

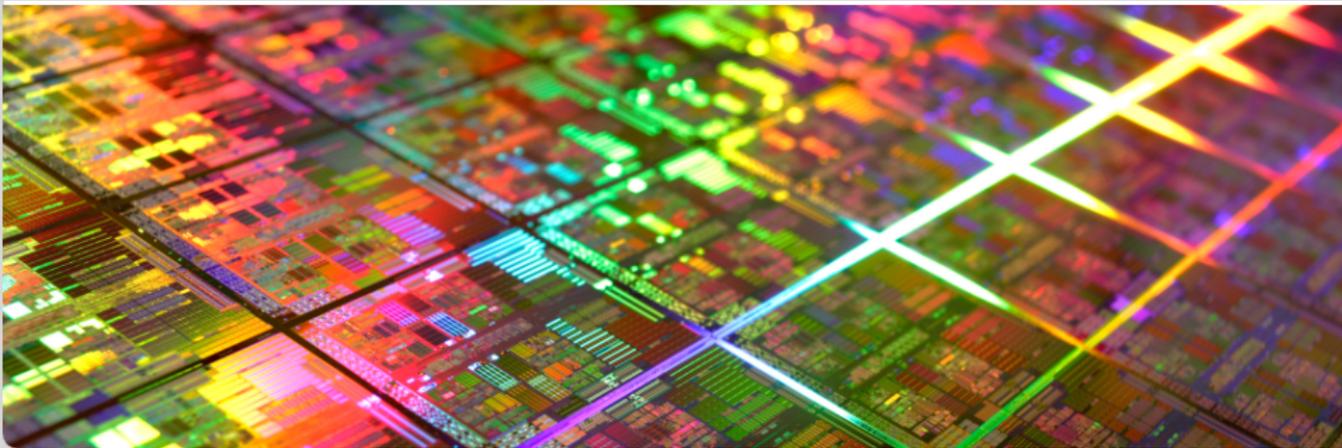
Zentralübung Rechnerstrukturen im SS 2016

Fehlertoleranz und Sprungvorhersage

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

2. Juni 2016



Fehlertoleranz

- Bestimmung der Zuverlässigkeit und Verfügbarkeit eines Systems

Sprungvorhersage

- Füllung der Pipeline sicherstellen
- Sprung hat zwei mögliche Ausgänge (T, NT)
- Sprungvorhersage:
 - Vorhersage des Ausgangs
 - Laden der Befehle vom vorhergesagten Ziel

Analyse von Architekturen

- Graphische Repräsentation einer Architektur durch **Zuverlässigkeitsblockdiagramm**
- Abbildung auf gleichwertige **Systemfunktion** (Strukturformel)
- Bestimmung des Fehlerbaums
- Transformation in Berechnungsformel

1. Blockdiagramm und Strukturformel

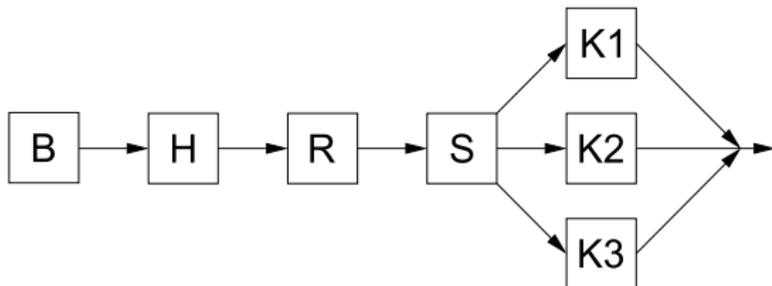
Gegeben sei ein portables Rechnersystem bestehend aus einer Batterie B , einer Hauptplatine H , der eigentlichen Recheneinheit R , dem Speicher S und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind die Batterie, die Hauptplatine, die Recheneinheit, der Speicher und mindestens eine Kommunikationskomponente erforderlich.

Erstellen Sie sowohl das Zuverlässigkeitsblockdiagramm als auch die Systemfunktion (Strukturformel), den Fehlerbaum und berechnen Sie die Funktionswahrscheinlichkeit.

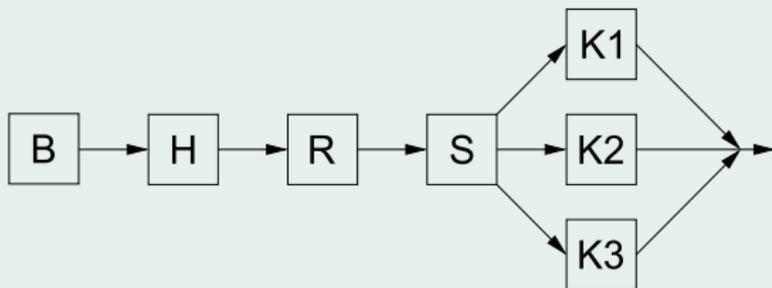
Aufgabe 1 – Fehlertoleranz

Gegeben sei ein portables Rechnersystem bestehend aus einer Batterie B , einer Hauptplatine H , der eigentlichen Recheneinheit R , dem Speicher S und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind die Batterie, die Hauptplatine, die Recheneinheit, der Speicher und mindestens eine Kommunikationskomponente erforderlich.

Zuverlässigkeitsblockdiagramm



Zuverlässigkeitsblockdiagramm



Systemfunktion (Strukturformel)

$$S = B \wedge H \wedge R \wedge S \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B \text{ and } H \text{ and } R \text{ and } S \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$$

Systemfunktion (Strukturformel)

$$S = B \wedge H \wedge R \wedge S \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B \text{ and } H \text{ and } R \text{ and } S \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$$

Fehlerbaum

Strukturbaum der Negation der Systemfunktion

$$\neg S = \neg(B \wedge H \wedge R \wedge S \wedge (K_1 \vee K_2 \vee K_3))$$

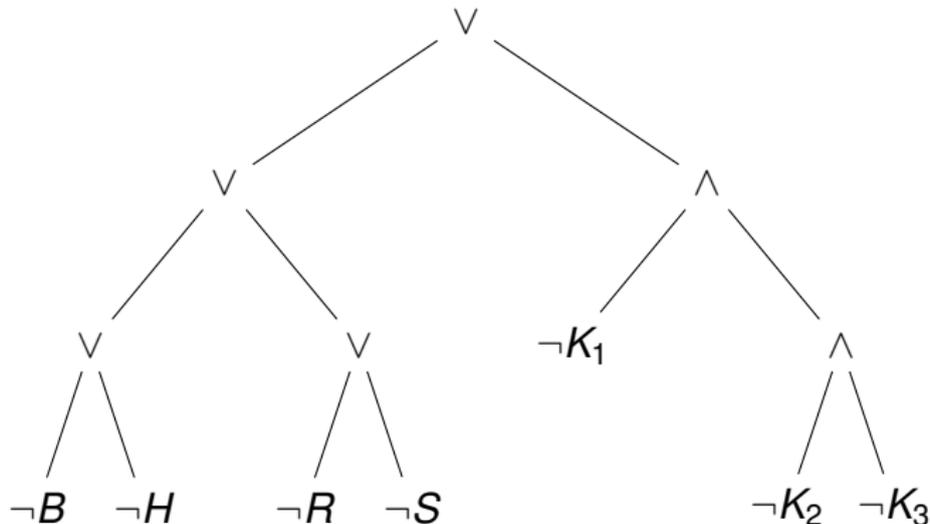
$$\neg S = \neg B \vee \neg H \vee \neg R \vee \neg S \vee \neg(K_1 \vee K_2 \vee K_3)$$

$$\neg S = \neg B \vee \neg H \vee \neg R \vee \neg S \vee (\neg K_1 \wedge \neg K_2 \wedge \neg K_3)$$

