

Inhaltsverzeichnis

14. Systolische Systeme (Autor: E. Zehendner)	703
14.1. Grundzüge der Systolik	703
14.1.1. Ein einführendes Beispiel: Matrixmultiplikation	704
14.1.2. Problemparameter und Feldparameter	706
14.1.3. Raumkoordinaten	706
14.1.4. Serialisierung generischer Operatoren	707
14.1.5. Zuweisungsfreie Notation	708
14.1.6. Elementare Berechnungen	709
14.1.7. Diskrete Zeitschritte	710
14.1.8. Äußere und innere Kommunikation	710
14.1.9. Fließbandverarbeitung	713
14.2. Raum-Zeit-Abbildung und systolisches Feld	714
14.2.1. Beispiel: Matrixprodukt ohne stationäre Variablen	714
14.2.2. Die Raum-Zeit-Abbildung als globale Sichtweise	715
14.2.3. Symbolische Bestimmung der Raumkoordinaten	717
14.2.4. Symbolische Berechnung der Gesamtausführungsdauer	719
14.2.5. Ableitung der Verbindungsstruktur	720
14.2.6. Bestimmung der Zellstruktur	721
14.3. Herleitung von Ein-/Ausgabeschemata	723
14.3.1. Von Datenstrukturindizes zu Iterationsvektoren	724
14.3.2. Momentaufnahmen von Datenstrukturen	725
14.3.3. Montieren der Ein-/Ausgabeschemata	726
14.3.4. Raum-Zeit-Abbildung induziert Datenraten	727
14.3.5. Ein-/Ausgabeexpansion und erweitertes Ein-/Ausgabeschema	728
14.3.6. Behandlung stationärer Variablen	728
14.3.7. Verschränkung von Berechnungen	729
14.4. Steuerungsaspekte	731
14.4.1. Steuerungslose Zellen	732
14.4.2. Zellen mit Globalsteuerung	732
14.4.3. Lokale Steuerung	732
14.4.4. Verteilte Steuerung	736
14.4.5. Das Zellprogramm als lokale Sichtweise	739
14.5. Lineare systolische Felder	743

14.5.1. Matrix-Vektor-Multiplikation	744
14.5.2. Sortieren	744
14.5.3. Lineares Gleichungssystem mit unterer Dreiecksmatrix	746
Literaturverzeichnis	749
Nameverzeichnis	750
Index	751

14. Systolische Systeme (Autor: E. Zehendner)

Systolische Felder — die wohl perfektteste Form von Spezialrechnern — sind in ihrer einfachsten Ausprägung nur zur wiederholten Ausführung einer einzigen Rechenoperation fähig. Dennoch besitzen sie eine Fülle praxisrelevanter Anwendungen, hauptsächlich in Gebieten iterativer Verfahren wie der numerischen Mathematik, der kombinatorischen Optimierung, der linearen Algebra, der algorithmischen Graphentheorie, der Bild- und Signalverarbeitung, der Sprach- und Textverarbeitung, und so weiter.

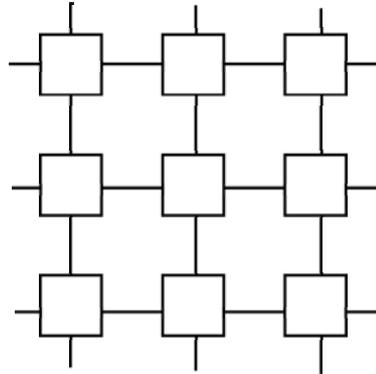
Ein systolisches Feld kann nämlich der Struktur dieses einen auszuführenden Algorithmus statisch so exakt angepasst werden, dass Ort und Zeit der Ausführung jeder Operation ein für alle Mal festliegt, miteinander kommunizierende Zellen direkt und permanent verbunden sind, das Schalten von Verbindungswegen überflüssig wird: Der Algorithmus ist direkt in Hardware gegossen. Die systolischen Algorithmen werden deshalb in diesem Zusammenhang auch als „Hardware-Algorithmen“ bezeichnet.

Der Begriff „systolische Algorithmen“ meint also nicht eine Menge konkreter Algorithmen zur Lösung eines bestimmten, fest eingegrenzten Berechnungsproblems (wie etwa „Sortieralgorithmen“), sondern stellt vielmehr einen besonderen Spezifikations-, Programmier-, Berechnungsstil dar. Algorithmen aus vielen verschiedenen Anwendungsgebieten können von der Art her „systolisch“ sein, ohne dass sich alle bekannten Algorithmen eines solchen Gebiets schon deshalb in eine zur systolischen Abarbeitung geeignete Form bringen lassen.

Es liegt daher nicht in der Absicht dieses Kapitels, „die“ systolischen Algorithmen oder auch nur die „wichtigsten“ systolischen Algorithmen vorzustellen. Stattdessen soll an Hand weniger, einfacher, aber typischer Beispiele die Grundlage für das allgemeine Verständnis systolischer Algorithmen gelegt werden.

14.1. Grundzüge der Systolik

Die Bezeichnung *systolisch* leitet sich aus dem Operationsprinzip der systolischen Architektur ab. Unter einer systolischen Arbeitsweise ist die intensive gemeinsame Anwendung von Pipelining und Parallelität zu verstehen, gesteuert von einem glo-



14.1. Abbildung. Rechteckiges systolisches Feld für Matrixmultiplikation: (a) Feldaufbau mit Dateneingabeschema; (b) Zellstruktur.

balen und völlig synchronen Takt, wodurch sich Ströme von Daten ergeben, die rhythmisch durch das Verbindungsnetzwerk pulsieren, gerade so wie Ströme von Blut vom menschlichen Herzen durch die Adern des Körpers getrieben werden. Pipelining findet dabei nicht nur in einer einzigen Richtung statt, sondern betrifft alle, sich in verschiedene Richtungen bewegend, sich in den Zellen des systolischen Feldes kreuzende Datenströme.

Ein *systolisches System* besteht typischerweise aus einem *Wirtsrechner* und dem eigentlichen systolischen Feld. Der Wirtsrechner ist konzeptionell von untergeordneter Bedeutung; er dient nur der Ansteuerung des systolischen Feldes sowie seiner Versorgung mit Daten. Das *systolische Feld* kann verstanden werden als spezialisiertes Netzwerk von Zellen, die mit Hilfe massiver Parallelität datenintensive Berechnungen mit hoher Geschwindigkeit durchführen. Das von den Zellen eines systolischen Feldes gemeinsam ausgeführte Programm stellt den *systolischen Algorithmus* dar.

So verschieden systolische Felder auch sein mögen, sie teilen doch meist eine Vielzahl von gemeinsamen Eigenschaften: Das diskrete Zeitschema, die synchrone Arbeitsweise, den regulären (häufig zweidimensionalen) geometrischen Aufbau, Kommunikation nur zwischen unmittelbar benachbarten Zellen sowie die Verwendung einfachster Steuerungsmechanismen.

In diesem Abschnitt sollen zunächst grundlegende Phänomene im Zusammenhang mit systolischen Feldern an einem durchgängigen Beispiel eingeführt und erläutert werden. Zu einem Berechnungsproblem gibt es häufig viele Lösungen durch verschiedene Ausprägungen von systolischen Feldern, deren beste meist sehr komplex sind. Für die folgenden Ausführungen steht jedoch nicht die Güte der gezeigten Variante im Vordergrund, sondern die Möglichkeit, wichtige Konzepte kompakt und intuitiv zu präsentieren.

14.1.1. Ein einführendes Beispiel: Matrixmultiplikation

Abbildung 14.1 zeigt ein aus 15 Zellen bestehendes, rechteckiges *systolisches Feld* zur Multiplikation einer $(3 \times N)$ -Matrix A mit einer $(N \times 5)$ -Matrix B . Der Parameter N

spielt für den *Aufbau* des systolischen Feldes in Abbildung 14.1 keine Rolle, wohl aber für das *Dateneingabeschema* sowie die *Gesamtausführungsdauer* des Algorithmus.

Das abgebildete Dateneingabeschema beruht auf der speziellen Wahl $N = 4$. Abbildung 14.1 zeigt also die Lösung des folgenden konkreten Problems:

$$A \cdot B = C$$

mit

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix},$$

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \end{pmatrix},$$

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \end{pmatrix},$$

wobei

$$c_{ij} = \sum_{k=1}^4 a_{ik} \times b_{kj} \quad 1 \leq i \leq 3, 1 \leq j \leq 5$$

Die Zellen des systolischen Feldes können untereinander Daten über *Verbindungen* austauschen, die in Abbildung 14.1(a) als Pfeile zwischen den Zellen eingezeichnet sind. *Randzellen* des systolischen Feldes können darüber hinaus mit der *Außenwelt* kommunizieren. Alle Zellen des systolischen Feldes verfügen über das gleiche Verbindungsmuster zur Kommunikation mit ihrer Umgebung. Durch den vollständig regelmäßigen Aufbau des systolischen Feldes (Platzierung der Zellen und Struktur der Verbindungen) ergeben sich *reguläre Datenflüsse* längs der verschiedenen Verbindungsachsen.

In Abbildung 14.1(b) ist die *interne Struktur einer Zelle* dargestellt. Zu sehen sind ein *Multiplizierwerk*, ein *Addierwerk*, drei *Register* und vier *Kanäle* sowie Leitungen zwischen diesen Einheiten. Alle Zellen besitzen dabei denselben Aufbau.

Jedes der Register RA, RB, RC kann einen Zahlenwert speichern. Die Bezeichnungen der Register sind hier suggestiv gewählt, im Prinzip aber beliebig. Die Register RA und RB erhalten ihre Werte von *Eingangskanälen*, in Abbildung 14.1(b) dargestellt durch kleine Kreise am linken beziehungsweise oberen Rand der Zelle. Ein solcher Kanal stellt eine Schnittstelle zur entsprechenden, von außen an die Zelle angeschlossenen Verbindung dar.

