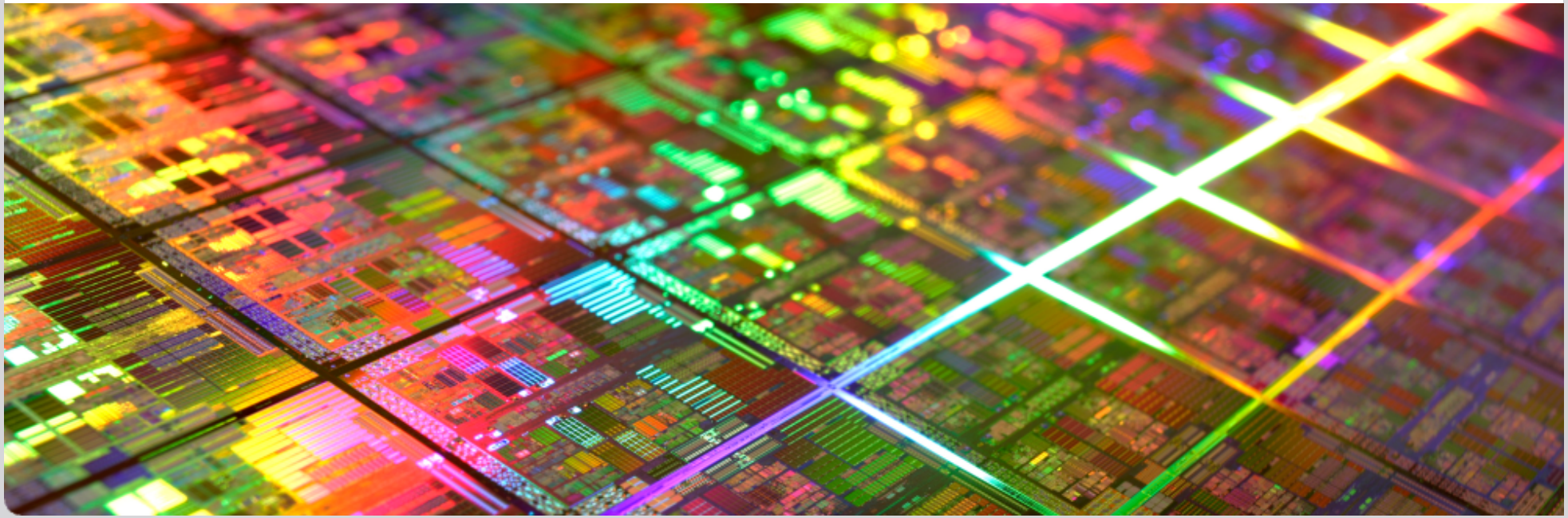
Karlsruher Institut für Technologie

Vorlesung Rechnerstrukturen im SS 2011
Sprungvorhersage

David Kramer, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

17. Mai 2011



Sprungvorhersage

Karlsruher Institut für Technologie

Bedingte Sprünge

- Problem: Ergebnis der Auswertung erst nach EX-Phase
- Pipeline-Stalling bis Auswertung
- Minimierung des Pipeline-Durchsatzes

→ Alternative: Sprungvorhersage

- Zwei grundsätzliche Arten von Sprungvorhersage
  - Statische Vorhersage (feste Sprungannahme)
  - Dynamische Sprungvorhersage
- Ziel: Umschiffung des Steuerkonflikts durch korrekte Sprungvorhersage
  - Maximierung des Pipeline-Durchsatzes
  - Minimierung von Flush/Wiederanlauf

Sprungvorhersage (forts.)

Karlsruher Institut für Technologie

Statische Sprungvorhersage

- Keine Vorhersage im eigentlichen Sinn, sondern Definition des Sprungverhaltens:
  - always not taken (i386)
  - always taken (i486)
  - Abhängig von Sprungrichtung (PPC405)
- Sprungvorhersage basiert auf Schleifenannahme
  - Rücksprung in den Schleifenkörper nehmen
  - Aussprung durch Abbruchkriterium nicht nehmen
  - Bei einfachem Modell somit nur Fehlvorhersage für Abbruchkriterium
- Für besten Erfolg Compilerunterstützung notwendig
  - Umformung der hochsprachlichen Schleifen analog zur verwendeten Sprungvorhersage
  - Ggf. Umformulierung von Fallunterscheidungen
  - Anpassung des Programms an Sprungvorhersage

Sprungvorhersage (forts.)