Fehlerbaum

Strukturbaum der Negation der Systemfunktion

$$\neg S = \neg B \vee \neg H \vee \neg R \vee \neg S \vee (\neg K_1 \wedge \neg K_2 \wedge \neg K_3)$$



Funktionswahrscheinlichkeit für 1-aus-n Systeme

- Betrachtung der **Ausfallwahrscheinlichkeit** (Nichtfunktionswahrscheinlichkeit):
 - Gegeben seien die Funktionswahrscheinlichkeiten $\varphi(B)$, $\varphi(H)$, $\varphi(R)$, $\varphi(S)$ und $\varphi(K)$
 - 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 1-aus-3 K-System damit:
 $(1 - \varphi(K))^3$
 - Anschließend: **Retransformation** in Funktionswahrscheinlichkeit
- Funktionswahrscheinlichkeit somit:

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

■ Zuverlässigkeitsblockdiagramm und Strukturformel:

Erfassung aller Funktionszustände

- Beispiel: 2-aus-3-System
- System funktionsfähig, wenn die Komponenten 1&2, 1&3, 2&3 oder 1&2&3 funktionsfähig sind
- System nicht funktionsfähig, wenn nur Komponente 1, 2, oder 3 funktionsfähig ist

■ 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 Komponenten
- $\varphi(V)$: Funktionswahrscheinlichkeit des Entscheiders (Voter)

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

2. Maßzahlen und Berechnung

a) Ein RAID6-System besteht aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu einem Datenverlust kommt. Unter der Annahme, die Funktionswahrscheinlichkeit pro Festplatte betrage $\varphi(F) = 0,99$, wie hoch ist die Chance auf Datenverlust?

■ allgemein:

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

■ mit $n = 8$, $m = 10$ und $\varphi(K) = \varphi(F) = 0,99$

$$\varphi_{10}^8 = \sum_{k=8}^{10} \binom{10}{k} * 0,99^k * 0,01^{(10-k)} = 0,999886$$

Chance auf Datenverlust somit: $1 - 0,999886 = 0,000114$

- **MTTF** – Mean Time to Failure:
 - mittlere Funktionszeit
 - Einheit, die nicht instand gesetzt wird (Ersatz)
- **MTBF** – Mean Time between Failures:
 - mittlere Zeit zwischen Ausfällen
 - Einheit, die wieder instand gesetzt wird
 - Betriebszeit zwischen zwei aufeinanderfolgenden Ausfällen
- **MTTR** – Mean Time to Repair/Recover:
 - mittlere Reparaturzeit
- Zeit zwischen zwei aufeinander folgenden Systemstarts:
 $MTBF + MTTR$

- **Punktverfügbarkeit** (V) eines Systems:
 - auch A für Availability
 - Wahrscheinlichkeit, ein System zu einem beliebigen Zeitpunkt fehlerfrei anzutreffen
 - unabhängig davon, ob es bis zu diesem Zeitpunkt bereits ausgefallen ist oder nicht

$$V = \frac{MTTF}{MTTF + MTTR} \quad \text{oder} \quad V = \frac{MTBF}{MTBF + MTTR}$$

- Für über die Zeit konstante Ausfallraten gilt außerdem:

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

2. Maßzahlen und Berechnung

b) Eine Festplatte habe eine MTTF von 20 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 .

$$\blacksquare \text{ MTTF} = 20a = (20 * 365 * 24 * 60) \text{ min} = 10\,512\,000 \text{ min}$$

$$\blacksquare \text{ MTTR} = (2 + 10 + 2) \text{ min} = 14 \text{ min}$$

$$\blacksquare V = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} = \frac{10\,512\,000 \text{ min}}{10\,512\,014 \text{ min}} = 0,9999987$$

Alterungseffekte:

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

Badewannenkurve

1 Frühphase

- Initialausfälle
- Fertigungsfehler, Bauteildefekte
- Ausfallrate exponentiell abfallend

2 Betriebsphase

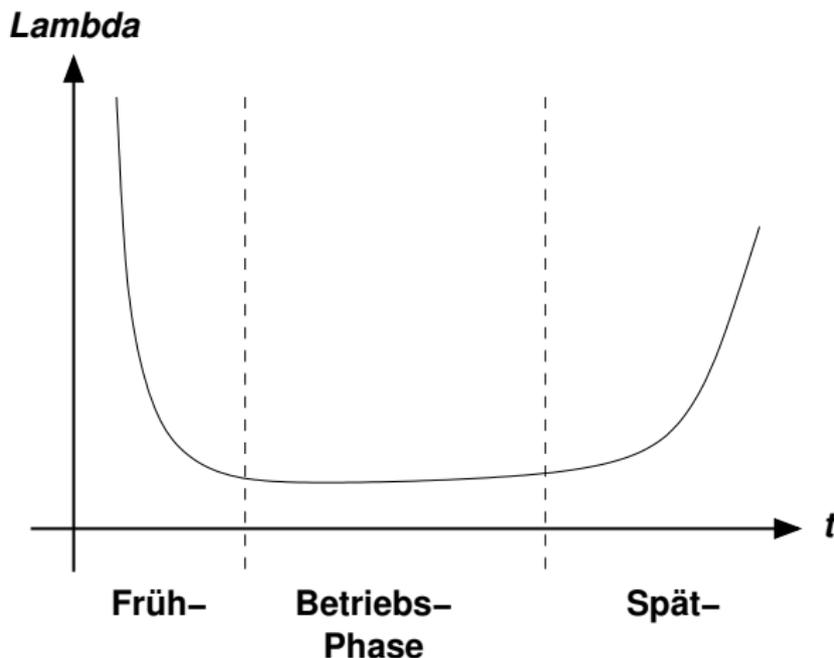
- Nahezu konstante Ausfallrate

3 Spätphase

- Alterungseffekte
- Ausfallrate exponentiell ansteigend

Aufgabe 1 – Fehlertoleranz – 2.c)

Alterungseffekte: Badewannenkurve



Rechnen mit zeitlich variabler Ausfallrate:

2. Maßzahlen und Berechnung

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.

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

Aufgabe 1 – Fehlertoleranz – 2.d)

Rechnen mit zeitlich variabler Ausfallrate:

- i. Wie groß ist die Ausfallrate für eine einzelne Komponente?

