

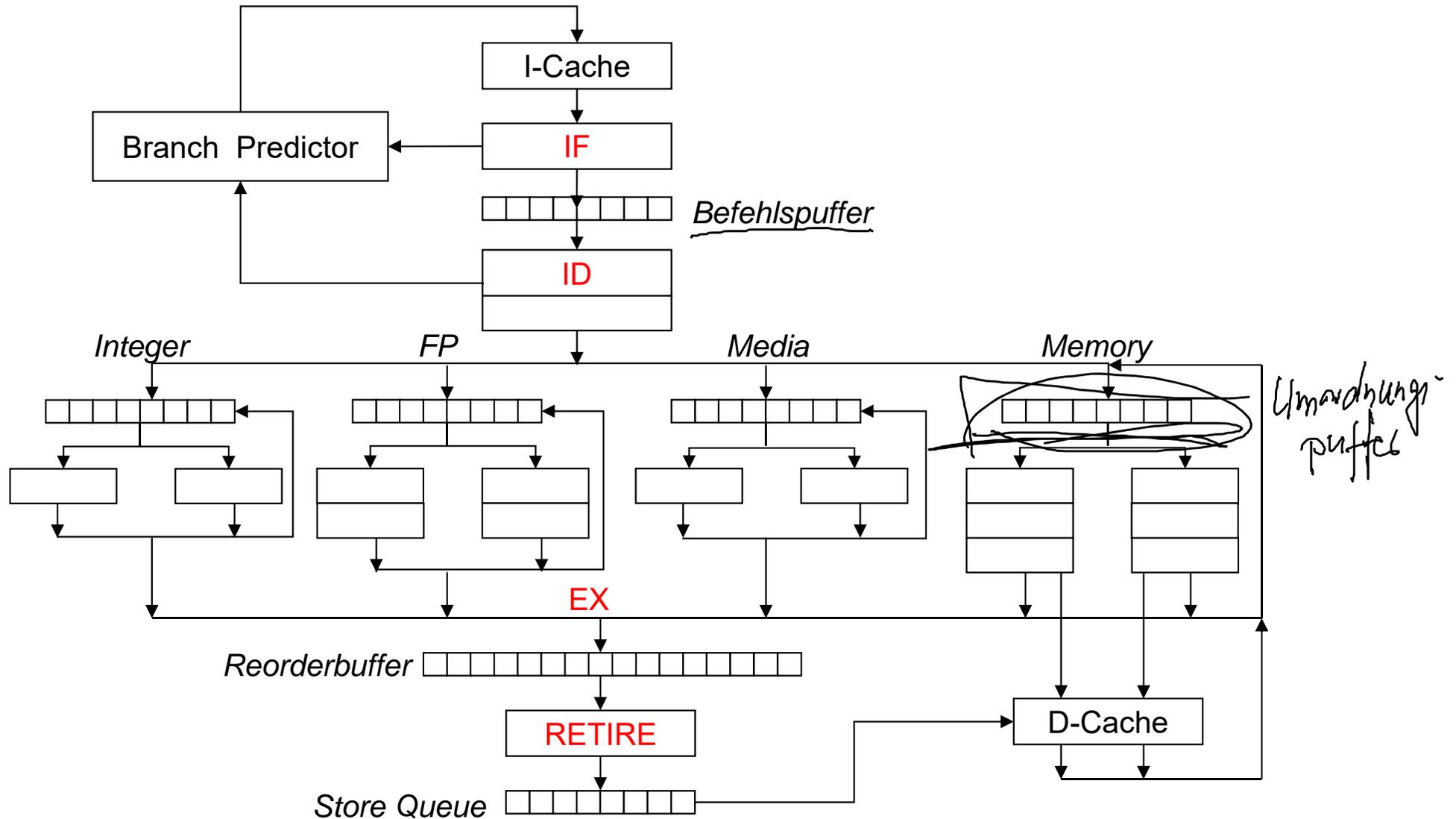
Kapitel 11

Parallele Rechnerstrukturen

Parallele Prozessorarchitekturen
Parallelrechner

11.3 Parallelismus auf Befehlsebene

Superskalärer Prozessor



11.3 Parallelismus auf Befehlsebene

Superskalarer Prozessor

■ Komponenten

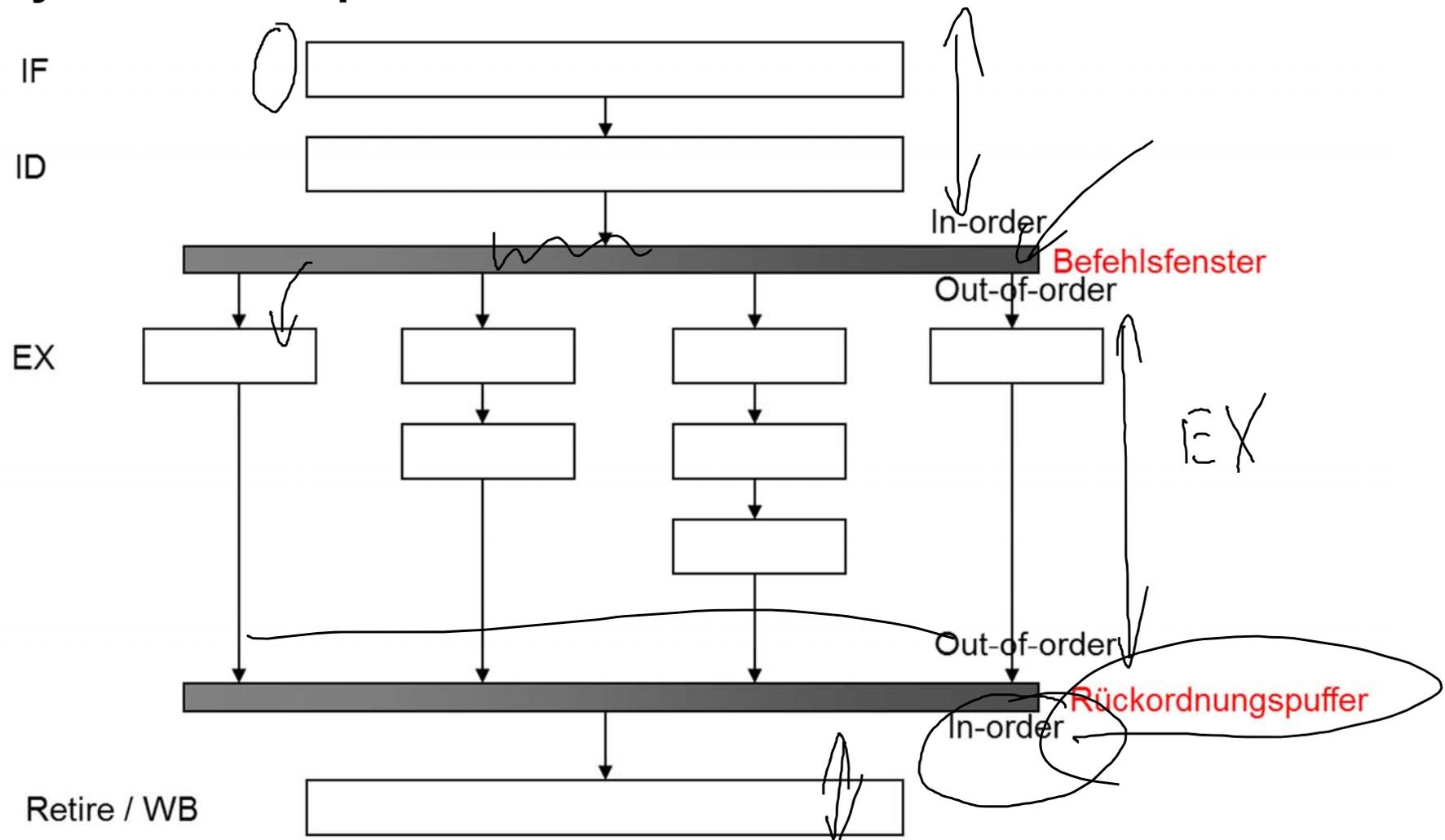
- **Befehlsholeinheit** (Instruction Fetch)
- **Dekodiereinheit** (Instruction Decode) mit **Registerumbenennung** (register renaming)
- **Zuordnungseinheit** (Instruction Issue)
- Unabhängige **Verarbeitungseinheiten** (Functional Units)
- **Rückordnungseinheit** (Retire Unit)
- **Register:**
 - Allzweckregister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

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

11.3 Parallelismus auf Befehlsebene

Superskalärer Prozessor

■ Dynamische Pipelines



11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ 1. In-order-Abschnitt

- Befehle werden entsprechend ihrer Programmordnung bearbeitet
- Umfasst
 - die **Befehlsholphase** (IF)
 - die **Dekodierphase** (ID)
 - die **Zuordnungsstufe** (Dispatch)
 - Dynamische Zuordnung der Befehle an die Ausführungseinheiten
 - Scheduler bestimmt die Anzahl der Befehle, die im nächsten Takt zugeordnet werden können

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Out-of-order-Abschnitt

- Ausführungsphase

■ 2. In-order-Abschnitt

- **Gültigmachen der Ergebnisse** entsprechend der ursprünglichen Programmordnung (**Retire**)
- Erhalten der korrekten Programmsemantik
 - Ausnahmeverarbeitung (precise interrupts)
 - Spekulation

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Befehlsholphase (IF Phase)

