Karlsruher Institut für Technologie

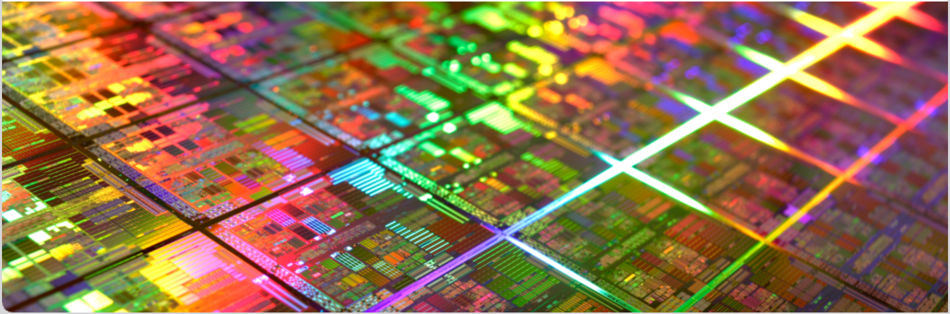
Zentralübung Rechnerstrukturen im SS 2013

Fehlertoleranz und Sprungvorhersage

Martin Schindewolf, Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

04. Juni 2013



Agenda

Karlsruher Institut für Technologie

Fehlertoleranz

Zuverlässigkeit/Verfügbarkeit eines Systems bestimmen

Sprungvorhersage

- Pipeline muss gefüllt werden
- Sprung hat zwei mögliche Ausgänge (T, NT)
- Sprungvorhersage sagt diesen Ausgang vorher

sorgt dafür, dass Befehle vom vorhergesagten Ziel geladen werden

Fehlertoleranz

Analyse von Architekturen

- Graphische Repräsentation einer Architektur durch **Zuverlässigkeitsblockdiagramm**
- Abbildung auf gleichwertige **Systemfunktion** (Strukturformel)
- evtl. Bestimmen des Fehlerbaums
- Transformierung in Berechnungsformel

Blockdiagramm / Strukturformel

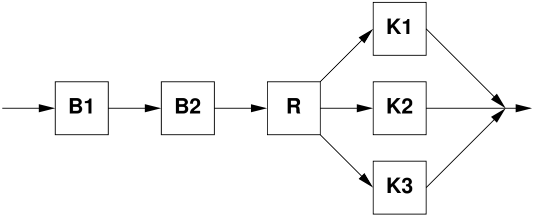
Aufgabe 1

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 Rechenenenheit und mindestens eine Kommunikationskomponente erforderlich.

Erstellen Sie Zuverlässigkeitsblockdiagramm, Systemfunktion (Strukturformel) und Fehlerbaum.

Aufgabe 1

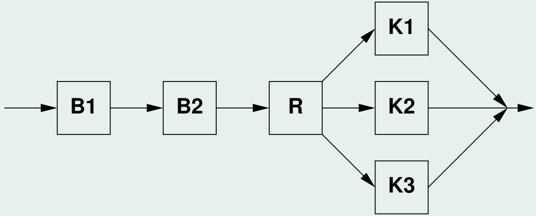
Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2, der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3. Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneneinheit und mindestens eine Kommunikationskomponente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



Zuverlässigkeitsblockdiagramm

Aufgabe 1 (forts.)

Zuverlässigkeitsblockdiagramm



Systemfunktion (Strukturformel)

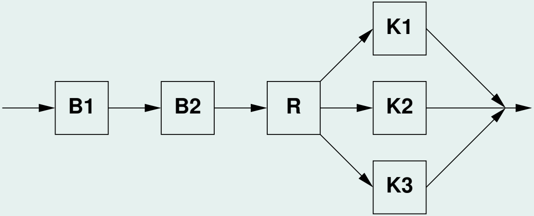
$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B_1 \; and \; B_2 \; and \; R \; and \; (K_1 \; or \; K_2 \; or \; K_3)$$

Aufgabe 1 (forts.)

Zuverlässigkeitsblockdiagramm



Systemfunktion (Strukturformel)

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

oder

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

Aufgabe 1 (forts. II)

Systemfunktion (Strukturformel)

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

oder

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

Fehlerbaum

Strukturbaum der Negation der Systemfunktion

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

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

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

Aufgabe 1 (forts. II)

Systemfunktion (Strukturformel)

$$S = B_1 \land B_2 \land R \land (K_1 \lor K_2 \lor K_3)$$

oder

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

Fehlerbaum

Strukturbaum der Negation der Systemfunktion

$$\neg S = \neg(B_1 \land B_2 \land R \land (K_1 \lor K_2 \lor K_3))$$

$$\neg S = \neg B_1 \lor \neg B_2 \lor \neg R \lor \neg(K_1 \lor K_2 \lor K_3)$$

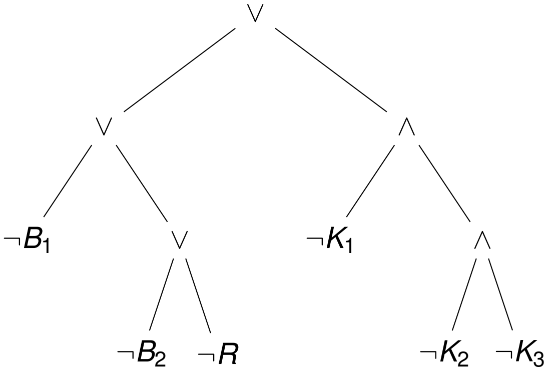
$$\neg S = \neg B_1 \lor \neg B_2 \lor \neg R \lor (\neg K_1 \land \neg K_2 \land \neg K_3)$$

