
Kapitel 12 Superskalare Rechner

12.1. Einleitung

Die Neunzigerjahre brachten noch einmal eine beachtliche Steigerung der Zahl von Transistorfunktionen auf einem Chip auf ca. 6 - 8 Millionen bei CPU's und eine Vergrößerung der Zahl der Anschlüsse nach außen durch Stiftgitter (pin-grid-array, PGA) und flip-chip bonding der Anschlüsse (Anschlußpunkte nach außen können überall auf dem Chip liegen und die Anbindung an die Keramikträgerplatte geschieht über Miniaturlötperlen).

Das eröffnete der Rechnerarchitektur neue Möglichkeiten:

- getrennte Befehls- und Datencaches mit 8 - 32 kB
- "Predecode" Caches zur schnellen Bestimmung des Befehlstyps
- viele Register intern, die dem Programmierer nicht sichtbar sind
- mehrere ALU's und FPU's
=> Parallelausführung von 2 - 6 Befehlen in einem Takt
- Scoreboard zur Auflösung von Datenhazards
- Sprungvorhersagetabelle mit "history bits" zur Entschärfung von Steuerhazards
- spekulative Befehlsausführung
- interne Wortbreiten von 64 Bit
- sehr breite Busse nach außen
- durch kleinere Linienbreiten (0.35 μm) höhere Clockfrequenzen > 200 MHz
- Pipelining für Befehle
- MMU an Bord des Chip.

Im Folgenden sollen anhand typischer Maschinen diese Konzepte näher erläutert werden.

12.2. Motorola MPC 601 - Power PC, (1993) - Befehlspipelining; MMU an Bord -

Mehrere Befehle (8 x 32 Bit) werden parallel aus einem Cache in eine Befehlswarteschlange geladen, die 8 Plätze umfaßt. Noch vor dem eigentlichen Dekodiervorgang in Platz Q_0 der Warteschlange werden die Befehle schon einmal grob sortiert: Sprungbefehle werden in Sprungvorhersagepuffer geladen und Gleitkommabefehle in eine eigene GK-Pipeline, während Integer-Befehle in eine normale Befehlspipeline einmünden.

Die in den Pipelines erzeugten Steuerbefehle gehen parallel in eine GK-Recheneinheit und eine Integer-Recheneinheit. Die GK-Recheneinheit hat 32 Register à 64 Bit und Schaltnetze für Addition und Multiplikation und ein Schaltwerk für die Division. Die Integer-Recheneinheit arbeitet mit 32 Registern zu 32 Bit und hat Schaltnetze für Addition, Multiplikation, Division, Booleans und Ausnahmebehandlungen.

Die Datenversorgung geschieht aus einem Cache (32 kB, 8-fach satzassoziativ, Satzlänge 64 Byte mit write-back). Zwischen dem Cache und der Recheneinheit liegt ein Interface, das von einer Warteschlange für Adressen gesteuert wird.

Die Adressierung des Cache geschieht über eine integrierte MMU. Sie wird getrennt angesprochen von Sprungadressen und Adressen, die Daten speichern oder laden wollen.

Sprungadressen (logische prefetch-Adressen) landen in einem Assoziativspeicher (instruction translation look aside buffer, ITLB), der die 4 zuletzt benutzten Befehlsadressen vorhält und bei Erfolg den Cache adressiert.

Mißerfolg adressiert, gleich den Adressen aus der Recheneinheit, einen Assoziativspeicher für die letzten 4 Blockadressen (block translation look aside buffer, BTLB) und einem allgemeinen Assoziativspeicher für die letzten 256 Adressübersetzungen (universal translation look aside buffer, UTLB). Bei Erfolg wird der Cache adressiert, sonst muß eine Tabellensuche eingeleitet werden.

