007

- **Ziel: Feedback an Veranstalter**
 - Lob (hören wir gerne)
 - Konstruktive Kritik / Verbesserungsvorschläge
- **Nicht Ziel: Frustrabbau**
 - Hilft gegen Magengeschwüre, aber leider nicht zur Verbesserung der Lehre
- **Bearbeitungszeit: 5-10 Minuten**

Warum Disziplin der Rechnerarchitektur?

- Fehlertoleranzaspekte betreffen Entwurf und Betrieb
 - Minimierung der Entwicklungskosten bei für die Anwendung ausreichender Zuverlässigkeit
 - Maximierung der Ausfallsicherheit
- Maßzahlen
 - Funktionswahrscheinlichkeit φ
 - Ausfallwahrscheinlichkeit $1 - \varphi$

Was bedeutet Verfügbarkeit?

Verfügbarkeit/Jahr	Ausfallzeit/Jahr
99%	87,6h (3,65d)
99,5%	43,8h (1,825d)
99,9%	8,76h
99,99%	52,56m
99,999%	5,26m

Hochverfügbarkeit: 99,999% (“Five Nines”)

- Statistische Verfahren auch im Low-Power-Bereich von Bedeutung

Schaltungsentwurf

Gegeben Sei ein NAND-Gatter mit zwei Eingängen. Die Eingabewerte seien gleichverteilt. Wie hoch ist die Wahrscheinlichkeit $p_{0 \rightarrow 1}$ für einen Pegelwechsel von 0 auf 1?

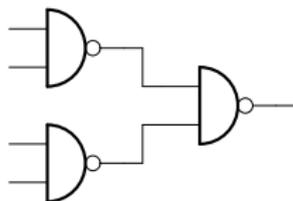
- Statistische Verfahren auch im Low-Power-Bereich von Bedeutung

Schaltungsentwurf

Gegeben Sei ein NAND-Gatter mit zwei Eingängen. Die Eingabewerte seien gleichverteilt. Wie hoch ist die Wahrscheinlichkeit $p_{0 \rightarrow 1}$ für einen Pegelwechsel von 0 auf 1?

- NAND: 0 wenn beide Eingänge 1, sonst 0
- Somit $p_0 = \frac{1}{4}$, $p_1 = \frac{3}{4}$
- $p_{0 \rightarrow 1}$ damit $p_0 * p_1 = \frac{3}{16}$

- NAND-basierte Funktion mit 4 Eingängen: **Auswirkung von Schaltwahrscheinlichkeiten**
(Grundsatzbeispiel ohne Betrachtung der logischen Funktion)



- $f_{NAND}(x, y) = \begin{cases} 0 & \forall x = y = 1 \\ 1 & \text{sonst} \end{cases}$
- $p_{0(1)} = \frac{1}{4}$
 $p_{1(1)} = 1 - p_{0(1)} = \frac{3}{4}$
- $p_0 = \frac{3}{4} * \frac{3}{4} = \frac{9}{16}$
 $p_1 = 1 - p_0 = \frac{7}{16}$
- $p_{0 \rightarrow 1} = p_0 * p_1 = \frac{63}{256}$

- NAND-basierte Funktion mit 4 Eingängen: **Auswirkung von Schaltwahrscheinlichkeiten**

- Hinter erstem Gatter:

$$p_{0(1)} = \frac{1}{4}, \quad p_{1(1)} = \frac{3}{4}$$

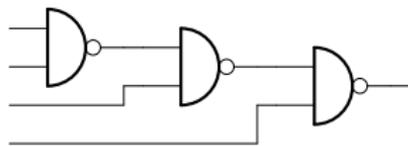
- $p_{0'} = p_{1(1)} * p = \frac{3}{4} * \frac{1}{2} = \frac{3}{8}$