Aufgabe 1 (forts. III)

Fehlerbaum

Strukturbaum der Negation der Systemfunktion

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



Funktionswahrscheinlichkeit

Aufgabe 1 (forts.)

Berechnen sie die Funktionswahrscheinlichkeit.

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

Strukturformel

- Gegeben seien die Funktionswahrscheinlichkeiten $\varphi(B)$, $\varphi(R)$ und $\varphi(K)$.

Umformung in Formel zur Berechnung:

- Funktionswahrscheinlichkeit eines **Seriensystems**:
  $\varphi(\wedge_{K\in\Lambda}) = \prod_{K\in\Lambda} \varphi(K)$, also $\varphi = \varphi(B) * \varphi(B) * \varphi(R) * \ldots$
- Funktionswahrscheinlichkeit eines **Parallelsystems**:
  $\varphi(\vee_{K\in\Lambda}) = \sum_{\emptyset\neq A\in\Lambda} (-1)^{1+\#A} * \varphi(\wedge_{K\in A} K)$

Funktionswahrscheinlichkeit (forts.)

Wie mit 1-aus-n umgehen?

- Betrachtung der **Ausfallwahrscheinlichkeit**
  - Umformung in Seriensystem gemäß boolescher Logik
    $(K_1 \vee K_2 \vee K_3) \rightarrow \neg(\neg K_1 \wedge \neg K_2 \wedge \neg K_3)$
  - $K \rightarrow \neg K$, entsprechend $\varphi(K) \rightarrow 1 - \varphi(K)$
  - Ausfallwahrscheinlichkeit für K-System damit: $(1 - \varphi(K))^3$
  - Anschließend: **Retransformation** in Funktionswahrscheinlichkeit

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

Redundanzsysteme

- **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

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

Redundanzsysteme (forts.)

- **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)

Aufgabe 2: Maßzahlen

Teilaufgabe a)

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Funktionswahrscheinlichkeit 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: $1 - 0.999886 = 0.000114$.

Aufgabe 2: Maßzahlen

Teilaufgabe a)

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Funktionswahrscheinlichkeit 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: $1 - 0.999886 = 0.000114$.

Weitere Taxonomien

- Mean Time to Failure (**MTTF**): mittlere Funktionszeit (**E(L)**)
- Mean Time to Repair (**MTTR**): mittlere Reparaturzeit (**E(B)**)
- 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:

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

Aufgabe 2

Teilaufgabe b)

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.

- $MTTF = 2a = (2 * 365 * 24 * 60)\,min = 1\,051\,200\,min$
- $MTTR = (2 + 10 + 2)\,min = 14\,min$
- $V = \frac{MTTF}{MTTF+MTTR} = \frac{1\,051\,200\,min}{1\,051\,214\,min} = 0,999997$

Aufgabe 2

Teilaufgabe b)

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.

- $MTTF = 2a = (2 * 365 * 24 * 60)\, min = 1\,051\,200\, min$
- $MTTR = (2 + 10 + 2)\, min = 14\, min$
- $V = \frac{MTTF}{MTTF+MTTR} = \frac{1\,051\,200\, min}{1\,051\,214\, min} = 0,999997$

Alterungseffekte (Teilaufgabe c)

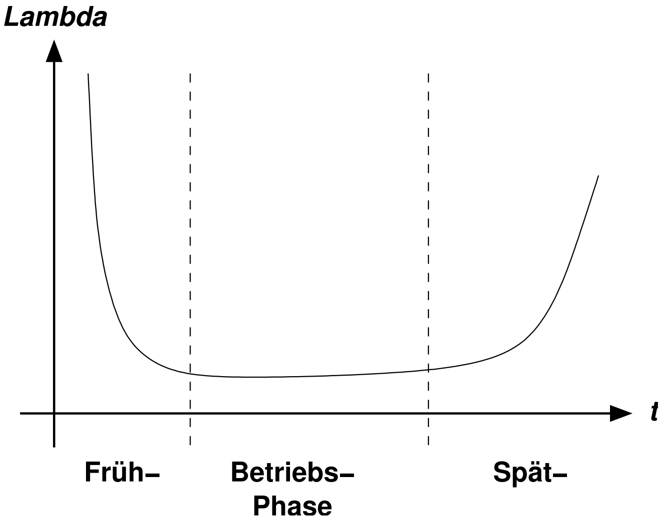
- **Konstante Ausfallrate ist vereinfachtes Modell**
- **Reale Systeme: variable Ausfallwahrscheinlichkeit** über Zeit

Badewannenkurve

1. **Frühphase**
   - Initialausfälle
   - Fertigungsfehler, Bauteildefekte
   - Ausfallrate exponentiell abfallend
2. **Betriebsphase**
   - Nahezu konstante Ausfallrate
3. **Spätphase**
   - Alterungseffekte
   - Ausfallrate exponentiell ansteigend

Alterungseffekte

Karlsruher Institut für Technologie



Badewannenkurve

Rechnen mit zeitlich variabler Ausfallrate

Karlsruher Institut für Technologie

Teilaufgabe d)

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.

