

Sprungvorhersagestrategien für Befehlspipelines

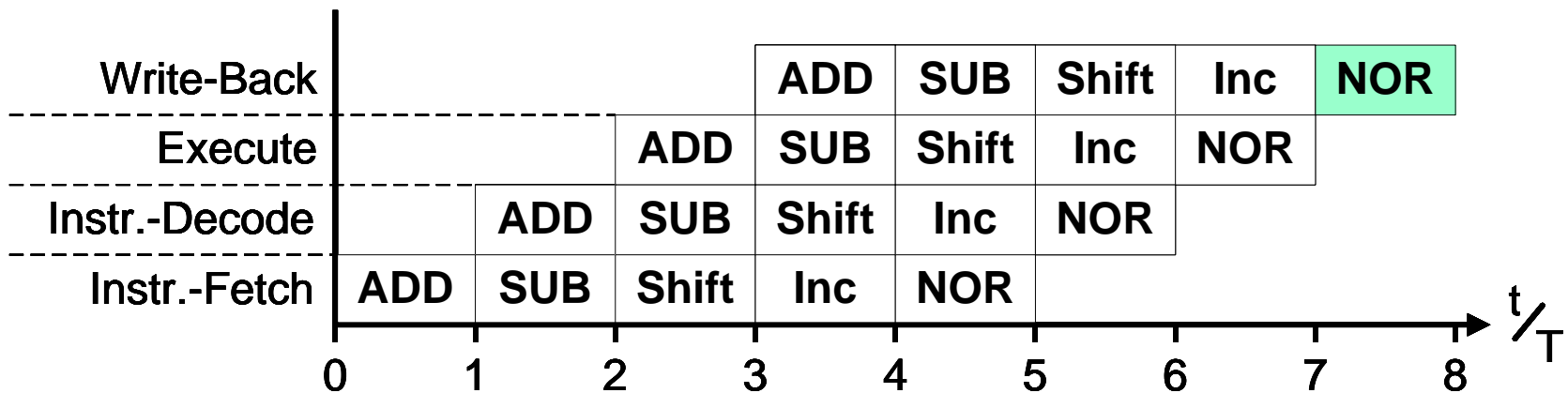
Robert Ramm

Dresden, 09.07.2010

- **Was sind Befehlspipelines?**
- **Warum Sprungvorhersage?**

Befehlspipelines

- Aufteilung der Maschinenbefehle in Teilaufgaben
(typisch : Fetch-, Decode-, Execution-, Write-Back-Phase)
- gleichzeitige Bearbeitung der Teilaufgaben unterschiedlicher Befehle
- dadurch wesentliche Verkürzung des kritischen Pfades
→ größere Taktfrequenz möglich



Pipeline-Hazards

Man unterscheidet zwischen 3 verschiedenen Art von Pipelinekonflikten.

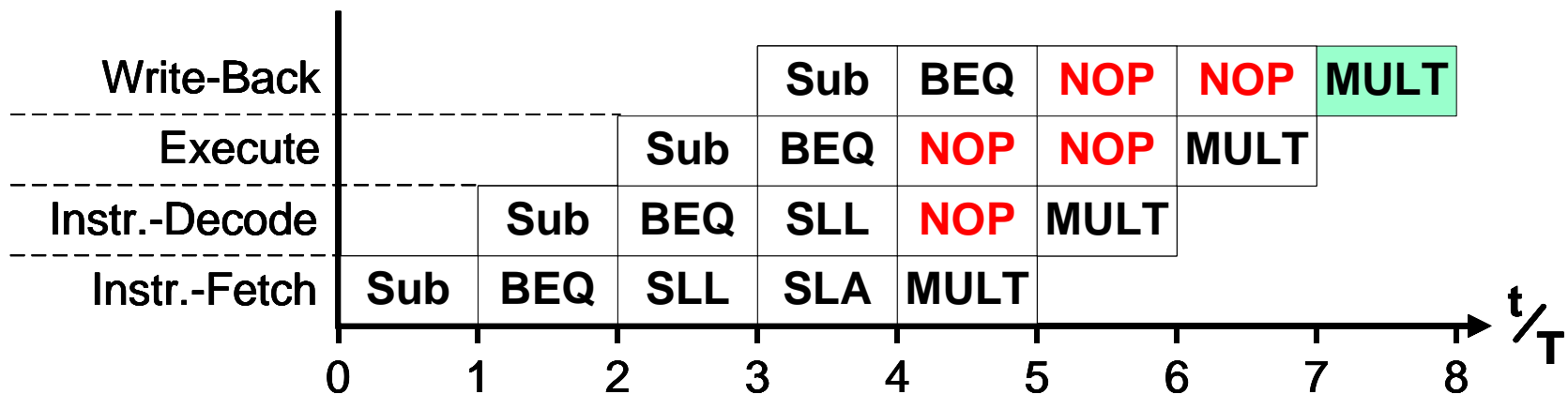
Ressourcenkonflikte: zwei oder mehr Stufen der Pipeline benötigen gleichzeitig Zugriff auf die gleiche Ressource

Datenkonflikte: durch Datenabhängigkeiten zwischen zwei oder mehr Befehlen im Befehlsfluss