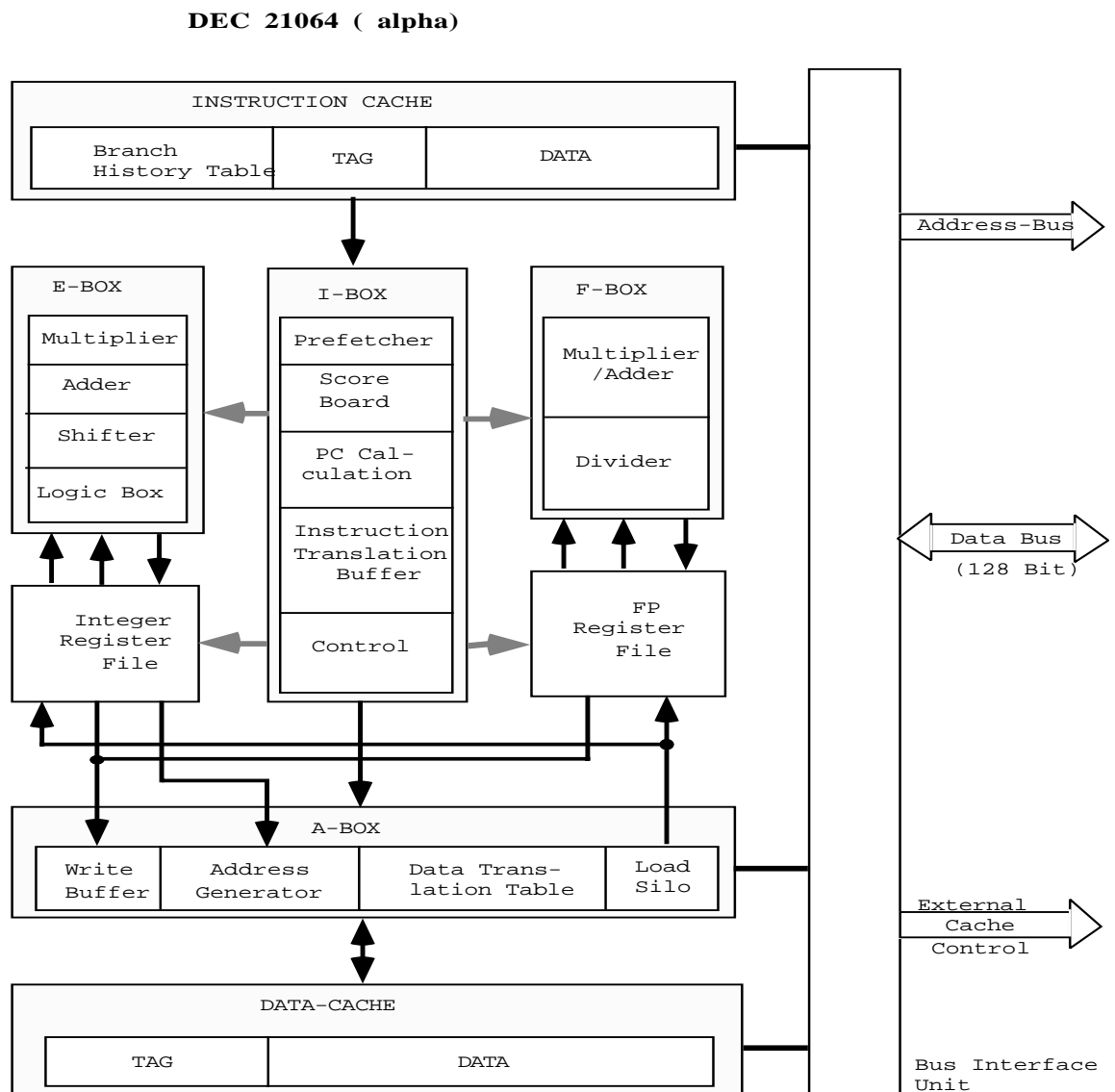
Auf dem Chip ist noch eine Speicher-Puffer-Einheit zu finden: Sie verwaltet den Zugriff auf den Cache, enthält die snoop-Einheit und zwei Warteschlangen für Lesen und Schreiben von Daten.

12.3. Digital Equipment Alpha 21064 (1992) - 64 Bit-Register, Programm- und Datencache -

Diese Maschine war der erste Mikroprozessor mit einer internen Wortlänge von 64 Bit. Nach außen hin hat der Rechner ein pin-grid-array mit 431 pins, davon werden 128 für Daten gebraucht und 64 für Adressen.