Rechnen mit variabler Ausfallrate (forts.)

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

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

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

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.

Rechnen mit variabler Ausfallrate (forts.)

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

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

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

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.

Rechnen mit variabler Ausfallrate (forts.)

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

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

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

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.

Rechnen mit variabler Ausfallrate (forts.)

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

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

   Einzelkomponente: $\mathbf{R(K,t) = R(t)}$

   somit gilt:
   $R(K,t) = R(S_{2v3},t)$
   $\leftrightarrow R = 3 * R^2 - 2 * R^3$
   $\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$
   Wegen $R(K,t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörenden Werte $t_2 = 0$ und $t_3 = \frac{ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{ln(2)}{\lambda})$.

Rechnen mit variabler Ausfallrate (forts.)

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

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

   Einzelkomponente: $\mathbf{R(K,t) = R(t)}$

   somit gilt:
   $R(K,t) = R(S_{2v3},t)$
   $\leftrightarrow R = 3 * R^2 - 2 * R^3$
   $\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$
   Wegen $R(K,t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörenden Werte $t_2 = 0$ und $t_3 = \frac{ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{ln(2)}{\lambda})$.

Rechnen mit variabler Ausfallrate (forts.)

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

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

Einzelkomponente: $\mathbf{R(K,t) = R(t)}$

somit gilt:
$R(K,t) = R(S_{2v3},t)$
$\leftrightarrow R = 3 * R^2 - 2 * R^3$
$\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$
Wegen $R(K,t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörenden Werte $t_2 = 0$ und $t_3 = \frac{ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{ln(2)}{\lambda})$.

Rechnen mit variabler Ausfallrate (forts.)

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

   Es gilt:

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

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

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

a) Vermeidung von Pipelineleerlauf

Um das Leerlaufen der Pipeline bei Kontrollflussbefehlen zu vermeiden, existieren statische, sowie dynamische Techniken, die jeweils zu verschiedenen Teilen durch Hardware und Software unterstützt werden.
Nennen Sie diese und stellen Sie die wesentlichen Unterschiede gegenüber.

1. Sprungvorhersage: Einführung

a) Vermeidung von Pipelineleerlauf

- Statische Techniken
  - **Statische Sprungvorhersage**
    - Always Not Taken
    - Always Taken
  - **Prädikation**
    - Verwenden von Prädikaten
    - Compiler-gestützt
    - Sprungbefehle werden vermieden, stattdessen spekulative Ausführung der mit Prädikat versehenen Befehle und Gültigmachen bei Evaluation des Prädikatregisters zu Wahr
    - Benötigte Hardware: Spekulative Ausführung, Prädikatregister

1. Sprungvorhersage: Einführung

a) Vermeidung von Pipelineleerlauf

- Dynamische Techniken - **Prädiktion**
  - Sprungvorhersage
  - Hardware-Technik
  - Sprungbefehle werden ausgeführt und die nächsten Befehle entsprechend der Vorhersage über den Sprungausgang spekulativ geladen und ausgeführt
  - Benötigte Hardware: Sprungzieltabelle, Prädiktor(en)

Weitere Informationen

- Brinkschulte, Ungerer: Mikrocontroller und Mikroprozessoren - Springer, 2. Auflage – Seite 301ff

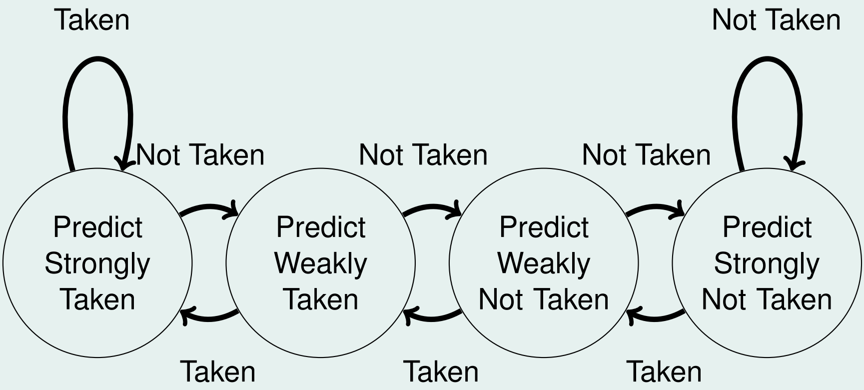
1. Sprungvorhersage: Einführung

b) 2-Bit-Prädiktoren

Zeichnen Sie einen 2-Bit-Prädiktor einmal mit Sättigungszähler und einmal mit Hysteresezähler.
Worin liegt die Motivation zur Verwendung eines Hysterese- anstelle eines Sättigungszählers?

