3.1 Elementare Datentypen, Operationen und ihre Realisierung (siehe 2.1)

3.2 Mikroprogrammierung

3.3 Einfache Implementierung von MIPS

3.4 Pipelining

3.5 Superskalare Befehlsausführung

Superskalarprozessoren

Erinnerung:

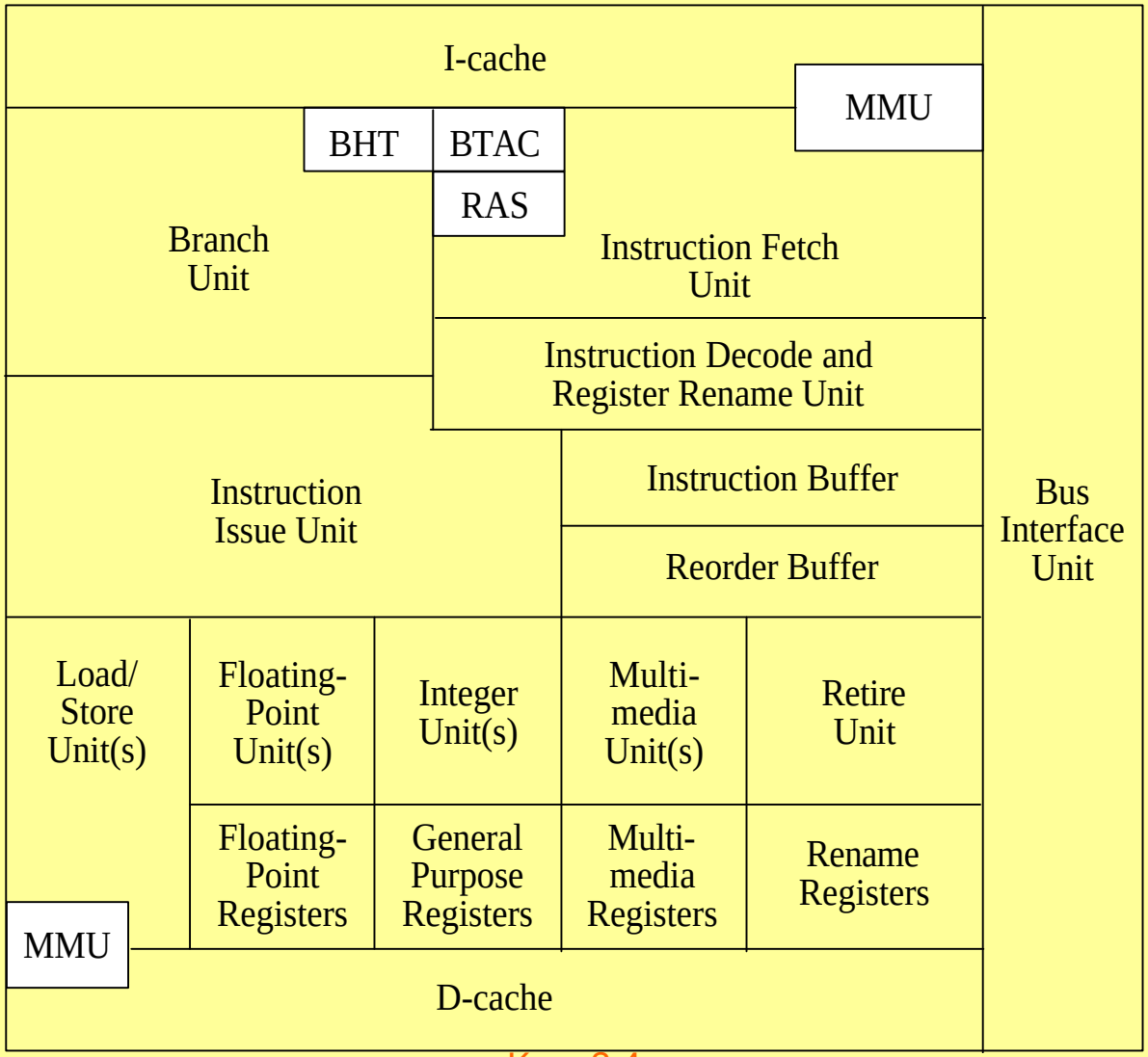
- Scoreboarding und Tomasulo sind die beiden single-issue Techniken, die out-of-order Ausführung ermöglichen
- Tomasulo ist ausgereifter, da es WAW und WAR Hazards auflöst und nicht nur behandelt (scoreboarding)

- Bisher: single-issue Prozessoren
- Jetzt: multiple-issue Prozessoren, d.h. superscalar und VLIW Proz.

