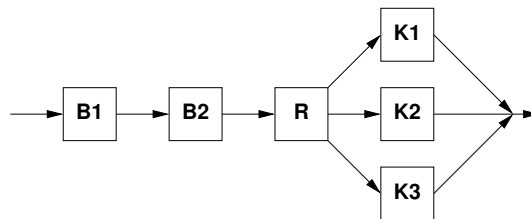


Zentralübung Rechnerstrukturen: Fehlertoleranz und Sprungvorhersage

Musterlösung zu Aufgabenblatt 3

Fehlertoleranz

1. Blockdiagramm und Strukturformel Zuverlässigkeitsblockdiagramm:



Strukturformel: $S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$
(alternativ: $S = B_1 \text{ and } B_2 \text{ and } R \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$)

Berechnung der Funktionswahrscheinlichkeit:

- Berechnung der Funktionswahrscheinlichkeit des parallelen Teilsystems K_1 bis K_3 durch Überführung in entsprechendes Seriensystem und Berechnung der Ausfallwahrscheinlichkeit:

$$(K_1 \vee K_2 \vee K_3) \rightarrow \neg(\neg K_1 \wedge \neg K_2 \wedge \neg K_3)$$

- $K \rightarrow \neg K$, entsprechend $\varphi(K) \rightarrow 1 - \varphi(K)$,
d.h. Ausfallwahrscheinlichkeit für K-System: $(1 - \varphi(K))^3$
- Transformierung in Funktionswahrscheinlichkeit: $\varphi(K) = 1 - ((1 - \varphi(K))^3)$
- Berechnung der Funktionswahrscheinlichkeit des kompletten Seriensystems:

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

2. Maßzahlen und Berechnung

a) 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: $1 - 0.999886 = 0.000114$.b) $MTTF = 2a = (2 * 365 * 24 * 60)min = 1051200min$ $MTTR = (2 + 10 + 2)min = 14min$

$$V = \frac{MTTF}{MTTF + MTTR} = \frac{1051200}{1051214} = 0,999997$$

c) Bei der Berechnung der Punktverfügbarkeit wird typischerweise von einer konstanten Ausfallrate ausgegangen; dies trifft näherungsweise für die Betriebsphase der Badewannenkurve zu.

d) – allgemein: $z(t) = \frac{f_L(t)}{R(t)} = \frac{1}{R(t)} \cdot \frac{d}{dt}(F_L(t)) = \frac{1}{R(t)} \cdot \frac{d}{dt}(1 - R(t)) = \frac{-R'(t)}{R(t)}$

$$\rightarrow z(t) = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda.$$

– 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: $R(K, t) = R(t)$

$$\text{somit gilt: } R(K, t) = R(S_{2v3}, t) \quad \Leftrightarrow \quad R = 3 * R^2 - 2 * R^3$$

$$\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$$

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

– Es gilt:

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

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

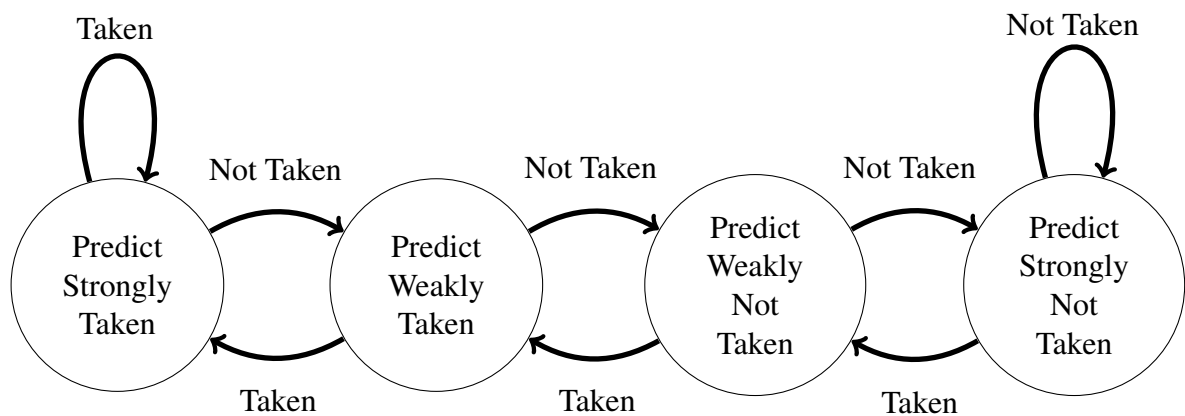
Sprungvorhersage

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

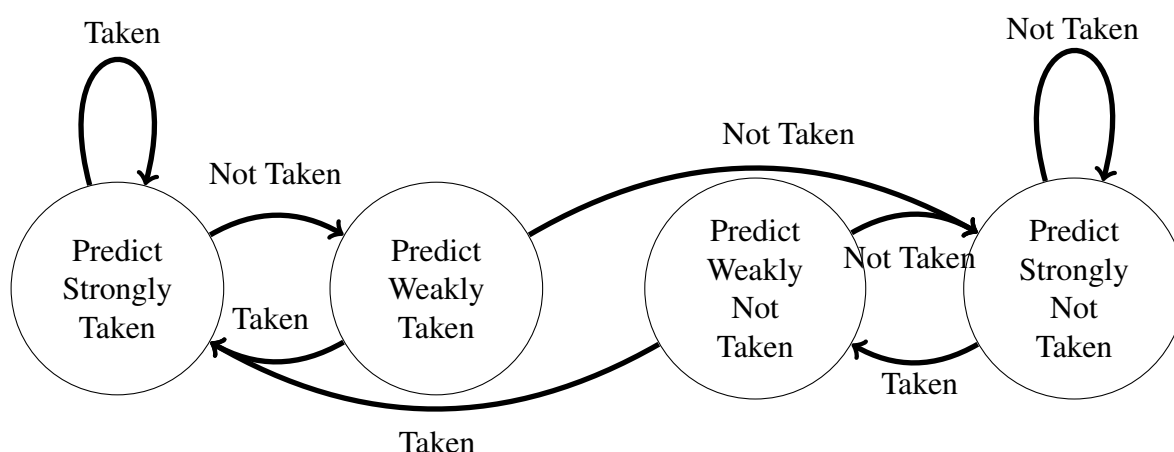
Technik:	Prädikation	Prädiktion
Beschreibung	Verwenden von Prädikaten	Sprungvorhersage
Art	Statisch	Dynamisch
Umsetzung	Compiler-gestützt	Hardware-Technik
Sprungbefehle	Keine, stattdessen spekulative Ausführung der mit Prädikat versehenen Befehle und Gültig-machen bei Evaluation des Prädikatregisters zu Wahr	Werden ausgeführt und die nächsten Befehle entsprechend der Vorhersage über den Sprungausgang geladen und spekulativ ausgeführt
Hardware	Spekulative Ausführung, Prädikatregister	Sprungzieltabelle, Prädiktor(en)

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

Sättigungszähler:



Hysteresezähler:



Der Hysteresezähler wechselt bei der zweiten Fehlvorhersage in den starken Zustand der entgegengesetzten Vorhersage, um Flattern zwischen den schwachen Zuständen zu vermeiden.

3. 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 seien.

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	–

In Klammern steht der Wert von R1 **vor** dem Sprung.

Hierbei müssen die letzten zwei Sprünge nicht berücksichtigt werden: Ihre Bedingung ist immer wahr und somit wird immer gesprungen.

Zwei-Bit-Prädiktor mit Sättigungszähler

Es werden **6** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	Sprung	Neue Vorh.	Prädiktion	Sprung	Neue Vorh.
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT
WNT	NT	SNT	SNT	T	WNT
WNT	T	WT	WT	NT	WNT

Zwei-Bit-Prädiktor mit Hysteresezähler

Es werden **7** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	Sprung	Neue Vorh.	Prädiktion	Sprung	Neue Vorh.
WNT	NT	SNT	SNT	T	WNT
WNT	T	T	T	NT	WT
WT	NT	SNT	SNT	T	WNT
WNT	T	T	T	NT	WT

4. Nehmen Sie nun an, dass jeder Sprung über einen eigenen Prädiktor verfüge (Lokale Prädiktion). Welchen Unterschied stellen Sie fest, worauf lässt sich dieser zurückführen?