Intern realisiert die Maschine eine Harvard-Architektur mit getrenntem Speicher für Daten und Programme in Form von zwei Caches von je 8 k Byte Größe. Der Datencache ist "direct mapped" mit "write through", der Programmcache "direct mapped".

Außerhalb des Chip sind ein weiterer Cache (2. Stufe) vorgesehen und eine MMU mit zwei getrennten Assoziativspeichern für Daten (DTC) mit 32 Einträgen voll assoziativ für Seitengrößen von 8 - 4096 k Byte und für Programme (ITC) mit 8 voll assoziativen Einträgen.



Der interne Prozessor besteht aus vier Blöcken:

- einer I-Box für Befehlsbehandlung mit Prefetch, Scoreboard, Befehlsbuffer, Berechnung des nächsten Befehls und der eigentlichen Steuerung, die parallel zwei Befehle ausgeben kann an die Integer-Box (E-Box), die GK-Einheit (F-Box) oder die Adress-Recheneinheit (A-Box).
- einer F-Box für GK-Operationen mit eigenem Registerfile, der GK-Befehle in einer 10-stufigen Pipeline abarbeitet.
- einer E-Box für Integer-Befehle mit einem Registerfile, der über drei Busse mit den Rechenwerken für Multiplikation, Addition/Subtraktion, Schiebebefehle und logische Befehle verbunden ist und in 7 Stufen Befehle abarbeitet. Das Registerfile erhält, wie auch die GK-Register, seine Daten aus der A-Box.

- einer A-Box, die die Adressrechnungen macht und Daten zum Schreiben und Lesen zwischenspuffert, bevor sie in den Datencache übertragen werden. Die A-Box unterstützt bedingte Abfragen im Speicher. Es gibt ein LL-SC-Paar als Befehle.

LL < address > : load and link; lade die adressierte Speicherzelle, schiebe die Adresse in ein Komparator-Register.

SC < address > : store conditional; schreibe das Wort zurück, wenn die Zelle nicht zwischenzeitlich referenziert wurde; sonst mache die Operation zwischen LL und SC rückgängig.

Das entspricht einer "set and test"-Operation bei CISC-Rechnern.

Durch die verwendete Technologie kann die Maschine mit Taktfrequenzen von 200 MHz laufen; verbraucht dann allerdings 30 W Leistung.

