Pipelining im Alpha 21164

Zusammenfassung von
Wolfgang Wiese
1/1998

Einleitung:

Ende 1994 stellt DEC (Digital Equipment Corporation) den 21164 Alpha Mikroprozessor vor.
Es ist ein mit 300 MHz getakteter Rechner, der auf die 0,5-Üm CMOS Technologie auf.
Der 16,5 x 18,1 mm-große Chip enthielt 9,3 Millionen Transistoren und konnte 1,2 Billionen Anweisungen pro Sekunde ausführen. Der Chip arbeitete 10 % schneller, als sein Vorgänger, der Alpha 21064, wenn dieser ebenfalls auf die 0,5-Üm CMOS-Technologie portiert worden wäre.
Damit war der Alpha 21164 damals einer der schnellsten Ein-Prozessor-Systeme überhaupt.

Der Schlüssel zu dieser großen Leistung lag -neben der Verbesserung der Konzepte vorheriger Versionen und der Portierung auf die CMOS-Technologie- in der superskalaren Instruktionsbereitstellung.

Superskalare Prozessoren besitzen mehrere nebenläufig nutzbare Funktionseinheiten, z.B. für die Verzweigungsvorhersage, für Load/Store-Operationen und für Fest- und Gleitkomma-Operationen.
Diese Einheiten sind in der Regel Pipelines mit gemeinsamen und getrennten Registerfiles.

Bild 1: Superskalare Struktur:

Beim Alpha 21164 ist somit auch der Chip in 5 Einheiten aufgeteilt: Der Instruktions-, Integer-, Fließpunkt- und Speichereinheit, sowie dem Cache-Control/bus interface (CBOX).

Die Instruktionseinheit (oder Befehlsbereitstellungs-Einheit) ist die eigentliche Steuerzentrale des Prozessors. Es kontrolliert alle Datenströme, Zugriffe auf Registerfiles und behandelt Ausnahmen, Traps und Interrups. Die Einheit besteht aus einer 4-stufigen Pipeline und besitzt schnelle Algorithmen zur Steuerung des Kontrollflusses.
Die Integer-Einheit führt alle Integer-Operationen aus. Sie berechnet virtuelle Adressen für Load/Store-Befehle, und führt alle Control-Anweisungen aus, mit Ausnahme der Fließpunkt-abhängigen Abfragen. Die Integer-Einheit enthält die Integer-Register und einige Integer-Rechenfunktionen, welche zum größten Teil wiederum aufgeteilt

wurden zwischen zwei parallelen Pipelines. Jeder dieser beiden Pipelines besteht aus 4 Stufen und besitzt einen Addierer und eine Funktion zur Behandlung der Boolschen Logik.
Die Fließpunkt-Einheit ist ähnlich aufgebaut wie die Integer-Einheit, besitzt allerdings viel mehr Stufen. Sie besteht aus dem Register-File und den beiden parallelen Ausführungspipeline: Eine Addier- und eine Multiplikations-Einheit. Beide Einheiten können Daten vom Format IEEE und VAX bearbeiten.
Die Speicher-Einheit übersetzt virtuelle Adressen zu physikalische Adressen und führt Load-,Store- und Speicherbelegungsfunktionen aus. Die Speicher-Einheit ist 9 Stufen tief.
In der CBOX werden die Lese- und Schreib-Anweisungen, die von der Speicher-Einheit kommen ausgeführt. Außerdem verwaltet die CBOX die Second- und Third-Level-Caches.

Wie oben bereits oben angerissen, ist die Pipeline des 21164 7 Stufen tief für Integer-Berechnungen, 9 Stufen tief für Fließpunkt-Berechnungen und bis zu 12 Stufen für Speicheranweisungen. Zusätzliche Stufen werden benötigt, wenn es zu Zugriffen ausserhalb des Chips kommt.

Bild 2: Die 21164-Pipelines:

Die Pipeline-Stufen:

Die Pipeline für Integer-Funktionen beginnt mit dem Lesen des Registerfiles in Stufe 3. Danach folgt die Ausführung in der Stufe 4. Stufen 5 und 6 bedienen abhängige Move-Anweisungen. In Stufe 6 werden außerdem die Ergebnisse ins Integer Input Register geschrieben. Das Integer Register File besitzt 40 Register, von denen 32 durch die Architektur bestimmt sind, und weitere 8 als versteckte Register für besondere Anweisungen. Die

Integer-Einheit erlaubt das setzen von Bypassen zu allen Teileinheiten, außer der Multiplikation-Einheit. Somit ist man in der Lage, Ergebnisse von jeder vorherigen Stufe zu lesen. Die meisten Anweisungen, welche in der Integer-Einheit ausgeführt werden, führen zu einer Latenzzeit von einem Zyklus. Abhängige Sprünge benötigen zwei Zyklen.

Die Pipeline für Fließpunkt-Berechnungen beginnt mit dem Lesen des Registers in Stufe 4. In den Stufen 5 bis 8 wird
die Anweisung ausgeführt. Die Ergebnisse werden in der letzten Stufe ins Register-File geschrieben. Die Latenzzeit für Fließpunkt-Anweisungen, außer denen der Division, beträgt 4 Zyklen. Die Latenzzeit für Division, ist -wie die Latenzzeit der Multiplikation bei Integer-Anweisungen- abhängig von der Breite der Daten.

Bei der Behandlung von Speicherzugriffen/-Anweisungen wird in Stufe 4 zuerst die virtuelle Adresse berechnet und dann und auf den Daten-Cache zugegriffen. In Stufe 5 wird der Cache-Zugriff beendet, die Adresse übersetzt, sowie ein Cache-Treffer berechnet. Falls ein Cache-Treffer vorkam, werden in Stufe 6 Daten ins Registerfile geschrieben. Im Falle eines „Cache-Miss“ beginnt in Stufe 6 ein Zugriff auf den Second-Level-Cache, der in Stufe 9 endet. Ab Stufe 10 wird der Cache mit Daten gefüllt. Eine abhängige Anweisungen kann dann in Stufe 12 ausgeführt werden.

Ausnahmebehandlung:

Bei vielen Situationen muß die Ausführung von Instruktionen unterbrochen werden. Dies nennt man „Trap“. Bei einer Trap wird die aktuelle Instruktion abgebrochen und dessen Zwischenergebnisse verworfen. Dann wird die Instruktion nochmals ab einer bestimmten -vorgegebenen- Adresse gestartet.
Gründe für Traps sind:

- Interrupts
- Fehler im Control-Fluß
- Logische Fehler (z.B. arithm. Overflow)
- Replay Traps (Speicherkonflikte, Bufferüberlauf, ...)

Bei einer Trap ist jeweils nur die Instruktion betroffen, bei der die Trap ausgelöst wurd, sowie dessen Unterfunktionen. Nicht aber andere Instruktionen in anderen Pipelines.

Quellen:

- *„Superscalar Instruction Execution in the 21164 Alpha Microprocessor“, IEEE Micro, April 1995*
- *„Rechnerarchitektur“, M. Dal Cin*