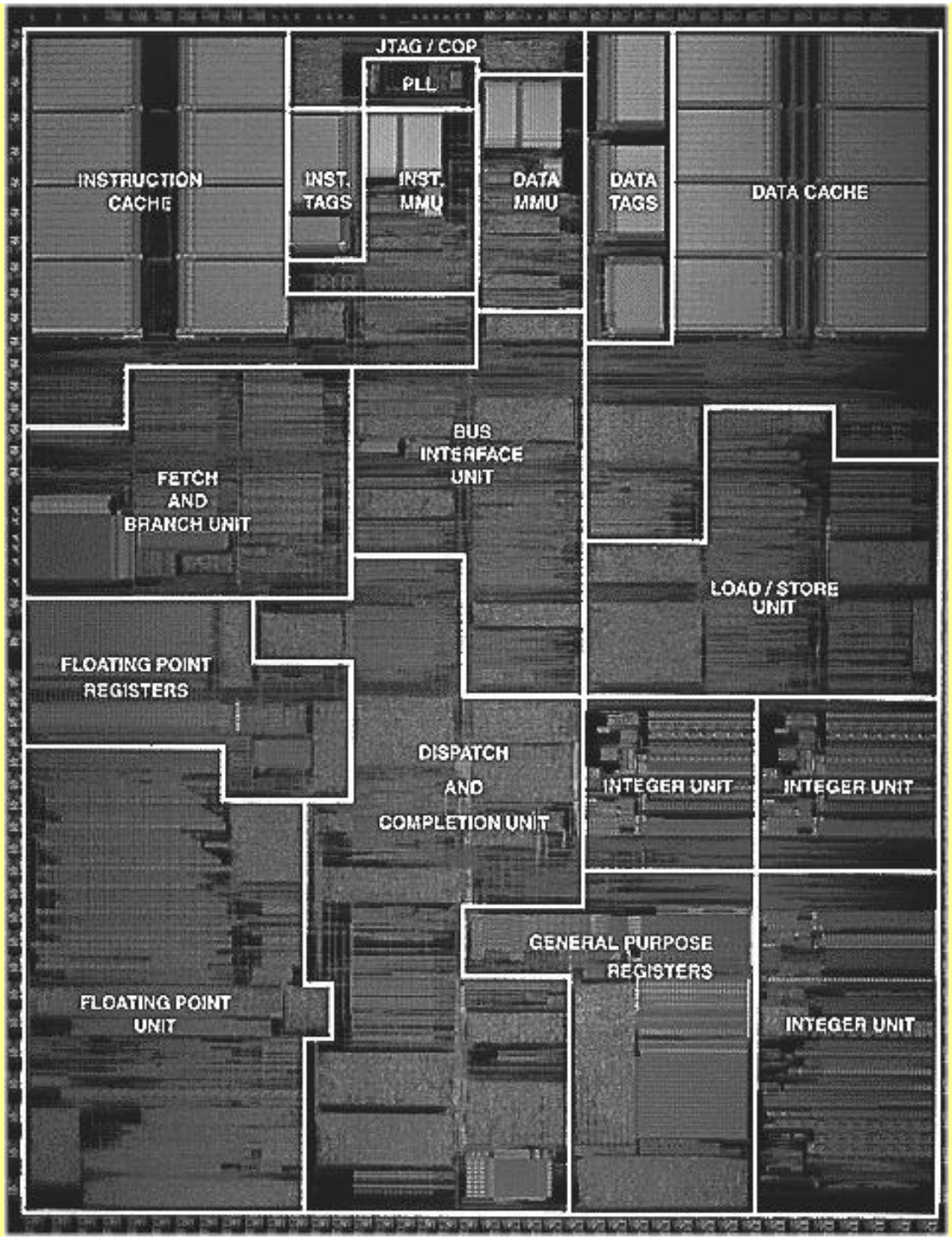
  - Die meisten heutigen Mikroprozessoren ordnen 4 bis 8 Befehle gleichzeitig an die FEs zu, mit erweitertem Tomasulo Algorithmus

  - VLIW wird häufig in Signalprozessoren eingesetzt.
  - VLIW wird von Intel als EPIC (explicitly parallel instruction computing) by Architektur (IA-64 ISA) in Itanium Prozessoren eingesetzt.

