Rechnerstrukturen

Vorlesung im Sommersemester 2006

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik

- Ringvorlesung Informatik im SS 2006
  - Jeweils Mo. 11:30 – 13:00 im HSaF

  - Was forschen die Karlsruher Informatiker/Innen?
    - Hat es Sie nicht schon immer interessiert, mehr über die Forschungsarbeiten der Karlsruher Informatikinstitute zu erfahren? Die Ringvorlesung Informatik bietet hier eine einmalige Chance: Karlsruher Professor/Innen stellen Highlights aus ihren Forschungsgruppen vor. Wozu all die lästigen Grundlagen, die Theorie und die Technik? Hier werden Ihnen faszinierende Forschungsfragestellungen präsentiert, die nur mit fundierten Kenntnissen der Grundlagen der Informatik zielstrebig zu lösen sind. Nehmen Sie das Informationsangebot wahr!

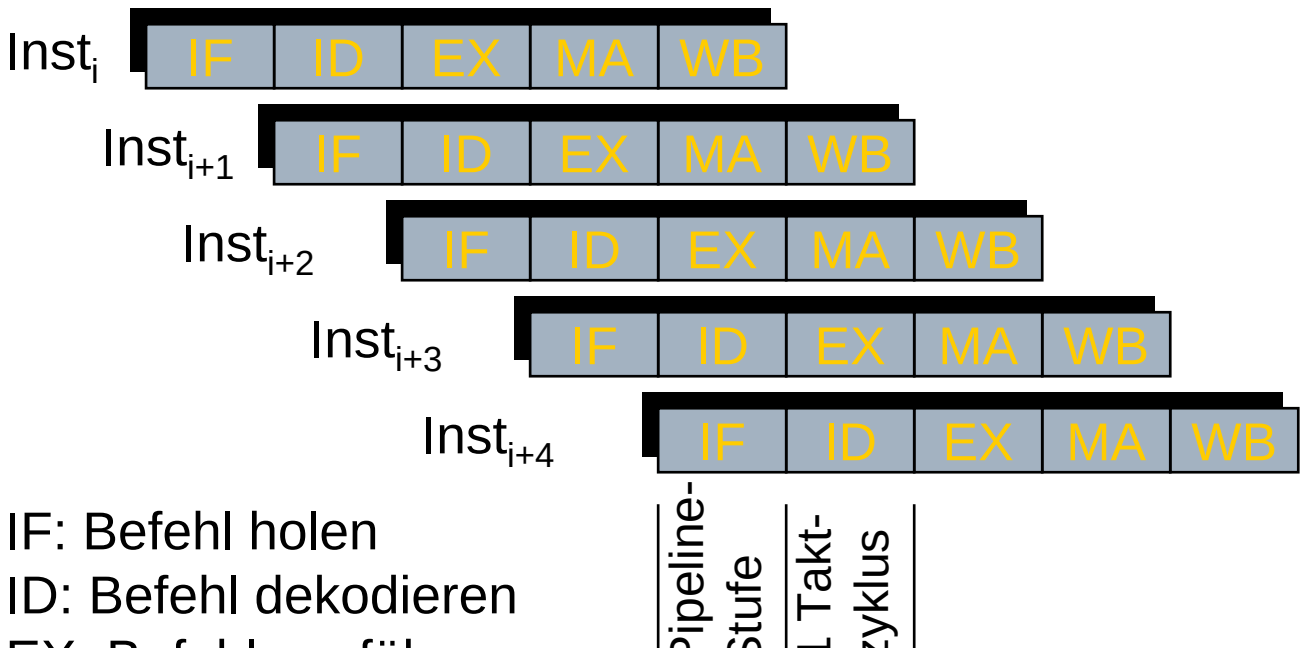
- Termine:

15.05.2006	Institut für Telematik	Abeck, Hartenstein, Juling, Zitterbart
29.05.2006	Institut für Betriebs- und Dialogsysteme	Bellosa, Prautzsch, A. Schmitt
12.06.2006,	Institut für Prozessrechentechnik, Automation und Robotik	Brinkschulte, Wörn
19.06.2006	Institut für Programmstrukturen und Datenorganisation	Reussner, Tichy
26.06.2006	Zentrum für Angewandte Rechtswissenschaft	Kühling, Sester
03.07.2006	Institut für Technische Informatik	Hanebeck, Karl
10.07.2006	Institut für Algorithmen und Kognitive Systeme	Calmet, Volllmar
17.07.2006	Fraunhofer-Institut für Informations- und Datenverarbeitung, Institut für Technische Informatik	Beyerer, Waibel
24.07.2006	Institut für Theoretische Informatik	Sanders, P. Schmitt, Wagner

- **Kapitel 2: Parallelismus auf Befehlsebene**

2.1: Pipelining

- Implementierung eines RISC-Befehlssatzes
  - k-stufige Befehlspipeline (k=5)



IF: Befehl holen
ID: Befehl dekodieren
EX: Befehl ausführen
MA: Speicherzugriff
WB: Zurückschreiben