Kontrollflusskonflikte: wenn durch bedingte oder unbedingte Sprünge Befehle in den vorderen Pipelinestufen annulliert werden müssen

Bsp. Kontrollfluss hazards

1	Sub	R1	R2	R2
2	BEQ	R3	R4	59
3	SLL	R3	R4	R3
4	SLA	R1	R4	R7
⋮				
⋮				
⋮				
59	Mult	R4	R5	R6



Statische Sprungvorhersage

Predict Not Taken

- Spekulative Annahme das Sprung nicht ausgeführt wird

Predict Taken

- Spekulative Annahme das Sprung ausgeführt wird
- Sprungziel muss möglichst früh bekannt sein

Rücksprungerkennung

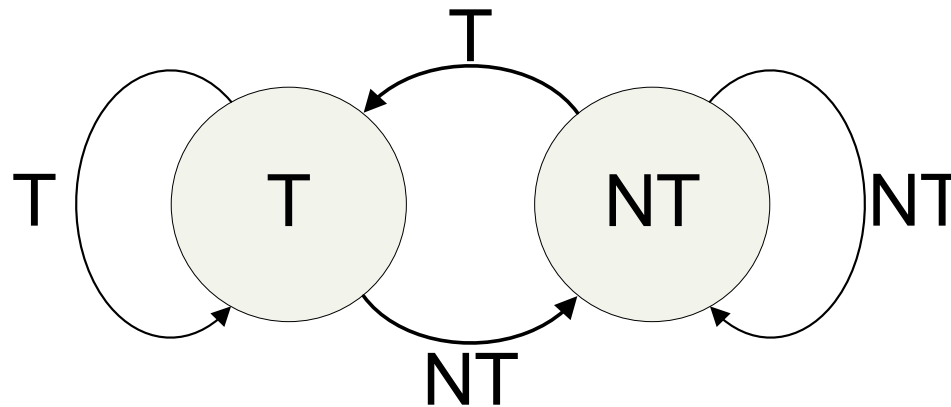
- Sprung als Verzweigend angenommen, wenn Sprungziel kleiner aktuellen PC

Delayed Branches

- Auffüllen der Zeit bis bekannt ist, ob der Sprung ausgeführt wird mit unabhängigen Befehlen