Komponenten eines Superskalarprozessors



Floorplan
des
PowerPC 604



Superskalare Befehlszuordnung

```
loop: add R1, R3, R5
       mul R3, R7, R3
       ld  R2, 299A4
       div R4, R2, R3
       sar R4
       add R4, R4, #66
      bne R4 R1 loop
```

Superskalare Befehlszuordnung

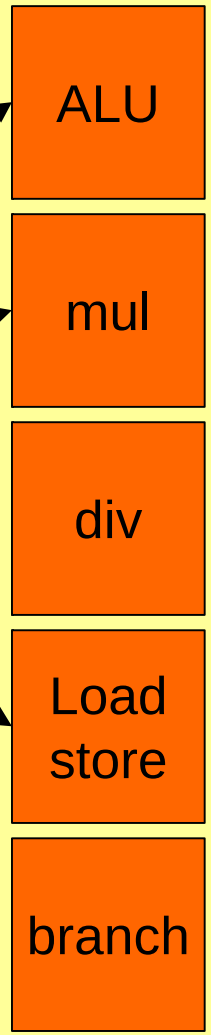
Instruktionsfenster

```
loop: add R1, R3, R5
       mul R3, R7, R3
       ld  R2, 299A4
       div R4, R2, R3
       sar R4
       add R4, R4, #66
      bne R4 R1 loop
```

Superskalare Befehlszuordnung

Instruktionsfenster

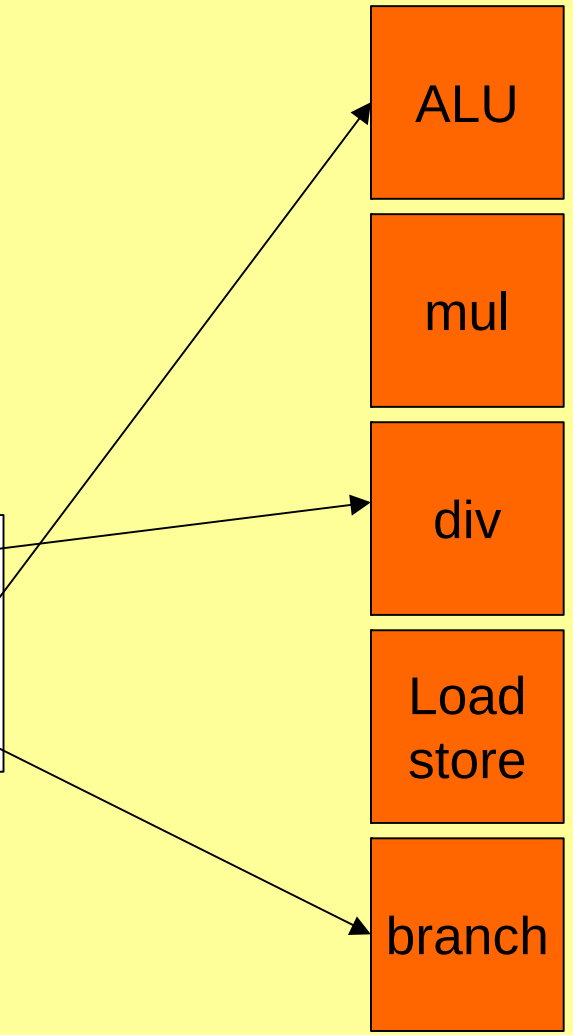
```
loop: add R1, R3, R5
      mul R3, R7, R3
      ld  R2, 299A4
      div R4, R2, R3
      sar R4
      add R4, R4, #66
     bne R4 R1 loop
```