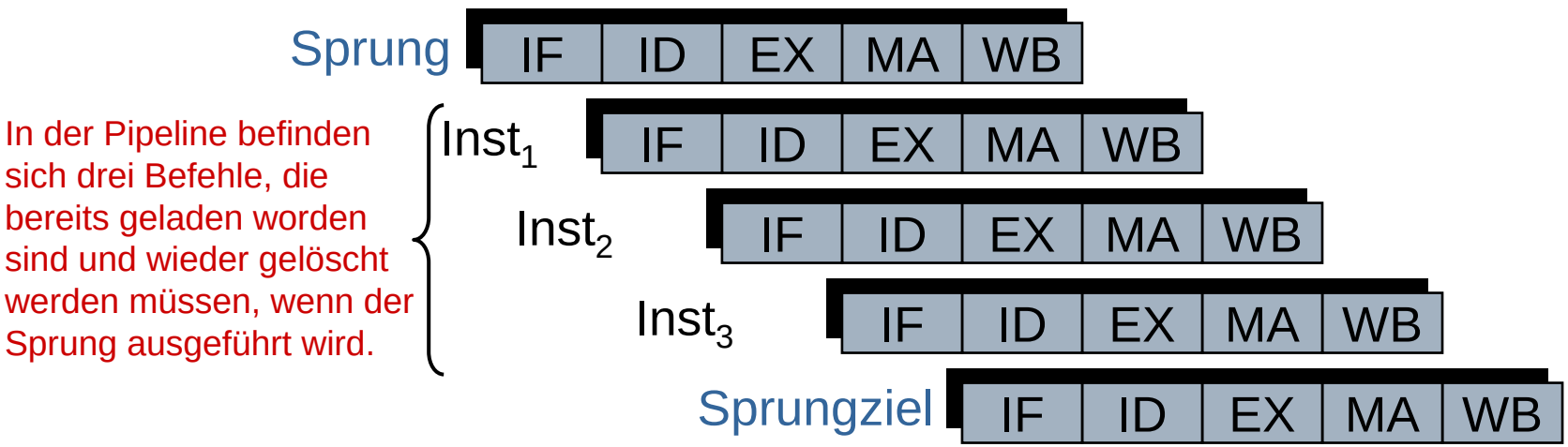
- Pipeline-Konflikte (Pipeline Hazards, Pipeline-Hemmnisse)
  - Situationen, die verhindern, dass die nächste Instruktion im Befehlsstrom im zugewiesenen Taktzyklus ausgeführt wird
    - Unterbrechung des taktsynchronen Durchlaufs durch die einzelnen Stufen der Pipeline
  - Verursachen Leistungseinbußen im Vergleich zum idealen Speedup
    - Erfordern ein Anhalten der Pipeline (Pipeline stall)
      - Bei einfacher Pipeline:
        - » Wenn eine Instruktion angehalten wird, werden auch alle Befehle, die nach dieser Instruktion zur Ausführung angestoßen wurden, angehalten
        - » Alle Befehle, die vor dieser Instruktion zur Ausführung angestoßen wurden, durchlaufen weiter die Pipeline

- Pipeline-Konflikte
  - Strukturkonflikte
    - Ergeben sich aus Ressourcenkonflikten
    - Die Hardware kann nicht alle möglichen Kombinationen von Befehlen unterstützen, die sich in der Pipeline befinden können
    - Beispiel:
      - Wenn ein Prozessor nur über einen Schreibeingang verfügt und unter bestimmten Umständen zwei Schreibvorgänge in einem Takt in der Pipeline auftreten

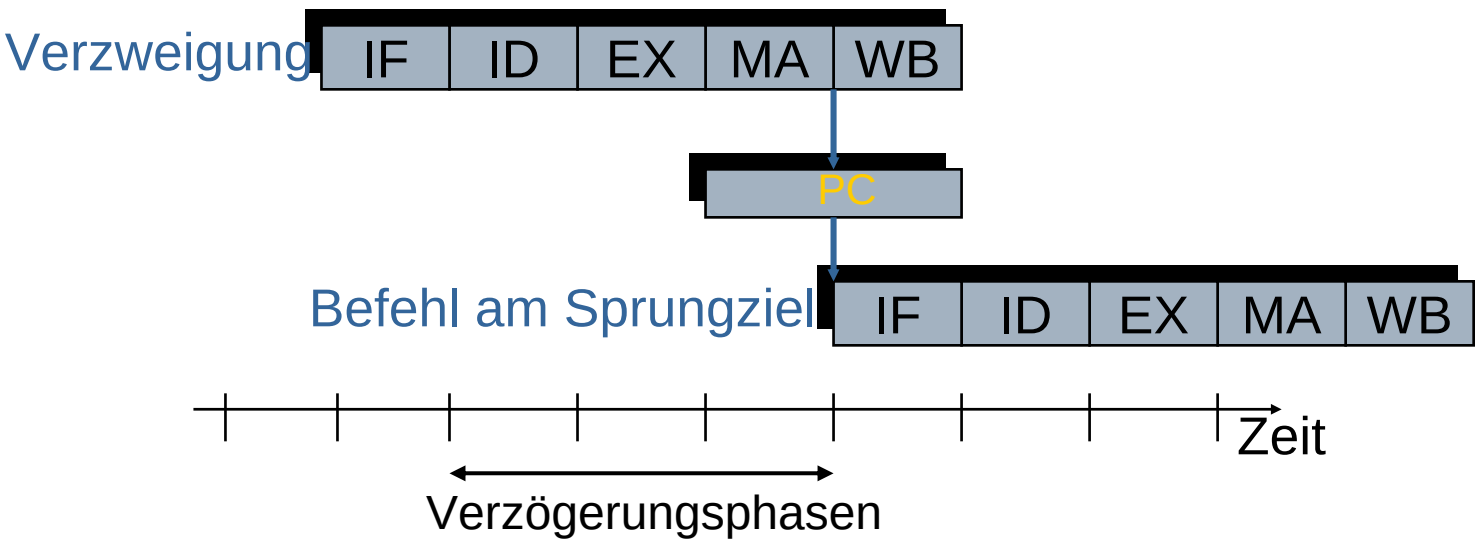
- Pipeline-Konflikte
  - Strukturkonflikte
    - Auflösen von Konflikten
      - Arbitrierung mit Interlocking
        - » Auflösung durch Hardware
        - » Anhalten eines Befehls, der den Konflikt verursacht
      - Einfügen von Leerzyklen
      - Ressourcenreplizierung
        - » Vervielfachung der Ressourcen
        - » Beispiel: mehrere Schreibeingänge für Registerdatei

- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Berechnung der Sprungadresse und des Sprungziels in der EX-Phase. Die Zieladresse ersetzt den PC in der MA-Phase.
      - Bei einer Verzweigung kann erst nach drei Takten mit der Ausführung der korrekten Befehlsfolge gestartet werden.
      - In der Pipeline befinden sich drei Befehle, die bereits geladen worden sind und wieder gelöscht werden müssen, wenn der Sprung ausgeführt wird.

- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Berechnung der Sprungadresse und des Sprungziels in der EX-Phase. Die Zieladresse ersetzt den PC in der MA-Phase.
      - Bei einer Verzweigung kann erst nach drei Takten mit der Ausführung der korrekten Befehlsfolge gestartet werden.



- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Lösung des Konflikts: Einfügen von Verzögerungsphasen

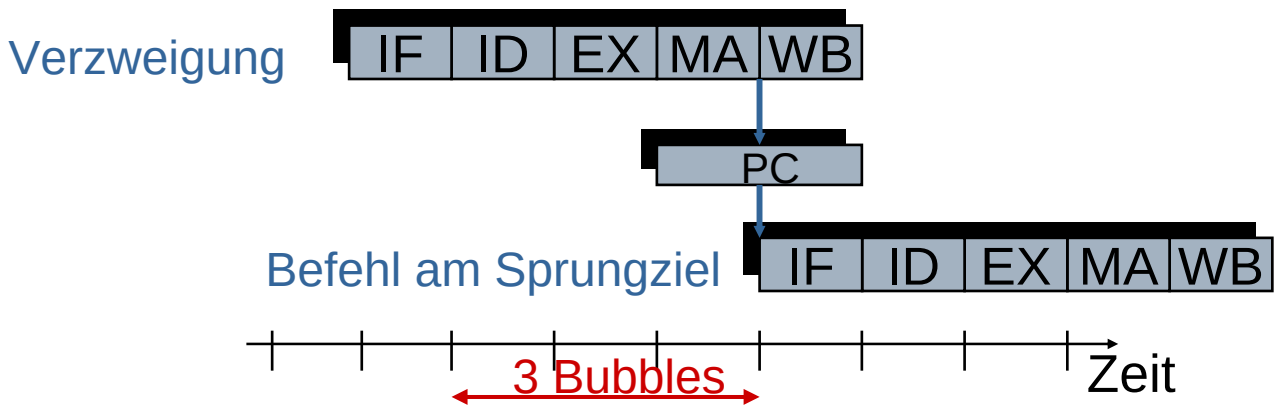


- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Lösung des Konflikts: Einfügen von Verzögerungsphasen
    - Problem: Vermeiden langer Wartezeiten
      - Möglichst frühe Auswertung der Bedingung und frühe Berechnung des Sprungziels: ID Phase ist geeignet
      - Aber
        - » Strukturkonflikt: ALU kann nicht für Berechnung des Sprungziels verwendet werden, weshalb eine zusätzliche ALU zur Sprungzielberechnung in ID-Phase notwendig ist.
        - » Datenabhängigkeit: zwischen arithmetischer Operation und nachfolgender Verzweigung; Konflikt mit Verzögerung
        - » Dekodieren, Zieladresse berechnen und PC schreiben sind in einer Pipeline-Stufe: Kritischer Pfad in der Dekodierphase, d.h. Verlängerung der Zykluszeit!

- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Lösung des Konflikts: Einfügen von Verzögerungsphasen
    - Problem: Vermeiden langer Wartezeiten
      - Mit einer Pipeline-Reorganisation wie beschrieben bleibt eine Verzögerungsphase!

      ➔Lösung: Statische und dynamische Techniken zur Konfliktauflösung!

- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Dynamische Technik zur Auflösung von Konflikten
      - Pipeline-Interlock
        - » Hardware zum Erkennen einer Verzweigung
        - » Hardware-Interlocking zum Anhalten des der Verzweigung folgenden Befehls



- Pipeline-Konflikte
  - Steuerflusskonflikte
    - Sprungvorhersage (Branch Prediction): Vorhersage des Verhaltens bei Verzweigungen
      - Beim Auftreten einer Verzweigung: Vorhersage des Sprungziels
      - Füllen der Verzögerungsphasen spekulativ mit Befehlen, die dem Sprung folgen oder die am Sprungziel stehen
      - Nach Auswertung der Sprungbedingung:
        - » Fortfahren mit der Ausführung ohne Verzögerung bei korrekter Vorhersage.
        - » Verwerfen der geholten Befehle bei falscher Vorhersage

- Pipeline-Konflikte
  - Sprungvorhersage
    - Statische Sprungvorhersage
      - Die Richtung der Vorhersage ist für einen Befehl immer gleich
    - Dynamische Sprungvorhersage
      - Die Vorhersage hängt von der Vorgeschichte ab.

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Statische Sprungvorhersage
      - Die Richtung der Vorhersage ist für einen Befehl immer gleich
      - Sprungvorhersage im Prozessor fest verdrahtet:
        - » Branch-taken: Annahme, dass die Verzweigung immer stattfindet.
        - » Branch-not-taken: Annahme, dass die Verzweigung nicht stattfindet.

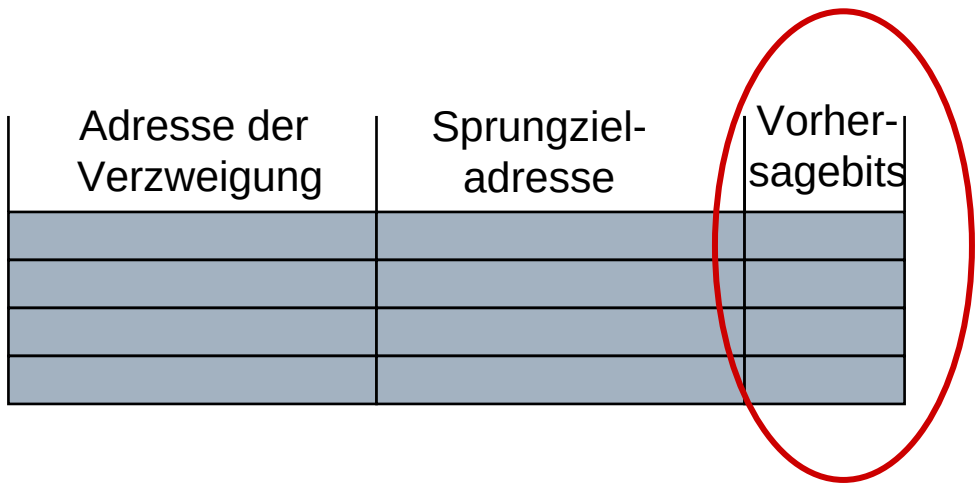
- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Statische Sprungvorhersage
      - Compiler-generierte Sprungvorhersage
        - » Kodierung der Vorhersage in einem Bit des Opcodes
        - » Holen der Befehle von der festgelegten Sprungrichtung
        - » Bei falscher Vorhersage werden die geholten Befehle wieder verworfen.
        - » Vorhersage aufgrund Programmanalyse oder Profiling

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Dynamische Vorhersage
      - Berücksichtigung des Programmverhaltens
        - » Die Richtung hängt von der Vorgeschichte der Verzweigung ab
      - Genauere Vorhersage möglich
      - Hoher Hardware-Aufwand!

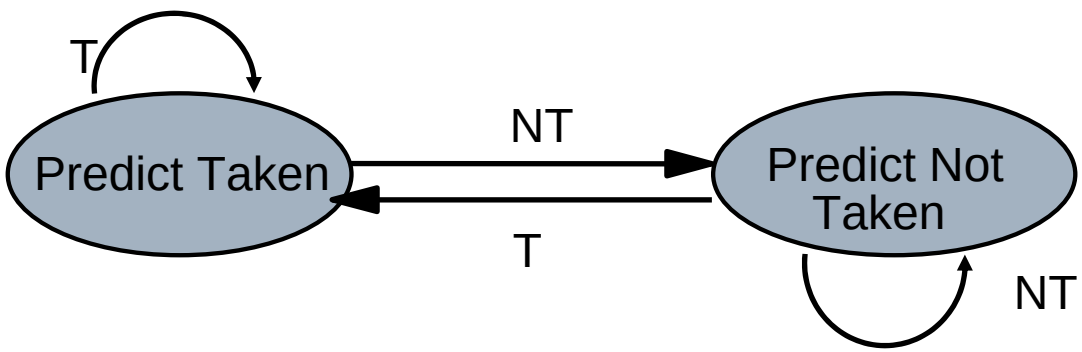
- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Dynamische Vorhersage
      - Sprungziel-Cache: Branch Target Address Cache (BTAC), Branch Target Buffer (BTB)
        - » Speichert die Adresse der Verzweigung und das entsprechende Sprungziel
        - » Steht in Verbindung mit IF-Phase

Adresse der Verzweigung	Sprungziel-adresse

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Dynamische Vorhersage
      - Sprungziel-Cache: Branch Target Address Cache (BTAC), Branch Target Buffer (BTB)
      - In Verbindung mit Branch Prediction Buffer, Branch History Table (BHT)
        - » Weiteres Feld für Vorhersagebit