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

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

- ii. Bestimmen Sie die 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.

Aufgabe 1 – Fehlertoleranz – 2.d)

Rechnen mit zeitlich variabler Ausfallrate:

- ii. 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: $R(K, t) = R(S_{2v3}, t)$

$$\Leftrightarrow R = 3 * R^2 - 2 * R^3$$

$$\Rightarrow R_1 = 0, R_2 = 1, R_3 = 0, 5$$

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 $[t_2, t_3) = [0, \frac{\ln(2)}{\lambda})$.

Aufgabe 1 – Fehlertoleranz – 2.d)

Rechnen mit zeitlich variabler Ausfallrate:

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

Es gilt:

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

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

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.

a) Vermeidung von Pipelineleerlauf

- Statische Sprungvorhersage
 - Richtung der Vorhersage für Befehl immer gleich
 - Always Taken, Always Not Taken
- Statische Sprungvorhersage – **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

a) Vermeidung von Pipelineleerlauf

- Dynamische Sprungvorhersage – **Prädiktion**
 - Vorhersage auf Grund von bisherigem Programmablauf
 - Sprungbefehle werden ausgeführt
 - Nächsten Befehle werden entsprechend der Vorhersage über den Sprungausgang spekulativ geladen und ausgeführt
 - Benötigte Hardware: Sprungzieltabelle, Prädiktor(en)

Weitere Informationen

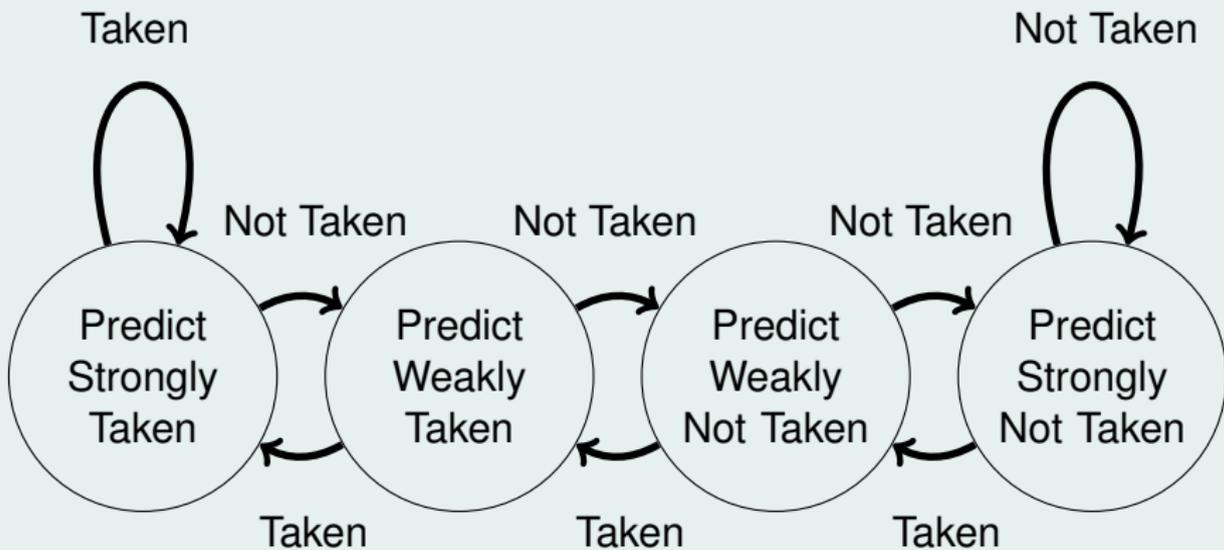
- Brinkschulte, Ungerer: Mikrocontroller und Mikroprozessoren, Kap. 2.4.6, 7.2

b) 2-Bit-Prädiktoren

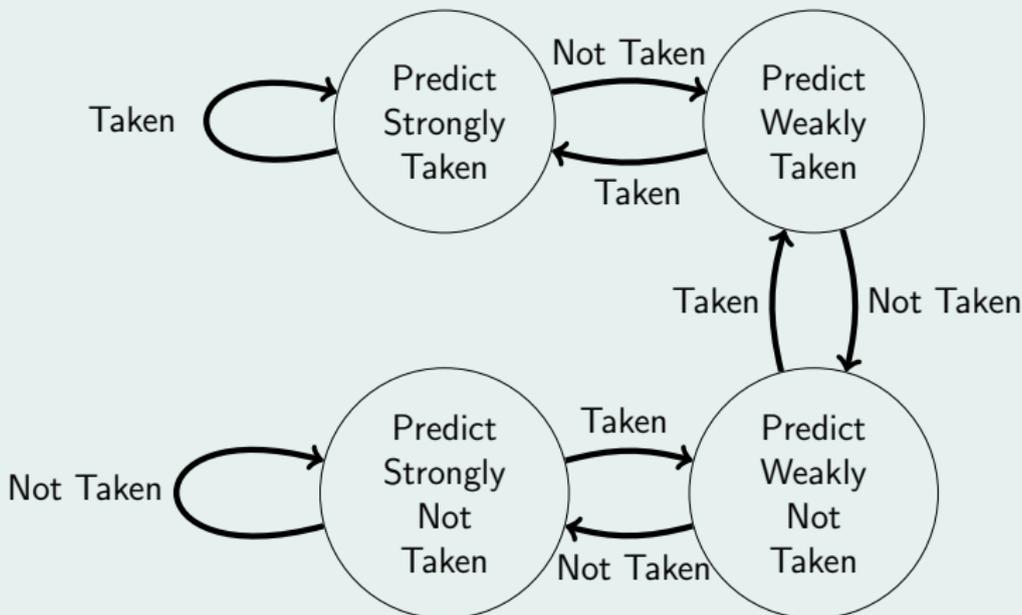
Zeichnen Sie einen 2-Bit-Prädiktor einmal mit Sättigungszähler und einmal mit Hysteresezähler.

Worin liegt die Motivation zur Verwendung eines Hysterese- anstelle eines Sättigungszählers?