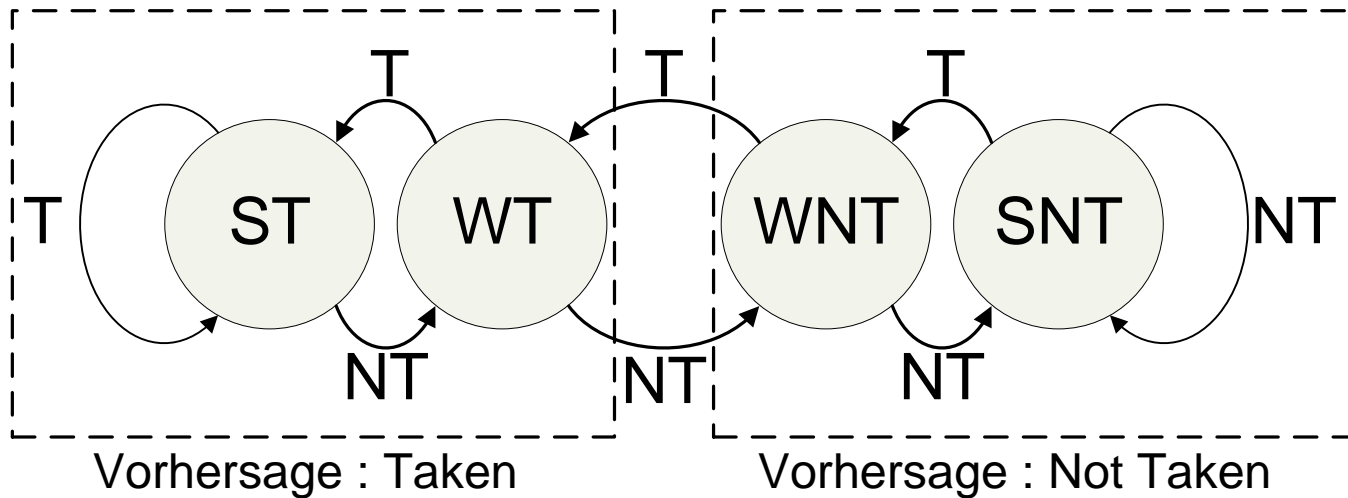
Dynamische Sprungvorhersage

1-Bit-Prädiktor



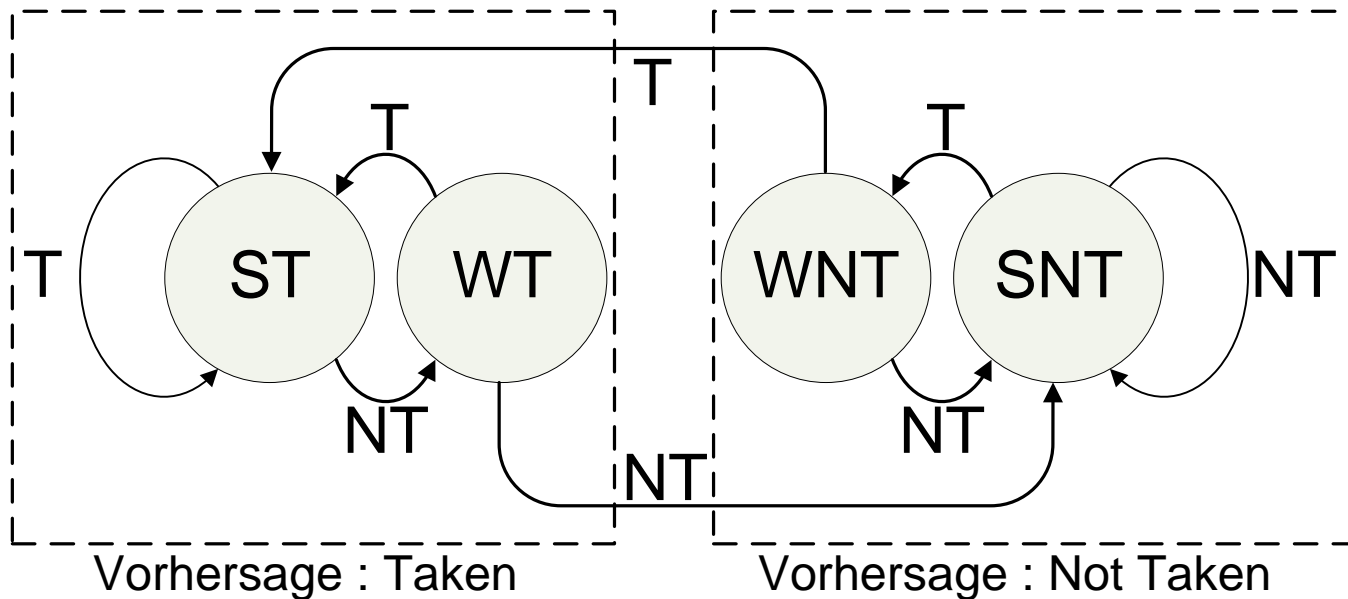
- 2 Zustände : taken (T) , not taken (NT)
- Zustand legt Sprungvorhersage fest
- Problem : zu kurze Historie
 - 100% Fehlvorhersage z.B. bei alternierenden Sprüngen

2-Bit-Prädiktor



- 4 Zustände : strongly taken (ST) , weakly taken (WT), weakly not taken (WNT), strongly not taken (SNT)
- Optimiert auf Schleifenaustritts- und Wiederanlaufverhalten

2-Bit-Prädiktor



- 4 Zustände : strongly taken (ST) , weakly taken (WT), weakly not taken (WNT), strongly not taken (SNT)
- Wechsel immer erst nach zwei Fehlvorhersagen

Zusammenfassung Prädiktoren

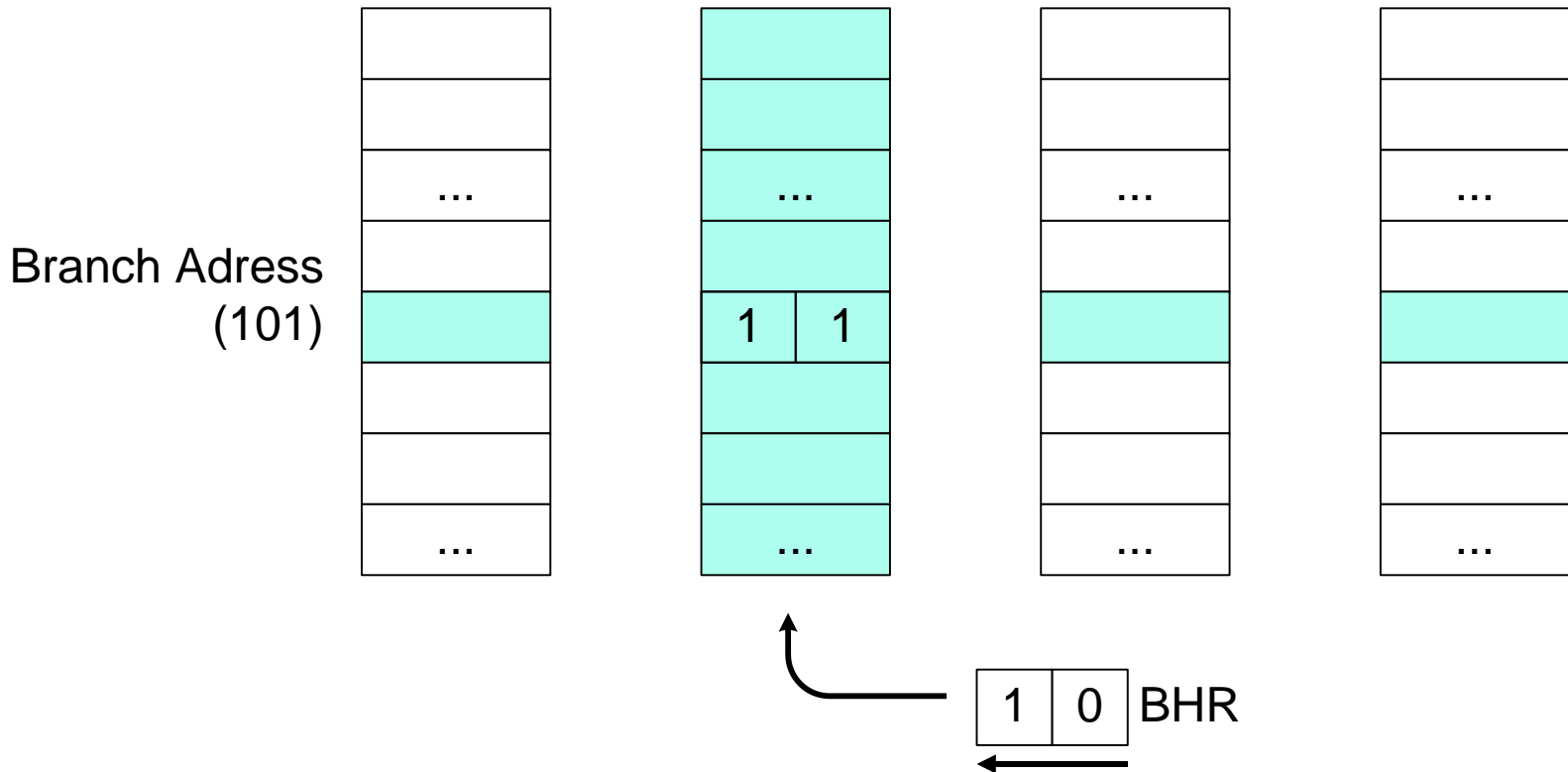
- Vorhersagegenauigkeit hängt stark von der Anwendung ab
 - hohe Trefferquote bei numerischen Problemen mit vielen Schleifen und wenigen if - then - else - Konstrukten
 - geringere Trefferquote bei allgemeinen Programm mit vielen if - then - else – Konstrukten
- weitere Vergrößerung der Zustandsmenge ohne praktischen Nutzen
 - Ursache: if - then – else - Konstrukte

