REALISIERUNG DES STREAMING - KONZEPTS

W.J.PAUL

I.Einleitung

Ein Computer besteht unter anderem aus Schaltwerk und Kernspeicher. Die Register des Schaltwerks arbeiten mit Flipflops,die viel schneller sind als die Ringkerne des Speichers.Durch dieses Mißverhältnis der Schaltzeiten entstehen im Schaltwerk Wartezeiten, in denen das Schaltwerk auf neue Daten aus dem Speicher wartet oder Daten nicht wegspeichern lassen kann, weil der Speicher noch mit dem vorhergehenden Lese- oder Schreibbefehl beschäftigt ist.Um dem abzuhelfen,unterteilt man den Speicher in mehrere selbständige Speicherwerke, in denen man gleichzeitig lesen oder schreiben kann (Speicherverschränkung).Bei einigen Listenoperationen ist es möglich, den Datenfluß zwischen Speicherwerken und Schaltwerk so zu organisieren, daß im Schaltwerk keine Wartezeiten entstehen.Hierbei werden Makrobefehle auf Hardwareebene realisiert.Das ist das Streamingkonzept.

Wir testen dieses Konzept an der folgenden charakteristischen Aufgabe: Gesucht ist die Realisierung eines Maschinenbefehls,der folgendes tut:

Im Speicher steht auf N aufeinanderfolgenden Speicherplätzen die Liste $A = A_1,\ldots,A_N$, auf anderen N aufeinanderfolgenden Plätzen eine Liste $B = B_1,\ldots,B_N$ .Für eine Liste $C = C_1,\ldots,C_N$ sind weitere N aufeinanderfolgende Plätze reserviert.Wir schreiben dem Schaltwerk die Adressen von A_1,B_1,C_1 in die Indexregister I_1,I_2,I_3 sowie die Listenlänge N in ein spezielles Register.

Dann soll eine Liste $A_1*B_1,\ldots,A_N*B_N$ berechnet und auf den Plätzen $C_1,\ldots,C_N$ abgespeichert werden. $*$ ist eine binäre Operation, die im Rechenwerk ausgeführt werden kann und deren Ausführungszeit von den Operanden abhängt.Im Rechenwerk dürfen hierbei vom Beginn der Berechnung von C_1 bis zum Ende der Rechnung für C_N keine Wartezeiten entstehen.

Wir werden folgendes ausrechnen:

i) Minimale Anzahl von Speicherwerken

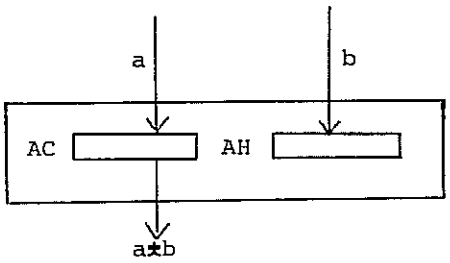
ii) Obere und untere Schranken für den Bedarf an Pufferspeicher zwischen Speicher und Rechenwerk

iii) Aufwand und zeitlicher Gewinnfaktor

II. Modelle für Rechenwerk und Speicher

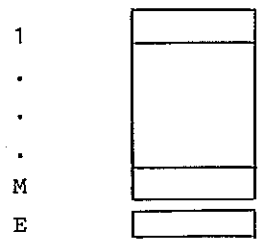
Wir betrachten das Rechenwerk als Verbraucher und Lieferanten von Daten.In den Akkumulator AC und den Hilfsakkumulator AH werden

Daten a und b geschrieben, und eine Rechnung, die AC:=AC*AH bewirkt,wird gestartet.Nach einer Anzahl von Schaltwerktakten t(a,b) steht im Akkumulator a*b.Wir bezeichnen die Mindestzeit zur Berechnung eines Elements C_i mit $tmin := \min_{a,b}\{t(a,b)\}$



Wir bemerken: wenn wir unseren Makrobefehl für die Operation * realisieren können,dann können wir ihn auch mit einer binären Operation als Parameter realisieren,sofern nur die minimale Ausführungszeit der zulässigen Operationen $\geq$ tmin ist.