$$p_{1'} = 1 - p_{0'} = \frac{5}{8}$$

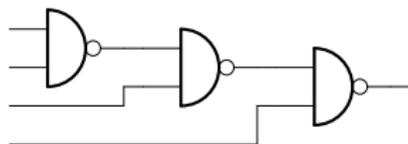
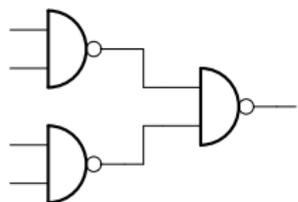
- $p_0 = p_{1'} * p = \frac{5}{8} * \frac{1}{2} = \frac{5}{16}$

$$p_1 = 1 - p_0 = \frac{11}{16}$$

- $p_{0 \rightarrow 1} = p_0 * p_1 = \frac{55}{256}$



- NAND-basierte Funktion mit 4 Eingängen: **Auswirkung von Schaltwahrscheinlichkeiten**



- $p_0 = \frac{9}{16}, p_1 = \frac{7}{16}$
- $p_{0 \rightarrow 1} = \frac{63}{256}$
- Höherer
Leistungsverbrauch
- Geringere Durchlaufzeit

- $p_0 = \frac{5}{16}, p_1 = \frac{11}{16}$
- $p_{0 \rightarrow 1} = \frac{55}{256}$
- Geringerer
Leistungsverbrauch
- Höhere Durchlaufzeit

- Graphische Repräsentation einer Architektur durch **Zuverlässigkeitsblockdiagramm**
- Abbildung auf gleichwertige **Strukturformel**
- Transformierung in Berechnungsformel

Beispielaufgabe

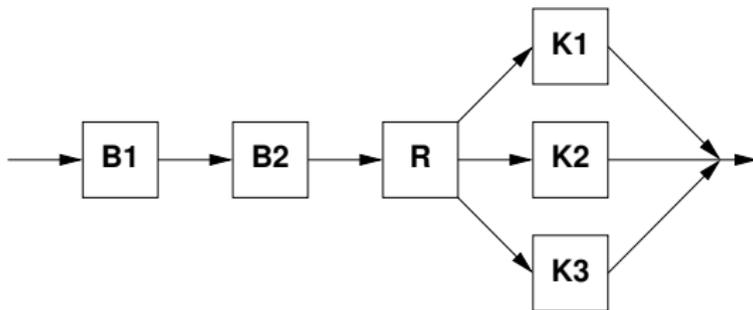
Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2 , der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneneinheit und mindestens eine Kommunikationskomponente erforderlich.

Erstellen Sie Zuverlässigkeitsblockdiagramm und Strukturformel.

Beispielaufgabe

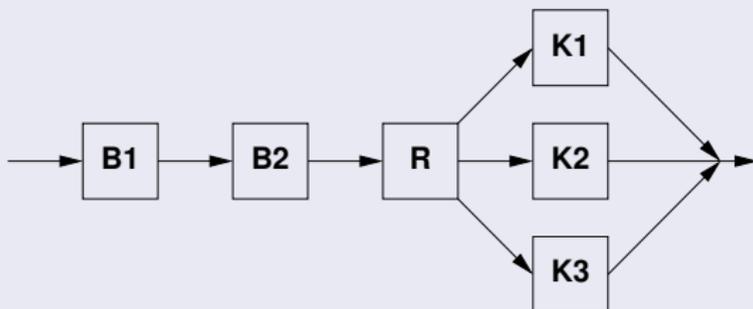
Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2 , der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneinheit und mindestens eine Kommunikationskomponente erforderlich.

Erstellen Sie Zuverlässigkeitsblockdiagramm und Strukturformel.