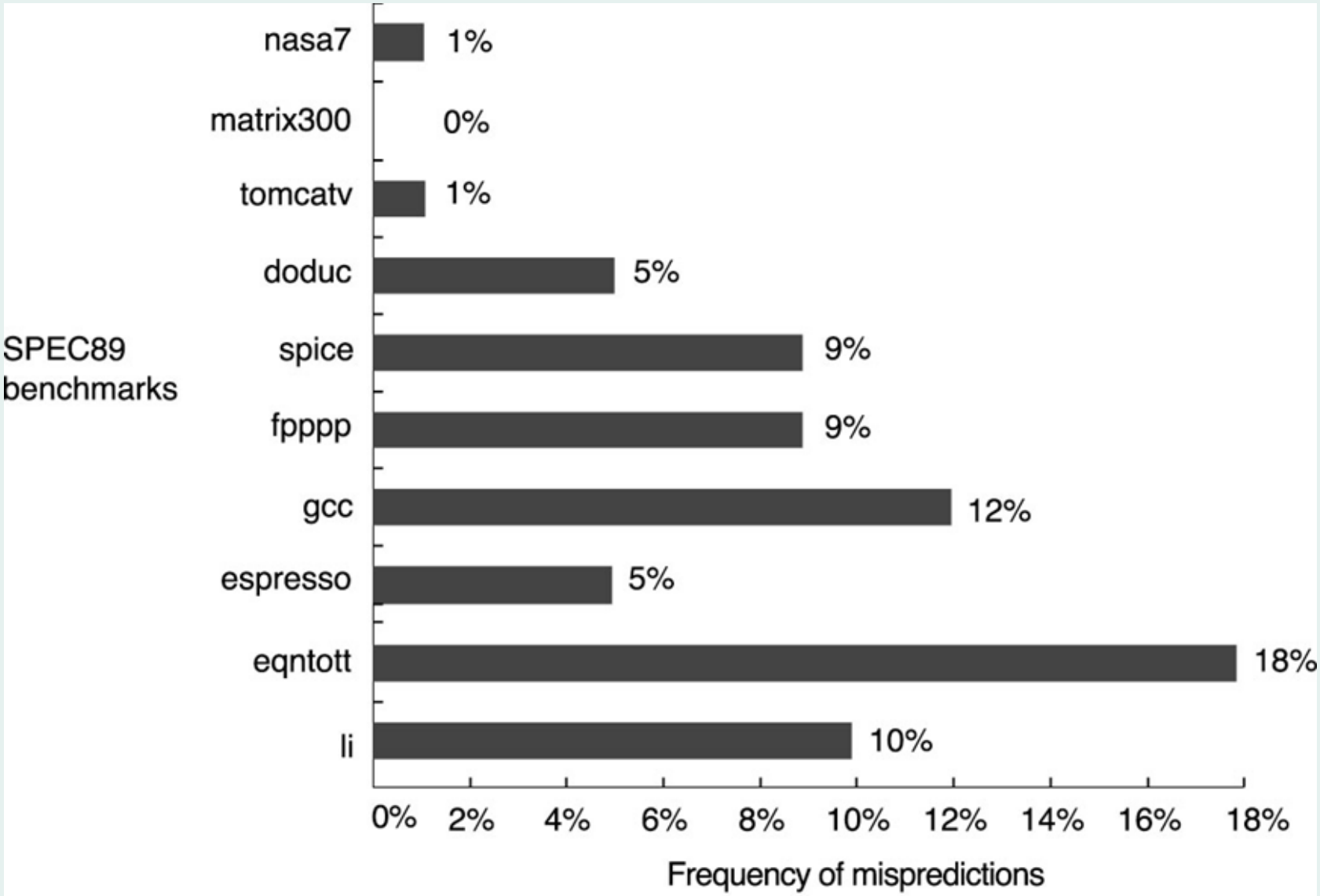
Karlsruher Institut für Technologie

Dynamische Sprungvorhersage

- Anpassung der Sprungvorhersage an laufendes Programm
- Einführung einer Sprunghistorie
- Etliche Konzepte mit variierender Komplexität
  - 1-Bit-Prädiktor
  - 2-Bit-Prädiktor (Sättigungszähler und Hysteresemodell)
  - Korrelationsprädiktoren
  - Zweistufig adaptive Prädiktoren
  - gshare, gselect
  - Hybridprädiktoren
- Vorhersagegenauigkeit in realen Programmen weit besser als statische Vorhersage
  - typischerweise $\gg$95%
- Aber: Hardwaretechnisch aufwendiger

Sprungvorhersage (forts.)