Ein Speicherwerk betrachten wir als Folge von M adressierbaren Zellen mit langer Zugriffszeit und ein schnelles Ein-Ausgaberegister E,das so schnell ist wie die Register des Schaltwerks.Ergeht an das Speicherwerk ein Lese- oder Schreibbefehl,wird ein Taktimpuls angeworfen,mit dem der Befehl ausgeführt wird.Nach einer Periode des Taktimpulses ist der Befehl ausgeführt.Diese Speicherzykluszeit ist ein Vielfaches eines Schaltwerktakts.Wir nennen dieses Vielfache F.



(F $\approx$ 1o bei heutigen Maschinen).Wir setzen $q := \lceil F/tmin \rceil$;das ist die maximale Anzahl von Elementen C_i,die während eines Speicherzyklus berechnet werden kann.

III. Minimaler Bedarf an Speicherwerken

Satz 1: Der minimale Bedarf an Speicherwerken zur Lösung unserer Aufgabe ist 3q.

Beweis über zwei Lemmata:

Lemma 1: Man braucht mindestens 3q Speicherwerke

Beweis: Man wähle a,b mit t(a,b)=tmin und setze A=a,a,... ,B=b,b,... mit genügend großer Listenlänge N:Alle F Takte verbraucht das Rechenwerk 2q Daten und liefert q Daten zum wegspeichern.Da ein Speicherwerk in F Takten nur ein Datum liefern oder wegspeichern kann,braucht man mindestens 3q Speicherwerke.

Lemma 2: Man kommt mit 3q Speicherwerken aus

Beweis: Man muß eine Realisierung mit 3q Speicherwerken angeben.

Wir unterteilen den Speicher in 3 Gruppen zu je q Speicherwerken.Die Adresse einer Zelle im Gesamtspeicher ist gegeben durch ein Tripel (g,s,a) mit:

g: Speichergruppe, $g \in \{1,2,3\}$

s: Speicherwerk in der Speichergruppe, $s \in [1:q]$

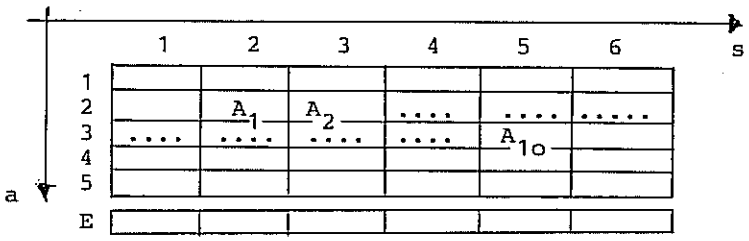
a: Adresse der Zelle im Speicherwerk, $a \in [1:M]$

Wir erklären die Nachfolgeradresse

$$N(g,s,a) = \begin{cases} (g,s+1,a) & \text{für } s \neq q \\ (g,1,a+1) & \text{für } s = q \end{cases}$$

Durch diese Anordnung der Adressen kann stets auf q aufeinanderfolgende Zellen gleichzeitig zugegriffen werden.

Beispiel einer Liste,die in einer Speichergruppe abgespeichert ist:



M=5,q=6,N=1o

Bezeichnung: Ein Indexregister I_j enthalte die Adresse (g,s,a).(g,s) bezeichnet ein Speicherwerk, a eine Adresse darin.Wir bezeichnen mit $E(I_j)$ das E-Register des Speicherwerks (g,s).

Folgende Indexregisterbefehle werden benutzt:

$I_j := N(I_j)$:Nachfolgeradresse ins Indexregister schreiben

Lesen(I_j) :$E(I_j)$:=Inhalt der Speicherzelle a im Speicherwerk (g,s)

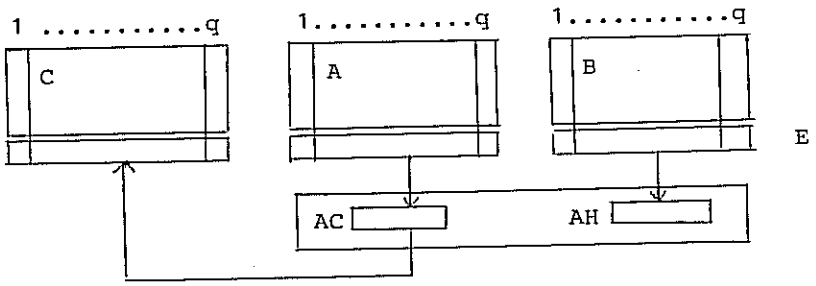
Schreiben(I_j):Inhalt der Zelle a im Speicherwerk (g,s):=$E(I_j)$

Die beiden letzten Befehle sind erst F Takte,nachdem sie gegeben wurden, ausgeführt.