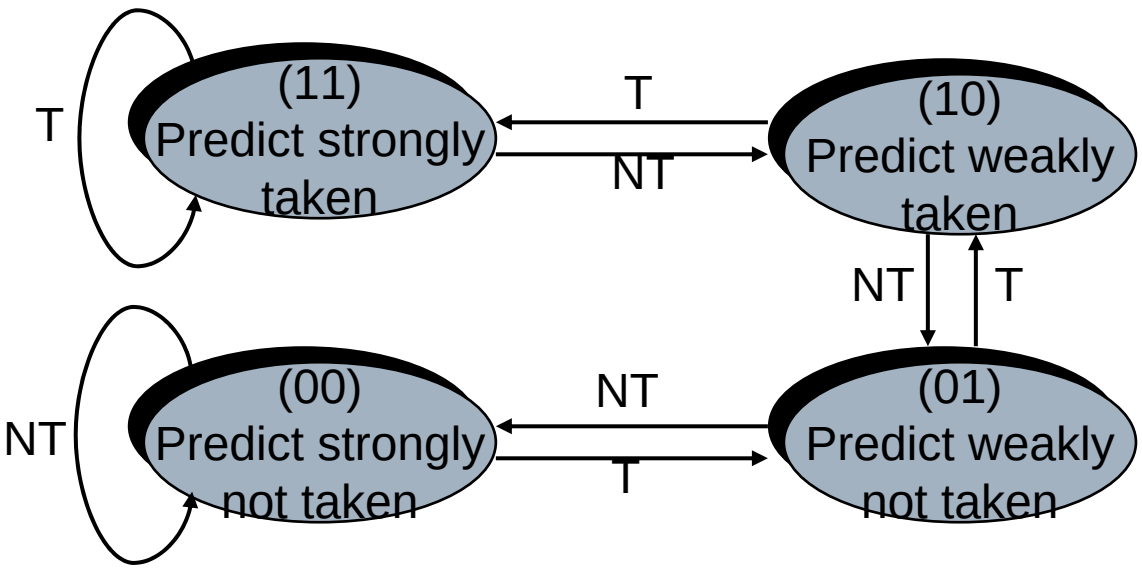


- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table
      - Vorhersagebit:
        - » Wenn das Bit gesetzt ist, wird angenommen, dass der Sprung ausgeführt wird.
        - » Wenn das Bit nicht gesetzt ist, wird angenommen, dass der Sprung nicht ausgeführt wird.
        - » Bei einer Fehlannahme: Invertieren des Bits

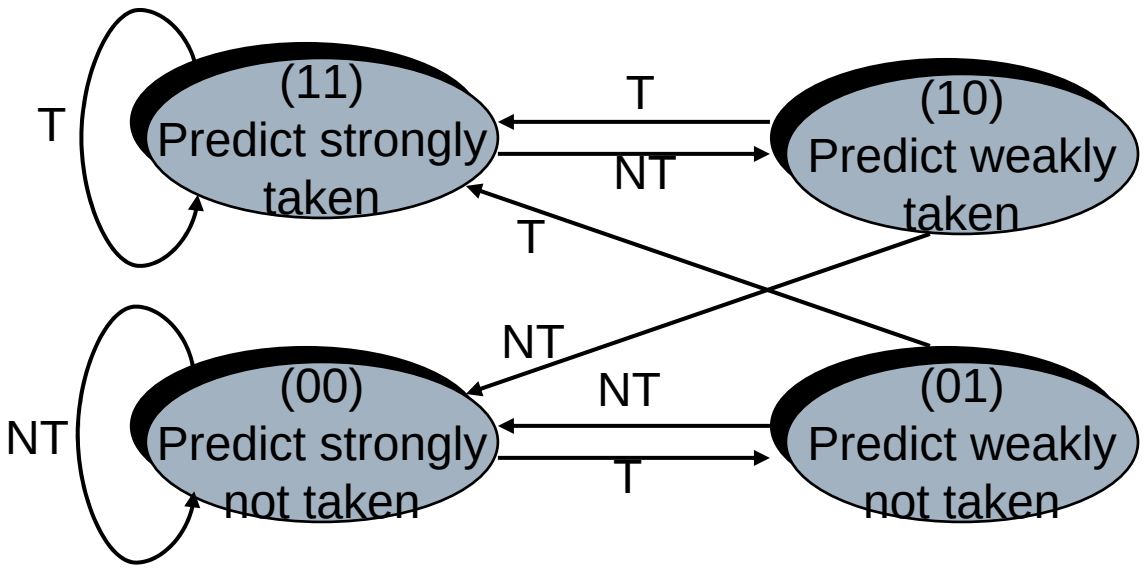


- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table: Zwei-Bit Predictor
      - Zwei Bit pro Eintrag für die Kodierung der Vorhersage → vier Zustände:
        - » *Sicher genommen (stronly taken)*
        - » *Vielleicht genommen (weakly taken)*
        - » *Vielleicht nicht genommen (weakly not taken)*
        - » *Sicher nicht genommen (stronly not taken)*
      - In einem sicheren Zustand sind zwei aufeinander folgende Fehlannahmen notwendig, um die Vorhersageannahme umzudrehen.