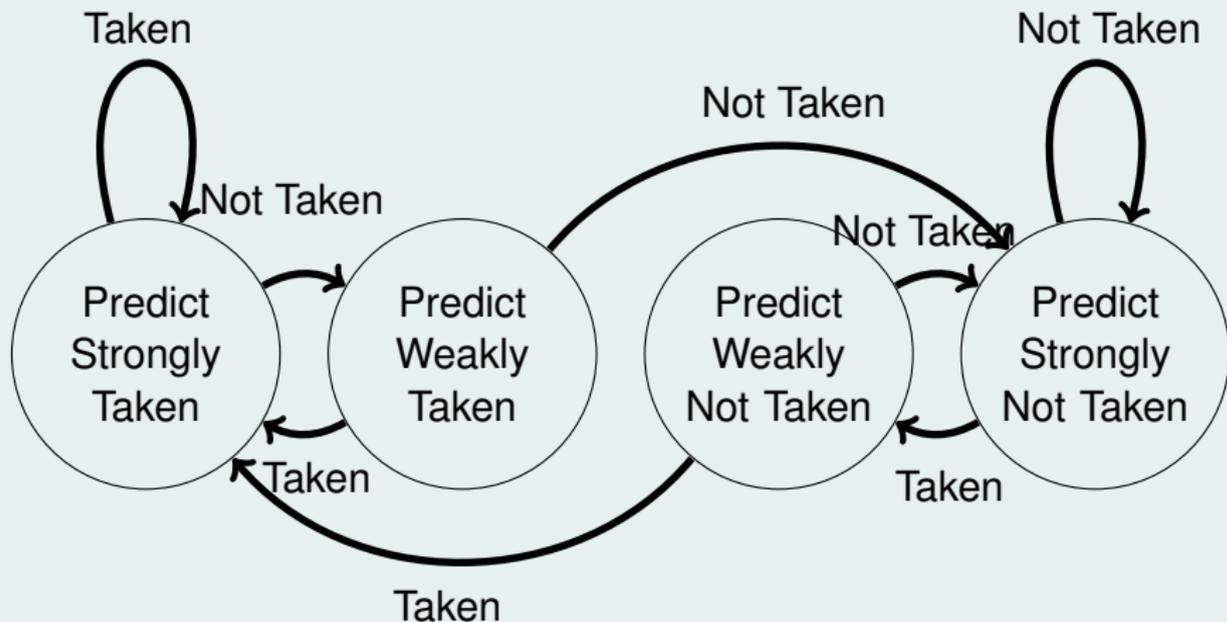
b) 2-Bit-Prädiktoren – Sättigungszähler



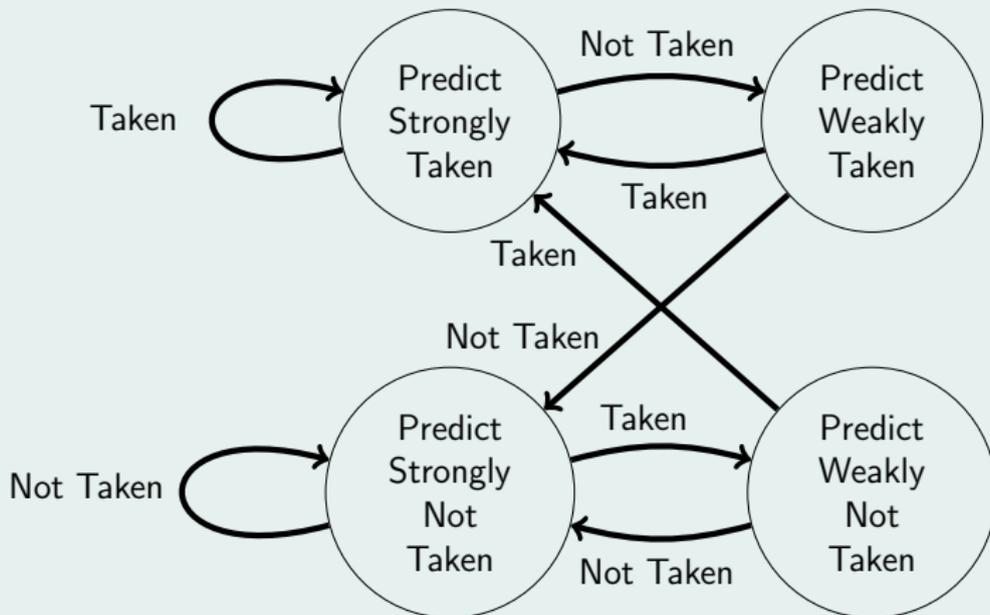
b) 2-Bit-Prädiktoren – Sättigungszähler



b) 2-Bit-Prädiktoren – Hysteresezähler



b) 2-Bit-Prädiktoren – Hysteresezähler



b) 2-Bit-Prädiktoren – Diskussion

Hysteresezähler:

- Aggressiveres Umschaltverhalten
- Wechsel in den starken Zustand der entgegengesetzten Vorhersage schon bei der zweiten Fehlvorhersage
- Vermeidet „Flattern“ zwischen Weakly-Zuständen

c) 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 die Prädiktoren mit Predict Weakly Not Taken (WNT) initialisiert seien.

```
1 INIT:  ADD  R1,R0,#0 ; R1=0
2       ADD  R2,R0,#2 ; R2=2
3
4 START:  BNE  R1,R0,L1 ; if (R1!=0) goto L1
5       ADD  R1,R0,#1 ; R1=1
6
7 L1:    SUB  R3,R1,R2 ; R3=R1-R2
8       BNE  R3,R0,L2 ; if (R1!=R2) goto L2
9       ADD  R1,R0,#0 ; R1=0
10      J    START ; goto START
11
12 L2:    ADD  R1,R0,#2 ; R1=2
13      J    START ; goto START
```

Aufgabe 2 – Sprungvorhersage I

c) Sprungverläufe

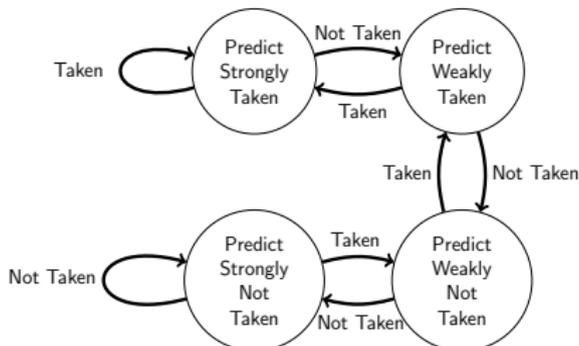
Sprung 1 Zeile 4 BNE L1	Sprung 2 Zeile 8 BNE L2	Sprung 3 Zeile 10 J START	Sprung 4 Zeile 13 J START
NT (R1=0)	T (R1=1)	–	T
T (R1=2)	NT (R1=2)	T	–
NT (R1=0)	T (R1=1)	–	T
T (R1=2)	NT (R1=2)	T	–

```
1  INIT:  ADD  R1,R0,#0  ; R1=0
2         ADD  R2,R0,#2  ; R2=2
3
4  START: BNE  R1,R0,L1  ; if (R1!=0) goto L1
5         ADD  R1,R0,#1  ; R1=1
6
7  L1:    SUB  R3,R1,R2   ; R3=R1-R2
8         BNE  R3,R0,L2  ; if (R1!=R2) goto L2
9         ADD  R1,R0,#0  ; R1=0
10        J    START    ; goto START
11
12  L2:    ADD  R1,R0,#2  ; R1=2
13        J    START    ; goto START
```

Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

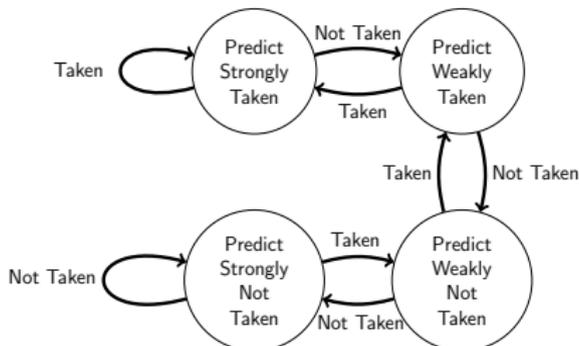
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT			T	
	T			NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