Beispiel: Vorhersagegenauigkeit

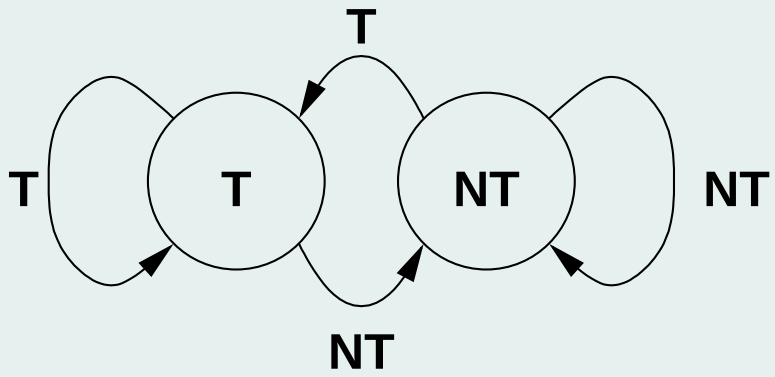


2-Bit-Prädiktor, 4096 Einträge

Quelle: Elsevier Science, 2003

Prädiktoren

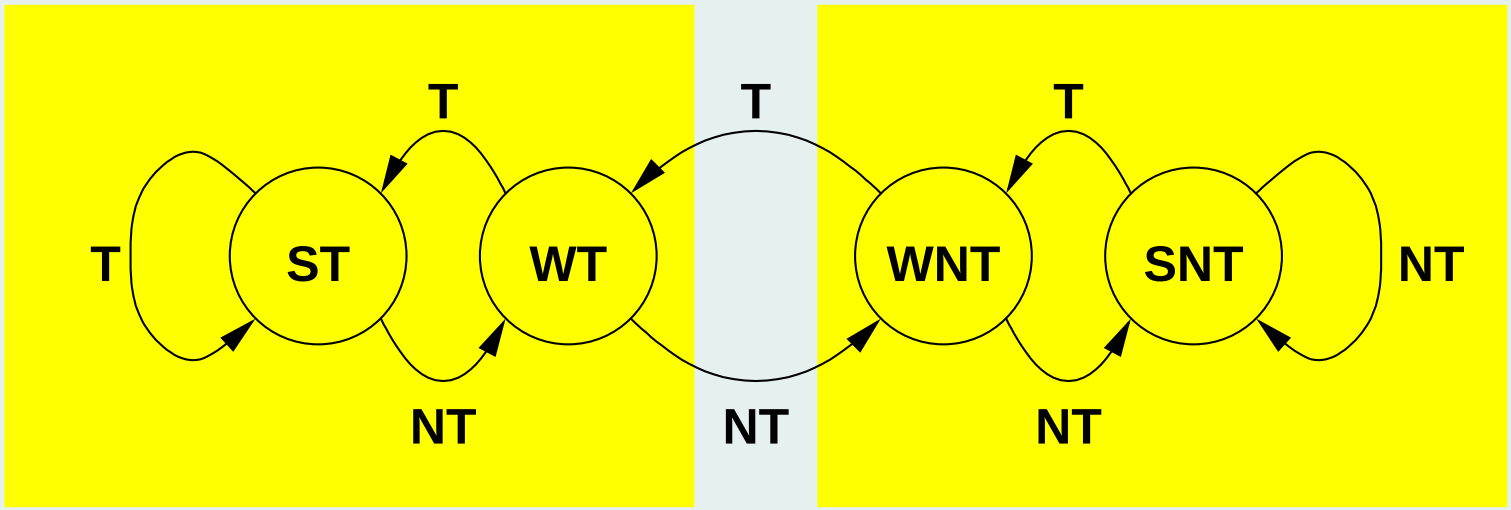
1-Bit-Prädiktor



- Zustandsautomat mit 2 Zuständen:
  taken (T), not taken (NT)
- Aktueller Zustand definiert Sprungvorhersage
- Zustandsübergang anhand tatsächlichem Sprungverlauf
- daher: Initialisierung entscheidet Anlaufverhalten
- Problem: Kurze Historie
  - Zwei Fehlvorhersagen bei verschachtelten Schleifen:
    Schleifenende und Wiedereintritt

Prädiktoren (forts.)