Zwei-Bit-Prädiktor mit Sättigungszähler

Es werden **6** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	Sprung	Neue Vorh.	Prädiktion	Sprung	Neue Vorh.
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT
WNT	NT	SNT	WNT	T	WT
SNT	T	WNT	WT	NT	WNT

Zwei-Bit-Prädiktor mit Hysteresezähler

Es werden **5** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	Sprung	Neue Vorh.	Prädiktion	Sprung	Neue Vorh.
WNT	NT	SNT	WNT	T	T
SNT	T	WNT	T	NT	WT
WNT	NT	SNT	WT	T	ST
SNT	T	WNT	ST	NT	WT

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.

5. Gegeben Sei nun ein (1,2)-Korrelationsprädiktor, der global verwendet werde. Das Schieberegister BHR sei mit NT initialisiert, und die beiden 2-Bit-Hystereseprediktoren mit Predict Weakly Taken. Füllen Sie untenstehende Tabelle für die oben ermittelten Sprungausgänge aus.

Es werden **5** Fehlvorhersagen gemacht.

Sprungzeile	Richtung	Aktuelle Vorhersage			Neue Vorhersage	
		Historie	Prädiktor	Vorh.	Akt. Historie	Akt. Prädiktoren
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)

1 Sprungvorhersage II

Anzahl an genommenen / nicht genommenen Sprüngen pro Sprungadresse:

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

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

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

1. Always Taken

Alle Sprünge mit Ausgang Not Taken werden falsch vorhergesagt.

2. Ein globaler 1-Bit Prädiktor, initialisiert mit Taken.

- 0x10C: Da Prädiktor initialisiert mit T $\Rightarrow$ Fehlvorhersage
Bei den weiteren Durchläufen bestimmt 0x134 die Vorhersage $\Rightarrow$ Prediction immer Taken $\Rightarrow$ 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
 - 0x120: Vorhersage wird von 0x114 bestimmt $\Rightarrow$ hier Vorhersage immer taken $\Rightarrow$ 50% Fehlvorhersagen (alle Sprünge mit Ausgang Not Taken)
 - 0x12C: Vorhersage wird von 0x120 bestimmt $\Rightarrow$ Vorhersage alterniert $\Rightarrow$ 50% Fehlvorhersagen
 - 0x134: Vorhersage wird von 0x114 bestimmt $\Rightarrow$ Vorhersage immer taken $\Rightarrow$ keine Fehlvorhersagen
3. Ein 1-Bit Prädiktor mit 32 Einträgen, die niedrigsten Bits der Instruktionsadresse werden zur Indizierung des Eintrags verwendet, initialisiert mit Taken.
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 $\Rightarrow$ 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 $\Rightarrow$ 200 Fehlvorhersagen
 - 0x120: Prädiktor alterniert ständig zwischen Taken und Nottaken. $\Rightarrow$ Ständige Fehlvorhersagen, außer bei Eintritt in die innere Schleife $\Rightarrow$ 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 $\rightarrow$ 100 Fehlvorhersagen
 - 0x134: Vorhersage wird von 0x114 bestimmt $\Rightarrow$ Vorhersage immer taken $\Rightarrow$ keine Fehlvorhersagen
4. Ein 2-Bit Prädiktor mit 16 Einträgen, die niedrigsten Bits der Instruktionsadresse werden zur Indizierung des Eintrags verwendet, initialisiert mit Strongly Taken.
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
- 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
- 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

Statistiken				Fehlvorhersagen			
Sprung	Bits	Taken	Not Taken	Taken	1Bit	1Bit (32 Einträge)	2Bit
0x10C	01100	1	100	100	100	100	100
0x114	10100	100	10000	10000	9900	200	201
0x120	00000	5000	5000	5000	5000	9900	7450
0x12C	01100	10000	0	0	5000	100	0
0x134	10100	100	0	0	0	0	100
				15100	20000	10300	7851

2 SimpleScalar

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

1. 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%

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

2. 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%

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