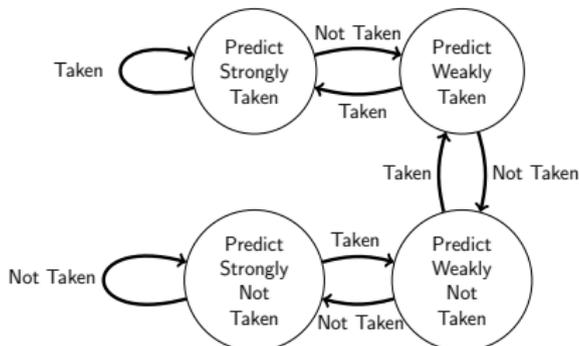
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	T
	T				
	NT				
	T				



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

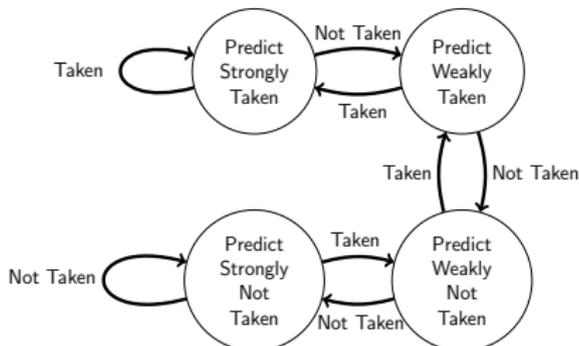
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T			NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

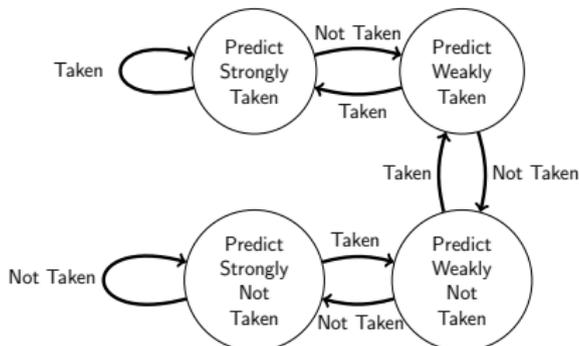
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

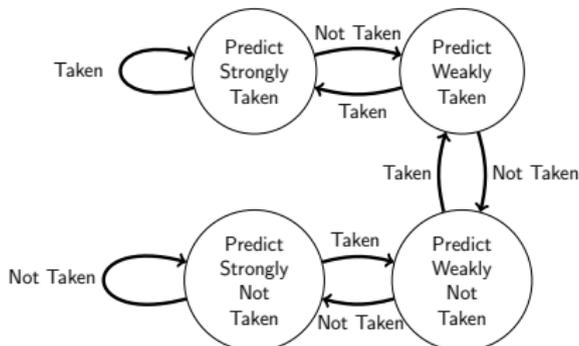
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

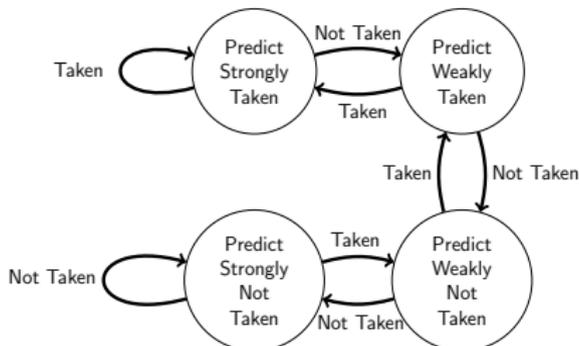
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT	SNT	SNT	T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

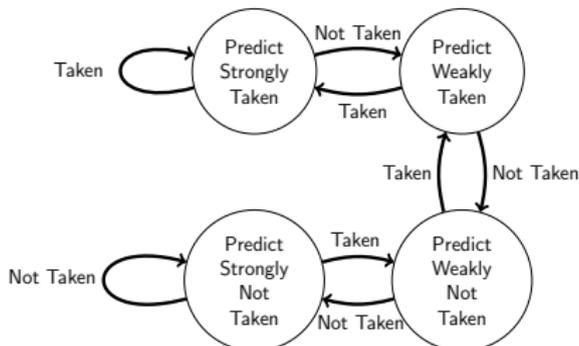
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT	SNT	SNT	T	WNT
WNT	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		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	



c) Gemeinsamer 2-Bit-Prädiktor mit Sättigungszähler

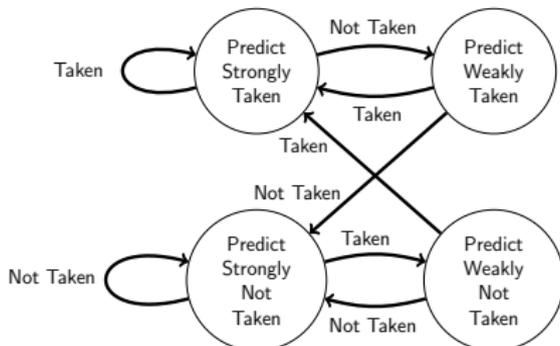
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT

⇒ Es werden **6** Fehlannahmen gemacht!

Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

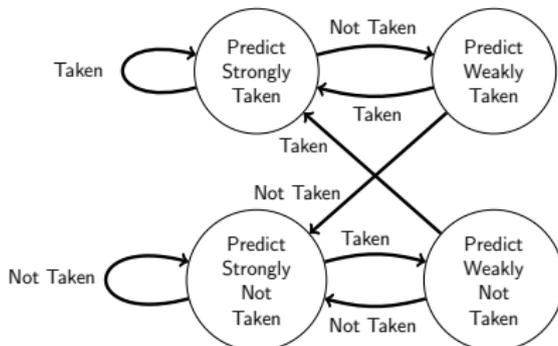
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT			T	
	T			NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

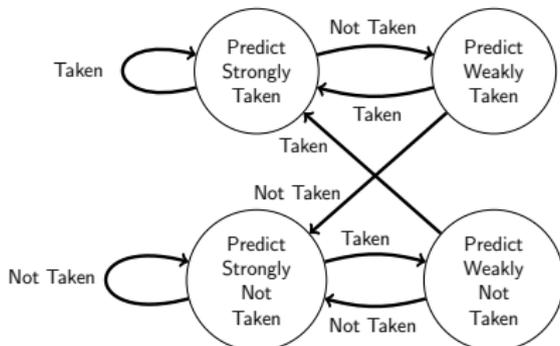
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	T
	T				
	NT				
	T				