2-Bit-Prädiktor mit Sättigungszähler

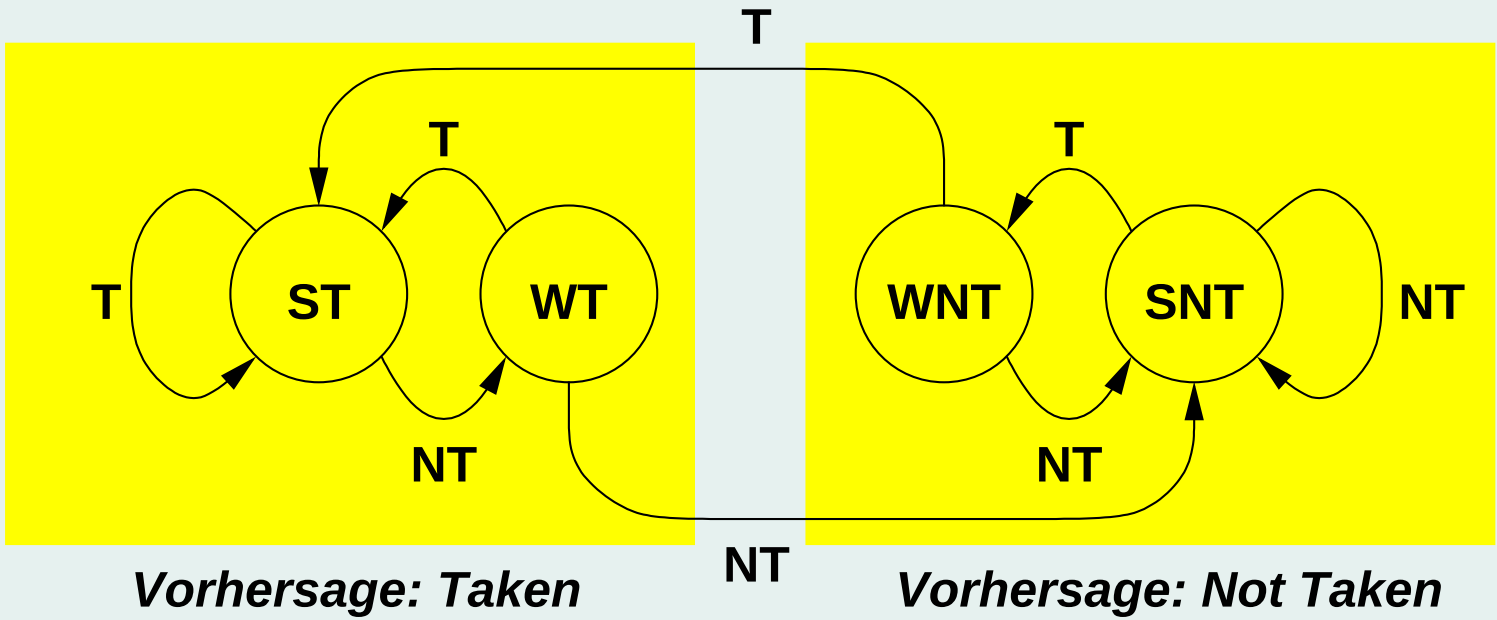


Vorhersage: Taken ***Vorhersage: Not Taken***

- Zustandsautomat mit 4 Zuständen: strongly taken (**ST**), weakly taken (**WT**), weakly not taken (**WNT**), strongly not taken (**SNT**)
- Wechsel der Vorhersage erst nach zwei Fehlvorhersagen
- Optimiert Schleifenaustritts- und Wiederanlaufverhalten

Prädiktoren (forts.)

2-Bit-Prädiktor mit Hysterese



- Zustandsautomat mit 4 Zuständen: strongly taken (**ST**), weakly taken (**WT**), weakly not taken (**WNT**), strongly not taken (**SNT**)
- Wechsel der Vorhersage erst nach zwei Fehlvorhersagen
- Aggressiveres Umschaltverhalten, vermeidet „Flattern“ zwischen Weakly-Zuständen

Prädiktoren (forts.)

n-Bit-Prädiktoren: Diskussion

- Qualität der 2-Bit-Prädiktoren variiert je nach Anwendungsgebiet
- Sehr gute Leistung bei numerischen Problemen (große Schleifen, kaum if/then/else-Konstrukte)
- Anzahl Fehlspekulationen bei allgemeinen Anwendungen höher: kurze Schleifen, Programmcode dominiert von Fallunterscheidungen
- Fehlspekulation bei SPEC89: zwischen 1% und 18% (gcc: 12%)
- Weitere Vergrößerung bringt praktisch keine Verbesserung der Sprungvorhersage
- Ursache: Korrelationen
  - if/then/else-Konstrukte

n-Bit-Prädiktoren

Problemfall: Korrelation

Code:

```