(m,n) - Korrelationsprädiktoren

- Auswertung der Sprunghistorie der vorangegangenen m Sprünge zusätzlichen zu der Verzweigung, für welche die Vorhersage zu treffen ist
- Zusätzliche Betrachtung der Beziehungen zwischen verschiedenen Sprüngen
- Speicherung des Sprungverlaufes in einem Branch-History-Register (BHR)
- Notation : (m,n) -Prädiktor nutzt das Verhalten der letzten m Sprünge für die Auswahl aus 2^m Einträgen wobei jeder Eintrag einen n -Bit Prädiktor für einen einzelnen Sprung darstellt.

Beispiel: (2,2) - Korrelationsprädiktor

Pattern History Tables



Zweistufige adaptive Prädiktoren

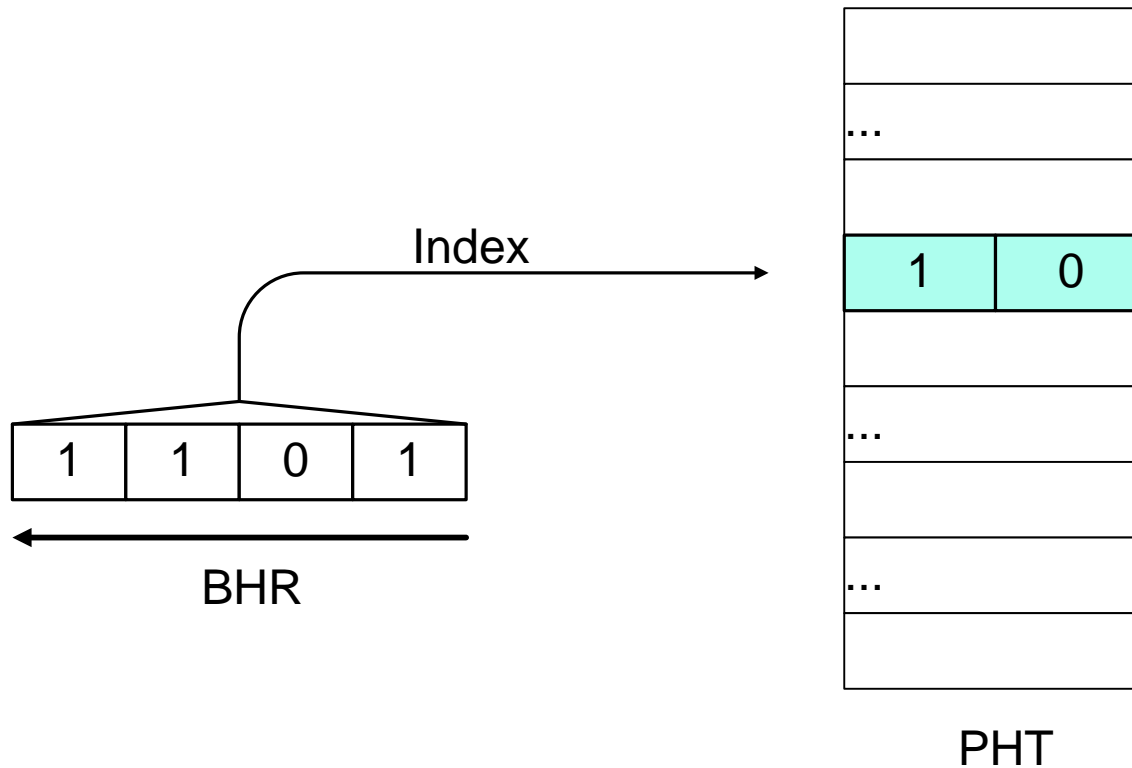
- Mechanismus besteht aus zwei Tabellenebenen: Erste Tabelle dient dazu, Vorhersagebits aus zweiter zu selektieren
- Berücksichtigung der Sprunghistorie
- Neun Varianten XAx :
 - X: Beschreibt erste Tabellenebene
 - A: Adpativ (konstant)
 - x: Organisationsform der zweiten Tabellenebene