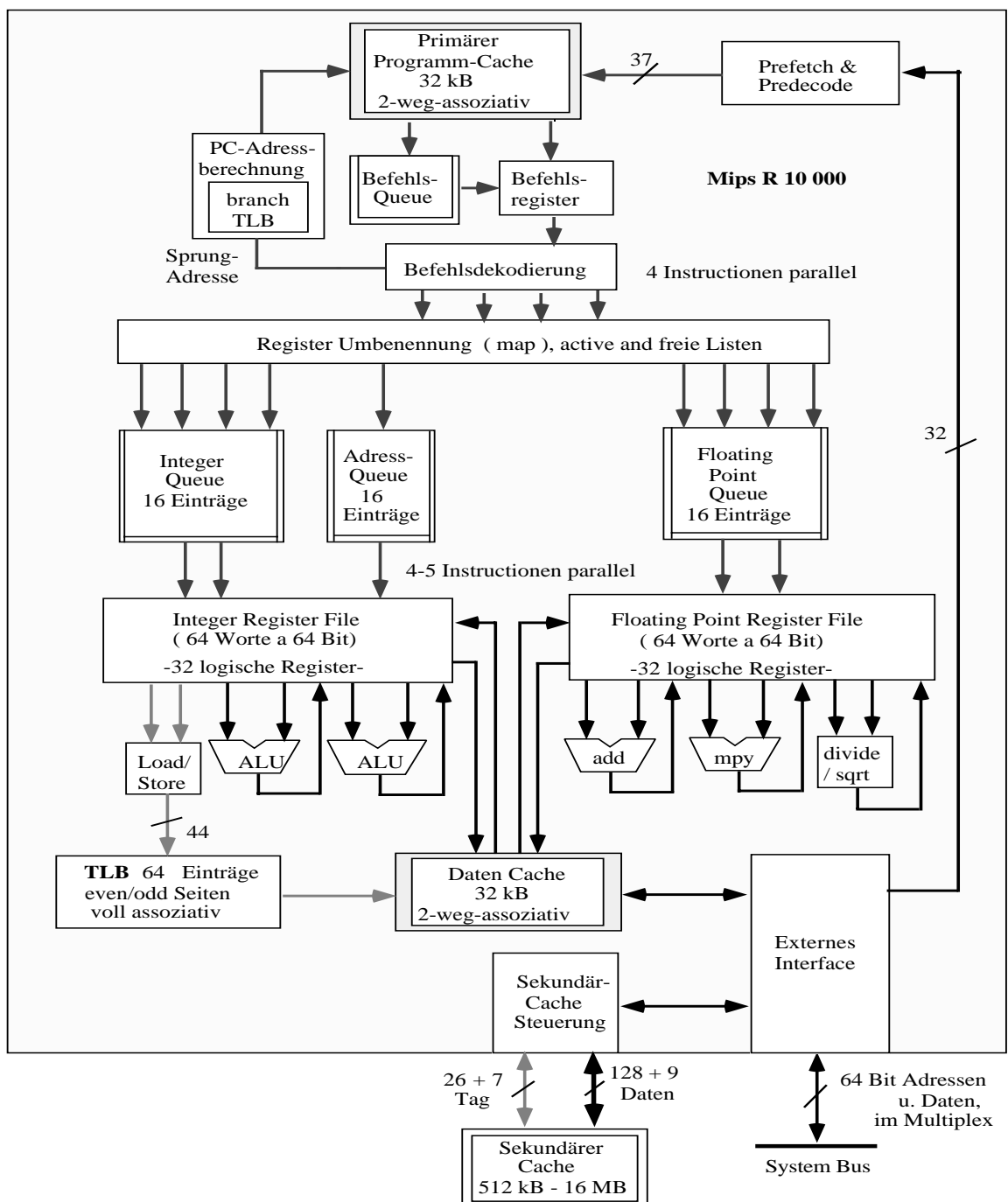
Merkmale und Leistungsdaten des Alpha-Prozessors

Technologie	0.75 μ CMOS
Zykluszeit (Takt)	6.6 ns (150 MHz) / 5 ns (200 MHz)
Chipgröße	13.9 x 16.8 mm ²
Zahl der Transistoren	1.68 Millionen
Gehäuse	431 pin grid array (PGA)
Leistungsaufnahme	23 W bei 150 MHz / 30 W bei 200 MHz
Betriebsspannung	3.3 V
Datencache	8 kByte, physikalisch, direct mapped, write through
Befehls-cache	8 kByte, physikalisch, direct mapped
DTC in der MMU	32 Einträge, voll assoziativ, Seitengröße 8 4096
ITC in der MMU	8 Einträge, voll assoziativ, Seitengröße 8 4096
Prozessorbus	128-bit Datenbus, 64 bit Adressbus
Adresse, virtuell	64
Adresse, phys.	34
Seitengröße	8 kByte
Befehlseinheit	gibt gleichzeitig 2 Befehle an A-, E- oder F-Box
Integer-Processor	7-stufige Pipeline
Gleitpunkt-Processor	10-stufige Pipeline

**12.4. Mips R 10000 (1995)
- ANDES-Architektur -**

Mit dem Fortschritt in der Produktionstechnik kann der R 10000 mit einem 0.5 µm CMOS-Prozeß und 4-Lagen-Verdrahtung 6.4 Millionen Transistorfunktionen auf einem Chip von 17.3 x 17.2 mm² Fläche unterbringen. Nach außen hin sind über einen pin-grid-array mit 527 Stiften Daten und Adressen mit 64 Bit Breite und mehr zugänglich.

Der Rechner reserviert 32 k Byte für einen Datencache (1. Stufe) und 32 k Byte für einen Programmcache (1. Stufe) auf dem Chip und braucht dafür 3.6 Millionen Transistorfunktionen.



Der interne Prozessor wird wesentlich geprägt durch eine spekulative Befehlsausführung mehrerer Befehle gleichzeitig, die ANDES-Struktur: architecture with non sequential dynamic execution schedule.