■ Befehlsbereitstellung

- Holen mehrerer Befehle aus dem Befehls-Cache in den Befehlsholpuffer
- Anzahl der Befehle, die geholt werden, entspricht typischer Weise der Zuordnungsbandbreite
- Welche Befehle geholt werden hängt von der Sprungvorhersage ab

■ Verzweigungseinheit

- Überwacht die Ausführung von Verzweigungen, Sprungbefehlen
- Speklatives Holen von Befehlen
- Spekulation über weiteren Programmverlauf wird von dynamischen Sprungvorhersagetechnik entschieden
- Verwendung der Vorgeschichte von Sprüngen
- Gewährleistet im Falle einer Fehlspekulation die Abänderung der Tabellen sowie das Rückrollen der fälschlicherweise ausgeführten Befehle

■ Befehlsholpuffer

- entkoppelt die IF Phase von der ID Phase

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Befehlsholphase (IF Phase)

■ Sprungvorhersage und spekulative Ausführung

■ Problem

- Hohe Zuordnungs- und Ausführungsbandbreite
- Etwa jeder 5.- 7. Befehl ist bedingter Sprungbefehl, der den kontinuierlichen Befehlsfluss in der Pipeline unterbrechen kann
- Unter Berücksichtigung der spekulativen Ausführung von Befehlen können sich mehrere Sprungbefehle in der Pipeline befinden

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

- **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

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

- **Sprungvorhersage (Branch Prediction):** Vorhersage des Verhaltens bei Verzweigungen
 - **Dynamische Sprungvorhersage**
 - Die Verzweigungsrichtung hängt von der Vorgeschichte der Verzweigung ab
 - Berücksichtigung des Programmverhaltens
 - Genaue Vorhersagen möglich
 - Hoher Hardware-Aufwand!

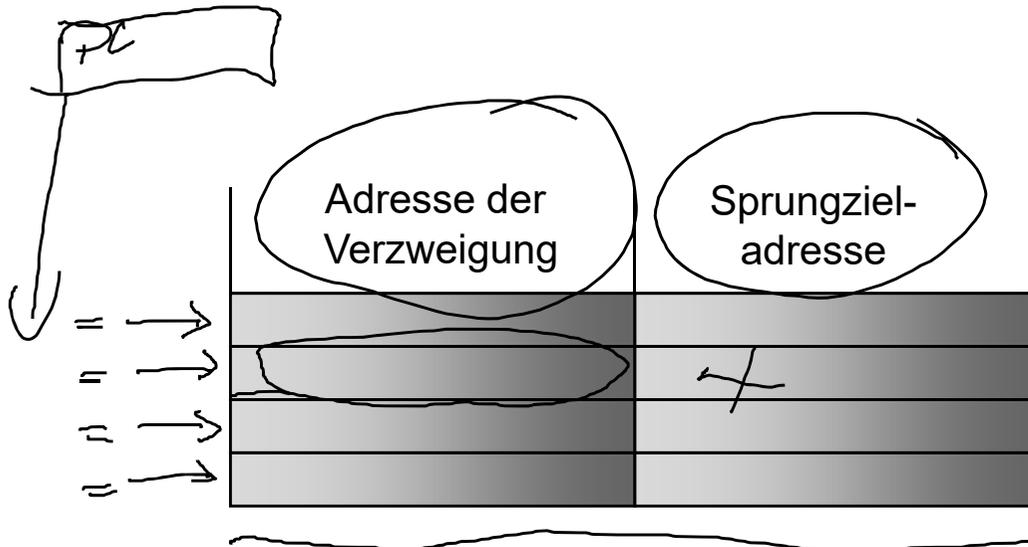
11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dynamische Sprungvorhersage

- Sprungziel-Cache: Branch Target Address Cache (BTAC), Branch Target Buffer (BTB)

- Speichert die Adresse der Verzweigung und das entsprechende Sprungziel

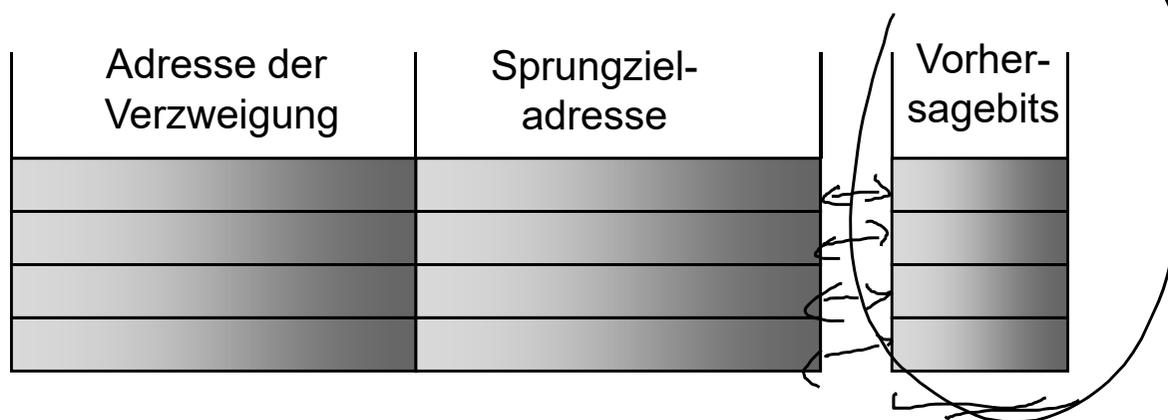


11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dynamische Sprungvorhersage