if(d==0)        /* Sprung 1 */
    d=1;
if(d==1)        /* Sprung 2 */
    ...
```

d vor S1	d==0?	S1	d vor S2	d==1?	S2
0	Ja	NT	1	Ja	NT
1	Nein	T	1	Ja	NT
2	Nein	T	2	Nein	T

- Beispiel zeigt korrelierten Sprung (S1∼S2) mit 1-Bit-Vorhersage
- Korrelationsinformation nicht von n-Bit-Prädiktoren auswertbar
- Keine Kenntnis über Sprungumgebung

Sprungvorhersagebeispiel

Karlsruher Institut für Technologie

Beispiel

Sprungverlauf eines Programms beim viermaligen Durchlaufen einer Endlosschleife.

Sprung 1 beqi R1,2	Sprung 2 beqi R1,2	Sprung 3 JMP start	Sprung 4 JMP start
NT (R1=1)	T (R1=2)	-	T
T (R1=2)	NT (R1=0)	T	-
NT (R1=1)	T (R1=2)	-	T
T (R1=2)	NT (R1=0)	T	-

Analyse: Sprünge 1 und 2 sind voneinander abhängig!

Sprungvorhersagebeispiel (forts.)

Karlsruher Institut für Technologie

Leistung des 1-Bit-Prädiktors

Es werden **6** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	**Sprung**	**P. neu**	**Prädiktion**	**Sprung**	**P. neu**
NT	**NT**	NT	T	**T**	T
NT	**T**	T	T	**NT**	NT
T	**NT**	NT	NT	**T**	T
NT	**T**	T	T	**NT**	NT

Leistung des 2-Bit-Prädiktors

Es werden **4** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	**Sprung**	**P. neu**	**Prädiktion**	**Sprung**	**P. neu**
WNT	**NT**	SNT	WT	**T**	ST
SNT	**T**	WNT	ST	**NT**	WT
WNT	**NT**	SNT	WT	**T**	ST
SNT	**T**	WNT	ST	**NT**	WT