2-Bit-Prädiktoren

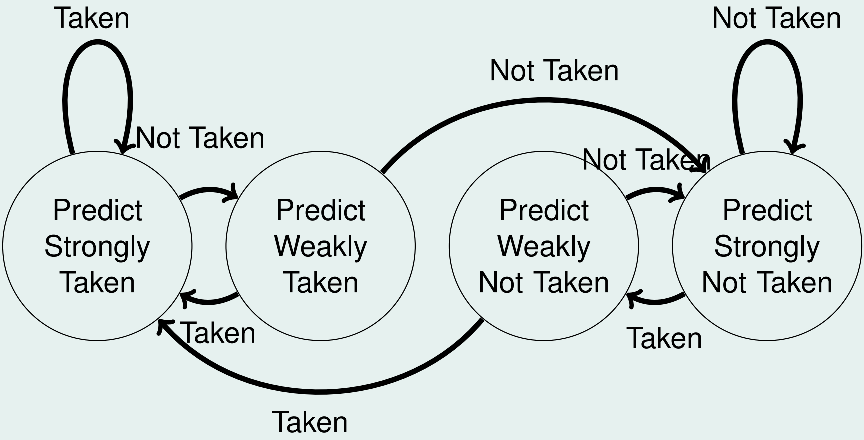
Sättigungszähler



2-Bit-Prädiktoren

Karlsruher Institut für Technologie

Hysteresezähler



2-Bit-Prädiktoren

Karlsruher Institut für Technologie

Diskussion

- Wechsel bei Hysterese der Vorhersage erst nach zwei Fehlvorhersagen
- Aggressiveres Umschaltverhalten, vermeidet "‘Flattern"’ zwischen Weakly-Zuständen

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

Beispiel: Gemeinsamer 2-Bit-Prädiktor

Füllen Sie die Tabelle für die Vorhersagen und Zustände der obigen zwei Prädiktoren und das unten angegebene Programm aus, wobei **alle Sprünge auf denselben Prädiktor zugreifen** und dieser mit Predict Weakly Not Taken (WNT) initialisiert sei.

```
INIT:    ADD   R1,R0, #0   ; R1=0
         ADD   R2,R0, #2   ; R2=2

START:   BNE   R1,R0,L1    ; if (R1!=0) goto L1
         ADD   R1,R0,#1    ; R1=1

L1:      SUB   R3,R1,R2    ; R3=R1-R2
         BNE   R3,R0,L2    ; if (R1!=R2) goto L2
         ADD   R1,R0,#0    ; R1=0
         J     START       ; goto START

L2:      ADD   R1,R0,#2    ; R1=2
         J     START       ; goto START
```

1. Sprungvorhersage: Einführung

c) Sprungverläufe

Sprung 1 Zeile 4 BNE L1	Sprung 2 Zeile 8 BNE L2	Sprung 3 Zeile 10 J START	Sprung 4 Zeile 13 J START
NT (R1=0)	T (R1=1)	–	T
T (R1=2)	NT (R1=2)	T	–
NT (R1=0)	T (R1=1)	–	T
T (R1=2)	NT (R1=2)	T	–

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT			T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	**NT**	SNT	SNT	T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	**NT**	SNT	SNT	T	WNT
WNT	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT

$\Rightarrow$ Es werden **6** Fehlannahmen gemacht!

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT			T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT	SNT	SNT	T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT	SNT	SNT	T	WNT
WNT	T			NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	

1. Sprungvorhersage: Einführung

2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT

$\Rightarrow$ Es werden **7** Fehlannahmen gemacht!

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor

Füllen Sie die Tabelle für die Vorhersagen und Zustände der obigen zwei Prädiktoren und das unten angegebene Programm aus, wobei **jeder Sprung über seinen eigenen Prädiktor verfüge** und dieser mit Predict Weakly Not Taken (WNT) initialisiert sei.

Welchen Unterschied stellen Sie fest und worauf lässt sich dieser zurückführen?

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT		WNT	T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	
SNT	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T		WT	NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	
WNT	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT		WNT	T	
	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	**NT**	SNT	WNT	T	
SNT	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	**NT**	SNT	WNT	T	WT
SNT	T		WT	NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	**NT**	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT

$\Rightarrow$ Es werden **6** Fehlannahmen gemacht!

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT		WNT	T	
	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	
SNT	T			NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T		ST	NT	
	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	
WNT	NT			T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	NT		WT	T	
	T			NT	

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	**NT**	SNT	WT	T	
SNT	T			NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	**NT**	SNT	WT	**T**	ST
SNT	T		ST	NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	**NT**	SNT	WT	**T**	ST
SNT	T	WNT	ST	NT	

1. Sprungvorhersage: Einführung

d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	**NT**	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	**NT**	SNT	WT	**T**	ST
SNT	T	WNT	ST	NT	WT

⇒ Es werden **5** Fehlannahmen gemacht!