Die aktuellen Werte der Register RA und RB werden als Operanden des Multiplizierwerks verwendet und gleichzeitig über *Ausgangskanäle* der Zelle weitergegeben (Kreise am rechten beziehungsweise unteren Rand). Das Ergebnis der Multiplikation wird dem Addierwerk zugeführt, dessen zweiter Operand aus dem Register RC stammt. Das Ergebnis der Addition überschreibt anschließend den bisherigen Wert von RC.

14.1.2. Problemparameter und Feldparameter

Die 15 Zellen des systolischen Feldes sind als rechteckiges Schema von drei Zeilen mal fünf Spalten organisiert (genau wie die Matrix C). Diese Dimensionen entsprechen direkt der Anzahl der Zeilen der Matrix A beziehungsweise der Anzahl der Spalten der Matrix B . Die *Größe des systolischen Feldes* ist also hier auf die *Größe der Datenstrukturen* des zu lösenden Problems *abgestimmt*. Müssten wir im allgemeinen Fall eine $(N_1 \times N_3)$ -Matrix A mit einer $(N_3 \times N_2)$ -Matrix B multiplizieren, so würden wir ein systolisches Feld mit N_1 Zeilen und N_2 Spalten benötigen.

Der in Abbildung 14.1 gezeigte Aufbau würde allerdings auch zulassen, ein systolisches Feld mit mehr als N_1 Zeilen oder mehr als N_2 Spalten zu benutzen. Dies ist für den Fall wichtig, dass wir ein *fest dimensioniertes* systolisches Feld zur Multiplikation von Matrizen variierender Dimension heranziehen wollen. Wir würden dann nur den jeweils benötigten, aus N_1 Zeilen und N_2 Spalten bestehenden, rechteckigen Ausschnitt des gesamten Feldes nutzen, der sich zum Beispiel ganz links oben in der Ecke der Feldes befindet. Die übrigen Zellen würden zwar ebenfalls arbeiten, jedoch keinen Beitrag zur Lösung des Gesamtproblems erbringen (andererseits aber auch keinen Schaden anrichten).

Die Größen N_1, N_2, N_3 stellen Parameter des zu lösenden Problems dar, weil die Anzahl der durchzuführenden Operationen von jedem von ihnen abhängt; sie sind also *Problemparameter*. Dagegen hängen Dimension und Aufbau des systolischen Feldes nur von den Größen N_1 und N_2 ab, die damit auch zu *Feldparametern* werden.

14.1. Bemerkung. Für die Matrixmultiplikation werden wir in Abschnitt 14.2 ein weiteres systolisches Feld kennenlernen, das in seinen Abmessungen von allen drei Problemparametern N_1, N_2, N_3 abhängt.

14.1.3. Raumkoordinaten

Wir wollen nun jeder Zelle eines systolischen Feldes eindeutige *Raumkoordinaten* zuordnen, als eine Charakterisierung der geometrischen Lage der Zelle relativ zum gesamten Feld. In einem rechteckigen systolischen Feld können wir zum Beispiel einfach die jeweiligen Zeilen- und Spaltennummern hierfür verwenden. Die mit c_{11} beschriftete Zelle in Abbildung 14.1 bekäme so die Koordinaten $(1, 1)$, die mit c_{12} beschriftete die Koordinaten $(1, 2)$, die mit c_{21} beschriftete die Koordinaten $(2, 1)$, und so weiter. Wir wollen die so gebildeten Raumkoordinaten als für den Rest dieses Abschnitts gegeben voraussetzen.

Im Prinzip ist es egal, wo sich dabei der Koordinatenursprung befindet, in welche Richtung die Koordinatenachsen gelegt werden, welche Raumrichtung der ersten Koordinate entspricht und welche der zweiten. In der gerade für das Beispiel vorgestellten Festlegung wurde die Aufzählung wie bei der Benennung von Matrixkomponenten gewählt, es steht also die erste Koordinate für die von oben nach unten ab der Position 1 nummerierten Zeilen, die zweite Komponente für die von links nach rechts ab der Position 1 nummerierten Spalten.

Wir hätten natürlich auch eine ganz andere Wahl für das Koordinatensystem treffen können. Aber das vorgestellte Schema passt einfach ideal zum vorliegenden systolischen Feld: Die Indizes des in einer Zelle berechneten Matrixelements c_{ij}

stimmen genau mit den Koordinaten dieser Zelle überein. Die eingegebenen Zeilen der Matrix A tragen dieselbe Nummer wie die erste Koordinate der Zellen, die sie durchlaufen; entsprechendes gilt für die zweite Koordinate bezüglich der Spalten der Matrix B . Sämtliche Verbindungen (und damit alle darüber realisierten Datenflüsse) sind achsenparallel und in Richtung aufsteigender Koordinaten orientiert.

Nicht immer ist es so klar, wie aussagefähige Raumkoordinaten bestimmt werden können; als Beispiel sei hier auf das systolische Feld aus Abbildung 14.3(a) verwiesen. Aber wie auch immer das Koordinatensystem gewählt wird: Es ist wichtig, dass sich der regelmäßige Aufbau des systolischen Feldes in den jeweiligen Koordinaten der Zellen augenfällig widerspiegelt. Deshalb werden eigentlich stets ganzzahlige Koordinaten benutzt. Die Koordinaten von Zellen mit minimalem euklidischen Abstand sollten sich darüber hinaus nur in einer Komponente und dort nur um den Wert 1 unterscheiden.

14.1.4. Serialisierung generischer Operatoren

In jeder aktiven Zelle (i, j) aus Abbildung 14.1 wird genau das Element c_{ij} der Ergebnismatrix C berechnet. Hierfür muss die Zelle das *Skalarprodukt*

$$\sum_{k=1}^4 a_{ik} \times b_{kj}$$

auswerten. Dies geschieht iterativ: In jedem Schritt wird ein Produkt $a_{ik} \times b_{kj}$ ermittelt und auf die bisherige Teilsumme für c_{ij} aufaddiert. Die Teilsumme muss natürlich zu Beginn der Berechnung auf Null (oder, falls dies gewünscht wird, auf einen anderen definierten Startwert) gesetzt werden. In Anlehnung an die klassische Notation imperativer Programmiersprachen würde sich die allgemeine Vorgehensweise damit etwa folgendermaßen beschreiben lassen:

MATRIXPRODUKT(N_1, N_2, N_3)

```

1  for  $i \leftarrow 1$  to  $N_1$ 
2    do for  $j \leftarrow 1$  to  $N_2$ 
3      do  $c(i, j) \leftarrow 0$ 
4        for  $k \leftarrow 1$  to  $N_3$ 
5          do  $c(i, j) \leftarrow c(i, j) + a(i, k) * b(k, j)$ 
```

Der Summenoperator \sum gehört zu den sogenannten *generischen Operatoren*, mit denen eine beliebige Anzahl von Operanden verknüpft werden kann. In dem systolischen Feld aus Abbildung 14.1 werden zwar alle zu einer bestimmten Summe gehörenden Additionen in derselben Zelle durchgeführt. Jedoch gibt es genügend andere Beispiele, bei denen die Einzeloperationen eines generischen Operators auf verschiedene Zellen verteilt sind (siehe etwa das systolische Feld in Abbildung 14.3).

14.2. Bemerkung. Weitere Beispiele für generische Operatoren wären: Produkt, Minimum, Maximum sowie die Booleschen Verknüpfungen Und, Oder und Exklusiv-Oder.

Generische Operatoren müssen daher im Allgemeinen *serialisiert* werden, bevor die auszuführenden Berechnungen den Zellen des systolischen Feldes zugeordnet werden können. Sie sind damit grundsätzlich anders zu behandeln als gewöhnliche Operatoren mit fester Stelligkeit, etwa die zweistellige Addition, da es bei der Durchführung dieser Aufteilung gewisse Freiheitsgrade gibt.

14.1.5. Zuweisungsfreie Notation

Zur Beschreibung von systolischen Programmen verwenden wir statt einer imperativen Form wie in Algorithmus [Matrixprodukt] besser eine *zuweisungsfreie Notation*, die auf einem *Gleichungskalkül* beruht. Dadurch vermeiden wir Seiteneffekte und können Parallelität direkt ausdrücken. Im vorliegenden Fall stört das Überschreiben der Programmvariablen $c(i, j)$. Diese ersetzen wir daher durch eine Folge von *Instanzen* $c(i, j, k)$, die für die nacheinander angenommenen Zustände der Programmvariablen $c(i, j)$ aus Algorithmus [Matrixprodukt] stehen. Es entsteht hierdurch eine sogenannte *Rekurrenzgleichung*. Wir können die allgemeine Matrixmultiplikation aus Algorithmus [Matrixprodukt] damit zum Beispiel durch folgende Ausdrücke zuweisungsfrei formulieren:

Eingabeoperationen

$$c(i, j, 0) = 0 \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2$$

Berechnungen

$$c(i, j, k) = c(i, j, k-1) + a(i, k) * b(k, j) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \quad (14.1)$$

Ausgabeoperationen

$$c_{ij} = c(i, j, N_3) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2$$

Das System (14.1) beschreibt explizit die Feinstruktur des ausgeführten systolischen Algorithmus. Mit der oberen Gleichung werden alle *Eingabedaten* beschrieben, mit der unteren alle *Ausgabedaten*; im systolischen Feld sind für diese Gleichungen keine Berechnungen, sondern nur Ein-/Ausgabeoperationen auszuführen. In der mittleren Gleichung sind die eigentlichen Berechnungen aufgeführt.

Zu jeder Gleichung des Systems gehört, rechts davon notiert, eine *Quantifizierung*, in der die Menge der abzuarbeitenden Belegungen der *Iterationsvariablen* i und j (sowie für die mittlere Gleichung auch k) angegeben wird; jede solche Menge wird eine *Trägermenge* genannt. Die Iterationsvariablen i, j, k der mittleren Gleichung können zu einem *Iterationsvektor* (i, j, k) zusammengefasst werden. Für die Ein-/Ausgaben hätte der Iterationsvektor zunächst nur die Komponenten i und j . Um eine geschlossene Darstellung zu erhalten, können wir diesen Vektor um eine dritte Komponente k erweitern, die aber einen festen Wert trägt. Eingaben sind dann durch $k = 0$ gekennzeichnet, Ausgaben durch $k = N_3$. Es ergibt sich damit:

Eingabeoperationen

$$c(i, j, k) = 0 \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0$$

Berechnungen

$$c(i, j, k) = c(i, j, k-1) + a(i, k) * b(k, j) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \quad (14.2)$$

Ausgabeoperationen

$$c_{ij} = c(i, j, k) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = N_3$$

Zu beachten ist, dass die Trägermengen für die Ein-/Ausgaben nun zwar ebenfalls formal die Dimension 3 besitzen, im geläufigen geometrischen Sinne faktisch aber nur zweidimensional sind.