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table: Zwei-Bit Predictor mit Sättigungszähler (Two Bit Predictor with Saturation Scheme)



- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table: Zwei-Bit Predictor mit Hysteresemethode (Two Bit Predictor with Hysteresis Scheme)



- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table: Zwei-Bit Predictor
      - Erweiterbar auf n Bit
        - » Experimente haben gezeigt, dass kaum Verbesserungen erzielbar sind.
      - Implementierbar im Branch Target Address Cache
      - Neben BTAC Verwendung einer Branch History Table als Vorhersagetabelle

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Branch Prediction Buffer, Branch History Table: Zwei-Bit Predictor
      - Fehlannahmen:
        - » Falsche Annahme für Verzweigung
        - » Durch die Indizierung wurde die Vergangenheit eines anderen Sprungbefehls betrachtet.
      - Gute Vorhersagen in technisch-wissenschaftlichen Programmen (Schleifen).
      - Hohe Fehlannahmerate bei Programmen, in denen die Sprünge miteinander in Beziehung stehen.

      ➔Führen zu komplexen Sprungvorhersagetechniken!

- Pipeline-Konflikte
  - Sprungvorhersage (Branch Prediction)
    - Zusammenfassung
      - Statische Vorhersage
        - » Hardware- oder Compiler-Techniken
      - Dynamische Vorhersage
        - » Berücksichtigung der Vorgeschichte
        - » Hohe Genauigkeit erreichbar
        - » Hoher Hardware-Aufwand
        - » Für superskalare Prozessoren ist eine möglichst genaue Vorhersagetechnik notwendig: komplexe Techniken

- Literatur:
  - Patterson/Hennessy: Rechnerorganisation und –entwurf - Die Hardware/Software-Schnittstelle. Deutsche Ausgabe herausgegeben von Arndt Bode, Wolfgang Karl, Theo Ungerer; Spektrum Akademischer Verlag, Heidelberg, 2005:
    - Kapitel 6
  - Binkschulte/Ungerer: Microcontroller und Mikroprozessoren. Springer-Verlag, Heidelberg, 2002:
    - Kapitel 2, insbesondere Abschnitte 2.3 und 2.4

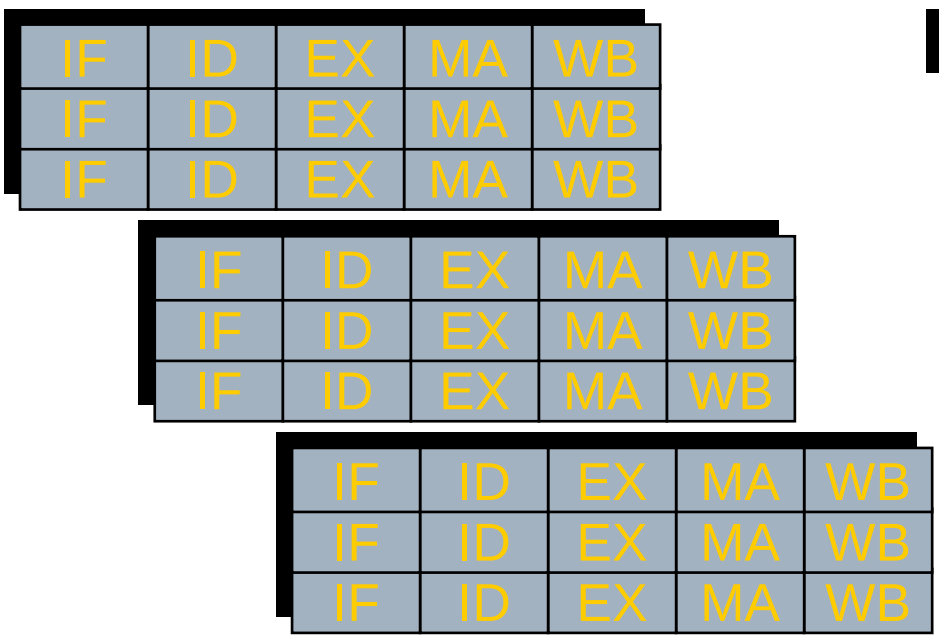
- **Kapitel 2: Parallelismus auf Befehlsebene**

