Prof. Dr. M. Jäger
FB MNI

21. März 2017

Lösungsvorschlag zu Übungsblatt 5 - Betriebssysteme

Aufgabe 1 (Von-Neumann-Architektur)

Lesen Sie im deutschen Wikipedia die Einträge zu folgenden Begriffen:

1. Von-Neumann-Architektur: http://de.wikipedia.org/wiki/Von-Neumann-Architektur
2. ALU: http://de.wikipedia.org/wiki/Arithmetisch-logische_Einheit
3. Rechenwerk: http://de.wikipedia.org/wiki/Rechenwerk
4. Von-Neumann-Zyklus: http://de.wikipedia.org/wiki/Von-Neumann-Zyklus
5. Adressierung (Rechnerarchitektur): http://de.wikipedia.org/wiki/Adressierung_(Rechnerarchitektur)

Beantworten Sie folgende Fragen auf der Basis der oben genannten Begriffsdefinitionen:

a) Was war die eigentliche Neuerung der Von-Neumann-Architektur gegenüber älteren Computern?

   Das Programm war nicht als Hardware realisiert, sondern wurde in den Haupsspeicher geladen wie die Daten, die es verarbeitet.

b) Was ist der Unterschied zwischen einem Prozessor, einem Rechenwerk und einer ALU?

   ALU ist Komponente des Rechenwerks (neben Registersatz). Rechenwerk ist Komponente des Prozessors (neben Steuerwerk, internem Bus)

c) Welche Operationen kann eine ALU ausführen?

   Minimal: ADD, AND, NOT. Normal: Ganzzahl-Arithmetik, div. logische Operationen und Vergleiche, Shift-Operationen.

d) Was bewirkt ein Sprungbefehl in der Hardware?

   Programmzähler wird geändert.

e) Nennen Sie die Funktion einiger Bits des Statusregisters.

   Anmerkung: Synonyme für Statusregister: PSW (Prozessorstatuswort), CCR (Condition Code Register)

   1) Übertragsbit (C für engl. carry bit): Zeigt den Übertrag an, der bei Addition der n-ten Bits (der höchsten Stelle) der Operanden entsteht. Es wird als Kennzeichen für Bereichsüberschreitung bei Betrachtung des Ergebnisses als vorzeichenlose Zahl (nur positive Ergebnisse) verwendet. Wird in 2er-Komplement-Darstellung gerechnet (auch negative Zahlen möglich), so spielt das Carry-Bit für die Bereichsüberschreitung keine direkte Rolle.
   2) Überlaufbit (V für engl. overflow bit): Zeigt Zahlenbereichsüberschreitung bei 2er-Komplement-Rechnung an (Übertrag der zweithöchsten Stelle der Operanden). Weitere mögliche Bezeichnungen sind neben V noch OF oder O.
   3) Nullbit (Z für engl. zero bit): Zeigt an, ob das Ergebnis der vorhergehenden Rechenoperation Null ist (wird 1, wenn Inhalt des Akkumulatorregisters 0 ist).

4) Negativbit (N): Wird gesetzt, wenn das Ergebnis der Operation als negative Zahl zu interpretieren sein kann (oberstes Bit des Akkumulatorregisters = 1).

f) Wie wird Gleitpunktarithmetik realisiert?

Software oder separater Gleitpunkt-Koprozessor (z.B. Intel 80486) oder in CPU interierte Hardwarelogik (Intel Pentium). GP-Artihmetik früher speziell für mathematische Probleme nötig, heute für alles Mögliche, z.B. Rendern von Zeichensätzen.

Früher schon bei 16 Bit CPUs 64 Bit FPUs.

Eine FP-Einheit kann neben Basisarithmetik: Logarithmus-, Wurzel-, Potenzrechnung und triginometrische Funktionen (teilw. mit Lookup-Tabellen).

Preise von Logik im Chip sanken, Preise von Verbindungen zwischen Chips nicht!

g) Welches sind die Teilschritte des Von-Neumann-Zyklus? Wie funktionieren die einzelnen Teilschritte?

1) FETCH – Befehlsabruf: Aus dem Speicher wird der nächste zu bearbeitende Befehl in das Befehlsregister geladen.
2) DECODE – Dekodierung: Befehlszähler wird vor dem Dekodieren um 1 erhöht. Der Befehl wird durch das Steuerwerk in Schaltinstruktionen für das Rechenwerk aufgelöst (übersetzt).
3) FETCH OPERANDS – Operandenabruf: Aus RAM oder ROM werden nun die Operanden geholt: die Werte, die durch den Befehl verändert werden sollen bzw. die als Parameter verwendet werden.
4) EXECUTE – Befehlsausführung: Die Operation wird vom Rechenwerk ausgeführt. An dieser Stelle wird, so vom Programm gewünscht, auch der Befehlszähler verändert (Sprungbefehl).
5) WRITE BACK – Rückschreiben der Daten: Die Ergebnisse der Berechnung werden in Register (oder auch Speicher) zurückgeschrieben, sofern notwendig.

h) Was ist der Von-Neumann-Flaschenhals?

Problem: CPU-Geschwindigkeit ist wesentlich schneller gestiegen als Bus-Geschwindigkeit. Bus wird zum Engpass des Rechners. Exklusive Busnutzung erzwingt Sequentialisierung von potentiell nebenläufig möglichen Aktionen.

Backus: “Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the von Neumann bottleneck. Not only is this tube a literal bottleneck for the data traffic of a problem, but, more importantly, it is an intellectual bottleneck that has kept us tied to word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand. Thus programming is basically planning and detailing the enormous traffic of words through the von Neumann bottleneck, and much of that traffic concerns not significant data itself, but where to find it.”

i) Was ist das Hauptmerkmal der Harvard-Architektur?