14.1.6. Elementare Berechnungen

Aus Gleichungen wie in System (14.2) lassen sich direkt die elementaren, also nicht weiter zergliederbaren, Operationen ablesen, die in den Zellen des systolischen Feldes durchzuführen sind. Es sind dies die in jeweils einer Gleichung des Systems beschriebenen Operationen, bezogen auf einen konkreten Trägerpunkt der zugehörigen Quantifizierung. Kommen dabei in einer Gleichung mehrere Teiloperationen vor, werden diese als zusammengehörig, das heißt als *Verbundoperation* aufgefasst, und immer gemeinsam in einem Arbeitsschritt in derselben Zelle abgearbeitet.

In der mittleren Gleichung von System (14.2) treten zum Beispiel die Multiplikation $a(i, k) * b(k, j)$ und die darauf bezogene Addition $c(i, j, k) = c(i, j, k-1) + \dots$ auf. Die zusammengehörenden Einzeloperationen Multiplikation und Addition werden in der Tat als „multiply-add“ durch die in Abbildung 14.1(b) gezeigte Zelle des systolischen Feldes ausgeführt.

Wir können nun jeder solchen elementaren Berechnung eine Bezeichnung zuordnen. Nennen wir diese ruhig ebenfalls Koordinaten. Als einfache Möglichkeit, zu geeigneten Koordinaten zu kommen, bieten sich die Iterationsvektoren (i, j, k) an, die in den Quantifizierungen angegeben sind.

In Anwendung auf das System (14.1) können wir zum Beispiel der Berechnung $c(i, j, k) = c(i, j, k-1) + a(i, k) * b(k, j)$ das Koordinatentripel (i, j, k) zuordnen. Ebenso können wir der Eingabe $c(i, j, k) = 0$ ebenfalls das Koordinatentripel (i, j, k) zuordnen; hier gilt allerdings immer $k = 0$. Die zugehörigen Trägermengen sind übrigens im Beispiel disjunkt gewählt.

Wenn wir immer nur die Iterationsvektoren als Bezeichner der Berechnungen und der Ein-/Ausgabeoperationen benutzen, brauchen wir zwischen diesen beiden Konzepten nicht mehr zu unterscheiden. Dies bedeutet dann aber auch, dass alle zu einem Trägerpunkt gehörenden Operationen zusammen wieder eine Verbundoperation darstellen — selbst wenn sie in verschiedenen Gleichungen vorkommen sollten und miteinander vielleicht gar nichts zu tun haben. Im Folgenden benutzen wir zur Vereinfachung immer nur die Iterationsvektoren als Koordinaten.

14.1.7. Diskrete Zeitschritte

Die einzelnen elementaren Berechnungsvorgänge in den systolischen Zellen finden immer in *diskreten Zeitschritten* statt. Alle solchen Zeitschritte in einem systolischen Feld sind gleich lang. Überdies arbeiten alle Zellen eines systolischen Feldes vollständig *synchron*, das heißt, sie beginnen und beenden ihre jeweiligen Kommunikations- und Berechnungsschritte alle gleichzeitig. Die einzelnen Zeitschritte einer Zelle schließen immer nahtlos aneinander an.

14.3. Bemerkung. *Nun wissen wir ja spätestens seit den Erkenntnissen von Albert Einstein, dass sich wirkliche Gleichzeitigkeit physikalisch nicht erzwingen lässt. Es geht hier in der Tat nur darum, dass die Zellen des systolischen Feldes einigermäßen zeitlich aufeinander abgestimmt arbeiten. Technisch wird dies durch Versorgung aller systolischen Zellen mit einem gemeinsamen Taktsignal garantiert, das alle Register des Feldes durchschaltet. Im Rahmen der dabei erreichbaren Genauigkeit findet die Kommunikation zwischen den Zellen hinreichend gleichzeitig statt, so dass bei korrespondierenden Lese- und Schreibvorgängen keine Daten verloren gehen. Es ist daher durchaus angemessen, bei theoretischen Überlegungen fest von einer konzeptionellen Gleichzeitigkeit auszugehen.*

Wir können deshalb die physikalische Zeit in Einheiten eines Zeitschritts diskretisieren und die Zeitschritte fortlaufend durchnummerieren. Dabei spielt es keine Rolle, wo der Ursprung der Zeitachse liegt; die Zeit läuft ja für alle Zellen synchron. Eine plausible Entscheidung ist es, den Zeitschritt der ersten Eingabe in eine Zelle als Zeit $t = 0$ zu wählen. Mit dieser Festlegung würde die durch (i, j, k) bezeichnete elementare Verbundoperation des Systems (14.1) zum Zeitpunkt $i + j + k - 3$ ausgeführt. Andererseits ist es genau so stimmig, den Koordinaten (i, j, k) den Zeitpunkt $i + j + k$ zuzuordnen; die Zeit erscheint dann lediglich global um drei Einheiten verschoben.

Nehmen wir also für das Folgende an, dass die Ausführung einer Berechnungsinstanz (i, j, k) zum Zeitpunkt $i + j + k$ erfolgt. Der erste Berechnungsschritt findet dann zur Zeit $t = 3$ statt, der letzte zur Zeit $t = N_1 + N_2 + N_3$. Die *Gesamtausführungsdauer* beträgt damit $N_1 + N_2 + N_3 - 2$ Zeitschritte.

14.1.8. Äußere und innere Kommunikation

Die zur Berechnung nötigen Daten für das systolische Feld befinden sich normalerweise zu Beginn der Berechnung noch nicht in den Zellen des Feldes. Sie müssen dem Feld vielmehr von der *Außenwelt* zugeführt werden. Diese Außenwelt besteht aus einem *Wirtsrechner*, in der Regel einem skalaren Steuerprozessor mit Zugriff zu einem zentralen *Datenspeicher*. Der Steuerprozessor holt zur richtigen Zeit die benötigten Daten aus dem Speicher, führt sie dem systolischen Feld in geeigneter Weise zu und schreibt die berechneten Ergebnisse wieder in den Speicher zurück.

Jeder Zelle (i, j) müssen für den k betreffenden Zeitschritt die Operanden a_{ik} und b_{kj} zur Verfügung stehen. Aber nur die Zellen der ersten Spalte des systolischen Feldes in Abbildung 14.1 erhalten die Elemente der Matrix A direkt als Eingabedaten von der Außenwelt. Alle übrigen Zellen sind darauf angewiesen, dass ihnen

die benötigten Werte a_{ik} von einer Nachbarzelle zugeleitet werden. Dies wird über die horizontalen *Verbindungen* zwischen benachbarten Zellen in Abbildung 14.1(a) bewerkstelligt. Das Element a_{ik} durchläuft nacheinander die Zellen $(i, 1), (i, 2), \dots, (i, N_2)$. Entsprechend wird auch der Wert b_{kj} nur über die Zelle $(1, j)$ eingebracht und von dort mittels der vertikalen Verbindungen über die Zellen $(2, j), (3, j), \dots$ bis zur Zelle (N_1, j) weitertransportiert. Eine Pfeilspitze in der Abbildung kennzeichnet, in welche *Richtung* eine solche Verbindung genutzt wird.

In verteilten/parallelen Architekturen ist es häufig problematisch, einen Wert in einem Zeitschritt über große Distanzen zu transportieren. Daher erfolgt in unserem Beispiel die Weitergabe des Wertes a_{ik} von Zelle (i, j) zu Zelle $(i, j + 1)$ nicht bereits in demselben Zeitschritt, in dem Zelle (i, j) diesen Wert von Zelle $(i, j - 1)$ oder der Außenwelt erhalten hat, sondern erst einen Zeitschritt später. Entsprechendes gilt auch für den Transport der Werte b_{kj} . Diese *Verzögerung* wird aus der Detailzeichnung der Zelle in Abbildung 14.1(b) ersichtlich: Jeder Pfad zum Durchreichen von Eingabedaten durch eine Zelle verläuft über ein Register, und jedes passierte Register bewirkt eine Verzögerung um genau einen Zeitschritt.

14.4. Bemerkung. Für systolische Architekturen wird generell vorgeschrieben, dass Pfade zwischen verschiedenen Zellen immer über mindestens ein Register verlaufen — auch wenn es nur um einen Datentransport zu einer Nachbarzelle geht. Alle Register der Zellen werden durch das globale Taktsignal des systolischen Feldes synchron durchgeschaltet. Dies führt zu einem charakteristischen, rhythmischen Datenverkehr auf allen Verbindungen des systolischen Feldes. Der medizinische Begriff der „Systole“ stand denn auch wegen der bildlichen Verwandtschaft mit pulsierenden Blutgefäßen Pate für den Namen des Konzeptes.