Zuverlässigkeitsblockdiagramm

Zuverlässigkeitsblockdiagramm



Strukturformel

$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B_1 \text{ and } B_2 \text{ and } R \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$$

$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

Strukturformel

- Gegeben seien die Funktionswahrscheinlichkeiten $\varphi(B)$, $\varphi(R)$ und $\varphi(K)$.
- Umformung in Formel zur Berechnung:
- Funktionswahrscheinlichkeit eines **Seriensystems**:
 $\varphi(\bigwedge_{K \in \Lambda}) = \prod_{K \in \Lambda} \varphi(K)$, also $\varphi = \varphi(B) * \varphi(B) * \varphi(R) * \dots$
- Funktionswahrscheinlichkeit eines **Parallelsystems**:
 $\varphi(\bigvee_{K \in \Lambda}) = \sum_{\emptyset \neq A \in \Lambda} (-1)^{1+\#A} * \varphi(\bigwedge_{K \in A} K)$

- **Wie mit 1-aus-n umgehen?**
- Betrachtung der **Ausfallwahrscheinlichkeit**
 - Umformung gemäß boolescher Logik
 $(K_1 \vee K_2 \vee K_3) \rightarrow \neg(K_1 \wedge K_2 \wedge K_3)$
 - entsprechend: $\varphi(K) \rightarrow 1 - \varphi(K)$, d.h.
Ausfallwahrscheinlichkeit für K-System: $(1 - \varphi(K))^3$
 - Anschließend: **Retransformation** in
Funktionswahrscheinlichkeit

- somit: $\varphi = \underbrace{\varphi(B) * \varphi(B) * \varphi(R)}_{\text{Seriensystem}} * \underbrace{(1 - (1 - \varphi(K))^3)}_{\text{Parallelsystem}}$

- Erhöhung der Systemsicherheit
- simpel, aber teuer: Vervielfachung der kritischen Komponenten
- **Hot Standby**
 - Backup-Komponenten rechnen immer parallel, jedoch wird nur ein Ergebnis verwendet
 - Hoher Energiebedarf
 - Verschleiß der Backup-Komponenten
 - Schnelle Umschaltzeit
- **Cold Standby**
 - Backup-Komponenten werden bei Bedarf aktiviert
 - Verbrauch (fast) wie Normalsystem
 - Kein Verschleiß (außer normaler Alterung)
 - Benötigt Zeit zum Hochfahren und Umschalten

- Kompromissentwurf: **Graceful Degradation**
 - System wird auf durchschnittlich benötigte Leistung hin aufgebaut mit Sicherheitszulage
 - Im Normalbetrieb: typischerweise Parallelbetrieb aller Einheiten bei gleichmäßiger Auslastung
 - Im Fehlerfall: Verteilung der verbleibenden Ressourcen
 - Im Fehlerfall weiterhin funktionsfähig, aber verringerte Gesamtleistung
- Problem: Wie **Fehlerzustand** erkennen?
- **Mehrfachsysteme mit Mehrheitsentscheider**
 - Mehrfache, unabhängige Durchführung von Systemaufgaben
 - Definition des korrekten Zustands über Mehrheitsentscheider
 - Identifikation fehlerhafter Berechnung bzw. von Fehlerzuständen
 - Klassische Ausführung:
Triple Modular Redundancy (TMR) *Tres faciunt collegium.*

● Alltagsimplementationen

- Batterien (mehr Spannung/Strom als zum Betrieb benötigt)
- RAID-Systeme
 - Mirroring (RAID-1): hot standby
 - Parität (RAID3-6): Entscheider-System
 - Erhöhung der Ausfallsicherheit

RAID	Anzahl Festpl.	Netto-kapazität	Ausfall-sicherheit
0	≥ 2	n	0
1	$\geq 2 * 1$	$\frac{n}{2}$	$\frac{n}{2}$
2	10	$\frac{8n}{10}$	2
3	≥ 2	$n-1$	1
4	≥ 2	$n-1$	1
5	≥ 3	$n-1$	1
6	≥ 4	$n-2$	2
DP	≥ 3	$n-2$	2

- **Zuverlässigkeitsblockdiagramm und Strukturformel:**

Erfassung aller Funktionszustände

- Beispiel: 2-aus-3-System
- System funktionsfähig, wenn 1&2, 1&3, 2&3, 1&2&3 funktionsfähig
- System nicht funktionsfähig, wenn nur 1, 2, oder 3 funktionsfähig.