1. Sprungvorhersage: Einführung

Karlsruher Institut für Technologie

Schlussfolgerung

Sprungeigene Prädiktoren können genauer arbeiten. Alternativ lässt sich die Information der beiden Sprünge in Korrelation setzen, um genau aus der gegenseitigen Beeinflussung weitere Genauigkeit zu erhalten.

Korrelationsprädiktoren

(m,n)-Korrelationsprädiktoren

- *m* Sprünge umfassende Historie
- Speicherung in Sprungverlaufsregister (Branch History Register, BHR) letzten *m* Sprünge
- Auswahl eines *n*-Bit-Prädiktors aus Sprungverlaufstabelle (Pattern History Table, PHT) anhand von Sprungadresse und BHR
- (Teil der) Sprungadresse selektiert Zeile in PHT
- BHR wählt eine von 2^m Spalten
- Ausgewähltes Speicherfeld enthält *n*-Bit-Prädiktor
- (1,2)-Korrelationsprädiktor: Globales 1-Bit-BHR, pro Sprung zwei separate 2-Bit-Prädiktoren

Korrelationsprädiktoren

Karlsruher Institut für Technologie

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4		NT	(WT, WT)			
8						
4						
8						
4						
8						
4						
8						

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(WT, WT)		NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	
4	NT	NT			NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)		NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	
4	NT	NT			NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

Karlsruher Institut für Technologie

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)		T	
4	T	T			T	
8	NT	T			NT	
4	NT	NT			NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	T	T	(WNT, **WT**)		T	
8	NT	T			NT	
4	NT	NT			NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)		NT	
4	NT	NT			NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)
4	NT	NT	(**WNT**, WT)		NT	
8	T	NT			T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

Karlsruher Institut für Technologie

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)
4	**NT**	NT	(**WNT**, WT)	NT	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)		T	
4	T	T			T	
8	NT	T			NT	

Korrelationsprädiktoren

Karlsruher Institut für Technologie

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)
4	**NT**	NT	(**WNT**, WT)	NT	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	T	T	(WNT, **WT**)		T	
8	NT	T			NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)
4	**NT**	NT	(**WNT**, WT)	NT	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)		NT	

Korrelationsprädiktoren

e) (1,2)-Korrelationsprädiktor

Zeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Historie	Prädiktor
4	NT	NT	(**WT**, WT)	T	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)
4	**NT**	NT	(**WNT**, WT)	NT	NT	(SNT, WT)
8	T	NT	(**SNT**, WT)	NT	T	(WNT, WT)
4	**T**	T	(WNT, **WT**)	T	T	(WNT, ST)
8	NT	T	(WNT, **ST**)	T	NT	(WNT, WT)

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

Aufgabe

Gegeben sei der folgende MIPS-Code. Beachten Sie, dass Register R0 in der MIPS ISA immer den Wert 0 hat. Beachten Sie weiterhin, dass MIPS-Instruktionen immer an Wortgrenzen ausgerichtet sind, d.h. die niedrigsten zwei Bits der Instruktionsadresse sind immer 0. In diesem Beispiel werden trotzdem die niederwertigsten Bits der Sprungadresse zur Indizierung der Sprungvorhersagetabellen verwendet.

2. Sprungvorhersage: Ausführliches Beispiel

Code

```
0x100   li    R2, 0                  ; v = 0
0x104   li    R3, 100                ; Loop bound for LoopI
0x108   li    R4, 0                  ; i = 0
  LoopI:
0x10C   beq   R4, R3, EndLoopI   ; Exit LoopI if i == 100
0x110   li    R5, 0                   ; j = 0
  LoopJ:
0x114   beq   R5, R3, EndLoopJ   ; Exit LoopJ if J == 100
0x118   add   R6, R5, R4         ; j + i
0x11C   andi  R6, R6, 1          ; (j+i)%2
0x120   bne   R6, R0, EndIf      ; Skip if (j+i)%2 != 0
0x124   add   R2, R2, R5         ; v +=j
  EndIf:
0x128   addi  R5, R5, 1          ; j++
0x12C   beq   R0, R0, LoopJ      ; Go back to LoopJ
  EndLoopJ:
0x130   addi  R4, R4, 1          ; i++
0x134   beq   R0, R0, LoopI      ; Go back to LoopI
  EndLoopI:
```

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

Aufgabe

Bestimmen Sie nun für diesen Assembler-Code die exakte Anzahl an Fehlvorhersagen der Sprungvorhersage, die während der Ausführung auftreten, wenn folgende Prädiktoren verwendet werden:

1. Ein Always-Taken Prädiktor
2. Ein globaler 1-Bit Prädiktor, initialisiert mit Taken.
3. Ein 1-Bit Prädiktor mit 32 Einträgen, die niedrigsten Bits der Instruktionsadresse werden zur Indizierung des Eintrags verwendet, initialisiert mit Taken.
4. Ein 2-Bit Prädiktor mit 16 Einträgen, die niedrigsten Bits der Instruktionsadresse werden zur Indizierung des Eintrags verwendet, initialisiert mit Strongly Taken.