Aufgabe 2 – Sprungvorhersage I

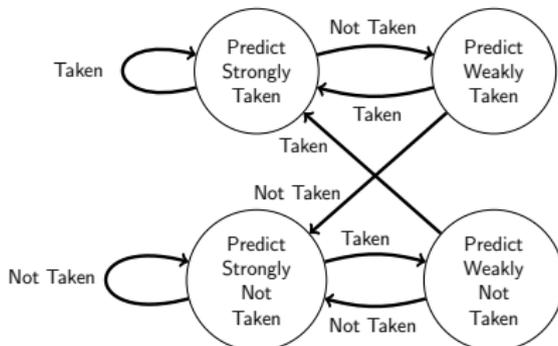
c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

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



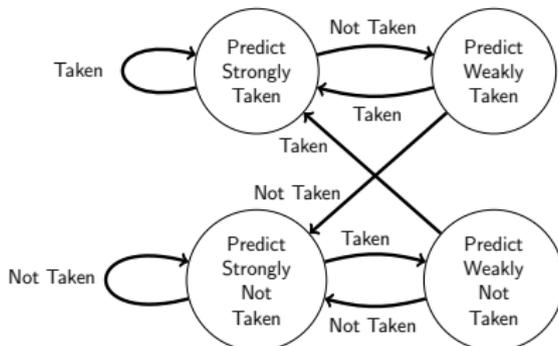
c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

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



c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

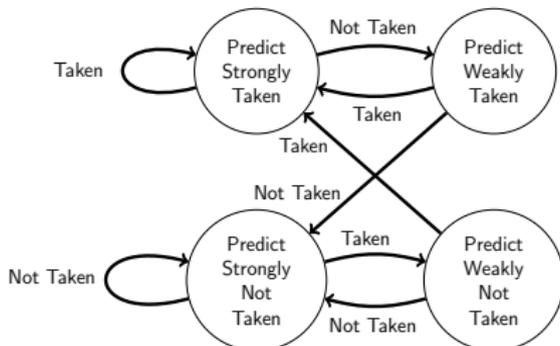
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	SNT	T	WNT
WNT	T	ST	ST	NT	WT
WT	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

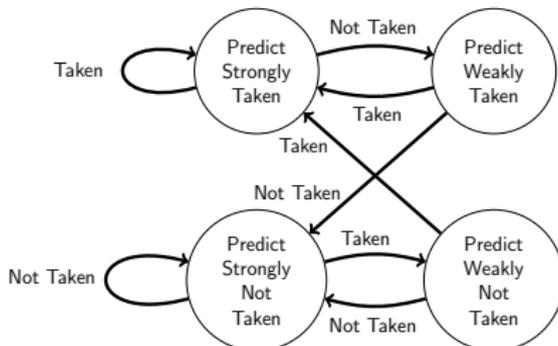
c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

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



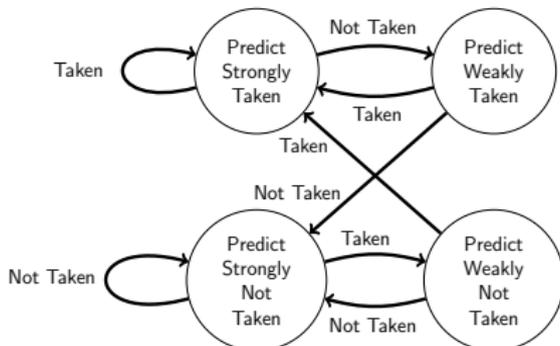
c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

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	



c) Gemeinsamer 2-Bit-Prädiktor mit Hysteresezähler

Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		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	



c) Gemeinsamer 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

⇒ Es werden **7** Fehlannahmen gemacht!

d) Sprungeigener 2-Bit-Prädiktor

Nehmen Sie nun an, dass **jeder Sprung über einen eigenen Prädiktor verfüge** (Lokale Prädiktion).

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

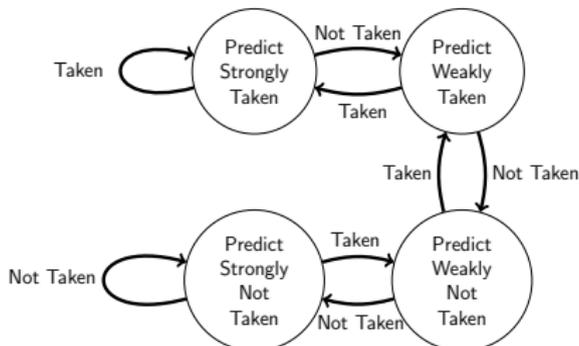
Bisherige Bedingungen:

- gleiches Programm wie vorher
- jeweils Initialisierung mit Predict Weakly Not Taken (WNT)
- je Durchlauf für Sättigungs- und Hysteresezaehler

Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

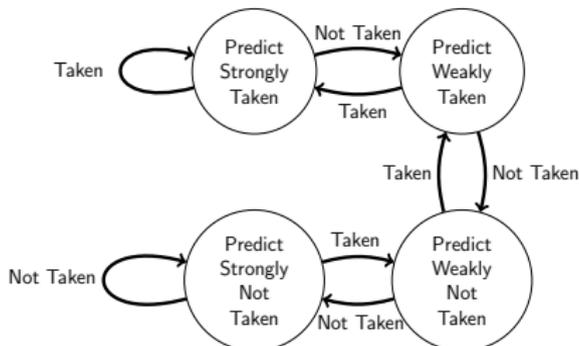
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT		WNT	T	
	T			NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

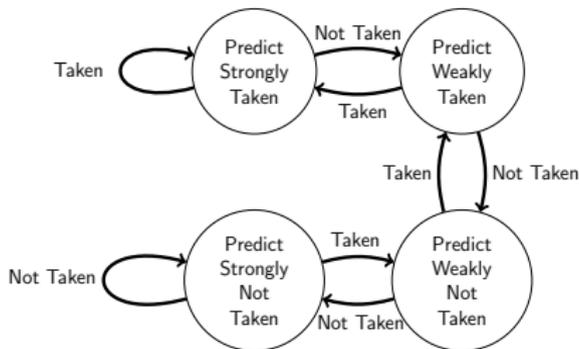
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	WNT	T	
SNT	T			NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

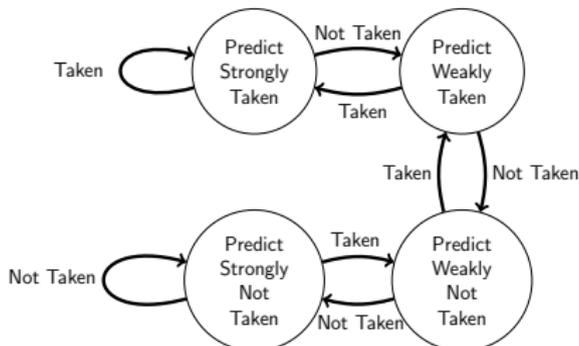
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T		WT	NT	
	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