- **Sprungziel-Cache: Branch Target Address Cache (BTAC), Branch Target Buffer (BTB)**
 - Speichert die Adresse der Verzweigung und das entsprechende Sprungziel
- **Sprungverlaufstabelle, Branch History Table (BHT)**
 - Festhalten des Verhaltens der Sprungbefehle während der Ausführung des Programms: **Prädiktoren**
 - Vorhersage des Verhaltens eines geholten Sprungbefehls



11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

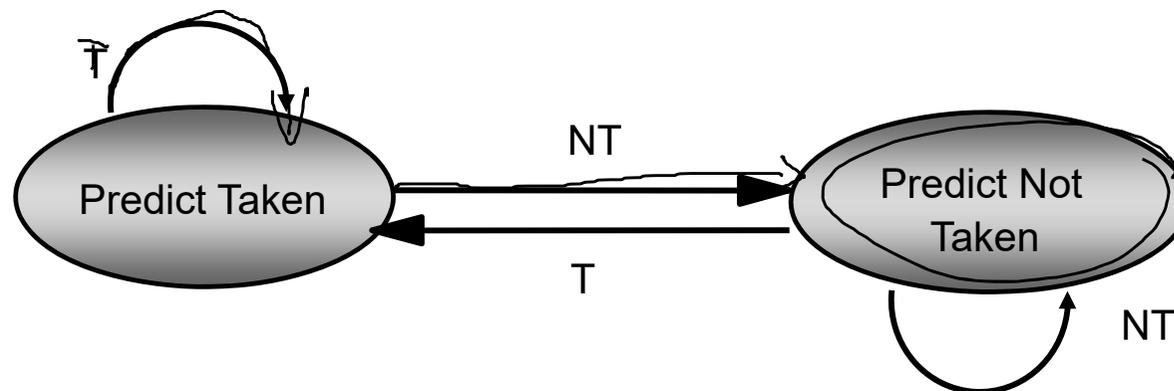
■ Dynamische Sprungvorhersage

■ Sprungverlaufstabelle, Branch History Table (BHT)

■ 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



11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dynamische Sprungvorhersage

- Sehr aufwendige Techniken für superskalare Prozessoren für die Gewährleistung einer möglichst genauen Vorhersage
 - (m,n)-Korrelationsprädiktoren
 - Zweistufige adaptive Prädiktoren
 - Gselect- und gshare-Prädiktoren
 - Hybridprädiktoren
- ↕
- Literatur zur Sprungvorhersage:
 - Brinkschulte/Ungerer: Microcontroller und Mikroprozessoren: Kap. 2.4.6, 7.2

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dekodierphase (ID Phase)

- Dekodierung der im Befehlspeicher abgelegten Befehle
 - Anzahl der Befehle, die dekodiert werden, entspricht typischer Weise der **Befehlsbereitstellungsbandbreite**

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dekodierphase (ID Phase)

■ Registerumbenennung

- Dynamische Umbenennung der Operanden- und Ergebnisregister
- Abbildung der nach außen hin sichtbaren Architekturregister in interne physikalische Register
 - Zur Laufzeit wird für jeden Befehl das jeweils spezifizierte Zielregister auf ein noch nicht belegtes physikalisches Register abgebildet
 - Nachfolgende Befehle, die dasselbe Architekturregister als Operandenregister verwenden, erhalten das entsprechende physikalische Register
 - Anzahl der Umbenennungsregister kann die Anzahl der Architekturregister überschreiten

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dekodierphase (ID Phase)

■ Registerumbenennung

■ Auflösung von Konflikten aufgrund von Namensabhängigkeiten:

■ Lese-nach-Schreib-Konflikt (Write-After-Read, WAR)

- Tritt auf, wenn Befehl j sein Zielregister beschreibt, bevor Befehl i den Operanden gelesen hat.
- D.h. der Befehl i liest einen falschen Wert

■ Schreib-nach-Schreib-Konflikt (Write-After-Write, WAW)

- Tritt auf, wenn Befehl j sein Zielregister beschreibt, bevor Befehl i das Ergebnis geschrieben hat.
- D.h. Der Befehl i liefert den Wert für das Zielregister, anstelle von j

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dekodierphase (ID Phase)

■ Registerumbenennung

■ Ursachen für diese Datenkonflikte:

■ Namensabhängigkeiten

- Treten auf, wenn zwei Instruktionen dasselbe Register dieselbe Speicherzelle (den Namen) verwenden, aber kein Datenfluss zwischen den Befehlen mit dem Namen verbunden ist.