2. Sprungvorhersage: Ausführliches Beispiel

Etwas Statistik

Anzahl an genommenen / nicht genommenen Sprüngen pro Sprungadresse:

- `0x10C`: for-Loop ⇒ 100 nicht genommen (i = 0 bis 99), dann einmal genommen
- `0x114`: Inner for-Loop ⇒ für jeden Durchlauf der äußeren Schleife: 100 nicht genommen (i = 0 bis 99), dann einmal genommen
  ⇒ Not Taken 100 * 100 = 10000 mal, Taken 100 * 1 = 100 mal

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

Etwas Statistik

Anzahl an genommenen / nicht genommenen Sprüngen pro Sprungadresse:

- `0x120`: Alterniert zwischen Taken und Not Taken
  $\Rightarrow$ 10000 Durchläufe $\Rightarrow$ Not Taken 5000 mal, Taken 5000 mal
- `0x12C`: Rücksprung $\Rightarrow$ wird hier immer genommen $\Rightarrow$ Taken 10000 mal (innere Schleife)
- `0x134`: Rücksprung $\Rightarrow$ wird hier immer genommen $\Rightarrow$ Taken 100 mal

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

Überblick

Sprung	Bits	Taken	Not Taken
`0x10C`	01100	1	100
`0x114`	10100	100	10000
`0x120`	00000	5000	5000
`0x12C`	01100	10000	0
`0x134`	10100	100	0

2. Sprungvorhersage: Ausführliches Beispiel

a) Always taken

Sprung	Bits	Taken	Not Taken	Fehlvorhersagen
`0x10C`	01100	1	100	100
`0x114`	10100	100	10000	10000
`0x120`	00000	5000	5000	5000
`0x12C`	01100	10000	0	0
`0x134`	10100	100	0	0
				15100

⇒ Total: 15100 Fehlvorhersagen

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

Code

```
0x100   li   R2, 0                ; v = 0
0x104   li   R3, 100              ; Loop bound for LoopI
0x108   li   R4, 0                ; i = 0
  LoopI:
0x10C   beq  R4, R3, EndLoopI     ; Exit LoopI if i == 100
0x110   li   R5, 0                 ; j = 0
  LoopJ:
0x114   beq  R5, R3, EndLoopJ     ; Exit LoopJ if J == 100
0x118   add  R6, R5, R4           ; j + i
0x11C   andi R6, R6, 1            ; (j+i)%2
0x120   bne  R6, R0, EndIf        ; Skip if (j+i)%2 != 0
0x124   add  R2, R2, R5           ; v +=j
  EndIf:
0x128   addi R5, R5, 1            ; j++
0x12C   beq  R0, R0, LoopJ        ; Go back to LoopJ
  EndLoopJ:
0x130   addi R4, R4, 1            ; i++
0x134   beq  R0, R0, LoopI        ; Go back to LoopI
  EndLoopI:
```

2. Sprungvorhersage: Ausführliches Beispiel

b) 1-Bit Prädiktor (global)

- `0x10C`: Da Prädiktor initialisiert mit T ⇒ Fehlvorhersage
  Bei den weiteren Durchläufen bestimmt `0x134` die Vorhersage
  ⇒ Prediction immer Taken ⇒ Sprung wird aber nur einmal genommen (am Ende des Programms)
  ⇒ **100 Fehlvorhersagen**
- `0x114`: Die Vorhersage wird bestimmt durch den Ausgang der Sprünge `0x10C` und `0x12C`
  Vorhersage von `0x10C` kommend: NT (100 mal)
  Vorhersage von `0x12C` kommend: T (10000 mal)
  Korrekt vorhergesagt werden die Sprünge, die von `0x10C` kommen und die Schleifenaustritte ⇒ **9900 Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

b) 1-Bit Prädiktor (global)

- `0x120`: Vorhersage wird von `0x114` bestimmt ⇒ hier Vorhersage immer taken
  ⇒ **50% Fehlvorhersagen** (alle Sprünge mit Ausgang Not Taken)
- `0x12C`: Vorhersage wird von `0x120` bestimmt ⇒ Vorhersage alterniert
  ⇒ **50% Fehlvorhersagen**
- `0x134`: Vorhersage wird von `0x114` bestimmt ⇒ Vorhersage immer taken ⇒ **keine Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

b) 1-Bit Prädiktor (global)

Sprung	Bits	Taken	Not Taken	Fehlvorhersagen
`0x10C`	01100	1	100	100
`0x114`	10100	100	10000	9900
`0x120`	00000	5000	5000	5000
`0x12C`	01100	10000	0	5000
`0x134`	10100	100	0	0
				20000

$\Rightarrow$ Total: 20000 Fehlvorhersagen

2. Sprungvorhersage: Ausführliches Beispiel

Adressabbildung

Abbildung der Sprungadresse auf die jeweiligen Prädiktoren anhand ihrer Sprungadresse (niederwertigsten Bits):