2.1: Nebenläufigkeit, Superskalartechnik

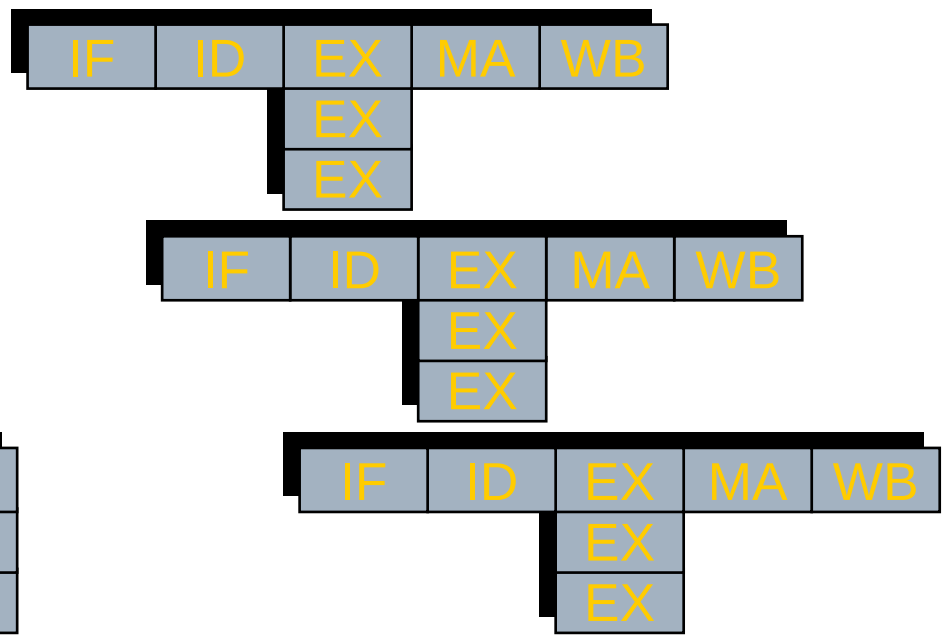
- Parallelismus auf Befehlsebene
  - Überblick
    - Nebenläufigkeit
      - Zu einem Zeitpunkt gleichzeitige Ausführung mehrerer Maschinenbefehle zu
      - Dynamische Ansätze
        - » Superskalare Mikroprozessoren
      - Statische Ansätze
        - » VLIW, EPIC

- Parallelismus auf Befehlsebene
  - Überblick
    - Nebenläufigkeit
      - Anzahl Befehle, die gleichzeitig zur Ausführung angestoßen werden (Issue parallelism IP): IP = n Befehle pro Zyklus