Korrelationsprädiktoren

- n-Bit-Prädiktoren sind sprungzentrisch
- Keine Betrachtung des umgebenden Sprungverlaufs

→ Einführung von Korrelationsprädiktoren Anfang der 1990er-Jahre

Korrelationsprädiktoren

- Pan et al., 1992: (m,n)-Korrelationsprädiktoren *(correlation-based predictors, correlating predictors)*
- Yeh und Patt, 1993: Zweistufig adaptive Prädiktoren *(bi-level adaptive predictors)*
- McFarling, 1994: gselect und gshare
- McFarling, 1993: Hybridprädiktoren

(m,n)-Korrelationsprädiktoren

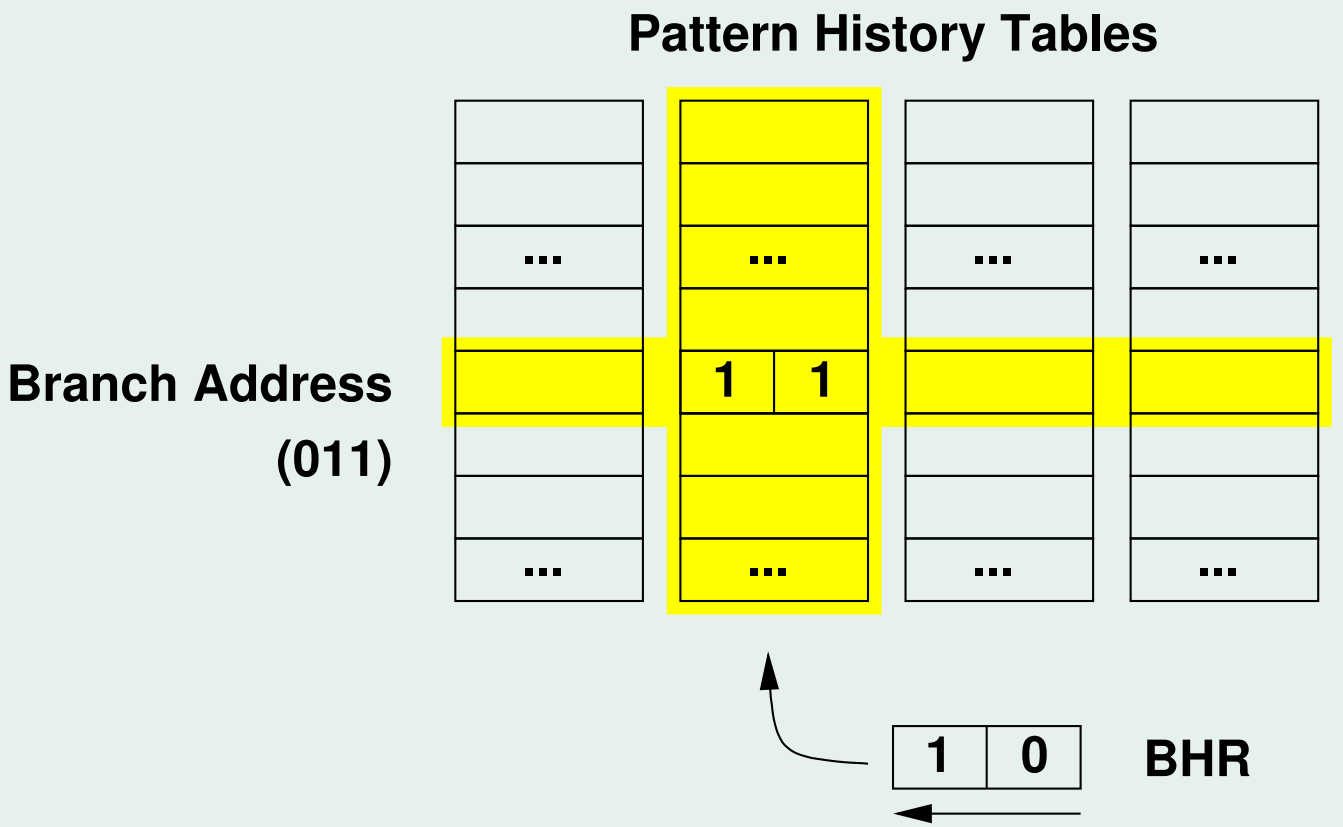
Karlsruher Institut für Technologie

(m,n)-Korrelationsprädiktoren

- Einführung einer m Sprünge umfassenden Historie
- Speicherung in Sprungverlaufsregister bzw. Branch History Register (BHR)
- BHR ist m-Bit breites Schieberegister, hält den Ausgang der letzten m Sprünge (1 - taken, 0 - not taken)
- Auswahl eines n-Bit-Prädiktors aus Sprungverlaufstabelle (Pattern History Table, PHT) anhand von Sprungadresse und BHR
- (Teil der) Sprungadresse selektiert Reihe
- BHR selektiert eine von 2^m Spalten
- Ausgewähltes Speicherfeld enthält n-Bit-Prädiktor

(m,n)-Korrelationsprädiktoren

(2,2)-Korrelationsprädiktor



Sprungvorhersagebeispiel (forts.)

Leistung des (1,1)-Korrelationsprädiktors

Es werden **keine** Fehlannahmen gemacht:

Sprung 1			Sprung 2		
Prädiktion	**Sprung**	**P. neu**	**Prädiktion**	**Sprung**	**P. neu**
(**NT**,T)	NT	(NT,T)	(**T**,NT)	T	(T,NT)
(NT,**T**)	T	(NT,T)	(T,**NT**)	NT	(T,NT)
(**NT**,T)	NT	(NT,T)	(**T**,NT)	T	(T,NT)
(NT,**T**)	T	(NT,T)	(T,**NT**)	NT	(T,NT)

Auswahl des (lokalen) Prädiktors erfolgt nach Ausgang des (globalen) letzten Sprungs (1-Bit BHR)

Alternative Techniken

Sprungvorhersagealternativen

- Prädikative Ausführung