In einer Speichergruppe stehe nun Liste A.Die Daten aus den E-Registern dieser Speichergruppe werden von AC werbraucht.In einer anderen Speichergruppe steht Liste B.Die E-Register dieser Gruppe geben ihre Daten an AH weiter.Liste C wird in der dritten Gruppe abgespeichert werden.In die E-Register dieser Gruppe werden die berechneten Daten C_i aus AC geschrieben.

Die Listen sind also auf die drei Speichergruppen verteilt.Um das zu erreichen,muß evtl. Liste A oder B umgespeichert werden.Man vermeidet

aber hierdurch,daß an ein Speicherwerk gleichzeitig zwei Lese- oder Schreibbefehle gegeben werden,die verschiedene Listen betreffen.



Wir realisieren unseren Makrobefehl nach dem Flußdiagramm auf der nächsten Seite.Was in einem stark eingerahmten Kästchen steht,soll während eines Schaltwerktakts ausgeführt werden.Es ist hier zu beachten, daß es sich um ein Flußdiagramm für Hardware handelt,wo speziell folgendes möglich ist:

i) Parallelarbeit
ii) einen Registerinhalt ändern und gleichzeitig den alten Registerinhalt weitergeben (so etwas geschieht etwa beim shiften)

Man wird erkennen: zwischen zwei Lese- oder Schreibbefehlen an das gleiche Speicherwerk werden stets q Daten C_i berechnet.Das dauert $\geq$ q.tmin $\geq$F Takte.Ein Speicherwerk ist also stets mit dem vorigen Befehl fertig,wenn ein neuer Befehl an es gegeben wird.Damit ist Satz 1 bewiesen.

Die Kosten zur Realisierung des Flußdiagramms sind etwa die des Festpunktmultiplizierwerks aus [I] ,Kapitel 4.3. Hinzu kommen die Kosten für das Aufspalten des Speichers in 3q kleine Speicherwerke.

Es wurden keine zusätzlichen Register zum Zwischenspeichern von Daten benötigt.

IV. Bedarf an Pufferspeicher

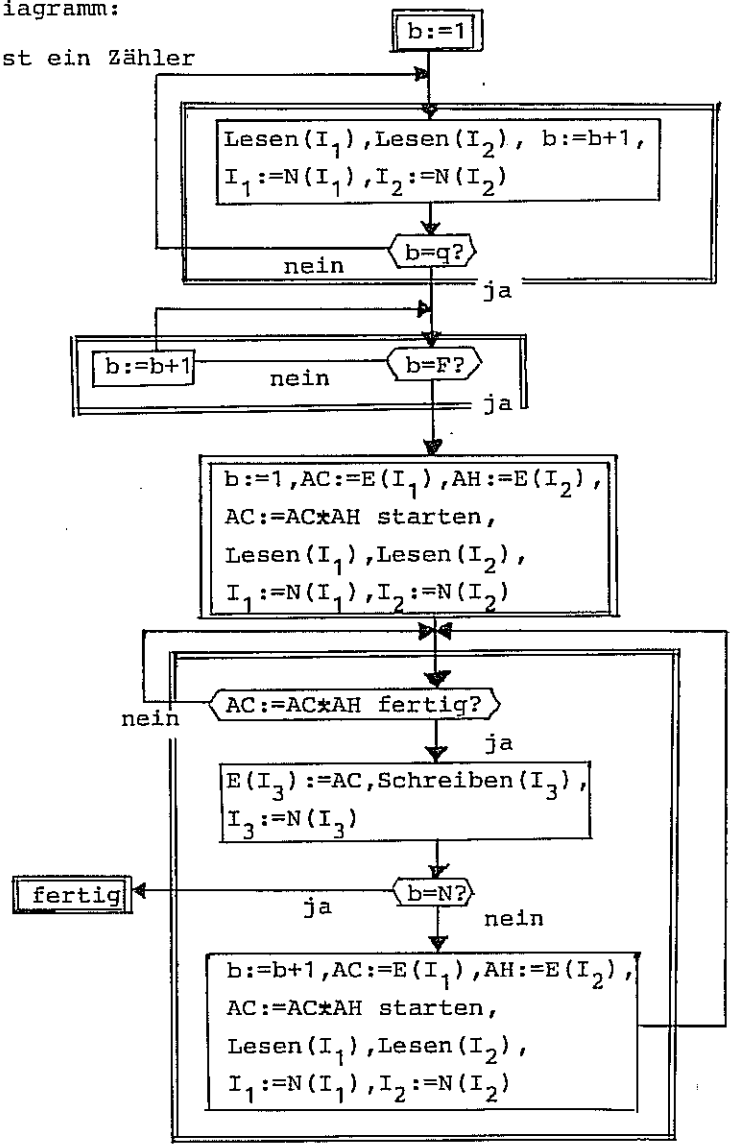
Hat man statt den vorne beschriebenen Speicherwerken,deren Speicherzyklus nach Belieben gestartet werden kann,Speicherwerke mit stehendem Speichertakt,wo ein Befehlszyklus nur alle F Takte beginnen kann - selbst wenn das Speicherwerk gerade nicht arbeitet -,dann läßt sich folgendes zeigen:

Satz 1: Die minimale Anzahl von Speicherwerken zur Lösung unserer Aufgabe ist 3q

Um mit 3q von diesen Speicherwerken unseren Makrobefehl zu realisieren, braucht man allerdings schnelle Register zum Zwischenspeichern von

Flußdiagramm:

b ist ein Zähler



Daten(schnell ist wieder zu verstehen als:so schnell wie die Register des Schaltwerks).

Satz 2: i)man löst die obige Aufgabe mit 3q Speicherwerken mit stehendem Speichertakt
ii)man berechnet die Elemente C_i in der Reihenfolge ihrer Indizes
dann braucht man mindestens q schnelle Register zum Zwischenspeichern von Daten

Satz 3: i)die Voraussetzungen von Satz 2 gelten
ii)der Speicher ist in 3 Speichergruppen zu 3q Speicherwerken aufgeteilt,auf welche die Listen A,B,C verteilt sind
iii) die Register zum Zwischenspeichern sind in drei kleine Pufferspeicher unterteilt: einer speichert nur Elemente A_i,ein zweiter nur Elemente B_i und der dritte nur Elemente C_i(das ist am einfachsten zu organisieren)
dann ist die minimale Anzahl von schnellen Registern zum Zwischenspeichern von Daten 2q

V.Zeitgewinn

Wir erklären die Schnelligkeit einer Maschine durch die Anzahl von Takten,die im Mittel benötigt wird,um einen Befehl des Maschinencodes abzuarbeiten.Diese Anzahl nennen wir t.Sei

t_m die mittlere Anzahl von Takten,die zur Ausführung eines Makrobefehls benötigt wird

t_n die mittlere Anzahl von Takten,die zur Ausführung eines nicht-Makrobefehls benötigt wird

g die relative Häufigkeit von Makrobefehlen im Programm

so ergibt sich:

$$t = g.t_m + (1-g)t_n \approx g.t_m + t_n \quad \text{da } g \ll 1$$

Wir berechnen t für eine Maschine E mit nur einem Speicherwerk und für eine Maschine S' mit verschränktem Speicher,die diesen aber nur bei der Ausführung von Makrobefehlen der Art C:=A*B ausnutzt.

Wir machen dazu folgende Hypothesen:

i) t_n=2F für E und S'

ii) die Rechnung für AC:=AC*AH dauert stets F Takte

Bemerkungen:

ad i) um ein Befehlswort abzuarbeiten,muß i.A. zwei mal auf den Speicher zugegriffen werden: einmal,um das Befehlswort ins Befehlsregister zu holen;dann,um ein Datum aus dem Speicher zu holen

oder hineinzuschreiben,das durch das Befehlswort bezeichnet wird.Das dauert 2F Takte.Zwischendurch können anfallende Rechnungen stattfinden.

ad ii) Unter * werden hauptsächlich arithmetische Operationen vorkommen.Die Ausführung arithmetischer Operationen dauert bei modernen Maschinen etwa so lang wie ein Speicherzyklus.