- `0x10C`: 01100
- `0x114`: 10100
- `0x120`: 00000
- `0x12C`: 01100
- `0x134`: 10100

32 Einträge benötigen 5 Bits
16 Einträge benötigen 4 Bits

2. Sprungvorhersage: Ausführliches Beispiel

Code

```
0x100   li   R2, 0                ; v = 0
0x104   li   R3, 100              ; Loop bound for LoopI
0x108   li   R4, 0                ; i = 0
  LoopI:
0x10C   beq  R4, R3, EndLoopI     ; Exit LoopI if i == 100
0x110   li   R5, 0                 ; j = 0
  LoopJ:
0x114   beq  R5, R3, EndLoopJ     ; Exit LoopJ if J == 100
0x118   add  R6, R5, R4           ; j + i
0x11C   andi R6, R6, 1            ; (j+i)%2
0x120   bne  R6, R0, EndIf        ; Skip if (j+i)%2 != 0
0x124   add  R2, R2, R5           ; v +=j
  EndIf:
0x128   addi R5, R5, 1            ; j++
0x12C   beq  R0, R0, LoopJ        ; Go back to LoopJ
  EndLoopJ:
0x130   addi R4, R4, 1            ; i++
0x134   beq  R0, R0, LoopI        ; Go back to LoopI
  EndLoopI:
```

2. Sprungvorhersage: Ausführliches Beispiel

c) 1-Bit Prädiktor (32 Einträge)

Sprünge `0x10C` und `0x12C` fallen zusammen, sowie Sprünge `0x114` und `0x134`

- `0x10C`: Bedingt durch die Initialisierung mit Taken und den Änderungen von `0x12C` auf Taken, wird hier nur die letzte Iteration korrekt vorhergesagt ⇒ **100 Fehlvorhersagen**
- `0x114`: Bedingt durch die Initialisierung mit Taken und den Änderungen von `0x134` auf Taken, erzeugt jede neue Iteration der äußeren Schleife eine Fehlvorhersage. Falsch vorhergesagt werden zudem die letzte Iteration der inneren Schleife. Alle anderen Iterationen werden korrekt vorhergesagt ⇒ **200 Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

c) 1-Bit Prädiktor (32 Einträge)

- `0x120`: Prädiktor alterniert ständig zwischen Taken und Not taken. ⇒ Ständige Fehlvorhersagen, außer bei Eintritt in die innere Schleife ⇒ **9900 Fehlvorhersagen**
- `0x12C`: Der erste Durchlauf jeder Iteration wird falsch vorhergesagt, da `0x10C` den Prädiktor auf Not taken setzt. Alle anderen Iterationen werden korrekt vorhergesagt → **100 Fehlvorhersagen**
- `0x134`: Vorhersage wird von `0x114` bestimmt ⇒ Vorhersage immer taken ⇒ **keine Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

c) 1-Bit Prädiktor (32 Einträge)

Sprung	Bits	Taken	Not Taken	Fehlvorhersagen
`0x10C`	01100	1	100	100
`0x114`	10100	100	10000	200
`0x120`	00000	5000	5000	9900
`0x12C`	01100	10000	0	100
`0x134`	10100	100	0	0
				10300

$\Rightarrow$ Total: 10300 Fehlvorhersagen

2. Sprungvorhersage: Ausführliches Beispiel

Code

```
0x100   li   R2, 0                ; v = 0
0x104   li   R3, 100              ; Loop bound for LoopI
0x108   li   R4, 0                ; i = 0
  LoopI:
0x10C   beq  R4, R3, EndLoopI     ; Exit LoopI if i == 100
0x110   li   R5, 0                 ; j = 0
  LoopJ:
0x114   beq  R5, R3, EndLoopJ     ; Exit LoopJ if J == 100
0x118   add  R6, R5, R4           ; j + i
0x11C   andi R6, R6, 1            ; (j+i)%2
0x120   bne  R6, R0, EndIf        ; Skip if (j+i)%2 != 0
0x124   add  R2, R2, R5           ; v +=j
  EndIf:
0x128   addi R5, R5, 1            ; j++
0x12C   beq  R0, R0, LoopJ        ; Go back to LoopJ
  EndLoopJ:
0x130   addi R4, R4, 1            ; i++
0x134   beq  R0, R0, LoopI        ; Go back to LoopI
  EndLoopI:
```

2. Sprungvorhersage: Ausführliches Beispiel

d) 2-Bit Prädiktor (16 Einträge)

Sprünge `0x10C` und `0x12C` fallen zusammen, sowie Sprünge `0x114` und `0x134`

- `0x10C`: Bedingt durch die Initialisierung mit Strongly Taken und den Änderungen von `0x12C` auf Strongly Taken, wird hier nur die letzte Iteration korrekt vorhergesagt
  $\Rightarrow$ **100 Fehlvorhersagen**
- `0x114`: Bedingt durch die Initialisierung mit Strongly Taken, Fehlvorhersage der ersten beiden Iterationen $\Rightarrow$ danach sagt der Prädiktor Not Taken vorher
  Beim Wechsel von der inneren zur äußeren Schleife wird sowohl `0x114`, als auch `0x134` genommen $\Rightarrow$ Vorhersage dann Weakly Taken $\Rightarrow$ Fehlvorhersage der neuen, ersten Iteration der inneren Schleife
  $\Rightarrow$ **201 Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

d) 2-Bit Prädiktor (16 Einträge)