- **Zuverlässigkeitsberechnung** direkt über:

$$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- Beispiel: 2-aus-3-System, n=2, m=3

$$\varphi_3^2 = \sum_{k=2}^3 \binom{3}{k} * \varphi(K)^k * (1 - \varphi(K))^{(3-k)}$$

- **Systeme mit Mehrheitsentscheider:**

$$\varphi_m^n = \varphi(V) * \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- $\varphi(K)$: Funktionswahrscheinlichkeit der Komponente
- $\varphi(V)$: Funktionswahrscheinlichkeit des Mehrheitsentscheiders (Voter)
- Entscheider ist **single point of failure!**
 - $\varphi(V)$ idealerweise $\rightarrow 1$
 - Voter vergleichsweise einfache Einheit, daher geringe Fehleranfälligkeit
 - Ggf. seinerseits Redundanzsystem (Teilauswertungen)

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Verfügbarkeit pro Festplatte betrage $\varphi(F) = 0,99$, wie hoch ist die Chance auf Datenverlust?

- allgemein:

$$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- $n=8, m=10, \varphi(K) = \varphi(F) = 0,99$ – also:

$$\varphi_{10}^8 = \sum_{k=8}^{10} \binom{10}{k} * 0.99^k * 0.01^{(10-k)} = 0.999886$$

Chance auf Datenverlust somit: $1 - 0.999886 = 0.000114$.

- Mean Time to Failure (**MTTF**): mittlere Funktionszeit
- Mean Time to Repair (**MTTR**): mittlere Reparaturzeit
- Mean Time between Failures (**MTBF**): mittlere Zeit zwischen zwei Ausfällen, $MTBF = MTTF + MTTR$
(für $MTTR \ll MTTF$ gilt somit: $MTBF \sim MTTF$)
- **Punktverfügbarkeit** eines Systems (V):
Wahrscheinlichkeit, ein System zu einem beliebigen Zeitpunkt fehlerfrei anzutreffen, unabhängig davon, ob es bis zu diesem Zeitpunkt bereits ausgefallen ist oder nicht.

$$V = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

- Für über die Zeit konstante Ausfallraten gilt außerdem:

$$\text{Ausfallrate } \lambda = \frac{1}{MTTF}$$

Eine Festplatte habe eine MTTF von 2 Jahren im Dauerbetrieb. Die Reparaturzeit (MTTR) setze sich zusammen aus der Zeit für das Herunterfahren des Rechners (2 Minuten), Austausch der Festplatte (10 Minuten) und anschließendes Hochfahren des Rechners (2 Minuten).

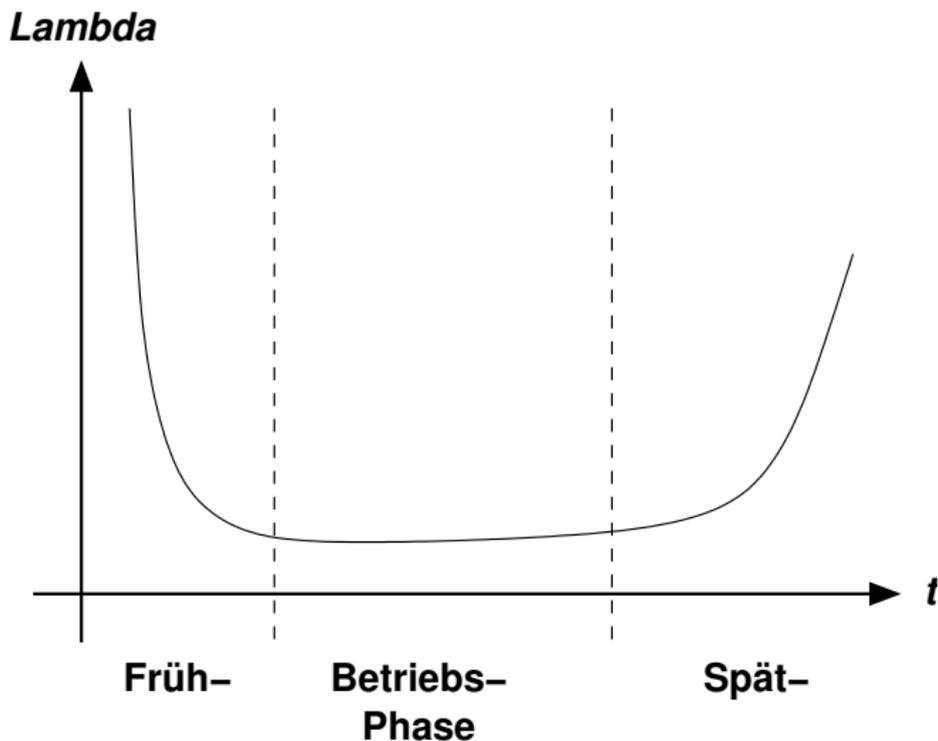
Berechnen Sie die Punktverfügbarkeit V .