Dynamische Ansätze:
Superskalartechnik



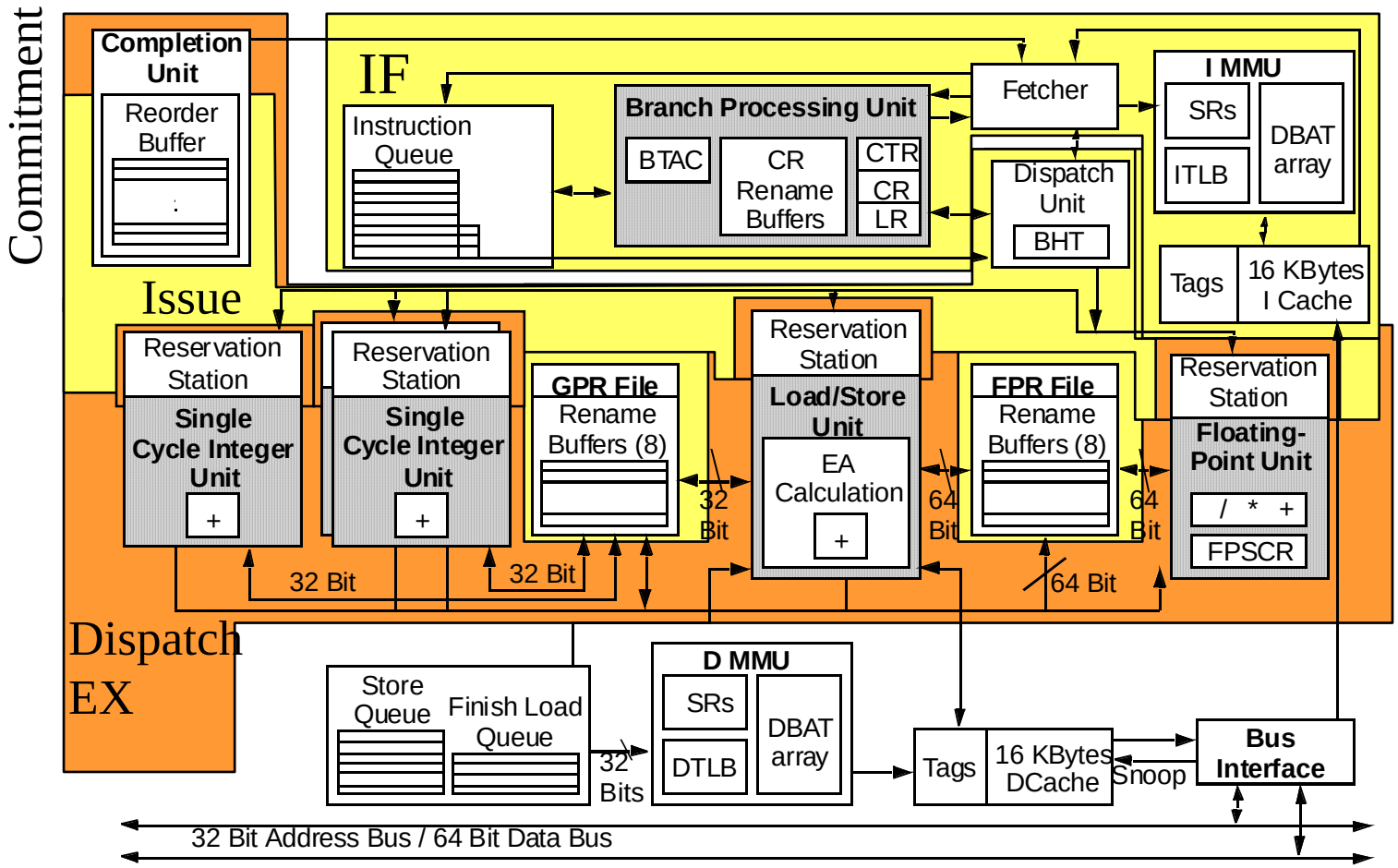
Statische Ansätze:
VLIW-Technik



- RISC → Superskalar
  - Mehrfachzuweisungsmethoden (multiple issue)
    - Die Superskalar-Technik ermöglicht es heute, pro Takt mehrere Befehle den Ausführungseinheiten zuzuordnen und eine gleiche Anzahl von Befehlsausführungen pro Takt zu beenden.
  - Superskalare RISC-Prozessoren:
    - RISC-Charakteristika werden auch heute noch weitgehend beibehalten
      - Lade-/Speicher-Architektur
      - Festes Befehlsformat (z. B.: Befehlslänge: 32 Bit)
    - Entwurfsziel: Erhöhung des IPC (Instruction per Cycle)
    - Heutige Mikroprozessoren nutzen Befehlsebenenparallelität durch die Pipelining- und Superskalartechnik

- Superskalarer Prozessor
  - Nützt den Parallelismus auf Befehlsebene aus
    - Vielstufige Befehlspipeline
    - Superskalartechnik
  - Eigenschaften:
    - Mehrere voneinander unabhängige Ausführungseinheiten
    - Zur Laufzeit werden pro Takt mehrere Befehle aus einem sequentiellen Befehlsstrom den Verarbeitungseinheiten zugeordnet und ausgeführt
    - Dynamische Erkennung und Auflösung von Konflikten zwischen Befehlen im Befehlsstrom ist Aufgabe der Hardware

- Superskalarer Prozessor: Beispiel



- Superskalarer Prozessor
  - Komponenten
    - Befehlsholeinheit (Instruction Fetch)
    - Dekodiereinheit (Instruction Decode) mit Registerumbenennung (Register renaming)
    - Zuordnungseinheit (Instruction Issue)
    - Unabhängige Verarbeitungseinheiten (Functional Units)
    - Rückordnungseinheit (Retire Unit)
    - Register:
      - Allzweckregister
      - Multimediaregister
      - Spezialregister

» *Anmerkung: Die Bezeichnungen der Einheiten sind bei den verschiedenen Prozessoren nicht einheitlich!*

- Superskalarer Prozessor
  - Komponenten
    - Umbenennungspufferregister (Rename Register)
    - Getrennte Befehls- und Daten-Cache-Speicher (I-Cache, D-Cache)
      - über Busschnittstelle mit externen Cache-Speicher oder Hauptspeicher verbunden
    - Interne Puffer
    - Befehlspuffer (Instruction Buffer)
    - Rückordnungspuffer (Reorder Buffer)
    - Speicherpuffer (Store Queue)

- Superskalarer Prozessor
  - Ausführungseinheiten:
    - Lade-/Speichereinheit
      - Zugriff auf den Daten-Cache
        - » Lesen bzw. Schreiben von Werten aus bzw. in den Daten-Cache
        - » Bei Fehlzugriff: Laden eines neuen Blocks über die Busschnittstelle
      - Speicherverwaltungseinheit (Memory-Management Unit)

- Superskalarer Prozessor
  - Ausführungseinheiten:
    - Eine oder mehrere Integer-Einheiten
      - Einstufige Einheiten
        - » Latenz und Durchsatz von 1
      - Organisiert als mehrstufige Pipeline
        - » Latenz von k (z.B. k=3) und Durchsatz von 1
      - Hängt von der Komplexität der Befehle ab

- Superskalarer Prozessor
  - Ausführungseinheiten:
    - Multimedia-Einheiten
      - Datenparallele Ausführung von Multimedia-Befehlen auf 8-Bit, 16-Bit oder 32-Bit Werten
      - Verwendung der Multimediaregister (64- und 128 Bit)
    - Gleitkomma-Einheiten
      - Ausführung von Gleitkomma-Befehlen
      - 64 Bit Gleitkomma-Arithmetik nach IEEE-754-Standard
      - Mehrstufige Pipeline (arithmetisches Pipelining), z.B. dreistufige Pipeline mit Latenz k=3 und Durchsatz von 1

- Superskalarer Prozessor
  - Verzweigungseinheit
    - Überwacht die Ausführung von Verzweigungen, Sprungbefehlen
    - Spekulatives Holen von Befehlen
      - Spekulation über weiteren Programmverlauf wird von dynamischen Sprungvorhersagetechnik entschieden
      - Verwendung der Vorgeschichte von Sprüngen

- Superskalarer Prozessor
  - Verzweigungseinheit
    - Sprungzielspeicher (Branch Target Adress Cache, BTAC)
      - enthält Adresse des Sprungbefehls und dessen Sprungziel und eventuell Vorhersagebits
    - Sprungverlaufstabelle (Branch History Table)
      - Aufzeichnen von Informationen über das Sprungverhalten bei der vorherigen Ausführung von Sprüngen
    - Rücksprungadresskeller
      - Speichert Rücksprungadressen von Unterprogrammaufrufen
    - Gewährleistet im Falle einer Fehlspekulation die Abänderung der Tabellen sowie das Rückrollen der fälschlicherweise ausgeführten Befehle