Das wird möglich durch zwei Maßnahmen:

- Es gibt mehr als eine ALU im Rechenwerk und getrennte GK-Rechenwerke. Damit können Befehle parallel ablaufen.
- Es gibt mehr Register als logisch durch Befehle ansprechbar: 64 Register à 64 Bit für Integer-Rechnungen; wobei das Programm 32 ansprechen kann, und 64 Register à 64 Bit für GK-Rechnungen, wobei ebenfalls nur 32 Register im Programm angesprochen werden können.

Eine eigene Einheit verwaltet die Umbenennung der Register (register renaming) und führt entsprechende Verwaltungslisten für belegte und freie Register.

Die Befehle (32-Bit RISC-Befehle) durchlaufen zunächst eine Prefetch- und Predecode-Einheit und werden mit einer zusätzlichen Kennung (tag) von 5 Bit versehen, bevor sie in den primären Programmcache von 32 k Byte eingelagert werden.

Von dort werden die zur Abarbeitung anstehenden Befehle über eine Warteschlange und Befehlsregister in die Dekodiereinheit gebracht. Hier werden 4 Befehle parallel zur Ausführung weitergeleitet.

Die Auswahl braucht nicht sequentiell zu sein, sondern kann dynamisch geschehen.

Die Folgeadressen der Befehle und insbesondere die Sprungadressen durchlaufen einen Assoziativspeicher (branch TLB) mit Charakterisierungen "häufig benutzt", "selten benutzt", "häufig unbenutzt" und "selten unbenutzt". Damit werden hohe Trefferquoten erreicht.

Die in der Umbenennungsstufe noch einmal umgesetzten Befehle laufen in Warteschlangen hinein, an deren Ende sie ausgeführt werden, indem mit den mehrfach vorhandenen ALU's die Registerinhalte verrechnet werden. Bis zu 5 Befehle können parallel ausgeführt werden.

Die Daten werden aus einem 32 kB Cache geladen, der über einen voll assoziativen Speicher angesprochen wird, der 64 Einträge von Seitenadressen im Datencache enthält.

Nach außen hin stellt ein externes Interface die Verbindung her zu einem Sekundärcache von 512 - 16.000 kB mit einer Zugriffsbreite von 128 Bit für Daten (+ 9 Bit Sicherung) und zu dem Systembus, auf den mit Daten und Adressen im Multiplex zugegriffen wird.

Während der Zugriff auf den Daten- und Programmcache mit der Clockfrequenz erfolgt und in einem Takt bis zu 4 Befehle gleichzeitig parallel vorbereitet und bis zu 5 parallel ausgeführt werden, dauert der Zugriff auf den Sekundärcache (2nd level cache) 15 Takte und auf den Hauptspeicher über den Bus 140 - 180 Takte.

Der Prozessor läuft mit 200 MHz Clockfrequenz und verbraucht 30 W bei einer Betriebsspannung von 3.3 V.

Die Leistung liegt bei 9 SPEC int 95 bzw. 19 SPEC fp 95.

12.5. AMD K 6 (1997) - x 86 RISC/CISC-Architektur, MMX execution unit -

x-86-Rechner der Pentium-Reihe von Intel müssen mit dem Befehlssatz leben. Ein superskalarer Rechner, pin-kompatibel zum Pentium-Pro, ist der K 6 von American Micro Devices.

Um die Abarbeitung der x-86-Befehle zu beschleunigen, werden sie durch einen Predecode-Cache hindurch in den Befehls-cache geladen. Im Predecode-Cache wird zu jedem x-86-Befehl ein zusätzliches 4-Bit Nibble erzeugt und hier gespeichert. Jeweils 16 Befehlsbytes und 16 Nibbles werden parallel aus dem Befehls-cache und Predecode-Cache geladen. Die Nibbles enthalten Informationen über Befehls-grenzen und ob Sprungbefehle entdeckt wurden.



Die Instruktionsbytes laufen in einen Puffer hinein, aus dem jeweils parallel 2 komplette Instruktionen pro Zyklus in zwei Dekodierer abgerufen werden. Sie werden hier in RISC-86-Befehle zerlegt.