Um diese verzögerte Weitergabe von Werten zu verdeutlichen, erweitern wir das Formelsystem (14.1) nochmals, indem wir nun auch für alle mehrfach *gelesenen* Werte wie a_{ik} separate Instanzen einführen (diese Vorgehensweise ist ganz typisch für den Entwurf systolischer Algorithmen):

Eingabeoperationen

$$\begin{aligned} a(i, j, k) &= a_{ik} & 1 \leq i \leq N_1, j = 0, 1 \leq k \leq N_3 \\ b(i, j, k) &= b_{kj} & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= 0 & 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0 \end{aligned}$$

Berechnungen

$$\begin{aligned} a(i, j, k) &= a(i, j - 1, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ b(i, j, k) &= b(i - 1, j, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= c(i, j, k - 1) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ &+ a(i, j - 1, k) * b(i - 1, j, k) \end{aligned} \quad (14.3)$$

Ausgabeoperationen

$$c_{ij} = c(i, j, k) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = N_3$$

Jede der Teilsummen $c(i, j, k)$ zur schrittweisen Berechnung von c_{ij} wird in einem bestimmten Zeitschritt berechnet und lediglich einmal, nämlich im nächsten Zeitschritt, benutzt. Zelle (i, j) muss daher über ein Register verfügen (RC in Abbildung 14.1(b)), in das der Wert von $c(i, j, k)$ einen Zeitschritt lang abgelegt wird. Anschließend wird $c(i, j, k)$ nicht länger benötigt, das betreffende Register kann also mit $c(i, j, k+1)$ überschrieben werden. Nach Abschluss der Berechnung des Skalarprodukts bleibt der Wert $c(i, j, N_3)$, also das auszugebende Resultat c_{ij} , in diesem Register stehen. Vor Beginn der Berechnung muss das Register *gelöscht*, das heißt mit dem Wert Null (oder einem anderen gewünschten Wert) beschrieben werden.

Die Werte a_{ik} und b_{kj} müssen dagegen nicht dauerhaft in Zelle (i, j) gespeichert werden. Wie aus dem Dateneingabeschema in Abbildung 14.1(a) ersichtlich, wird jede Zeile der Matrix A gegenüber der vorhergehenden um einen Zeitschritt verzögert eingegeben. Ebenso wird jede Spalte der Matrix B gegenüber der vorhergehenden um einen Zeitschritt verzögert. Damit kommen die Werte $a(i, j-1, k)$ und $b(i-1, j, k)$ exakt im Zeitschritt der Berechnung von $c(i, j, k)$ in Zelle (i, j) an, werden in die Register RA beziehungsweise RB geschrieben, daraus aber sofort wieder für die Multiplikation verwendet sowie an die Nachbarzellen weitergereicht. Nachdem die Werte a_{ik} und b_{kj} in Zelle (i, j) multipliziert wurden, sind sie für diese Zelle nicht mehr von Belang. Sie brauchen daher in der Zelle auch nicht länger verfügbar gehalten werden. RA und RB werden deshalb im nächsten Zeitschritt mit neuen Werten überschrieben.

Es ist aus diesen Ausführungen schon ersichtlich, dass wir zur Reduzierung des Speicherbedarfs einer Zelle bemüht sein müssen, alle Berechnungs- und Kommunikationsvorgänge räumlich und zeitlich so zu koordinieren, dass die Speicherung von Werten durch sofortige Verwendung und Weitergabe auf ein möglichst kurzes Zeitintervall beschränkt wird. Eine solche Koordinierung wird, außer durch den allgemeinen Aufbau des systolischen Feldes, im Wesentlichen durch die Wahl eines geeigneten Ein-/Ausgabeschemas und die Festlegung einer passenden Anzahl von Verzögerungsschritten in den Zellen ermöglicht. Abbildung 14.1(b) zeigt diesbezüglich nur die kleinstmögliche Verzögerung um einen Zeitschritt.

Das *Ein-/Ausgabeschema* des Beispiels entstand geometrisch durch Anwendung einer *Scherung* auf die Matrizen A und B . Die dadurch freiwerdenden Plätze in den Eingabeströmen wurden hier durch Null-Elemente gefüllt, da diese für die vorliegende Berechnung unschädlich sind. Die Länge der Eingabeströme ist vom Problemparаметer N_3 abhängig.

Die Elemente der Matrix C werden laut Abbildung 14.1 *stationär* berechnet, das heißt, die einzelnen Berechnungsschritte zum Aufbau eines endgültigen Matrixelements erfolgen alle in derselben Zelle. *Stationäre Variablen* bewegen sich während der Berechnung überhaupt nicht im systolischen Feld. Die Ergebnisse müssen abschließend in einem zusätzlichen Vorgang zu den Rändern des Feldes transportiert werden, da sie erst von dort an die Außenwelt abgeliefert werden können. Hinzu kommt die Notwendigkeit einer Initialisierung des Registers für c_{ij} . Die Lösung dieser beiden Zusatzaufgaben ist mit einem hohen Aufwand an Zeit und auch Material verbunden, was in Abschnitt 14.4 noch genauer studiert werden wird.

14.1.9. Fließbandverarbeitung

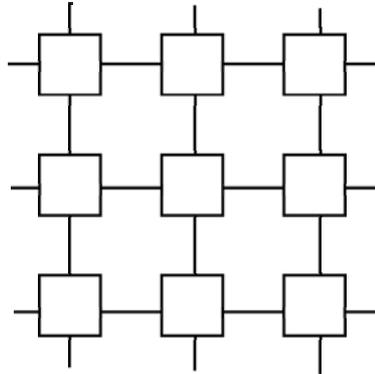
Die charakteristische Arbeitsweise mit diskreten, gleich langen, global synchronisierten Zeitschritten und strikter zeitlicher Separierung der Zellen durch Register weist die systolischen Felder als spezielle Anwendung der *Fließbandverarbeitung* aus. Die Register der Zellen entsprechen hierbei den bekannten *Pipeline-Registern*. Klassische Pipelines sind allerdings immer lineare Strukturen, während systolische Felder (wie im Beispiel gezeigt) häufig einen zweidimensionalen Aufbau haben. Ein *mehrdimensionales systolisches Feld* kann aber durchaus als eine Menge miteinander verflochtener Pipelines aufgefasst werden. Damit sollte klar sein, dass sich grundsätzliche Eigenschaften der eindimensionalen Fließbandverarbeitung auch bei mehrdimensionalen systolischen Felder wiederfinden.

Ein typischer Effekt der Fließbandverarbeitung ist die reduzierte *Auslastung* am Beginn und am Ende einer Berechnung. Zunächst ist die Pipeline leer, keine der Pipelinestufen arbeitet. Dann erhält die erste Stufe Daten und nimmt die Berechnung auf; alle weiteren Stufen sind immer noch unbeschäftigt. Im nächsten Zeitschritt gibt die erste Stufe Daten an die zweite Stufe weiter und nimmt selbst neue Daten entgegen; nur diese beiden Stufen arbeiten. Dies geht so weiter, bis schließlich alle Stufen in jedem Zeitschritt Daten erhalten und verarbeiten, die Pipeline also erstmalig voll ausgelastet ist. Nach einer Folge von Zeitschritten mit voller Last, deren Dauer von der Länge des Datenstroms abhängt, reißt die Eingabe ab, die erste Stufe der Pipeline hört damit auf zu arbeiten. Im nächsten Zeitschritt stellt auch die zweite Stufe die Arbeit ein, und so weiter, bis schließlich keine der Stufen mehr arbeitet. Die Phasen reduzierter Aktivität schmälern die durchschnittliche Leistung der gesamten Pipeline, und der relative Wert dieser Leistungseinbuße ist dabei um so größer, je mehr Stufen die Pipeline in Relation zur Länge des Datenstroms besitzt.

Wir können dieses Phänomen in seiner speziellen Ausprägung beim zweidimensionalen systolischen Feld an Abbildung 14.1 recht gut studieren. Auch hier gibt es quasi unbeschäftigte Zellen zu Beginn und am Ende der Berechnung. Im ersten Zeitschritt verrichtet nur die Zelle (1,1) sinnvolle Arbeit; alle übrigen Zellen führen zwar auch Berechnungen durch, die aber wie Leerschritte wirken (müssen). Im zweiten Schritt kommen die Zellen (1,2) und (2,1) hinzu; dieser Zeitschritt ist in Abbildung 14.2(a) dargestellt. Dies geht so weiter, bis schließlich auch die Zelle (N_1, N_2) zur Berechnung beiträgt. Sobald das letzte eigentliche Datenelement die Zelle (1,1) verlassen hat, erbringt diese keinen Beitrag mehr, sondern reproduziert nur noch den berechneten Wert von c_{11} . Schritt für Schritt fallen immer mehr Zellen aus der aktiven Arbeit, bis schließlich nur noch die Zelle (N_1, N_2) einen letzten notwendigen Berechnungsschritt erbringt; Abbildung 14.2(b) zeigt diesen finalen Zeitschritt.

Übungen

14.1-1. Wie müsste das Dateneingabeschema aus Abbildung 14.1(a) verändert werden, wenn auf demselben systolischen Feld eine (2×6) -Matrix mit einer (6×3) -Matrix multipliziert werden sollte? Könnte die Berechnung auch so organisiert werden, dass die Ergebnismatrix in der rechten unteren Ecke des systolischen Feldes anfallen würde?



14.2. Abbildung. Zwei Momentaufnahmen zu Abbildung 14.1 (Ausschnitte).

14.1-2. Warum ist es für die Matrixmultiplikation in der Fassung von Abbildung 14.1 unschädlich, nicht benötigte Plätze in den Eingabeströmen mit Nullen zu füllen?

14.1-3. Wenn das systolische Feld aus Abbildung 14.1 als Pipeline interpretiert werden sollte: Welche Anzahl von Stufen müsste dann angesetzt werden, um das Verhalten angemessen zu beschreiben?

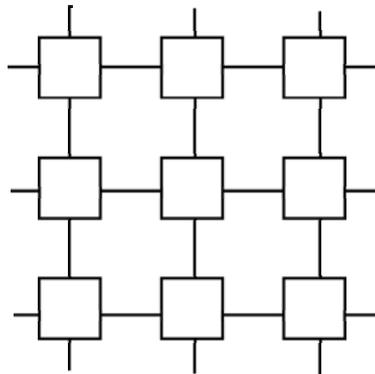
14.2. Raum-Zeit-Abbildung und systolisches Feld

Der im vorhergehenden Abschnitt benutzte Zugang reicht zwar für ein erstes Verständnis in der Regel aus, genügt aber nicht mehr, wenn die Eigenschaften systolischer Felder quantitativ exakt erfasst und bewertet werden sollen. Insbesondere das Auftreten *parametrischer Aufgabenstellungen* erfordert mathematische Hilfsmittel. In diesem Abschnitt werden daher zentrale Konzepte einer auf *linearen Abbildungen* basierenden Theorie *uniformer* systolischer Algorithmen formal behandelt.