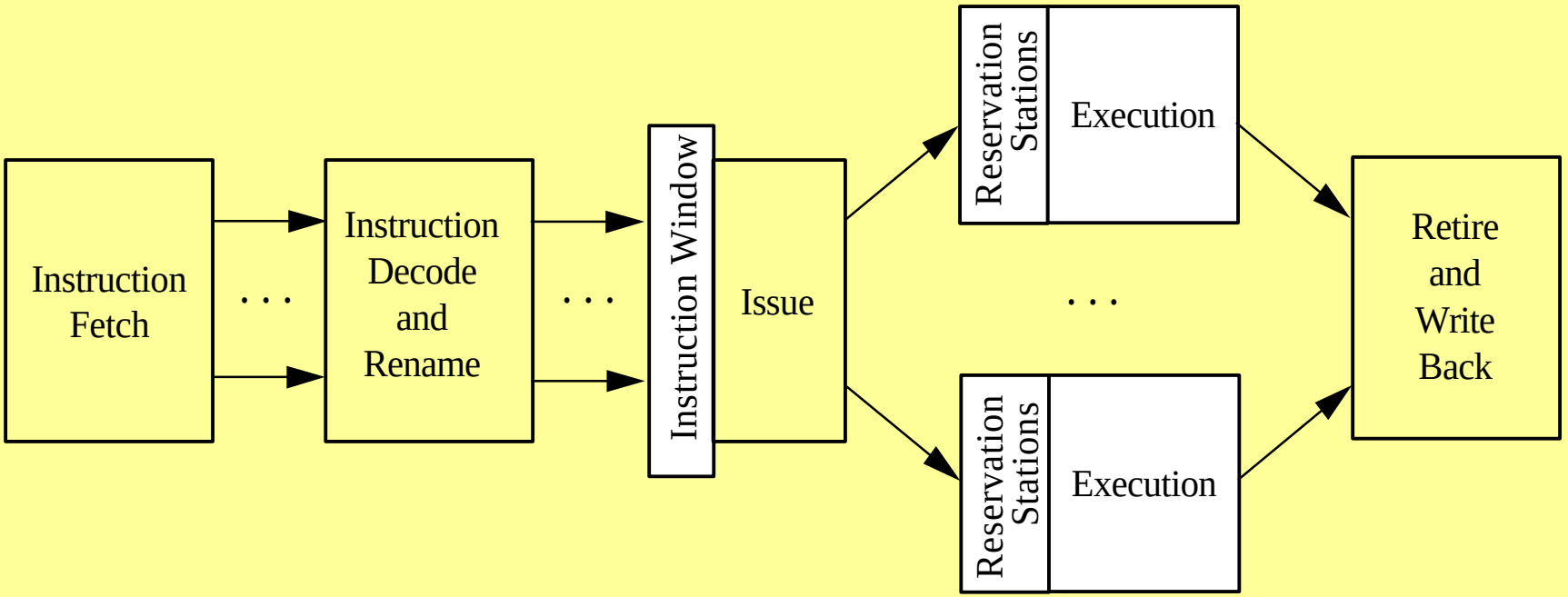
Superskalare Befehlszuordnung

Instruktionsfenster

```
loop: add R1, R3, R5
      mul R3, R7, R3
      ld  R2, 299A4
      div R4, R2, R3
      sar R4
      add R4, R4, #66
      bne R4 R1 loop
```

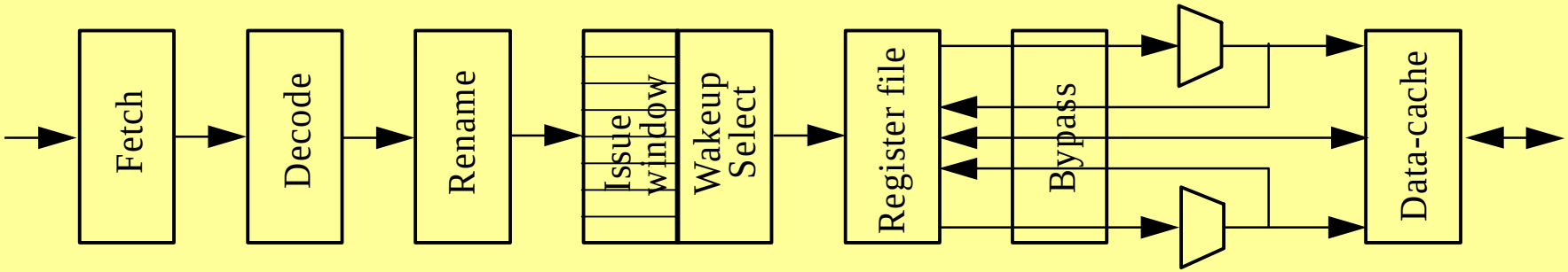


Superskalare Pipeline
(PowerPC- und erweiterter Tomasulo Algorithmus)