Einige werden direkt dekodiert, andere sprechen einen Mikrocode-ROM an; Sprungbefehle suchen die Sprungadresse in einer Sprungvorhersagetabelle (branch prediction table) mit 8192 Einträgen und einen Sprung-Ziel-Cache mit 16 Einträgen. Die Sprungvorhersage benutzt ein 2-Bit-Tag: überwiegend gesprungen, selten gesprungen, selten nicht gesprungen, überwiegend nicht gesprungen. Parallel können pro Zyklus 4 RISC-86-Befehle sequentiell verarbeitet werden.

Die RISC-86-Befehle durchlaufen eine Befehls-Steuereinheit (instruction control unit, ICU) und werden in einem Befehlspool mit 24 Einträgen gesammelt.

Bevor sie ausgeführt werden, muß eine Umbenennung der Register geschehen (register renaming).

Es gibt 48 Register, während der x-86-Code nur 8 Register direkt manipulieren kann und indirekt Zugriff hat auf ein Statusregister, den Befehlszähler und 4 Segmentregister. Die Befehlsausführung geschieht out-of-order und mit spekulativer Sprungbefehlsausführung.

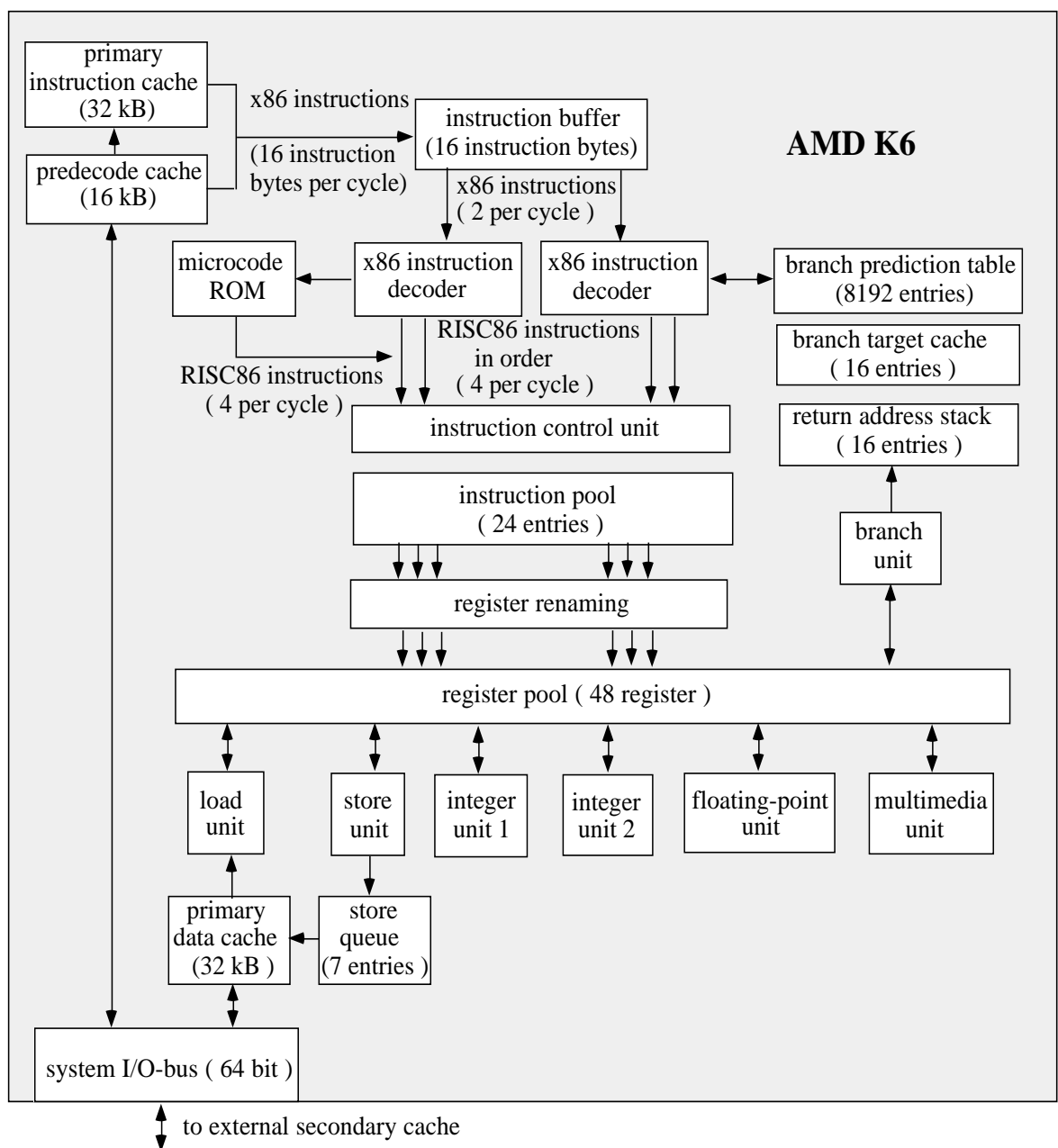
Um bei Unterprogramm-sprüngen das PSW nicht auf einem Stack im Speicher retten zu müssen, hat die Maschine einen Rückkehradresstack der Tiefe 16 an Bord.