14.2.1. Beispiel: Matrixprodukt ohne stationäre Variablen

Das System (14.3) kann außer durch das systolische Feld aus Abbildung 14.1 durch viele weitere Arten systolischer Felder berechnet werden. In Abbildung 14.3 ist zum Beispiel ein solches systolisches Feld dargestellt. Obwohl dieselbe Funktion berechnet wird wie bei dem System in Abbildung 14.1, ist das Erscheinungsbild ein völlig anderes:

- Die Zahl der Zellen ist nun wesentlich größer, insgesamt 36, statt 15.
- Die Kontur des Feldes ist hexagonal, statt rechteckig.
- Jede Zelle hat nun drei Eingänge und drei Ausgänge.
- Die Zuführung der Daten verläuft deutlich anders als in Abbildung 14.1(a).
- Und schließlich: Die Matrix C fließt nun ebenfalls durch das ganze Feld.



14.3. Abbildung. Hexagonales systolisches Feld für Matrixmultiplikation: (a) Felddaufbau und Prinzip der Datenein-/ausgabe; (b) Zellstruktur.

Die Zellstruktur in Abbildung 14.3(b) scheint zunächst nicht wesentlich verschieden von der in Abbildung 14.1(b). Dennoch sind die Unterschiede relevant: Es gibt keine *zyklischen Wege* in der neuen Zelle, *stationäre Variablen* können deshalb nicht mehr vorkommen. Stattdessen hat die Zelle nun drei Eingangs- und drei Ausgangskanäle, über die Elemente aller drei Matrizen durch die Zelle fließen. Die Kommunikationsrichtung der Kanäle an der rechten und linken Seite der Zelle hat sich verändert, die Zuordnung der Matrizen zu den Kanälen ebenfalls.

14.2.2. Die Raum-Zeit-Abbildung als globale Sichtweise

Wie hängt nun das System (14.3) mit der Abbildung 14.3 zusammen? In Abschnitt 14.1 konnten wir die Arbeitsweise des dort beschriebenen systolischen Feldes ohne Hilfsmittel gut nachvollziehen. Für das vorliegende Beispiel ist dies aber wesentlich schwieriger — nun sind wir wohl besser motiviert für den Einsatz mathematischer Hilfsmittel.

Wir können jeder auszuführenden Operation — um ihre Abarbeitung im systolischen Feld zu beschreiben — zwei fundamentale Kenngrößen zuordnen: den Zeitschritt, in dem die Operation ausgeführt wird, und die Zelle, in der die Operation ausgeführt wird. Wie noch ersichtlich werden wird, sind mit der Festlegung dieser *Raum-Zeit-Abbildung* die wesentlichen Freiheitsgrade des Entwurfs bereits ausgeschöpft; nahezu alle weiteren Gestaltungsmerkmale folgen zwingend aus der Raum-Zeit-Abbildung.

Wie schon für das systolische Feld in Abbildung 14.1 erfolgt auch im systolischen Feld der Abbildung 14.3 die Ausführung einer Instanz (i, j, k) zum Zeitpunkt $t = i + j + k$. Wir können dies auch als Skalarprodukt eines *Zeitvektors*

$$\pi = (1 \quad 1 \quad 1) \tag{14.4}$$

mit dem Iterationsvektor $v = (i, j, k)$ beschreiben,

$$t = \pi \cdot v, \quad (14.5)$$

hier also

$$t = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} i \\ j \\ k \end{pmatrix} = i + j + k. \quad (14.6)$$

Die Raumkoordinaten $z = (x, y)$ der ausgeführten Operationen im Beispiel aus Abbildung 14.1 lassen sich gemäß unserer Festlegung in Abschnitt 14.1.3 als $z = (i, j)$ aus den Iterationsvektoren $v = (i, j, k)$ herleiten. Die dabei gewählte Abbildung stellt eine *Projektion* des Raums \mathbf{R}^3 längs der k -Achse dar. Diese lineare Abbildung kann durch eine *Projektionsmatrix*

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (14.7)$$

beschrieben werden. Die Raumkoordinaten erhalten wir nun durch Multiplikation der Projektionsmatrix mit dem Iterationsvektor v , was durch

$$z = P \cdot v \quad (14.8)$$

beschrieben wird.

Die *Projektionsrichtung* wird durch jeden Vektor dargestellt, der zu allen Zeilen der Projektionsmatrix orthogonal verläuft,

$$P \cdot u = \vec{0}. \quad (14.9)$$

Zur Projektionsmatrix P aus (14.7) ist zum Beispiel $u = (0, 0, 1)$ ein möglicher *Projektionsvektor*.

Projektionen werden sehr häufig zur Beschreibung von Raumkoordinaten beim Entwurf systolischer Felder eingesetzt. Auch in unserem Beispiel aus Abbildung 14.3(a) entstehen die Raumkoordinaten durch Projektion aus den Iterationsvektoren. Hier ist eine Projektionsmatrix durch

$$P = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \end{pmatrix} \quad (14.10)$$

gegeben. Ein zugehöriger Projektionsvektors wäre $u = (1, 1, 1)$.

Projektionsmatrix und Zeitvektor können wir in einer Matrix T zusammenfassen, die dann die sogenannte *Raum-Zeit-Abbildung* beschreibt,

$$\begin{pmatrix} z \\ t \end{pmatrix} = \begin{pmatrix} P \\ \pi \end{pmatrix} \cdot v = T \cdot v. \quad (14.11)$$

Die ersten beiden Zeilen von T werden durch die Projektionsmatrix P gegeben, die dritte Zeile durch den Zeitvektor π .

Für das Beispiel aus Abbildung 14.1 lautet die Matrix T der Raum-Zeit-Abbildung

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad (14.12)$$

für das Beispiel aus Abbildung 14.3 stattdessen

$$T = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \quad (14.13)$$

Die Raum-Zeit-Abbildung kann als *globale Sichtweise* auf das systolische System verstanden werden. Durch Anwendung einer (hier linearen, durch T beschriebenen) Raum-Zeit-Abbildung auf ein System von Rekurrenzgleichungen ergeben sich direkt die äußeren Merkmale des systolischen Feldes, also seine *Architektur* (Raumkoordinaten, Verbindungsstruktur, Zellstruktur).

14.5. Bemerkung. *Statt rein linearer Abbildungen kommen alternativ auch affine Abbildungen in Frage, die dann noch einen vektoriellen Anteil tragen, $T \cdot v + h$. Affine Abbildungen sind aber nicht nötig, solange wir alle Iterationsvektoren mit derselben Raum-Zeit-Abbildung behandeln.*

14.2.3. Symbolische Bestimmung der Raumkoordinaten

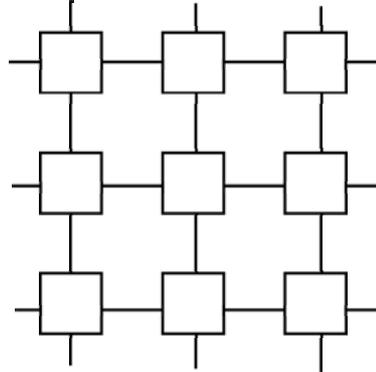
Wenn die Trägermengen zahlenmäßig vorgegeben und insbesondere klein sind, kann über die Beziehung (14.11) leicht die konkrete Menge der Raumkoordinaten ausgerechnet werden. Sind aber die Trägermengen parametrisch wie im System (14.3) gegeben, müssen die Positionen der Zellen *symbolisch* bestimmt werden. Die folgenden Ausführungen widmen sich speziell diesem Problem.

Wir betrachten jede Zelle als geometrisch durch einen Punkt mit den Raumkoordinaten $z = (x, y)$ in dem zweidimensionalen Raum \mathbf{R}^2 dargestellt. Aus jedem Iterationsvektor v der Trägermenge S erhalten wir mit Beziehung (14.8) die Raumkoordinaten z eines bestimmten Prozessors, $z = P \cdot v$, die durch v symbolisierte Operation wird auf die Zelle z projiziert. Die Menge der so erhaltenen Raumkoordinaten $P(S) = \{P \cdot v : v \in S\}$ gibt die Positionen aller für die Funktionsweise notwendigen Zellen des systolischen Feldes an.

Üblicherweise werden Trägermengen verwendet, die als Menge aller ganzzahligen Punkte eines konvexen Gebietes (hier im \mathbf{R}^3) darstellbar sind (*dichte konvexe Trägermengen*). Die konvexe Hülle einer solchen Trägermenge (mit einer endlicher Anzahl von Trägerpunkten) ist ein *Polytop*, dessen Ecken Trägerpunkte sind. Polytope werden durch beliebige lineare Abbildungen wieder in Polytope überführt. Nun können wir uns zunutze machen, dass jede Projektion eine lineare Abbildung ist. Die Ecken des sich ergebenden Polytops sind dabei Projektionen von Ecken des Ausgangspolytops.

14.6. Bemerkung. *Nicht alle Ecken des Ausgangspolytops müssen bei einer Projektion auch auf Ecken des sich ergebenden Polytops abgebildet werden, siehe zum Beispiel Abbildung 14.4.*

Das Gitter \mathbf{Z}^3 wird bei der Projektion mit einer ganzzahligen Projektionsmatrix P in das Gitter \mathbf{Z}^2 überführt, wenn sich P durch Wahl eines ganzzahligen Zeitvektors π zu einer *unimodularen Matrix* T ergänzen lässt. Eine dichte konvexe Trägermenge wird dadurch (bis auf für die Anwendung irrelevante Ausnahmen) in eine dichte



14.4. **Abbildung.** Ergebnis der Projektion einer rechteckigen Trägermenge.

konvexe Menge von Raumkoordinaten überführt, die durch die Ecken des umschließenden Polytops vollständig charakterisiert ist. Wir brauchen also P nur auf die Ecken der konvexen Hülle von S anzuwenden, um die Dimensionen des systolischen Feldes zu errechnen.