- Es gibt zwei Arten von Namensabhängigkeiten zwischen zwei Befehlen i und j :

■ Gegenabhängigkeit (Anti dependence)

■ ADD R2, R3, R4

■ XOR R3, R5, R6

■ Ausgabeabhängigkeit (Output dependence)

■ ADD R2, R3, R4

■ XOR R2, R5, R6

HW

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Dekodierphase (ID Phase)

- Schreiben der Befehle in ein Befehlsfenster (instruction window)
- Folge:
 - Befehle sind durch die Sprungvorhersage frei von Steuerflussabhängigkeiten
 - Befehle sind aufgrund der Registerumbenennung frei von Namensabhängigkeiten

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Zuordnungsphase (Dispatch)

- Zuführung der im Befehlsfenster wartenden Befehle zu den Ausführungseinheiten
- Zuordnung bis zur maximalen Zuordnungsbandbreite pro Takt
- Dynamische Auflösung der Konflikte aufgrund von echten Datenabhängigkeiten und Ressourcenkonflikten

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Zuordnungsphase (Dispatch)

- Zuführung der im Befehlsfenster wartenden Befehle zu den Ausführungseinheiten
- Zuordnung bis zur maximalen Zuordnungsbandbreite pro Takt
- **Dynamische Auflösung der Konflikte aufgrund von echten Datenabhängigkeiten und Ressourcenkonflikten**
 - **Lese-nach-Schreib-Konflikt (Read-After-Write, RAW)**
 - Tritt auf, wenn Befehl j sein Quellregister liest, bevor Befehl i das Ergebnis geschrieben hat.

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Zuordnungsphase (Dispatch)

■ Ursachen für Datenkonflikte:

■ Echte Datenabhängigkeit (true dependence, flow dependence)

- Ein Befehl j ist datenabhängig von einem Befehl i , wenn eine der folgenden Bedingungen gilt:

- Befehl i produziert ein Ergebnis, das von Befehl j verwendet wird, oder
- Befehl j ist datenabhängig von Befehl k und Befehl k ist datenabhängig von Befehl i (Abhängigkeitskette)

■ Beispiel:

LOOP:

L.D	F0,0(R1)
ADD.D	F4,F0,F2
S.D	F4,0(R1)
D.ADDUI	R1 R1,#-8
BNE	R1 R2,LOOP

Abhängigkeit \circ tritt in Pipeline auf, in der der Vergleich in der ID Phase stattfindet

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Zuordnungsphase (Dispatch)

■ Umordnungspuffer (Reservierungstabellen, reservation stations)

- Liegen vor den Verarbeitungseinheiten
- Jede Ausführungseinheit hat seinen eigenen Umordnungspuffer oder mehrere Ausführungseinheiten teilen sich einen Umordnungspuffer
- Zuordnung eines Befehls an Umordnungspuffer kann nur erfolgen, wenn ein freier Platz vorhanden ist, ansonsten müssen die nachfolgenden Befehle warten (Auflösen von Ressourcenkonflikten)

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

Umordnungspuffer (Reservierungstabelle, Reservation Station)

- Jeder Eintrag enthält jeweils Felder für:
 - die Werte der zwei Quelloperanden (Src1, Src2);
 - die Nummern (Namen, Tags) der Reservierungstabellen derjenigen Funktionseinheiten, welche die Quelloperanden für die auszuführende Operation liefern werden (RS1, RS2), falls der Operand noch berechnet wird
 - jeweils ein Flag für jeden Operanden, das anzeigt, ob ein Operand verfügbar ist (Vld1, Vld2) und
 - einen Namen (destination tag) für das Ziel (Dest).

x $\text{fadd } r1, r2, r3$
 $r4 \text{ sub } r5, r1, r2$

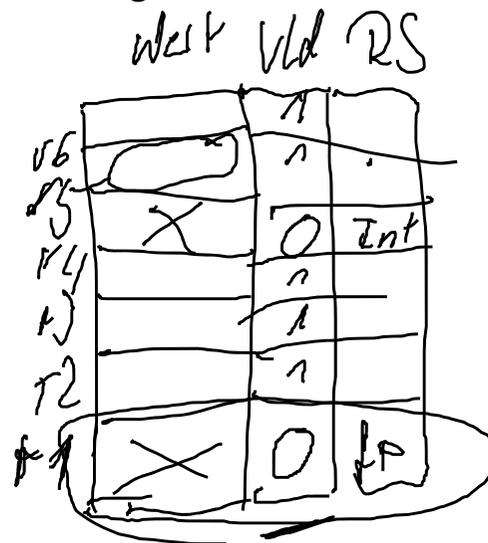
	Quelloperand 1			Quelloperand 2			Ziel
	Vld1	Src1	RS1	Vld2	Src2	RS2	Dest
Befehl n	1	(r2)		1	(r2)		r1
Befehl n+1	0	(r1)	fp	1	(r6)	x	r5
Befehl n+2							