Daten, die geladen werden sollen, durchlaufen eine Ladeeinheit, die auf den primären Datencache von 32 kB zugreift. Abzuspeichernde Daten durchlaufen eine Speichereinheit und werden in einer Warteschlange der Tiefe 7 vor dem Datencache gesammelt.

Die Maschine enthält 7 parallel arbeitende Recheneinheiten inklusive einer FPU und einer eigenen Einheit für Multimedia-Befehle (MMX-Einheit). Sie kennt 10 neue Befehle, die 3-D-Graphik, MPEG-2 Dekompression, Bewegungsvorhersage und Pixelmanipulation unterstützen wie z. B. Befehle der Art $a := a + b \cdot c$ oder Byteoperationen parallel auf 4 Byte.

Sie hat eine 4 - 5-stufige Pipeline, die in der Mitte breit wird:

- 2 x-86-Befehle gehen in die Dekoder
- 4 RISC-86-Befehle gehen in die Instruktionseinheit
- 6 Befehle gehen maximal an die Recheneinheiten
- 4 Operationen können zugleich fertig werden
- 2 Speicheroperationen können durchgeführt werden;
dazu Rückschreiben der Register mit forwarding des reorder-buffers.



Aufgebaut mit einem 0.35 µm CMOS-Prozeß mit 5 Lagen Metallisierung hat die Maschine eine Clockfrequenz von 180 MHz und 6 Millionen Transistorfunktionen auf

dem Chip.

Der Rechner ist pin-kompatibel zum P 54 C Pentium und erreicht die Leistung des Pentium Pro.

Die Leistungen superskalärer Rechner Ende 1996 zeigt die folgende Tabelle.

Hersteller	RISC μ P	Clock (MHz)	SPECfp95	Caches
DEC	21164 Alpha	500	21.1	8 kB I, D; 96 kB 2L
Exponential Technology	x704	466, 533	NA	2 kB I, D; 32 kB 2L
Intel	Pentium	200	3.92	8 kB I, D
	Pentium Pro	200	6.2	8 kB I, D; 256 kB 2L
HP	PA-8000	180	20.2	1-4 MB I, D off chip
IBM / Motorola	Power PC 604e	225	7.5	16 kB I, D
MIPS	R 10 000	200	19	32 kB I, D
	R 8 000	75	> 30	16 kB I, D
Sun	UltraSPARC I	200	11.4	16 kB I, D
	UltraSPARC II	250	NA	16 kB I, D