Berechnung von t_m für S':

Es müssen $I_1, I_2, I_3, N, *$ vorbesetzt werden.	1oF Takte
Dazu sind 5 Befehle notwendig	
Berechnung von Liste C	N.F Takte
Vorher vergehen F Takte,bis A_1 und B_1 verfügbar sind.Nachher muß man F Takte warten, bis C_N weggespeichert ist.	2F Takte
Die Wahrscheinlichkeit,daß Liste A und B in der gleichen Speichergruppe stehen,ist 1/3.Also muß bei jedem dritten Makrobefehl zusätzlich eine Liste umgespeichert werden.Das kann man auf Hardwareebene wieder als Makrobefehl realisieren.Jedes Datum,das umgespeichert wird,bleibt nur einen Takt in AC (tmin=1,q=F).	1/3(N+2F) Takte

Insgesamt ergibt sich bei mittlerer Listenlänge $\overline{N}$:

$$t_{mS'} = 12{,}7F + \overline{N}(F+1/3)$$

Berechnung von t_m für E:

Simuliert man C:=A*B auf einer Einadreßmaschine,wird im Mittel $\overline{N}$ mal folgende Befehlsfolge durchlaufen:

1) Befehl $AC:=A_i$ lesen und ausführen	2F Takte
2) Befehl $AC:=AC*B_i$ lesen und ausführen	2F Takte
3) Befehl AC wegspeichern lesen und ausführen	2F Takte
4) Befehl,abzufragen,ob man fertig ist,lesen	F Takte
5) Abfragen,ob man fertig ist.Das kann man mit Indexregistern machen.Kein Speicherzugriff nötig	-
6) drei Befehle lesen,Adressen zu erhöhen,um richtig auf die nächsten Listenelemente zuzugreifen bzw. richrig abzuspeichern	3F Takte
7) die drei Adressen erhöhen;mir Indexregistern	-
8) Befehl lesen,den in 5) abgefragten Zähler zu andern	F Takte
9) Zähler ändern (Indexregister)	-

Insgesamt: $t_{mE} = 11F\overline{N}$

Man erhält einen Gewinnfaktor

$$G_{S'/E}=\frac{t_E}{t_{S'}} = \frac{11gF\overline{N} + 2F}{g((F+1/3)\overline{N} + 12,7F) + 2F}$$

Sei V eine schnellere Maschine als E mit verschränktem Speicher,so daß $t_{nV} = a.t_{nE} = 2aF$ mit $a<1$.Die Größe von a hängt davon ab,wie gut der verschränkte Speicher ausgenutzt ist.Sei weiter S eine Maschine,bei der das Streamingkonzept realisiert ist und die außerdem mit ihrem verschränkten Speicher den gleichen Zeitgewinn erzielt wie V,d.h. $t_{nS}=2aF$.

Dann erhält man $t_V = a.t_E$

$$t_{mS} = g((F+1/3)N + 2,7F + 1oaF) + 2aF$$

$$\boxed{G_{S/V}= \frac{11gFN + 2F}{g((F+1/3)N/a + 2,7F/a + 1oF) + 2F}}$$

Beispiel: $a=3/4$, $g=1/2o$, $N=1o$, $F=1o$; dann ist $G_{S/V}=2,2$; $G_{S'/E}=2,4$

Beim Streaming wurde der verschränkte Speicher dazu benutzt,Makrobefehle auf Hardwareebene so zu realisieren,daß im Rechenwerk keine Wartezeiten auftraten.Als Alternative A böte sich an,die Makrobefehle zwar auf Hardwareebene zu verdrahten,womit man den Zeitverlust bei der Schleifenverwaltung umgeht, und die Forderung,daß keine Wartezeiten entstehen, fallenzulassen.Zur Berechnung eines Elements C_i wären 3 Speicherzugriffe nötig,weitere F Takte dauert die eigentliche Rechnung AC:=AC*AH.Man kommt also auf $t_{mA}= 4NF + 1oF$,denn man hat ja noch $I_1,\ldots,N$ vorzubesetzen.

Es ist $$G_{A/E}= \frac{11gFN + 2F}{g(4NF + 1oF) + 2F}$$

Beispiel mit den obigen Zahlenwerten: $G_{A/E}= 1,5$

Die Organisation hiervon kostet etwa die Hälfte des vorne zitierten Festpunktmultiplizierwerks.Außerdem muß man den Speicher nicht in so viele Speicherwerke aufteilen,wie das beim Streaming nötig ist.

VI.Literatur

[I] G.Hotz:
Informatik:Rechenanlagen
Teubner Studienbücher Informatik

[II] W.J.Paul:
Verschiedene Realisierungen des Streamingkonzepts und damit verbundene Optimierungsfragen; Diplomarbeit am Institut für Informatik der Universität des Saarlandes