Separater Daten- und Befehlsspeicher

j) Welche Vor- und Nachteile hat die Harvard-Architektur gegenüber der Von-Neumann-Architektur?

VORTEILE: Besser parallelisierbare Teilbefehle, Read-Only-Schutz der Befehle: Klare Trennung von Daten und Befehlen erhöht Betriebssicherheit (Buffer-Overflow-Security-Aspekt)

NACHTEILE: ggf. einer der Speicher voll, der andere ziemlich leer

k) Was ist „OpCode-Prefetching“? Welcher Speicher wird dafür verwendet?

Bei modernen Prozessoren können mehrere Befehle aus dem Speicher in einen Zwischenspeicher (Prefetch-Registerblock) geladen werden, während der aktuelle Befehl noch decodiert wird. Dieses Verfahren wird als OpCode Prefetching (dt. Operationscode-Vorabruf) bezeichnet.

Vorteil: Deutliche Steigerung der Verarbeitungsgeschwindigkeit.

Nachteil: Bei Programmverzweigungen muss der Befehl evtl. wieder entfernt werden.

Vergleiche auch: Execution Trace Cache = Speicher für aus der Dekodierung berechnete Mikroinstruktionen

l) Denken Sie, dass auch ein Daten-Prefetching sinnvoll sein könnte? (Begründung)

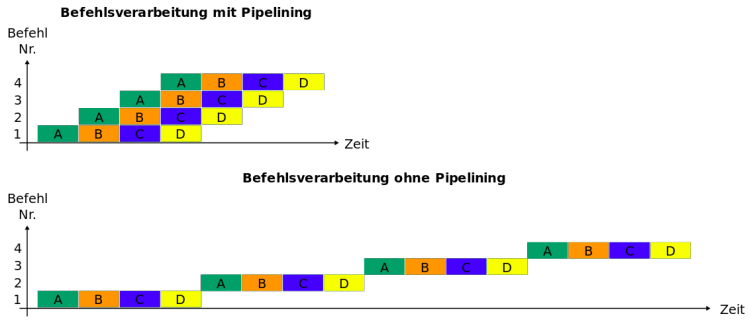
Ja. Es gibt viele typische Situationen, in denen Programme physikalisch hintereinander gespeicherte Daten nacheinander verarbeiten, z.B. die Elemente eines Arrays.

Aufgabe 2 (Pipelining)

Lesen Sie im deutschen Wikipedia folgende Artikel zum Pipelining: Pipeline (Prozessor) (http://de.wikipedia.org/wik
und Pipeline-Hazard (http://de.wikipedia.org/wiki/Pipeline-Hazard).

Hinweis: Die Erläuterungen zur geschickten Ausnutzung der Pipeline durch Software im Pipeline-Artikel sind nicht relevant.

a) Erklären Sie, wie eine 4-stufige Pipeline funktioniert.



b) Wieviele Pipeline-Stufen sind heute möglich?

31 Stufen bei Intel NetBURST (Sackgasse, zu teuer), 14 bei Intel Core

c) Welche Arten von Pipeline-Konflikten gibt es?

1) Ressourcenkonflikte, wenn eine Stufe der Pipeline Zugriff auf eine Ressource benötigt, die bereits von einer anderen Stufe belegt ist
2) Datenkonflikte
   auf Befehlsebene: Daten, die in einem Befehl benutzt werden, stehen nicht zur Verfügung
   auf Transferebene: Registerinhalte, die in einem Schritt benutzt werden, stehen nicht zur Verfügung
3) Kontrollflusskonflikte, wenn die Pipeline abwarten muss, ob ein bedingter Sprung ausgeführt wird oder nicht

d) Wie vermeidet man Ressourcenkonflikte?

Ressourcenkonflikte lassen sich durch Hinzufügen zusätzlicher Funktionseinheiten lösen.

Viele Datenkonflikte lassen sich durch Forwarding lösen, wobei Ergebnisse aus weiter hinten liegenden Pipeline-Stufen nach vorn transportiert werden, sobald diese verfügbar sind (und nicht erst am Ende der Pipeline).

Die Anzahl Kontrollflusskonflikte lässt sich durch eine Sprungvorhersage (engl. branch prediction) reduzieren. Hierbei wird spekulativ weitergerechnet, bis feststeht, ob sich die Vorhersage als richtig erwiesen hat. Im Falle einer falschen Sprungvorhersage müssen in der Zwischenzeit ausgeführte Befehle verworfen werden (pipeline flush), was besonders bei Architekturen mit langer Pipeline (wie etwa bei Intel Pentium 4 oder IBM Power5) viel Zeit kostet. Deshalb besitzen diese Architekturen sehr ausgeklügelte Techniken zur Sprungvorhersage, so dass die CPU nur in weniger als einem Prozent der stattfindenden Sprünge den Inhalt der Befehlspipeline verwerfen muss.

e) Geben Sie ein Beispiel für einen „Read after Write“-Pipeline-Konflikt an.

Read after Write (RAW) oder Echte Abhängigkeit:

Ein Operand wurde verändert und kurz darauf gelesen. Da der erste Befehl den Operanden evtl. noch nicht fertiggeschrieben hat (Pipeline-Stufe „store“ ist weit hinten), würde der zweite Befehl falsche Daten verwenden. Ein „Shortcut“ im Datenweg der Pipeline kann den Hazard vermeiden. Bei problematischeren Situationen, wenn beispielsweise ein Rechenergebnis zur Adressierung verwendet wird oder bei berechneten und bedingten Sprüngen, ist ein Anhalten der Pipeline aber unumgänglich.

Beispiel
R1 = R2+R3
R4 = R1+1

Anmerkung: Hyperthreading vs. Pipelining vs. Mehrkern-Multithreading