- Instruktionen im Instruktionsfenstersind frei von Kontrollabhängigkeiten (branch prediction), und frei von Namensabhängigkeiten (register renaming)
- Nur (echte) Datenabhängigkeiten uns Strukturkonflikte müssen behandelt werden

Superskalare Pipeline
ohne Reservation Stations



Fetch	Decode	Rename	Wakeup Select	Reg Read	Execute Bypass	Dcache Access	Reg Write Commit

Issue (Zuweisung)

- Die **Zuweisungslogik** untersucht die wartenden Befehle im Befehlsfenster und weist sie simultan den FEs zu
  (Zuordnungsbandbreite = Zahl der zugewiesenen Instruktionen pro Zyklus)
- Die Programmreihenfolge der zugewiesenen Instruktionen wird im reorder buffer gespeichert
- die Instruktionszuweisung :
  - kann **in-order** oder **out-of-order** sein
  - berücksichtigt Datenabhängigkeiten und Resourcenbeschränkungen
  - kann in mehrere Stufen eingeteilt werden
    - prüfen der Strukturkonflikte / prüfen der Datenabhängigkeiten
    - evl. Werden Datenabhängigkeiten in reservation stations (RS) vor den funktionalen Einheiten (FE) behandelt

Reservation Station(s)

- Verschiedene Definitionen in der Literatur:
  - A **reservation station** is a buffer for a **single** instruction with its operands (original Tomasulo paper, Flynn's book, Hennessy/Patterson book).
  - A **reservation station** is a buffer (in front of one or more FUs) with **one or more entries** and each entry can buffer an instruction with its operands (e.g. PowerPC literature).
- Abhängig vom Prozessor sind RS Puffer für mehrere FEs oder jede FE hat ihre eigene RS
- Befehele warten auf ihre Operanden in den RS wie in Tomasulos Algorithmus

Dispatch
(PowerPC- and enhanced Tomasulo-Scheme)

- Eine Instruktion wird „**dispatched“** von einer RS in die FE wenn alle Operanden zur Verfügung stehen und die Ausführung beginnt
- Sind alle Operanden verfügbar und die FE ist nicht aktiv, dann startet die Ausführung direkt im Zyklus nach der Zuordnung
- ..., d.h. „dispatch“ ist normalerweise keine Pipelinestufe
- Ein Befehl wartet in einer RS keinen bis mehrere Zyklen
- Dispatch und Ausführung werden nicht zwingend in der Programmreihenfolge durchgeführt

- Es gibt auch andere Definitionen von „dispatch“

Completion

- Wenn die FE die Ausführung beendet hat und das Ergebnis bereit steht zum Zurückschreiben (evl. Forwarding), dann heißt sie „**complete**“.
- Befehlsbeendigung findet nicht in der Programmreihenfolge statt.
- Während der Beendigung wird die RS freigegeben und der Zustand der Ausführung wird im **reorder buffer** abgelegt

Commitment ("Verbindlichkeit")

- Nach der Beendigung werden die Befehle in ihre Programmreihenfolge zurücksortiert (**committed**)
- Ein Befehl kann zurücksortiert werden:
  - wenn alle vorherigen Instruktionen bereits zurücksortiert wurden oder im aktuellen Zyklus zurücksortiert werden können
  - wenn kein Interrupt vor der Instruktion stattgefunden hat
  - wenn sich der Befehl nicht mehr auf einem spekulativen Pfad befindet (branch prediction)
- Durch das Zurücksortieren wird das Ergebnis endgültig in das Zielregister geschrieben,
  - üblicherweise durch das Rückschreiben des Wertes aus einem Renameregister in ein Architekturregister

Retirement ("Ruhestand")

- Ein Befehl „geht in den Ruhestand“ (**retires**) wenn der entsprechende Eintrag im reorder buffer freigegeben wird
  - weil die Instruktion comitted wird (Ergebnisse werden permanent gemacht)
  - weil die Instruktion entfernt wird (Ergebnisse werden verworfen)

Definition "Superscalar"

- **Definition**:

  Superscalar machines are distinguished by their ability to (dynamically) issue multiple instructions each clock cycle from a conventional linear instruction stream.