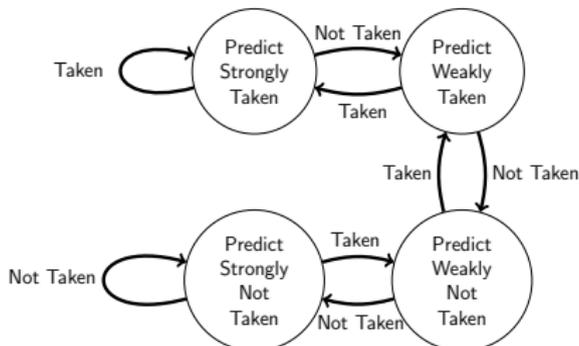
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	
WNT	NT			T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

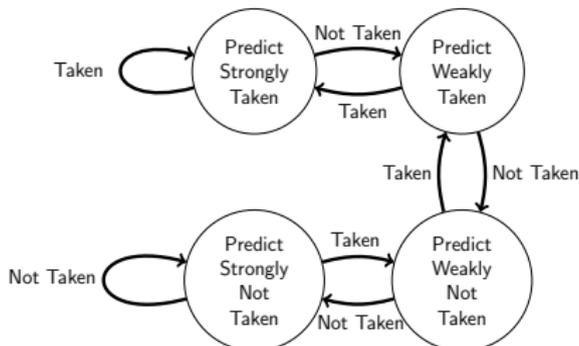
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT		WNT	T	
	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

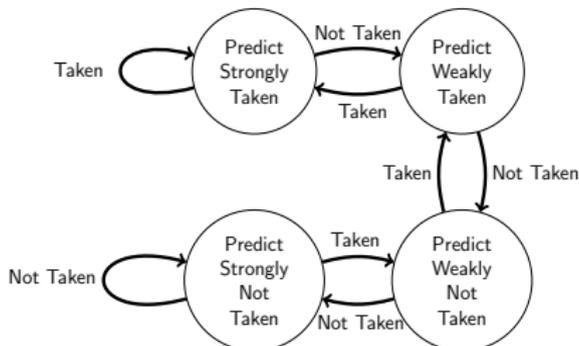
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT	SNT	WNT	T	
SNT	T			NT	



Aufgabe 2 – Sprungvorhersage I

d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

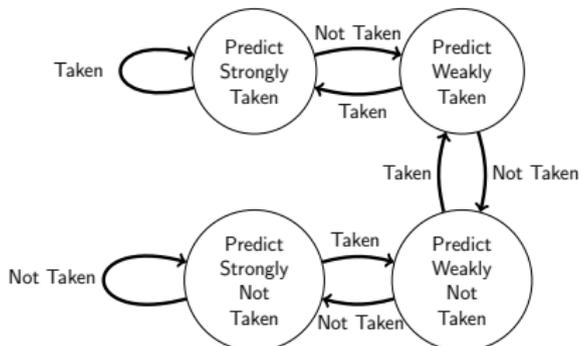
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT	SNT	WNT	T	WT
SNT	T		WT	NT	



Aufgabe 2 – Sprungvorhersage I

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	



d) Sprungeigener 2-Bit-Prädiktor mit Sättigungszähler

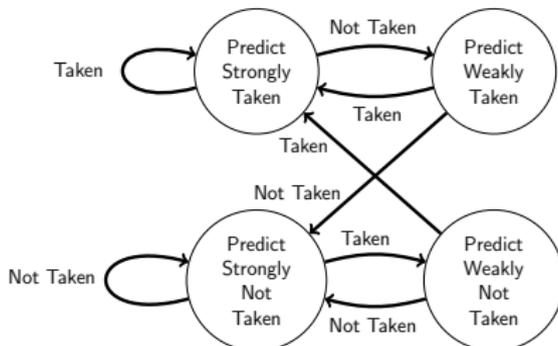
Sprung 1			Sprung 2		
Präd.	Sprung	Neue P.	Präd.	Sprung	Neue P.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT

⇒ Es werden **6** Fehlannahmen gemacht!

Aufgabe 2 – Sprungvorhersage I

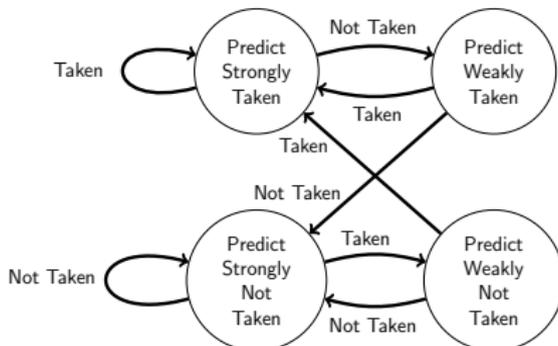
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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



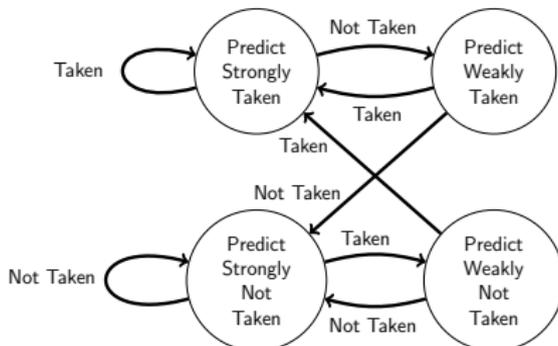
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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



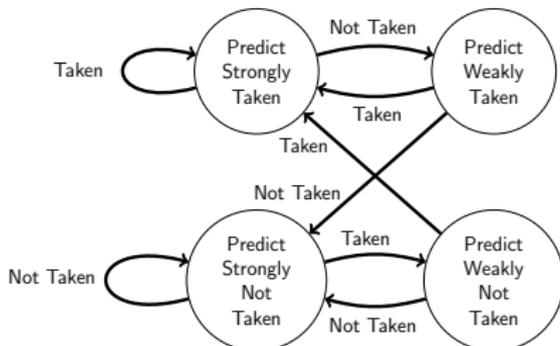
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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



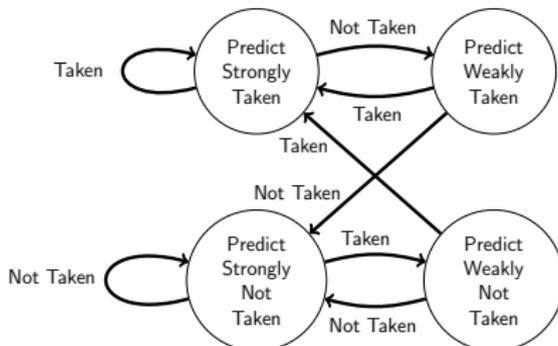
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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