Eine Festplatte habe eine MTTF von 2 Jahren im Dauerbetrieb. Die Reparaturzeit (MTTR) setze sich zusammen aus der Zeit für das Herunterfahren des Rechners (2 Minuten), Austausch der Festplatte (10 Minuten) und anschließendes Hochfahren des Rechners (2 Minuten).

Berechnen Sie die Punktverfügbarkeit V .

- $MTTF = 2a = (2 * 365 * 24 * 60)min = 1051200min$
- $MTTR = (2 + 10 + 2)min = 14min$
- $V = \frac{MTTF}{MTTF+MTTR} = \frac{1051200}{1051214} = 0,999997$

- **Konstante Ausfallrate ist vereinfachtes Modell**
- **Reale Systeme: variable Ausfallwahrscheinlichkeit**
über Zeit
- **Badewannenkurve**
 - **Frühphase**
 - Initialausfälle
 - Fertigungsfehler, Bauteildefekte
 - Ausfallrate exponentiell abfallend
 - **Betriebsphase**
 - Nahezu konstante Ausfallrate
 - **Spätphase**
 - Alterungseffekte
 - Ausfallrate exponentiell ansteigend



Badewannenkurve

Gegeben sei ein 2-aus-3-System, dessen Komponenten zufallsverteilt mit gleicher Rate ausfallen. Die Überlebenswahrscheinlichkeit einer Komponente wird durch die Formel $R(t) = e^{-\lambda \cdot t}$, $t > 0$ beschrieben.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?
- 2 Bestimmen Sie die Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.
- 3 Bestimmen Sie λ derart, dass die mittlere Lebensdauer für das gegebene 2-von-3-System $\frac{5}{6}$ beträgt.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

allgemein: $z(t) = \frac{d F_L(t)}{dt R(t)} = \frac{d 1-R(t)}{dt R(t)}$

somit: $z(t) = \frac{d 1-R(K,t)}{dt R(K,t)} = \lambda.$

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

$$\text{allgemein: } z(t) = \frac{d F_L(t)}{dt R(t)} = \frac{d 1-R(t)}{dt R(t)}$$

$$\text{somit: } z(t) = \frac{d 1-R(K,t)}{dt R(K,t)} = \lambda.$$

- 2 Bestimmung der Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.

Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

- 2 Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

- 2 Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.
2-von-3-System:

$$R(S_{2v3}, t) = \sum_{k=2}^3 \binom{3}{k} R(t)^k [1 - R(t)]^{3-k} = 3 * R(t)^2 - 2 * R(t)^3$$

Einzelkomponente: $R(K, t) = R(t)$

- 2 Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.
2-von-3-System:

$$R(S_{2v3}, t) = \sum_{k=2}^3 \binom{3}{k} R(t)^k [1 - R(t)]^{3-k} = 3 * R(t)^2 - 2 * R(t)^3$$

Einzelkomponente: $R(K, t) = R(t)$

somit gilt: $R(K, t) = R(S_{2v3}, t) \leftrightarrow R = 3 * R^2 - 2 * R^3$
 $\rightarrow R_1 = 0, R_2 = 0.5, R_3 = 1$

Wegen $R(K, t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörigen Werte $t_2 = 0$ und $t_3 = \frac{\ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{\ln(2)}{\lambda})$.