	global PHT	per-addr. PHT	per-set PHT
global BHR	GAg	GAp	GAs
per-addr. BHT	PAg	PAp	PAs
per-set BHT	SAg	SAp	SAS

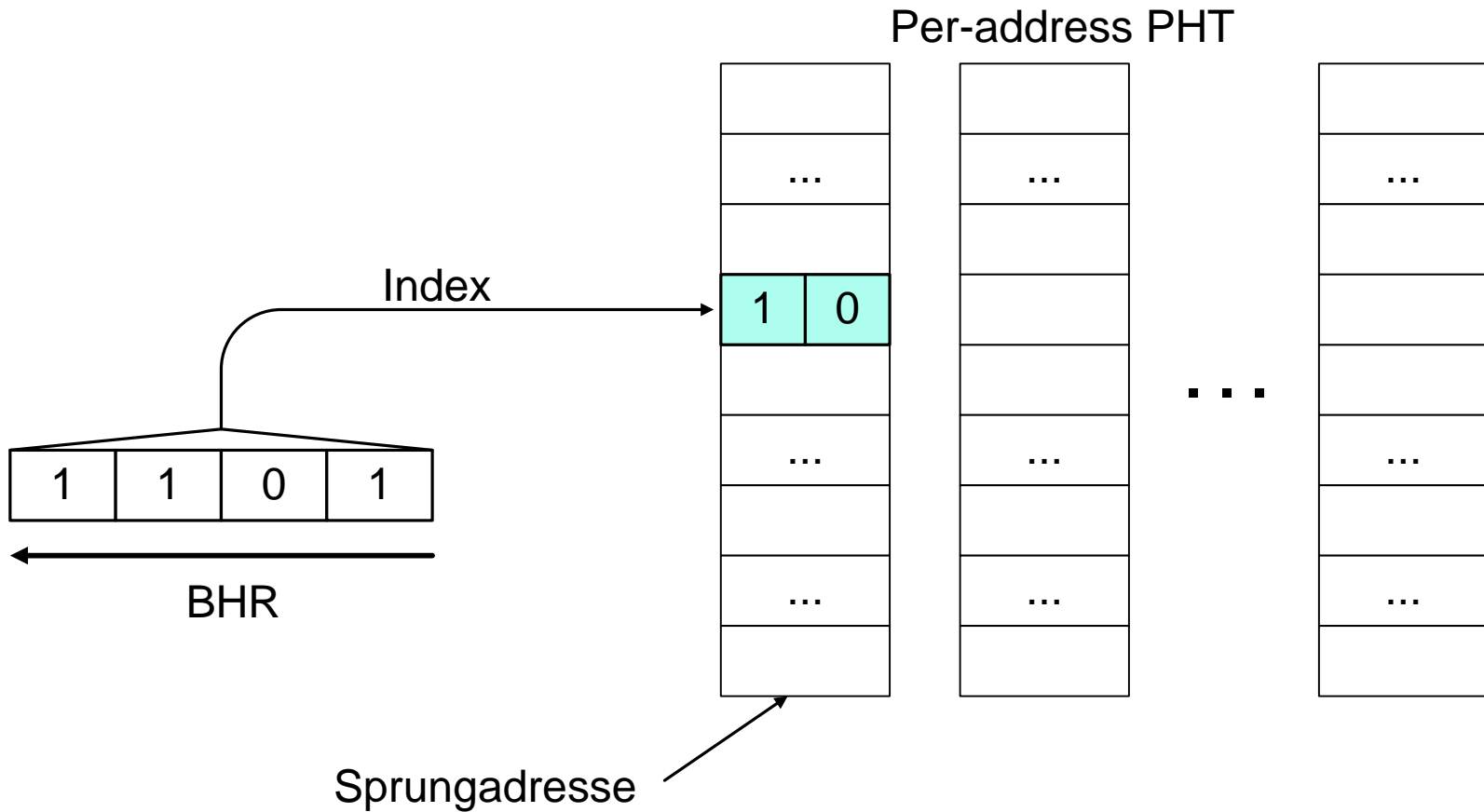
GAX

- sind Klasse der adaptiven Prädiktoren mit globalem Verlauf
- **GAg (k):**
 - 1. Tabellenebene: globales Branch-History-Register
 - 2. Tabellenebene: eine globale Pattern-Historie-Table
 - Vollständig unabhängig von Adresse des Sprungbefehls
- **GAp (k):** Zusätzliche Verwendung der Sprungadresse zur Adressierung der Pattern-Historie-Table
- **GAs(k,2ⁿ):** Verwendung von n Bits der Sprungadresse zur zusätzlichen Adressierung der Pattern-Historie-Table