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

Registerdatei

- Jedes Register einer Registerdatei enthält
 - das Feld für den Wert (Registerinhalt)
 - einen Namen (destination tag, Dest), der mit dem Ergebnis assoziiert ist, und
 - ein Bit, das anzeigt, ob der Wert für das Register gerade berechnet wird.



Hand-drawn instruction examples:

$t \rightarrow \text{fadd } (r1, r2), r3$

$t+1 \rightarrow \text{sub } r5, r1, r6$

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Zuordnungsphase (Dispatch)

■ Rückordnungspuffer (reorder buffer)

- Festhalten der ursprünglichen Befehlsanordnung
- Eintragen der Befehle, die die Dekodierphase verlassen und in das Befehlsfenster eingetragen werden
- Während der folgenden Phasen, die ein Befehl zu durchlaufen hat, wird dessen jeweiliger Ausführungsstand protokolliert.

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Befehlsausführung

- Ausführung der im Opcode spezifizierten Operation und Speichern des Ergebnisses im Zielregister (Umbenennungsregister)
- Einzyklusoperationen
 - Ausführung benötigt einen Taktzyklus
- Mehrzyklusoperationen
 - Ausführung einer Operation auf einer Ausführungseinheit kann mehrere Zyklen dauern
 - Ausführungs-Pipeline, arithmetische Pipeline

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Completion

- Eine Instruktion beendet ihre Ausführung, wenn das Ergebnis für nachfolgende Befehle bereitsteht (Forwarding, Puffer)
- **Completion** heißt: eine Befehlsausführung ist „vollständig“
 - Erfolgt unabhängig von der Programmordnung!
- Bereinigung der Reservierungstabellen
- Aktualisierung des Zustands des Rückordnungspuffers (Reorder Buffer)
 - Es kann eine Unterbrechung angezeigt sein.
 - Es kann ein vollständiger Befehl angezeigt werden, der von einer Spekulation abhängt.

11.3 Parallelismus auf Befehlsebene

Superskalare Prozessorpipeline

■ Rückordnungsstufe (Retire)

■ Commitment:

- Nach der Vervollständigung beenden die Befehle ihre Bearbeitung (Commitment), d.h. die Befehlsresultate werden in der Programmreihenfolge gültig gemacht
- Ergebnisse werden in den Architekturregistern dauerhaft gemacht, d.h. aus den internen Umbenennungsregistern (Schattenregistern) zurück geschrieben.

■ Bedingungen für Commitment:

- Die Befehlsausführung ist vollständig
- Alle Befehle, die in der Programmordnung vor dem Befehl stehen, haben bereits ihre Bearbeitung beendet oder beenden ihre Bearbeitung im selben Takt.
- Der Befehl hängt von keiner Spekulation ab.
- Keine Unterbrechung ist vor oder während der Ausführung aufgetreten

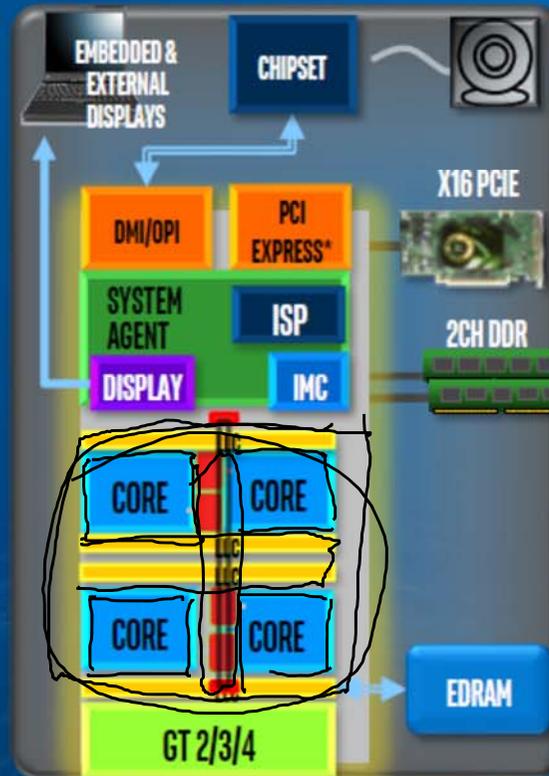
Superskalartechnik

Fallstudie: Intel Skylake

INTEL'S SKYLAKE MICROARCHITECTURE

HIGH LEVEL VIEW

- Increased chipset I/O throughput, Tablet I/Os, Audio DSP Upgrade, Sensor Hub
- Higher resolution display
- Bigger/wider core, better instruction per clock, improved power efficiency
- Enhanced ring/LLC for improved throughput