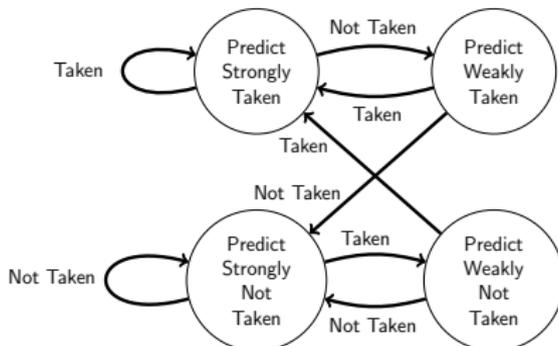
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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



d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

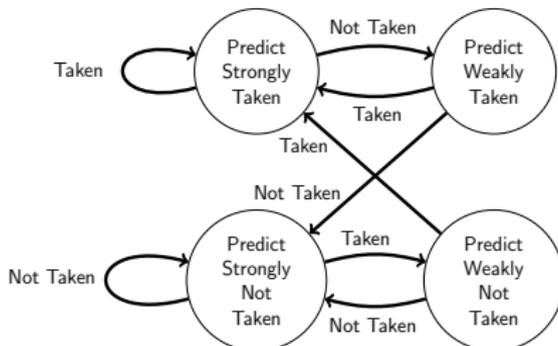
Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		Sprung	Neue P.
WNT	NT	SNT	WNT	T	ST
SNT	T	WNT	ST	NT	WT
WNT	NT	SNT	WT	T	
SNT	T			NT	



Aufgabe 2 – Sprungvorhersage I

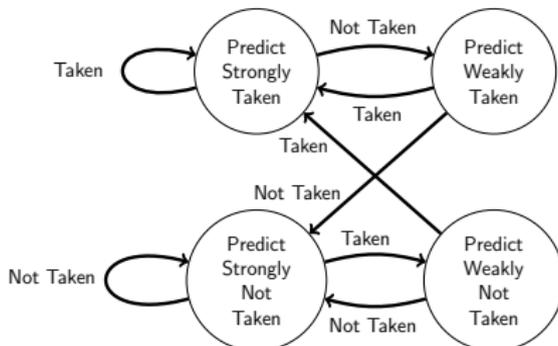
d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

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	



d) Sprungeigener 2-Bit-Prädiktor mit Hysteresezähler

Präd.	Sprung 1		Präd.	Sprung 2	
	Sprung	Neue P.		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	



Aufgabe 2 – Sprungvorhersage I

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!

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.

e) (m,n) -Korrelationsprädiktoren

- m Sprünge umfassende Historie
- Speicherung der letzten m Sprünge in Sprungverlaufsregister (Branch History Register, BHR)
- 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

Aufgabe 2 – Sprungvorhersage I

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						

Aufgabe 2 – Sprungvorhersage I

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		

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

Aufgabe 2 – Sprungvorhersage I

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	

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	

Aufgabe 2 – Sprungvorhersage I

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)

⇒ Es werden **5** Fehlvorhersagen gemacht!

Aufgabe 3 – Sprungvorhersage II

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.

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.

Aufgabe 3 – Sprungvorhersage II

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
```

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 $100 * 100 = 10.000$ mal
 - Taken $100 * 1 = 100$ mal

Etwas Statistik

Anzahl an genommenen / nicht genommenen Sprüngen pro Sprungadresse:

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

Aufgabe 3 – Sprungvorhersage II

Etwas Statistik – 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

Aufgabe 3 – Sprungvorhersage II

1. 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

Aufgabe 3 – Sprungvorhersage II

Code – globaler 1-Bit Prädiktor

```
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
```

2. 1-Bit Prädiktor (global)

- `0x10C`: Da Prädiktor initialisiert mit T \Rightarrow Fehlvorhersage
 - Bei den weiteren Durchläufen bestimmt `0x134` die Vorhersage
 - Prediction immer Taken
 - Sprung wird aber nur einmal genommen (am Ende des Programms)

\Rightarrow **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

\Rightarrow **9900 Fehlvorhersagen**

Aufgabe 3 – Sprungvorhersage II

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
```

2. 1-Bit Prädiktor (global)

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

2. 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

⇒ Total: 20000 Fehlvorhersagen

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

Aufgabe 3 – Sprungvorhersage II

Code – 1-Bit Prädiktor (32 Einträge)

```
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
```

3. 1-Bit Prädiktor (32 Einträge)

Sprünge $0x10C$ und $0x12C$ sowie Sprünge $0x114$ und $0x134$ fallen zusammen

- $0x10C$: 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**

Aufgabe 3 – Sprungvorhersage II

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
```

3. 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**

Aufgabe 3 – Sprungvorhersage II

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

⇒ Total: 10300 Fehlvorhersagen

Aufgabe 3 – Sprungvorhersage II

Code – 2-Bit Prädiktor (16 Einträge)

```
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
```

4. 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
⇒ **100 Fehlvorhersagen**
- 0x114: Bedingt durch die Initialisierung mit Strongly Taken, Fehlvorhersage der ersten beiden Iterationen ⇒ danach sagt der Prädiktor Not Taken vorher
 - Beim Wechsel von der inneren zur äußeren Schleife wird sowohl 0x114, als auch 0x134 genommen ⇒ Vorhersage dann Weakly Taken
 - Fehlvorhersage der neuen, ersten Iteration der inneren Schleife
⇒ **201 Fehlvorhersagen**

Aufgabe 3 – Sprungvorhersage II

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
```

4. 2-Bit Prädiktor (16 Einträge)

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

Aufgabe 3 – Sprungvorhersage II

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
```

4.) 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**

4. 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

⇒ Total: 7851 Fehlvorhersagen

b) Matrix-Multiplikation

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

Verwenden Sie folgende Sprungvorhersageeinheiten:

- Always taken
- Always nontaken
- 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.

b) Matrix-Multiplikation

	taken	not taken	bimod 1	bimod 16	2lev
Misses	17178	2131193	17305	17071	16988
Rate	99,60%	50,39%	99,60%	99,60%	99,60%

⇒ Der Korrelationsprädiktor bietet die höchste Trefferrate (am wenigsten Misses) und sollte daher gewählt werden.

Unter Umständen weichen die mit dem Virtualbox-Image erzielten Ergebnisse geringfügig von den folgenden Werten ab.

c) 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 Aufgabenteil b)

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

c) MiBench Benchmark Suite

		taken	not taken	bimod 1	bimod 16	2lev
basicmath	Misses	11142634	13658568	11907656	10001635	2400198
	Rate	66,06%	58,40%	63,73%	69,53%	92,69%
qsort	Misses	2517179	3101204	2946756	1485265	446568
	Rate	68,51%	61,20%	63,13%	81,42%	94,29%
susan -s	Misses	238448	1650033	247977	117405	116422
	Rate	87,42%	12,97%	86,92%	93,81%	93,86%
susan -e	Misses	11721	72864	11755	6273	3406
	Rate	87,29%	21,01%	87,26%	93,20%	96,31%
susan -c	Misses	20484	27833	8895	5547	2409
	Rate	62,97%	49,68%	83,92%	89,97%	95,64%

⇒ Der Korrelationsprädiktor bietet die höchste Trefferrate und sollte daher gewählt werden.

Unter Umständen weichen die mit dem Virtualbox-Image erzielten Ergebnisse geringfügig von den folgenden Werten ab.

Übung #4 – Di 07.06.2016

- Pipelining
- Superskalartechniken

Zentralübung Rechnerstrukturen im SS 2016

Fehlertoleranz und Sprungvorhersage

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

2. Juni 2016