- `0x120`: In der ersten Iteration, der Prädiktor alterniert ständig zwischen Strongly Taken und Weakly Taken.
  $\Rightarrow$ 50% Fehlvorhersagen
  Sprungverlauf am Ende der Iteration: NT - T | T $\Rightarrow$ Vorhersage zu Strongly Taken
  Zweite Iteration: Prädiktor alterniert zwischen ST und WT $\Rightarrow$ 50% Fehlvorhersagen
  Sprungverlauf am Ende der zweiten Iteration: T - NT | NT $\Rightarrow$ Wechsel der Vorhersage zu Weakly Not Taken
  Dritte Iteration: Prädiktor alterniert zwischen WT und WNT $\Rightarrow$ 100% Fehlvorhersagen
  $\Rightarrow$ 51 Iteration * 50 Fehlvorhersagen + 49 ungerade Iterationen * 100 Fehlvorhersagen $\Rightarrow$ **7450 Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

d) 2-Bit Prädiktor (16 Einträge)

- `0x12C`: Bedingt durch die Initialisierung mit Strongly Taken ⇒ **keine Fehlvorhersagen** (auch die Fehlvorhersage von `0x10C` ändert an der Vorhersage Taken nichts)
- `0x134`: Sprung `0x114` hält den Prädiktor bei der Vorhersage Not Taken ⇒ **100 Fehlvorhersagen**

2. Sprungvorhersage: Ausführliches Beispiel

Karlsruher Institut für Technologie

d) 2-Bit Prädiktor (16 Einträgen)

Sprung	Bits	Taken	Not Taken	Fehlvorhersagen
`0x10C`	01100	1	100	100
`0x114`	10100	100	10000	201
`0x120`	00000	5000	5000	7450
`0x12C`	01100	10000	0	0
`0x134`	10100	100	0	100
				7851

$\Rightarrow$ Total: 7851 Fehlvorhersagen

3. SimpleScalar

3.2) Matrix-Multiplikation

Gegeben Sei der Quellcode `mm-std.c` für eine Standard Matrix-Matrix-Multiplikation. Übersetzen Sie diesen Quellcode zunächst mit Cross-Compiler und simulieren Sie anschließend diese Anwendung mit dem `sim-bpred`. Verwenden Sie folgende Sprungvorhersageeinheiten:

- Always taken
- Always nottaken
- Globaler 2 bit-Prädiktor
- Prädiktortabelle mit 2 bit-Prädiktoren und 16 Einträgen
- Globaler Korrelationsprädiktor mit 1 bit-History und einem 2 bit-Prädiktor

Welche Sprungvorhersageeinheit würden Sie für diese Anwendung auswählen? Begründen Sie Ihre Antwort.

3. SimpleScalar

Karlsruher Institut für Technologie

3.2) Matrix-Multiplikation

	taken	not taken	bimod 1	bimod 16	2lev
Count	17172	2131190	17307	17067	16985
Rate	99,6%	50,4%	99,59%	99,6%	99,6%

$\Rightarrow$ Der Korrelationsprädiktor bietet die höchste Trefferrate und sollte daher gewählt werden.

Aufgabe 3: SimpleScalar

3.3) MiBench Benchmark Suite

Vergleichen Sie nun die Benchmarks `basicmath`, `qsort` und `susan` aus dem Automotive-Teil der Benchmark-Suite hinsichtlich ihrer Leistung mit verschiedenen Sprungvorhersageeinheiten. Verwenden Sie hier dieselben Prädiktoren wie in Aufgabe 3.2.

Welche Vorhersageeinheit würden Sie für jede Anwendung wählen? Welche Sprungvorhersageeinheit liefert insgesamt die beste Leistung?

3. SimpleScalar

3.3) MiBench Benchmark Suite

		taken	not taken	bimod 1	bimod 16	2lev
basicmath	Count	11142629	13658564	11907657	10001627	2400196
	Rate	66%	58,4%	63,7%	69,5%	92,7%
qsort	Count	2120041	2406870	2245820	1173120	446567
	Rate	68,5%	64,2%	66,6%	82,6%	93,4%
susan -s	Count	238434	1650032	247979	117397	116419
	Rate	87,4%	13,0%	86,9%	93,8%	93,9%
susan -e	Count	11707	72863	11757	6265	3403
	Rate	87,3%	21%	86,9%	92,8%	96%
susan -c	Count	20470	27832	8897	5539	2406
	Rate	63%	49,7%	83,3%	89,4%	95,1%

$\Rightarrow$ Der Korrelationsprädiktor bietet die höchste Trefferrate und sollte daher gewählt werden.

Fragen?

Fragen?