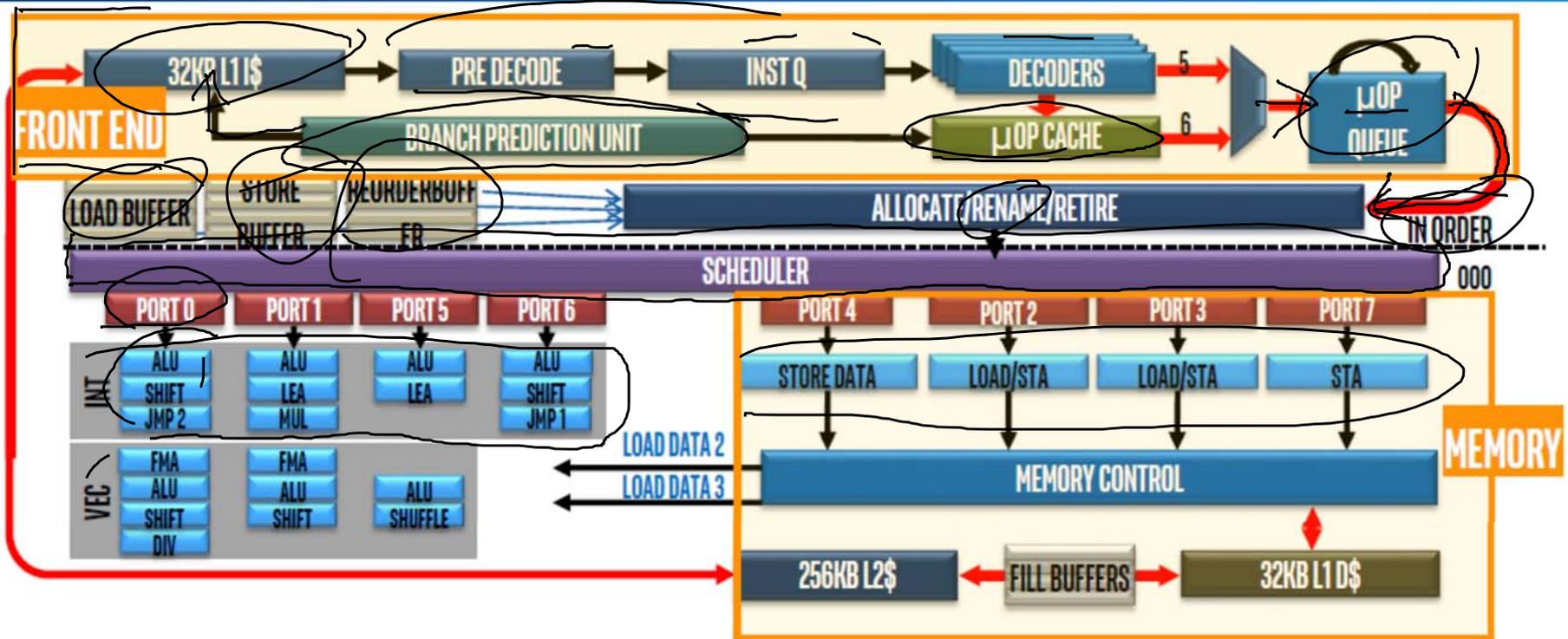
- Integrated camera ISP
- Extended overclocking capabilities
- Faster DDR Memory
- Advanced Processor Graphics GT3 + eDRAM, GT4 + eDRAM; OpenCL™ 2.0 API, DirectX® 12, OpenGL* 4.4

Quelle: http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.90-High-Perform-Epub/HC28.23.911-Skylake-Doweck-Intel_SK3-r13b.pdf