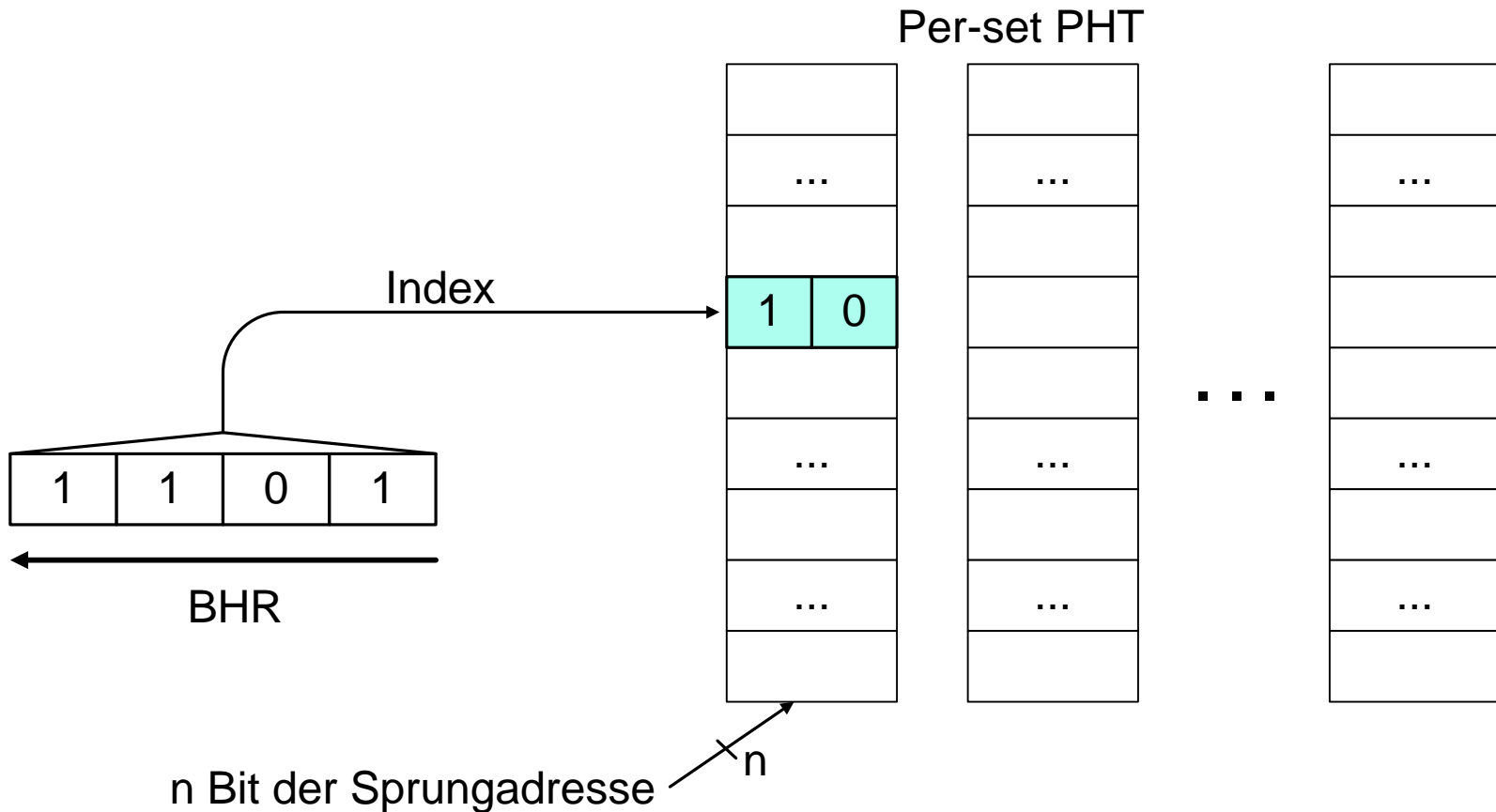
GAg (4) - Prädiktor



GAP (4) - Prädiktor



GAs (4,2ⁿ) - Prädiktor

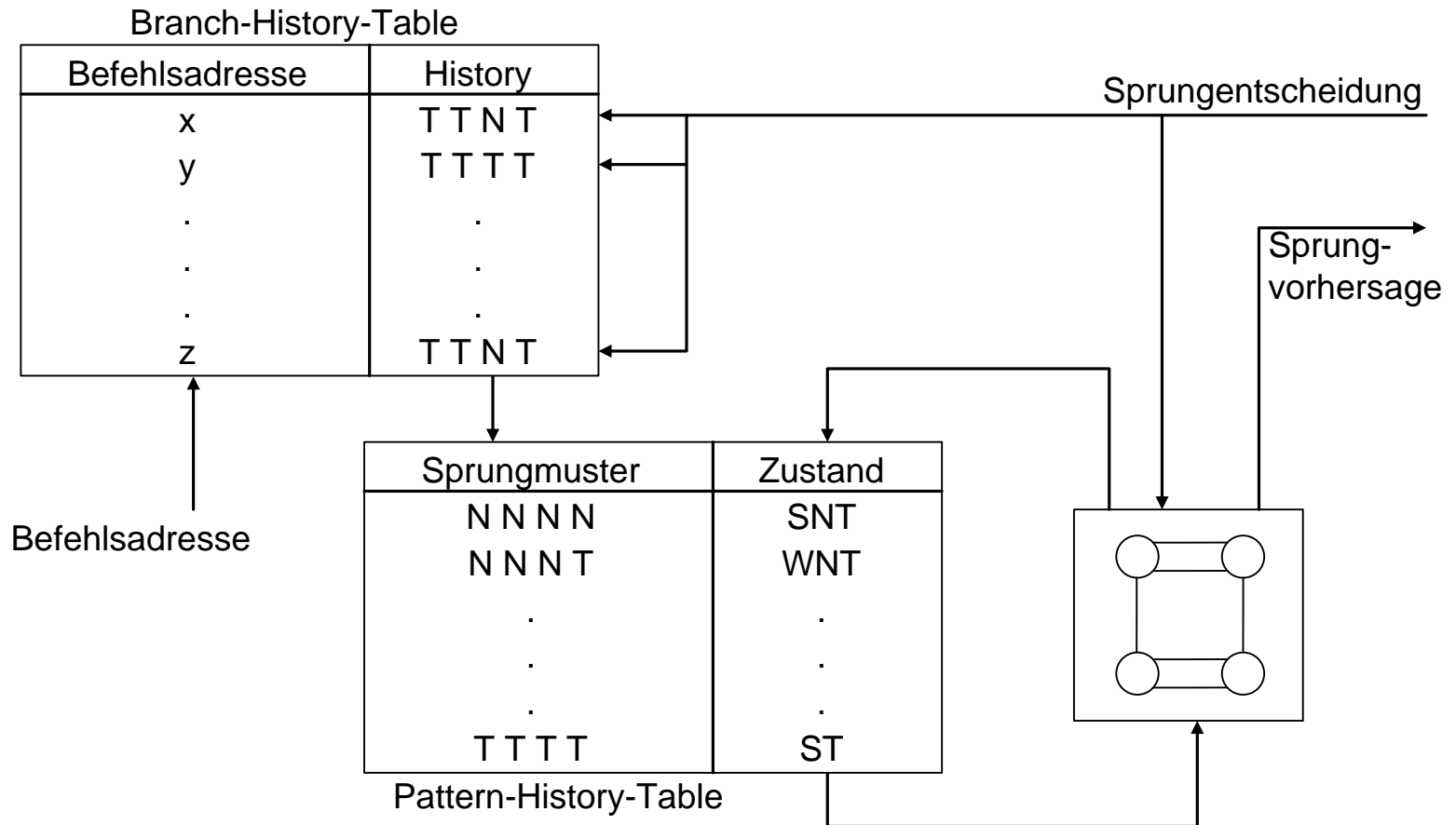


PAx / SAx

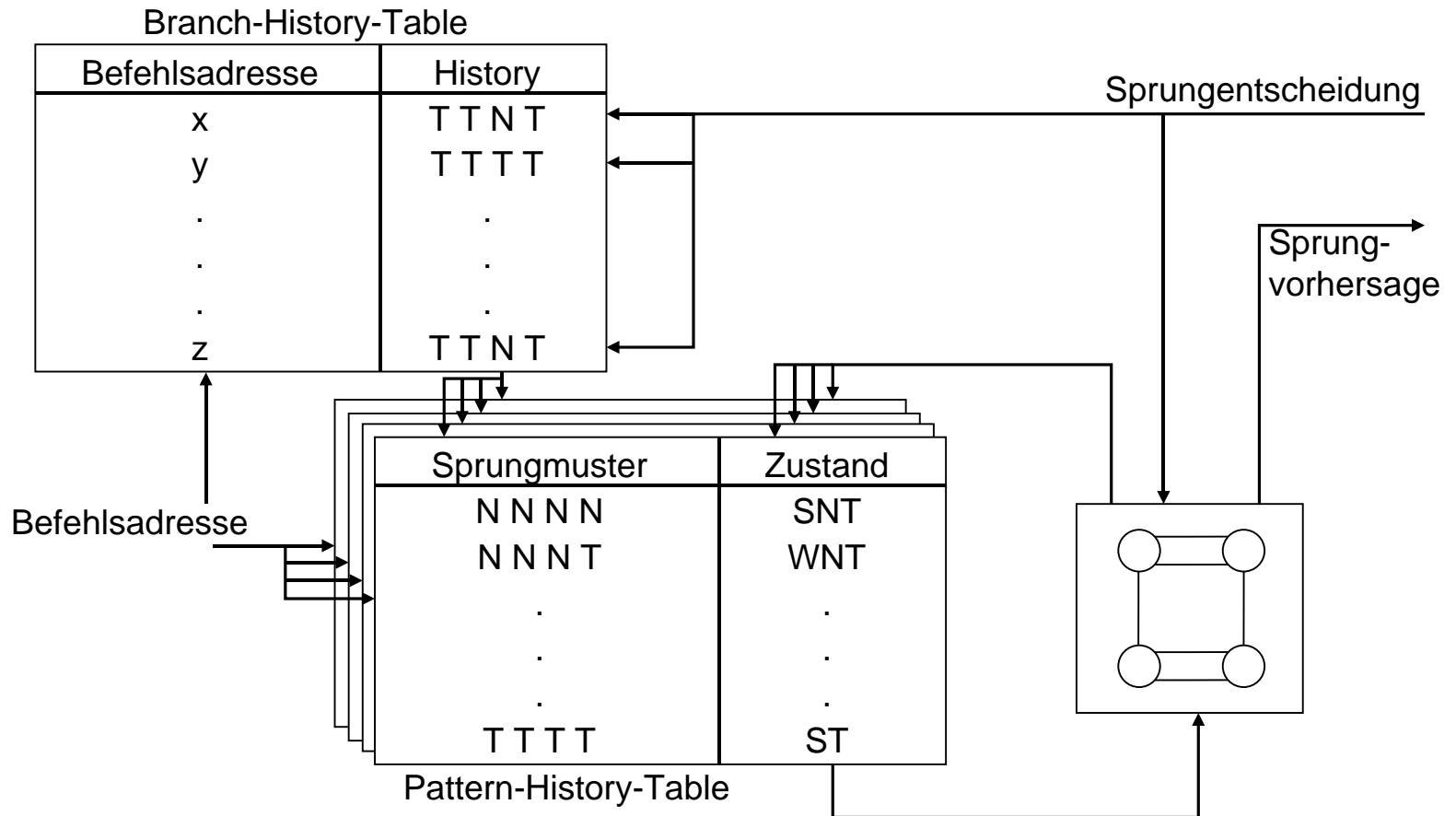
- **PAx** : Klasse der Prädiktoren mit per-address-Verlaufsschema
- Sprungadresse wählt Eintrag aus BHT aus
- Eintrag der BHT selektiert globale (PAG) oder per-[address,set]-PHT (PAP, PAs)
- Nur Historie des Sprungbefehls von Relevanz
→ keine Korrelation und Unabhängig vom Verlauf anderer Sprünge