14.7. Bemerkung. Eine Matrix heißt unimodular, wenn sie quadratisch ist, nur ganzzahlige Einträge aufweist und die Determinante ± 1 besitzt. Unimodulare Matrizen sind invertierbar, ihre Inversen sind ebenfalls unimodular.

Wir wenden diese Methode auf die ganzzahlige Trägermenge

$$S = [1, N_1] \times [1, N_2] \times [1, N_3] \quad (14.14)$$

aus System (14.3) an. Die Ecken der konvexen Hülle sind hier

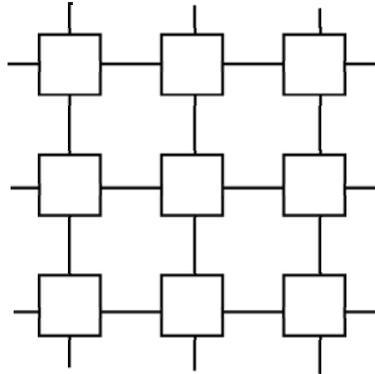
$$\begin{aligned} &(1, 1, 1), (N_1, 1, 1), (1, N_2, 1), (1, 1, N_3), \\ &(1, N_2, N_3), (N_1, 1, N_3), (N_1, N_2, 1), (N_1, N_2, N_3). \end{aligned} \quad (14.15)$$

Für die Projektionsmatrix P aus (14.10) haben die Ecken der Projektion die Positionen

$$\begin{aligned} &(N_3 - 1, 0), (N_3 - 1, 1 - N_1), (0, 1 - N_1), \\ &(1 - N_2, N_2 - N_1), (1 - N_2, N_2 - 1), (N_3 - N_2, N_2 - N_1). \end{aligned} \quad (14.16)$$

Da S acht Ecken hat, das Bild $P(S)$ aber nur sechs, wird klar, dass zwei Ecken von S in das Innere des Gebietes abgebildet werden, für die Ausmaße also keine Rolle spielen; dies sind die Ecken $(1, 1, 1)$ und (N_1, N_2, N_3) . Die Zusammenhänge sind in Abbildung 14.4 dargestellt.

Konkret für $N_1 = 3$, $N_2 = 5$ und $N_3 = 4$ ergeben sich die Eckpunkte $(3, 0)$, $(3, -2)$, $(0, -2)$, $(-4, 2)$, $(-4, 4)$ und $(-1, 4)$. Wir sehen, dass Raumkoordinaten nicht unbedingt positiv sein müssen. Auch ist die Wahl des Koordinatenursprungs, der hier im Inneren des Polytops liegt, nicht unbedingt offensichtlich.



14.5. Abbildung. Zerlegung der Menge von Raumkoordinaten.

Als Ergebnis der Projektion erhalten wir ein Sechseck mit parallelen gegenüberliegenden Seiten. Auf den begrenzenden Kanten liegen jeweils N_1 , N_2 oder N_3 ganzzahlige Punkte. Anders als im ersten Beispiel sind hier also alle Problemparameter auch gleichzeitig Feldparameter. Der Umriss dieses systolischen Feldes wird *hexagonal* genannt.

Die Fläche F dieses Gebiets beträgt, komplexitätstheoretisch betrachtet, $O(F) = N_1 N_2 + N_1 N_3 + N_2 N_3$. Sie hängt damit von allen drei Matrixdimensionen in gleicher Weise ab. Dies ist also ganz anders als in Abbildung 14.1(a), wo die Komplexität (für das gleiche Problem!) nur $O(F) = N_1 N_2$ beträgt.

Wir zählen nach dieser überschlägigen Berechnung nun die Zellen noch ganz exakt ab. Für diese Abzählung ist es hilfreich, das gesamte Gebiet zunächst in Teilgebiete zu zerlegen, bei denen die Anzahl der in ihnen enthaltenen Zellen leicht bestimmbar ist (Abbildung 14.5). Die Punkte $(0,0)$, $(N_3 - 1, 0)$, $(N_3 - 1, 1 - N_1)$ und $(0, 1 - N_1)$ beranden ein Rechteck aus $N_1 N_3$ Zellen. Verschieben wir diese Punktmenge um $N_2 - 1$ Zellen nach oben und um $N_2 - 1$ Zellen nach rechts, überstreichen wir das ganze Gebiet. Bei jeder Verschiebung um eine Zelle nach oben und rechts kommen aber nur $N_1 + N_3 - 1$ Zellen hinzu. Insgesamt ergibt dies $N_1 N_3 + (N_2 - 1)(N_1 + N_3 - 1) = N_1 N_2 + N_1 N_3 + N_2 N_3 - (N_1 + N_2 + N_3) + 1$ Zellen.

Für $N_1 = 3$, $N_2 = 5$ und $N_3 = 4$ erhalten wir damit die Zahl von 36 Zellen, wie in Abbildung 14.3(a) bereits ersichtlich.

14.2.4. Symbolische Berechnung der Gesamtausführungsdauer

Die symbolische Berechnung der *Gesamtausführungsdauer* eines systolischen Algorithmus kann mit einem ähnlichen Ansatz wie in Abschnitt 14.2.3 erfolgen. Die Zeitabbildung nach Beziehung (14.5) ist ebenfalls eine lineare Abbildung. Die Zeitschritte der ersten und der letzten Berechnung erhalten wir als Minimum beziehungsweise Maximum der Menge $\pi(S) = \{\pi \cdot v : v \in S\}$ von Berechnungszeitpunkten. Es genügt dabei nach oben Gesagtem, für v alle Ecken der konvexen Hülle von S einzusetzen.

Die *Gesamtausführungsdauer* errechnet sich dann nach der Formel

$$t_{\Sigma} = 1 + \max P(S) - \min P(S). \quad (14.17)$$

Die Addition der 1 ist unbedingt nötig, da sowohl der erste als auch der letzte Zeitschritt zur Berechnung zählen.

Für das Beispiel in Abbildung 14.3 ergibt sich mit Beziehung (14.6) und den in (14.15) aufgezählten Ecken des Polytops die Menge von Bildern $\{3, 2+N_1, 2+N_2, 2+N_3, 1+N_1+N_2, 1+N_1+N_3, 1+N_2+N_3, N_1+N_2+N_3\}$. Aus den Grundannahmen $N_1, N_2, N_3 \geq 1$ folgt dann das Minimum 3 und das Maximum $N_1+N_2+N_3$, somit eine Gesamtausführungsdauer von $N_1+N_2+N_3-2$ Zeiteinheiten (wie für das systolische Feld in Abbildung 14.1 — schließlich stimmen Trägermenge und Zeitvektor überein).

Für die konkreten Problemparametern $N_1 = 3$, $N_2 = 5$ und $N_3 = 4$ errechnet sich eine Gesamtausführungsdauer von $12 - 3 + 1 = 10$ Zeitschritten.

14.2.5. Ableitung der Verbindungsstruktur

Die *Verbindungsstruktur* des systolischen Feldes wird durch Anwendung der Raum-Zeit-Abbildung auf die *Datenabhängigkeiten* des Problems induziert. Jede Datenabhängigkeit ist die Folge der direkten Verwendung einer Variableninstanz zur Berechnung einer Instanz derselben oder einer anderen Variablen.

14.8. Bemerkung. *Im Gegensatz zur Ausgangssituation bei der Datenabhängigkeitsanalyse für imperative Programmiersprachen, wie sie von hoch optimierenden Compilern durchgeführt werden muss, sind Datenabhängigkeiten hier immer nur Flussabhängigkeiten. Dies ist eine Konsequenz der von uns verwendeten zuweisungsfreien Notation.*

Die Datenabhängigkeiten lassen sich jeweils aus der gleichzeitigen Betrachtung der rechten und der linken Seite einer quantifizierten Gleichung der zuweisungsfreien Notation ablesen. Wir analysieren zunächst die Gleichung $c(i, j, k) = c(i, j, k-1) + a(i, j-1, k) * b(i-1, j, k)$ aus System (14.3).

Der Wert $c(i, j, k)$ wird aus den Werten $c(i, j, k-1)$, $a(i, j-1, k)$ und $b(i-1, j, k)$ errechnet. Es liegt damit ein Datenfluss von $c(i, j, k-1)$ zu $c(i, j, k)$, ein Datenfluss von $a(i, j-1, k)$ zu $c(i, j, k)$ und ein Datenfluss von $b(i-1, j, k)$ zu $c(i, j, k)$ vor.

Die hier relevanten Eigenschaften eines solchen Datenflusses können durch einen *Abhängigkeitsvektor* beschrieben werden. Dieser ist der Differenzvektor zwischen dem Iterationsvektor der berechneten Variableninstanz und dem Iterationsvektor der dazu jeweils benötigten Variableninstanz.

Der Iterationsvektor für $c(i, j, k)$ ist (i, j, k) , der für $c(i, j, k-1)$ ist $(i, j, k-1)$. Als Differenzvektor ergibt sich somit

$$d_C = \begin{pmatrix} i \\ j \\ k \end{pmatrix} - \begin{pmatrix} i \\ j \\ k-1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (14.18)$$

Entsprechend folgen

$$d_A = \begin{pmatrix} i \\ j \\ k \end{pmatrix} - \begin{pmatrix} i \\ j-1 \\ k \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (14.19)$$

und

$$d_B = \begin{pmatrix} i \\ j \\ k \end{pmatrix} - \begin{pmatrix} i-1 \\ j \\ k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \quad (14.20)$$

In der Gleichung $a(i, j, k) = a(i, j-1, k)$ aus System (14.3) ist nicht direkt erkennbar, welches die berechnete Variableninstanz und welches die zu ihrer Berechnung notwendige Variableninstanz ist. Hier sehen wir also den Unterschied zwischen Gleichungen und Zuweisungen. Mit der Festlegung, dass $a(i, j, k)$ aus $a(i, j-1, k)$ durch Kopieren berechnet werden soll, erhalten wir denselben Abhängigkeitsvektor d_A wie in Beziehung (14.19). Entsprechendes gilt für die Gleichung $b(i, j, k) = b(i-1, j, k)$.