- Im Gegensatz zu VLIW Prozessoren; diese benutzen ein langes Befehlswort das eine feste Anzahl von Befehlen enthält, die gleichzeitig geladen, dekodiert, zugewiesen und ausgeführt werden

Zusammenfassung

Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen

Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen
- Es kann mehr als ein Befehl zugeordnet werden (deshalb superskalar im Gegensatz zu skalar)

Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen
- Es kann mehr als ein Befehl zugeordnet werden (deshalb superskalar im Gegensatz zu skalar)
- Die Zahl der Befehle, die gleichzeitig zugewiesen wird, ist von der Hardware bestimmt und definiert die maximale Zuordnungsbandbreite

Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen
- Es kann mehr als ein Befehl zugeordnet werden (deshalb superskalar im Gegensatz zu skalar)
- Die Zahl der Befehle, die gleichzeitig zugewiesen wird, ist von der Hardware bestimmt und definiert die maximale Zuordnungsbandbreite
- Die Zuornung kann in oder außerhalb der Programmreihenfolge stattfinden

Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen
- Es kann mehr als ein Befehl zugeordnet werden (deshalb superskalar im Gegensatz zu skalar)
- Die Zahl der Befehle, die gleichzeitig zugewiesen wird, ist von der Hardware bestimmt und definiert die maximale Zuordnungsbandbreite
- Die Zuornung kann in oder außerhalb der Programmreihenfolge stattfinden
- Einr Voraussetzung ist die Existenz mehrerer funktionaler Einheiten