- 3 Bestimmen Sie λ derart, dass die mittlere Lebensdauer für das gegebene 2-von-3-System $\frac{5}{6}$ beträgt.

Es gilt:

$$\mathbf{MTTF} = \int_0^{\infty} \mathbf{R(S, t)} dt, \quad \lambda = \frac{1}{\mathbf{MTTF}}, \quad R(K, t) = e^{-\lambda t}$$

$$MTTF = \int_0^{\infty} R(S_{2v3, t}) dt = \frac{5}{6\lambda} \rightarrow \lambda = 1$$

- **Vorlesungsfolie 3-21:** Statt `END Spec`; muss natürlich `END architecture`; **oder** `END architecture Architecture-Name`; stehen.
- **Vorlesungsfolie 6-9:** Durchsatz T einer Pipeline
 - $n + k - 1$ ist die Anzahl der Taktzyklen, die ein Programm mit n Befehlen auf einer Architektur mit $k - \text{stufiger}$ Pipeline benötigt.
 - $k - 1$ ist die sich durch die Pipeline ergebende Latenz
 - Der Durchsatz T einer k -stufigen Pipeline ergibt sich zu
$$T = \frac{n}{n+k-1}$$
 - $k = 1$: $T = 1$
 - $k > 1$: Einfluss der Pipeline-Latenz verliert sich mit zunehmender Programmgröße
- **Übungsaufgabe 4e** runderneuert zwecks besserer Nachvollziehbarkeit
- Bitte ggf. neu herunterladen und ausdrucken.

● Vorlesungen

- Mikroprozessoren II
Mikroprozessoren für eingebettete Systeme
(vorauss. montags, 14 Uhr)
- High-Performance Heterogeneous Computing
(vorauss. mittwochs, 15:45 Uhr)

● Praktika

- Multicore-Technologie
- Multicore-Programmierung

● Seminar

- High-Performance Reconfigurable Computing

● Studien- und Diplomarbeiten

- Parallele Programmierung
- Hardwareentwurf
- Monitoring und Datenanalyse
- Organic Computing
- ...

Fragen/Interesse? → buchty@ira.uka.de

- **8. August 2006, 14 Uhr**
- Verteilt auf bis zu 4 Hörsäle
 - Audimax A/B (30.95)
 - HSaF (50.35)
 - Neue Chemie (30.46)
 - Architektur HS37 (20.40)
- Hörsaaleinteilung wird nach
Anmeldeschluss (20. Juli, 12 Uhr)
bekanntgegeben
- Sitzplatzeinteilung wird direkt am Klausurtermin
bekanntgegeben

- Dauer der Klausur: **60 Minuten** (reine Bearbeitungszeit)
- Klausur besteht aus **6 Aufgaben** zu thematischen Schwerpunkten aus Vorlesung und Übung
- 60 Punkte erzielbar, **20 Punkte zum Bestehen** notwendig
- Faustregel: pro Punkt eine Minute Bearbeitungszeit
- **Keine Hilfsmittel zugelassen** (auch keine Wörterbücher)

- Dauer der Klausur: **60 Minuten** (reine Bearbeitungszeit)
- Klausur besteht aus **6 Aufgaben** zu thematischen Schwerpunkten aus Vorlesung und Übung
- 60 Punkte erzielbar, **20 Punkte zum Bestehen** notwendig
- Faustregel: pro Punkt eine Minute Bearbeitungszeit
- **Keine Hilfsmittel zugelassen** (auch keine Wörterbücher)

Don't panic!