- **SAx** : Klasse der Prädiktoren mit per-set-Verlaufsschema
- Teil der Sprungadresse adressiert BHT, ansonsten wie Pax
- Interferenzen möglichen, da nur Teil der Sprungadresse benutzt wird

PAg (4) - Prädiktor



PAp (4) - Prädiktor



gselect - und gshare - Prädiktoren

- 1993 von McFarling eingeführt
- gehören zur Gruppe der Korrelationsprädiktoren
- **gselect**
 - Adressierung der Pattern-Historie-Table durch Konkatenation des BHR und der Sprungbefehlsadresse (bzw. Teilen davon)
- **gshare**
 - Verknüpfung durch bitweise XOR anstatt durch Konkatenation
→ weniger Interferenz bei gleicher Länge

Adressteil	BHR	gselect 4/4	gshare 8/8
000 0000	0000 0001	0000 0001	0000 0001
1111 1111	0000 0000	1111 0000	1111 1111
1111 1111	1000 0000	1111 0000	0111 1111

Hybridprädiktoren

- Prädiktoren haben unterschiedliche Stärken, d.h. sie sind auf jeweils eine Klasse von Sprungbefehlen optimiert
- Kombination mehrerer unterschiedlicher Prädiktoren plus Selektor-Prädiktor → Selektor wählt zu verwendenden Prädiktor aus
- Kombination besser als der Beste der beiden Einzelprädiktoren bei gleichen Hardwarekosten
- Bsp.:
 - McFarling: PAp-Prädiktor mit gshare
 - Young & Smith : Compilerbasierte statische Vorhersage mit zweistufig adaptiven Prädiktor

Branch – Target - Buffer

- Mit bisherigen Methoden wahrscheinliche Sprungentscheidung bekannt, nicht jedoch das wahrscheinliche Sprungziel
 - Sprungziel muss erst berechnet werden
 - zusätzliche Takte Verzögerung
- Lösung: Sprungziele bei erstmaliger Sprungdurchführung speichern
- Nicht möglich für indirekte Sprünge, da Sprungziel bei jedem Sprung verschieden z.B. return-Instruction, ...
- → Lösung für return : Prozessorinterner return-Stack

Verwendung in Prozessoren

always not taken	Intel i486
always taken	Sun SuperSPARC
Rücksprung genommen	HP PA-7x00, PPC405
1-Bit Prädiktor	DEC Alpha 21064, AMD K5
2-Bit-Prädiktor	PPC604, MIPS R10000, Motorola 68060, NexGen 585
2-stufige adaptive Prädiktoren	Intel Ppro , PII , AMD K6
gshare	Intel PIII , AMD Atholn
Hybridprädiktoren	DEC Alpha 21264

Quellen

- http://www.kreissl.info/ra_07.php
- <http://www.springerlink.com/content/6we891xr5b6jeb7/fulltext.pdf>
- http://www.tecchannel.de/server/prozessoren/402311/risc_pipelines_im_detail/index10.html
- <http://capp.itec.kit.edu/teaching/mp/ss06/mp06-05.pdf>

- Mikrocontroller und Mikroprozessoren, Uwe Brinkschulte, Theo Ungerer, Springer-Verlag
- Rechnerorganisation und –entwurf, David A. Patterson, John L. Hennessy, Elsevier