Besitzt eine Variableninstanz den Iterationsvektor v , so wird sie in der Zelle $P \cdot v$ berechnet. Wird zu ihrer Berechnung eine andere Variableninstanz mit Iterationsvektor v' benötigt, liegt also eine Datenabhängigkeit mit Abhängigkeitsvektor $d = v - v'$ vor, so wurde diese in der Zelle $P \cdot v'$ berechnet. Es ist daher eine Kommunikation von Zelle $z' = P \cdot v'$ zu Zelle $z = P \cdot v$ durchzuführen. In systolischen Feldern wird die Kommunikation durch Herstellen einer direkten, statischen Verbindung zwischen den kommunizierenden Zellen realisiert. Wegen der Linearität der Abbildung (14.8) gilt $z - z' = P \cdot v - P \cdot v' = P \cdot (v - v') = P \cdot d$.

Ist $P \cdot d = \vec{0}$, so erfolgt die Kommunikation in der berechnenden Zelle selbst, das heißt, nur in der Zeit, nicht aber im Raum. Die Weitergabe von Werten in der Zeit geschieht über Register der berechnenden Zelle.

Gilt dagegen $P \cdot d \neq \vec{0}$, wird eine Kommunikation zwischen verschiedenen Zellen nötig. Zu allen Zellen des systolischen Feldes muss dann eine Verbindung mit Flussrichtung $P \cdot d$ bereitgestellt werden. Tragen wir den Vektor $P \cdot d$ am Raumpunkt z der berechnenden Zelle entgegen seiner Flussrichtung an, so gelangen wir zum Raumpunkt z' der z versorgenden Zelle.

Gibt es mehrere Abhängigkeitsvektoren d , werden für jeden von ihnen die entsprechenden Verbindungen benötigt. Nehmen wir als Beispiel die Beziehungen (14.18), (14.19), (14.20) und (14.10), so gilt $P \cdot d_A = (-1, 1)$, $P \cdot d_B = (0, -1)$ und $P \cdot d_C = (1, 0)$. Die den drei Vektoren $P \cdot d$ entsprechenden Verbindungen sehen wir in Abbildung 14.3(a) an jeder Zelle angetragen. Es resultiert eine hexagonale (statt der bereits bekannten orthogonalen) Verbindungsstruktur.

14.2.6. Bestimmung der Zellstruktur

Wir übertragen nun die räumlichen Überlegungen aus Abschnitt 14.2.5 auf die zeitlichen Zusammenhänge. Eine Variableninstanz mit Iterationsvektor v wird im Zeitschritt $\pi \cdot v$ berechnet. Wird zu ihrer Berechnung eine andere Variableninstanz mit Iterationsvektor v' benötigt, so wurde diese im Zeitschritt $\pi \cdot v'$ berechnet. Die dem Abhängigkeitsvektor $d = v - v'$ entsprechende Kommunikation hat damit in genau

$\pi \cdot v - \pi \cdot v'$ Zeitschritten zu erfolgen.

Da die Abbildung (14.6) linear ist, gilt $\pi \cdot v - \pi \cdot v' = \pi \cdot (v - v') = \pi \cdot d$. Da jede Kommunikation nach dem systolischen Prinzip über mindestens ein Register zu erfolgen hat und die Abhängigkeitsvektoren d festliegen, ist die Wahl eines Zeitvektors π durch die Vorschrift

$$\pi \cdot d \geq 1 \quad (14.21)$$

eingeschränkt.

Für $P \cdot d = \vec{0}$ müssen zur Speicherung des jeweils benötigten Wertes in allen Zellen *Register* bereitgestellt werden. Da ein solches Register bereits im nächsten Zeitschritt mit einem neuen Wert überschrieben wird, muss der alte Wert in ein weiteres Register übertragen werden, falls $\pi \cdot d \geq 2$ gilt. Da sich dieser Vorgang für jeden gespeicherten Wert über $\pi \cdot d$ Zeitschritte wiederholt, benötigt die Zelle dafür auch genau $\pi \cdot d$ Register, die ein Wert nacheinander durchläuft, bevor er an eine andere Zelle übertragen wird. Entsprechend zur eben geschilderten Situation verläuft der Transport im Falle $P \cdot d \neq \vec{0}$ ebenfalls über $\pi \cdot d$ Register, die aber nicht unbedingt alle in der berechnenden Zelle liegen müssen.

Für jeden Abhängigkeitsvektor d werden entsprechende Register benötigt. In Abbildung 14.3(b) finden wir an der Zelle drei Eingänge, die den Abhängigkeitsvektoren d_A , d_B und d_C entsprechen. Es gilt nämlich für diese $P \cdot d \neq \vec{0}$, wegen (14.6) und (14.4) jeweils mit $\pi \cdot d = 1$. Somit ist für jeden Abhängigkeitsvektor d genau ein Register nötig. Wegen der Regularität des Systems (14.3) entsprechen den drei Eingängen jeder Zelle auch drei Ausgänge auf den bezüglich der Zellmitte genau gegenüberliegenden Positionen.

Da die Anzahl der Abhängigkeitsvektoren d durch ein System wie (14.3) statisch beschränkt ist und für jeden von ihnen die Größe $\pi \cdot d$ einen festen, meist kleinen Wert besitzt, benötigt eine Zelle generell nur wenige Register.

Die drei Ein- und Ausgänge der Zelle machen nun die Verwendung von drei bewegten Matrizen möglich. Anders als in Abbildung 14.1 erfolgt die Berechnung eines Skalarprodukts $\sum_{k=1}^4 a_{ik} \times b_{kj}$ hier also nicht in einer einzigen Zelle, sondern verteilt über das systolische Feld. Es war hierzu zwingend notwendig, die Summe in eine Folge von Einzeladditionen aufzulösen. Dies stellt den Fall eines verteilten generischen Operators dar.

Außer den drei Eingängen mit ihren Registern und den drei Ausgängen zeigt Abbildung 14.3(b) ein Multiplizierwerk in Serie mit einem Addierwerk. Diese beiden Einheiten werden durch die Anwendung der Abbildung (14.8) auf die Trägermenge S der Gleichung $c(i, j, k) = c(i, j, k - 1) + a(i, j - 1, k) * b(i - 1, j, k)$ des Systems (14.3) in jeder Zelle induziert. Da laut dieser Gleichung die Bildung der Summe erst nach erfolgter Berechnung des Produkts erfolgen kann, ergibt sich die in Abbildung 14.3(b) gezeichnete Abfolge der beiden Operatoren.

Die Herkunft jedes der benötigten Operanden wird an der Projektion seines zugehörigen Abhängigkeitsvektors abgelesen. So gehört zu $a(i, j - 1, k)$ hier der Abhängigkeitsvektor $d_A = (0, 1, 0)$. Dessen Projektion $P \cdot d_A = (-1, 1)$ ist die Flussrichtung der Matrix A . Der zu lesende Wert muss daher aus der, vom berechnenden Prozessor gesehen, entgegengesetzten Richtung $(1, -1)$ erwartet werden, also vom Kanal in der linken unteren Ecke der Zelle kommen (aber durch das Register RA). Entsprechend kommt $b(i - 1, j, k)$ von rechts (durch das Register RB) und $c(i, j, k - 1)$ von oben

(durch das Register RC). Die entsprechenden Werte $a(i, j, k)$, $b(i, j, k)$ und $c(i, j, k)$ werden in die gegenüberliegenden Richtungen durch Kanäle ausgegeben: nach rechts oben, links und unten.

Benutzen wir andererseits die Projektionsmatrix P aus (14.7), so erhalten wir für d_C die Projektion $(0, 0)$. Zusammen mit $\pi \cdot d_C = 1$ ergibt sich dann die Notwendigkeit genau eines Registers RC für jedes Element der Matrix C . Dieses Register liefert für die Berechnung von $c(i, j, k)$ den Wert $c(i, j, k-1)$ und nimmt nach dieser Berechnung den Wert $c(i, j, k)$ auf. Dies ist an der Zelle aus Abbildung 14.1(b) nachvollziehbar. Abbildung 14.1(a) zeigt dementsprechend, dass für die Matrix C keine Verbindungen zwischen den Zellen erforderlich sind: Die Matrix ist *stationär*.

Übungen

14.2-1. Zu einer Projektionsrichtung u passen immer viele verschiedene Projektionsmatrizen P .

a. Zeigen Sie, dass auch

$$P = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}$$

eine Projektionsmatrix zur Projektionsrichtung $u = (1, 1, 1)$ ist.

- b.** Berechnen Sie mit dieser Projektionsmatrix die Projektion der Trägermenge von System (14.3).
- b.** Die hierbei entstehenden Raumkoordinaten unterscheiden sich von denen aus Abschnitt (14.2.3). Warum sind die in beiden Fällen erhaltenen Punktmengen dennoch topologisch äquivalent?
- b.** Untersuchen Sie die beiden Anordnungen der Zellen auf Gemeinsamkeiten und Unterschiede.

14.2-2. Führen Sie alle Überlegungen des Abschnitts 14.2 für das System (14.3) und die Raum-Zeit-Matrix

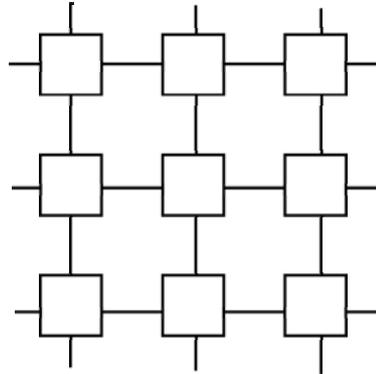
$$T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

durch.

14.3. Herleitung von Ein-/Ausgabeschemata

In Abbildung 14.3(a) wird das *Ein-/Ausgabeschema* nur cursorisch durch Angabe der *Flussrichtungen* für die Matrizen A, B, C beschrieben. Die notwendigen Details zum Verständnis der Ein-/Ausgabevorgänge werden nun in Abbildung 14.6 nachgeliefert.