Superskalartechnik

Fallstudie: Intel Skylake

SKYLAKE CORE

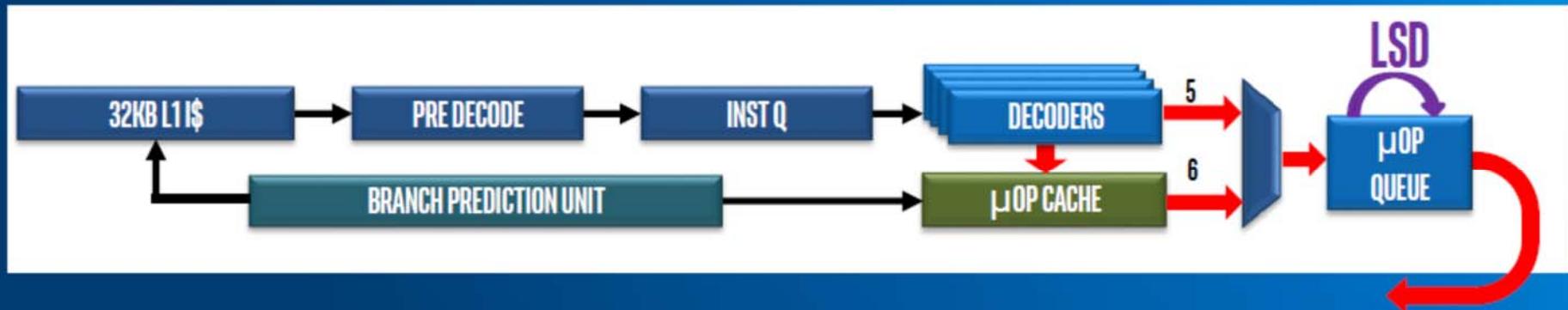


Quelle: http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.90-High-Perform-Epub/HC28.23.911-Skylake-Doweck-Intel_SK3-r13b.pdf

Superskalartechnik

Fallstudie: Intel Skylake

SKYLAKE CORE: FRONT-END



IMPROVED FRONT-END

- Increased bandwidth of Instruction Decoders and μ op-cache
- Higher capacity, improved Branch Predictor
- Reduced penalty for wrong direct jump target prediction
- Faster instruction prefetch
- Increased capacity of the μ op queue / Loop Stream Detector

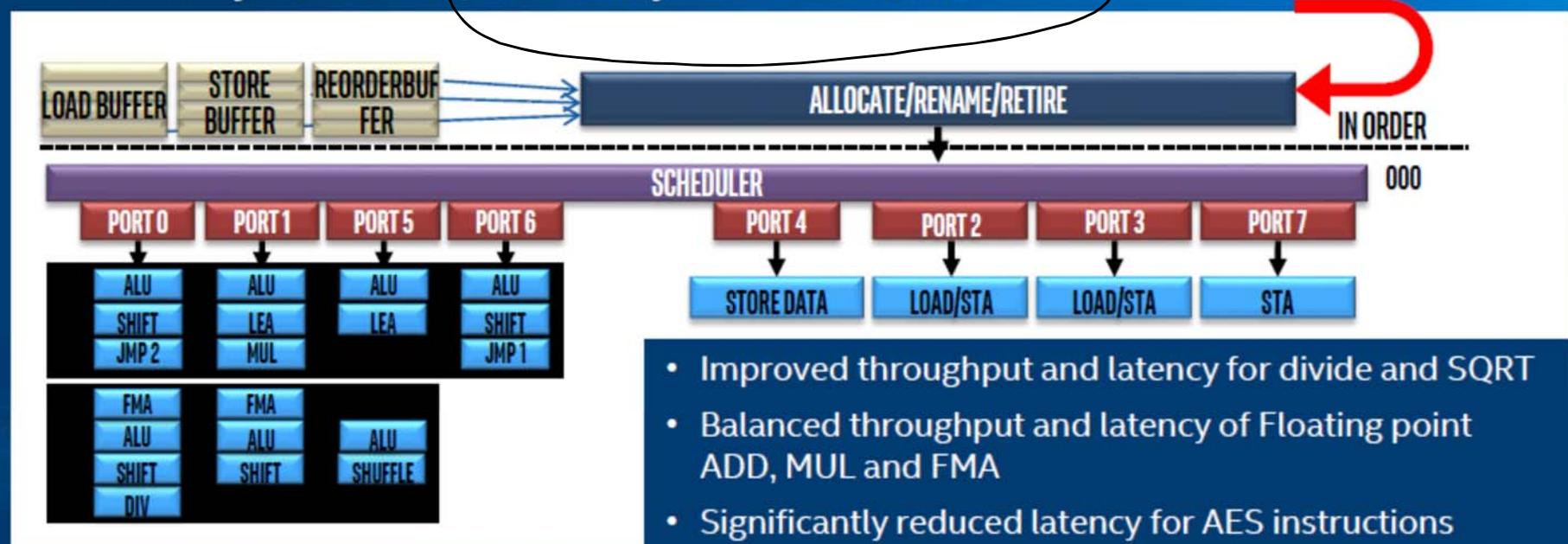
Quelle: http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.90-High-Perform-Epub/HC28.23.911-Skylake-Doweck-Intel_SK3-r13b.pdf

Intel Skylake

SKYLAKE CORE: OUT-OF-ORDER EXECUTION

Deeper Out-of-Order buffers extract more instruction parallelism

- 97 entry scheduler, 224 entry Reorder Buffer



Quelle: http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.90-High-Perform-Epub/HC28.23.911-Skylake-Doweck-Intel_SK3-r13b.pdf