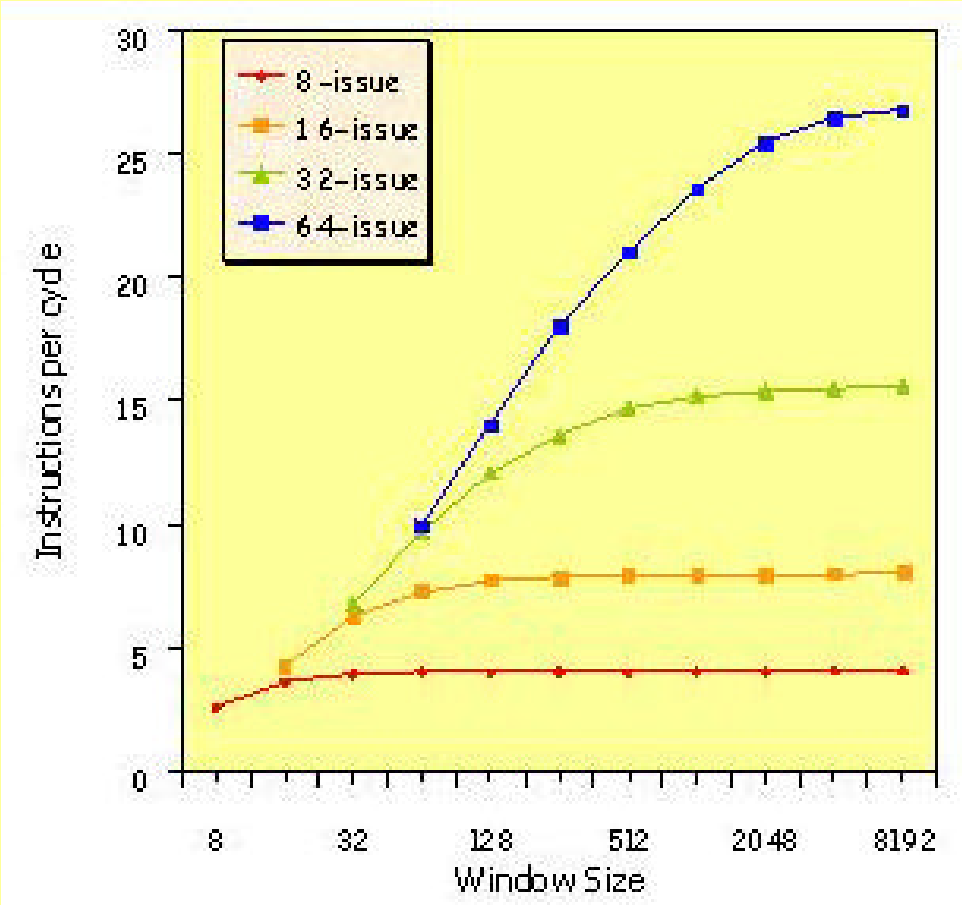
Zusammenfassung

- Befehle werden aus einem sequentiellen Befehlsstrom entnommen und dynamisch von der Hardware and FEs zugewiesen
- Es kann mehr als ein Befehl zugeordnet werden (deshalb superskalar im Gegensatz zu skalar)
- Die Zahl der Befehle, die gleichzeitig zugewiesen wird, ist von der Hardware bestimmt und definiert die maximale Zuordnungsbandbreite
- Die Zuornung kann in oder außerhalb der Programmreihenfolge stattfinden
- Einr Voraussetzung ist die Existenz mehrerer funktionaler Einheiten
- Die superskalar Technik ist eine Mikroarchitekturtechnik, kein externes Architekturmerkmal

Zuordnungsbandbreite und Fenstergröße

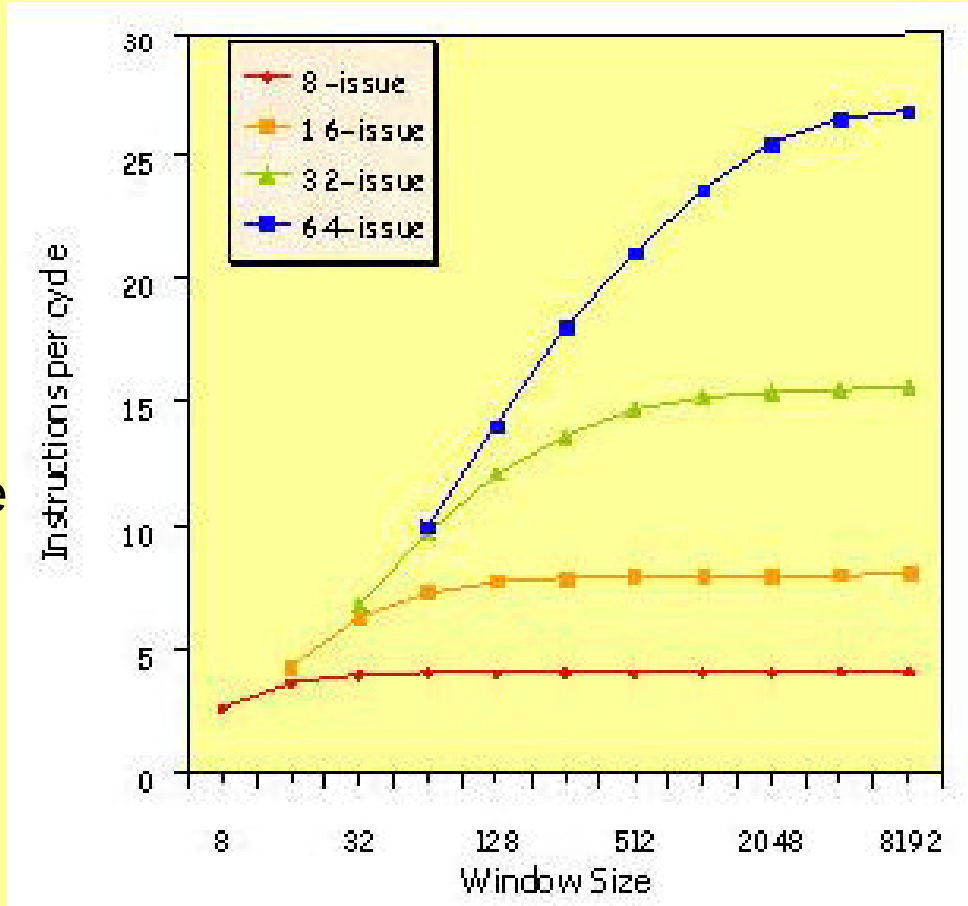
- optimale Fetch-Einheit
- perfekte Zuordnungseinheit
- perfekte Sprungvorhersage
- jede FE mit eigener Pipeline
- realistische Latenzzeiten

[Gupta et. al]



Zuordnungsbandbreite und Fenstergröße

- Wie viel Befehlsebenenparallelität steckt in Programmen (Spec95)?



- optimale Fetch-Einheit
- perfekte Zuordnungseinheit
- perfekte Sprungvorhersage
- jede FE mit eigener Pipeline
- realistische Latenzzeiten

[Gupta et. al]