  - Parallele Ausführung von Entscheidungspfaden
  - Kopplung der Ergebnisse an Prädikatregister
  - Bedingungstest und Verwerfen der Fehlergebnisse
  - Beispiel: Intel IA64
  - Balancierte Pfade erforderlich
  - Hardwareaufwand (Einheiten, Register)
- Bedingte Ausführung
  - Kopplung der Befehlsausführung an Bedingung
  - Umwandlung des Befehls in NOP, falls nicht erfüllt
  - Günstig in der Implementierung
  - Compilerunterstützung
  - Beispiel: ARM

Architekturüberblick

Sprungvorhersagetechnik	Beispielarchitekturen
keine Sprungvorhersage	Intel i8086, praktisch alle 8- & 16-Bit-Prozessoren sowie Mikrocontroller
Statische Verfahren	
always not taken	Intel i386
always taken	Intel i486, Sun SuperSPARC
Rücksprung genommen, Aussprung nicht	HP PA-7x00, PPC405
Dynamische Verfahren	
1-Bit-Prädiktor	DEC Alpha 21064, AMD K5
2-Bit-Prädiktor	PPC604, MIPS R10000, Cyrix 6x86, M2, NexGen 586, Motorola 68060
Bi-level Adaptive	Intel Pentium Pro, Pentium II, AMD K6
gshare	Intel Pentium III, AMD Athlon
Hybridprädiktoren	DEC Alpha 21264
Prädikation	Intel IA-64, ARM, TI TMS320C6201, weitere DSPs
Mehrpfadausführung	IBM 360/91, IBM 3090 (Großrechner)

Sprungvorhersage

Karlsruher Institut für Technologie

Zusammenfassung

- Sprungzielcache (BTAC/BTB) zur frühestmöglichen Sprung- und Sprungzielerkennung
- Statische Vorhersagemethoden
  - Vorgabe der Sprungannahme, ggf. in Abhängigkeit der Sprungrichtung (always taken, always not taken, reverse taken/forward not taken)
  - Compilerunterstützung
- Dynamische Vorhersagemethoden
  - Berücksichtigung der Sprung-Vorgeschichte (BHT)
  - Miteinbezug der globalen Sprunghistorie (BHR)
  - Hoher Hardwareaufwand, aber auch hohe Genauigkeit
- Sprungvermeidung
  - Prädikative Ausführung
  - Bedingte Ausführung