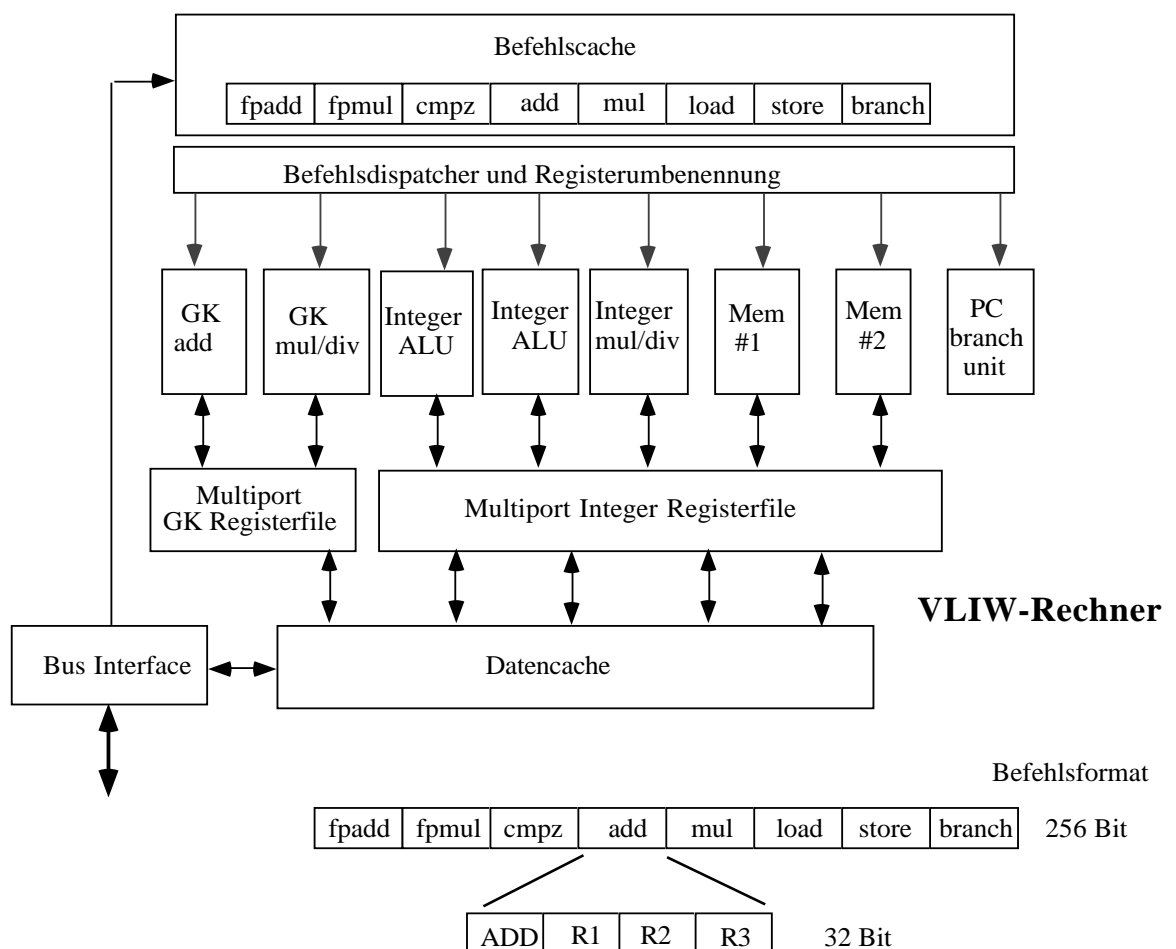
Computer Design, November 1996

12.6. VLIW-Rechner

Mit einem Compiler, der Code erzeugt, der zu Gruppen von unabhängig abarbeitbaren Befehlen zusammengefaßt wird, kann ein Rechner diese Befehle im Prinzip parallel abarbeiten und damit schnell werden.

Man erhält ein sehr langes Befehlswort (256 Bit), das aus 8 Operationsfeldern mit RISC-Befehlen besteht. Jeder Teilbefehl kann eine unabhängige Recheneinheit treiben. Die Sprungvorhersage wird dem Compiler aufgelastet, der sehr viel komplexer wird.

Alle Versuche, solche Maschinen bislang zum Erfolg zu führen, scheiterten.



Der Name der Architektur bezieht sich auf das lange Befehlswort: very long instruction word, VLIW.

Der Compiler muß hier die Optimierungen machen, z. B. loop-unrolling und software-pipelining. Die Programmierung hat Ähnlichkeit mit horizontaler Mikroprogrammierung, wo ebenfalls viele Aktionen parallel durchgeführt werden können.