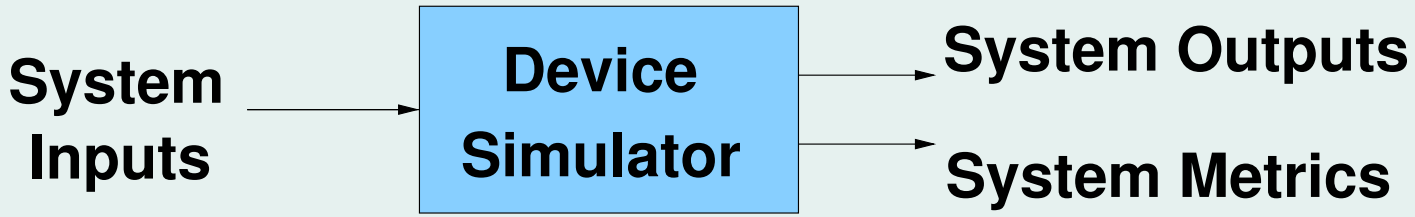
Literatur

- Brinkschulte/Ungerer:
  Mikrocontroller und Mikroprozessoren
  Springer, 2. Auflage, 2007
- Tse-Yu Yeh, Yale N. Patt:
  Two-level adaptive training branch prediction
  ACM, 1991

Prozessorarchitektursimulatoren

Was ist ein Prozessorarchitektursimulator?

- Ein Werkzeug, welches das Verhalten eines Prozessors nachahmt.



Warum werden Simulatoren verwendet?

- Simulatoren beschleunigen die Exploration eines Entwurfraumes
- Ermöglichen die parallele Entwicklung von Hard- und Software
- Im Vergleich zu HW-Prototypen: Leichte Erweiterbarkeit

Simulatoren - Taxonomie

Karlsruher Institut für Technologie

Welche Arten von HW-Simulatoren gibt es?

- Funktionale Simulatoren
  - Kein Zeitmodell
  - Erzeugung von Statistiken
  - Trace-Driven
  - Exceution-Driven
    - Emulation
    - Direct Execution
- Leistungs-/Mikroarchitektur-Simulatoren
  - Simulation von komplexen Mikroarchitekturen
  - Cycle timer - Zyklengenau
- Full-System-Simulation
  - Simulation eines vollständigen Systems (z.B. Prozessor inkl. Betriebssystem und Peripherie)

SimpleScalar

Karlsruher Institut für Technologie

Computer Architecture Research Test Bed

- Compiler, Assembler, Linker, Libraries und Simulatoren
- Zielarchitektur: SimpleScalar PISA (RISC-Format)
- angepaßte GNU Werkzeuge: gcc, binutils
- Kommandozeilen-basiert

Simulatoren

- sim-fast / sim-safe (einfach funktional)
- sim-profile (Erstellung von Statistiken)
- sim-cache / sim-cheetah (Cache-Simulation)
- sim-bpred (Prädiktor-Simulation)
- sim-outorder (Zyklengenaue Simulation, OoOE-Engine)

SimpleScalar (forts.)

Usage

- Übersetzung:
  ```
  sslittle-na-sstrix-gcc -o output input.c
  ```
- Ausführung:
  ```
  sim-bpred [-sim opts] program [-program_opts]
  ```

Options `sim-bpred` (Auswahl)

```
-bpred <string> # Default bimod
# predictor type {nottaken|taken|bimod|2lev|comb}
-bpred:bimod <int> # 2048
# bimodal predictor config (<table size>)
-bpred:2lev <int list...> # 1 1024 8 0
# 2-level predictor config
#(<l1size> <l2size> <hist_size> <xor>)
```