Das Ein-/Ausgabeschema in Abbildung 14.6 weist gegenüber der Abbildung 14.1(a) eine Reihe neuer Phänomene auf. Durch jede gewöhnliche Randzelle fließt zwar auch hier ein Eingabe- beziehungsweise ein Ausgabedatenstrom, für die Ecken des systolischen Feldes sind es je zwei Datenströme. Allerdings liegen nun die zu einer Matrix gehörenden Eingabezellen nicht mehr alle auf einer geraden Linie, sondern



14.6. Abbildung. Detailliertes Ein-/Ausgabeschema zu Abbildung 14.3(a).

auf zwei aneinandergrenzenden Kanten.

Die Datenstrukturen in Abbildung 14.6 besitzen auch andere Neigungswinkel als in Abbildung 14.1(a). Außerdem kommen die Matrizen A und B in Abbildung 14.6 mit einer gegenüber Abbildung 14.1(a) um zwei Drittel reduzierten Datenrate pro Datenstrom an.

Mit einiger Mühe ist es auch hier prinzipiell möglich, das passende Ein-/Ausgabeschema zum vorliegenden systolischen Feld elementweise zu konstruieren. Sicherer ist jedoch der Weg über eine formale Herleitung. Die folgenden Abschnitte sind der Vorstellung der einzelnen Verfahrensschritte hierfür gewidmet.

14.3.1. Von Datenstrukturindizes zu Iterationsvektoren

Zunächst einmal muss der Zusammenhang zwischen den abstrakten Datenstrukturen und den konkreten Variableninstanzen der zuweisungsfreien Darstellung hergestellt werden.

Jedes Element der Matrix A ist gekennzeichnet durch einen Zeilenindex i und einen Spaltenindex k . Diese beiden Indizes können wir zu einem *Datenstrukturvektor* $w = (i, k)$ zusammenfassen. Das Element a_{ik} entspricht im System (14.3) den Instanzen $a(i, j, k)$ mit beliebigem j . Die Koordinaten dieser Instanzen liegen im \mathbf{R}^3 alle auf einer Geraden längst der Richtung $q = (0, 1, 0)$. Der formale Übergang vom Datenstrukturvektor (i, k) zu den Koordinaten (i, j, k) wird hier also beschrieben durch die Abbildung

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} i \\ k \end{pmatrix} + j \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (14.22)$$

Bei der in System (14.3) benutzten Form stimmt der Koordinatenvektor (i, j, k) jeder Variableninstanz genau mit dem Iterationsvektor des Trägerpunktes überein, an dem die Berechnung dieser Variableninstanz erfolgt. Wir können deshalb die Formel (14.22) auch als Beziehung zwischen Datenstrukturvektor und Iterationsvektor auffassen. Abstrakt formuliert können die betreffenden Iterationsvektoren v also nach

einer Formel

$$v = H \cdot w + \lambda \cdot q + p \quad (14.23)$$

aus dem Datenstrukturvektor w hergeleitet werden. Der affine Vektor p ist in unserem Beispiel zwar stets der Nullvektor, wird aber in allgemeineren Fällen benötigt.

Wegen $b(i, j, k) = b_{kj}$ lautet die Darstellung für die Matrix B entsprechend

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} k \\ j \end{pmatrix} + i \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (14.24)$$

Was nun Matrix C anbelangt, so steht zwar jede Variableninstanz $c(i, j, k)$ unter Umständen für einen anderen Wert. Jedoch können alle Instanzen $c(i, j, k)$ mit festem Indexpaar (i, j) als zu dem Matrixelement c_{ij} gehörig betrachtet werden, da sie direkt aus der Serialisierung des Summenoperators zur Berechnung von c_{ij} hervorgegangen sind. Für C ergibt sich entsprechend Formel (14.23) also

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} i \\ j \end{pmatrix} + k \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (14.25)$$

14.3.2. Momentaufnahmen von Datenstrukturen

Jede der drei Matrizen A, B, C wird durch zwei Richtungen auf den Datenstrukturindizes generiert: längs einer Zeile beziehungsweise längs einer Spalte. Der Differenzvektor $(0, 1)$ beschreibt dabei den Übergang von einem Element zum nächsten Element derselben Zeile, das heißt in der nächsten Spalte: $(0, 1) = (x, y + 1) - (x, y)$. Entsprechend steht der Differenzvektor $(1, 0)$ für den Übergang von einem Element zum nächsten Element derselben Spalte und nächsten Zeile: $(1, 0) = (x + 1, y) - (x, y)$.

Ein-/Ausgabeschemata der in den Abbildungen 14.1(a) und 14.6 gezeigten Form stellen *Momentaufnahmen* dar: Alle Positionen der gezeigten Datenelemente relativ zur Lage des systolischen Feldes beziehen sich auf denselben Zeitpunkt.

Wie in Abbildung 14.6 erkennbar ist, gehen die rechteckigen Matrixformen der abstrakten Datenstrukturen bei dieser Momentaufnahme in Parallelogramme über. Dies liegt an der Linearität der benutzten Raum-Zeit-Abbildung. Auch diese Parallelogramme können durch Differenzvektoren längs ihrer Kanten beschrieben werden.

Nun wollen wir Differenzvektoren Δw aus Datenstrukturvektoren in räumliche Differenzvektoren Δz für die Momentaufnahme übersetzen. Dazu müssen wir durch Wahl des Parameters λ in Beziehung (14.23) Paare konkreter Iterationsvektoren v, v' bestimmen, die mittels der gewählten Raum-Zeit-Abbildung auf denselben Zeitpunkt abgebildet werden. Welcher Zeitpunkt das nun ist, spielt zunächst keine Rolle. Wir setzen also an

$$\pi \cdot v = \pi \cdot v' \quad \text{mit} \quad v = H \cdot w + \lambda \cdot q + p \quad \text{und} \quad v' = H \cdot w' + \lambda' \cdot q + p. \quad (14.26)$$

Dies ergibt

$$\pi \cdot H \cdot (w - w') + (\lambda - \lambda') \cdot \pi \cdot q = 0, \quad (14.27)$$

also

$$\Delta\lambda = (\lambda - \lambda') = -\pi \cdot H \cdot (w - w') / \pi \cdot q. \quad (14.28)$$

Der gesuchte räumliche Differenzvektor Δz ergibt sich somit aus dem Differenzvektor der Datenstruktur $\Delta w = w - w'$ wegen der Linearität aller beteiligten Abbildungen durch

$$\Delta z = P \cdot \Delta v = P \cdot H \cdot \Delta w + \Delta\lambda \cdot P \cdot q, \quad (14.29)$$

also

$$\Delta z = P \cdot H \cdot \Delta w - (\pi \cdot H \cdot \Delta w / \pi \cdot q) \cdot P \cdot q. \quad (14.30)$$

Wir bestimmen nun aus Formel (14.30) die räumlichen Differenzvektoren Δz für die Matrix A . Wie oben eingeführt, gilt

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, q = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, P = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \end{pmatrix}, \pi = (1 \ 1 \ 1).$$

Daraus folgt mit $\pi \cdot q = 1$ zunächst

$$\Delta z = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \Delta w + \Delta\lambda \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \text{mit} \quad \Delta\lambda = - (1 \ 1) \cdot \Delta w.$$

Für die Zeilen haben wir den Differenzvektor $\Delta w = (0, 1)$, dies ergibt den räumlichen Differenzvektor $\Delta z = (2, -1)$. Entsprechend ergibt sich aus $\Delta w = (1, 0)$ für die Spalten $\Delta z = (1, -2)$. Vergleichen wir nun mit Abbildung 14.6, sehen wir, dass die Zeilen von A tatsächlich längs des Vektors $(2, -1)$ verlaufen, die Spalten längs des Vektors $(1, -2)$.

Genauso ergibt sich $\Delta z = (-1, 2)$ für die Zeilen von B und $\Delta z = (1, 1)$ für die Spalten von B . Entsprechend $\Delta z = (-2, 1)$ für die Zeilen von C und $\Delta z = (-1, -1)$ für die Spalten von C .

Damit sind wir nun in der Lage, Ein-/Ausgabeschemata — zunächst noch für jede Matrix separat — zu erstellen.

14.3.3. Montieren der Ein-/Ausgabeschemata

Zwar liegt die Erscheinungsform der Matrizen A, B, C für die Momentaufnahme damit fest. Allerdings müssen die Matrizen nun noch in ihrer Lage relativ zum systolischen Feld ausgerichtet werden (und damit auch relativ zueinander). Es gibt eine einfache grafische Methode, dies zu bewerkstelligen.

Wir wählen zunächst einen beliebigen Iterationsvektor, sagen wir $v = (1, 1, 1)$. Diesen bilden wir mittels der Projektionsmatrix P auf die Zelle seiner Berechnung ab,

$$z = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Dem Iterationsvektor $(1, 1, 1)$ sind die Berechnungen $a(1, 1, 1)$, $b(1, 1, 1)$ und

$c(1,1,1)$ zugeordnet, die wiederum zu den Datenelementen a_{11} , b_{11} und c_{11} korrespondieren. Wir legen nun die gefundenen Ein-/Ausgabeschemata der Matrizen A, B, C so über das systolische Feld, dass die zugehörigen Einträge a_{11} , b_{11} und c_{11} alle in Zelle $z = (0,0)$ stehen.

Eigentlich wären wir damit nun schon fertig. Dieses Ein-/Ausgabeschema überlappt sich so jedoch mit den Zellen des systolischen Feldes und ist deshalb nicht gut erkennbar. Wir ziehen daher die Ein-/Ausgabeschemata aller Matrizen gleichzeitig solange um jeweils einen Platz entgegen der jeweiligen Datenströme aus dem systolischen Feld zurück, bis keine Überlappung mehr auftritt. Damit erhalten wir genau das in Abbildung 14.6 vorgestellte Ein-/Ausgabeschema.

Alternativ zu dieser hübschen grafischen Methode bestünde auch wieder die Möglichkeit, die überlappungsfreie Platzierung der einzelnen Ein-/Ausgabeschemata formal zu berechnen.

Erst mit Angabe des Ein-/Ausgabeschemas kann die tatsächlich benötigte Anzahl von Zeitschritten korrekt berechnet werden. Der erste relevante Zeitschritt beginnt mit der ersten Dateneingabe. Der letzte relevante Zeitschritt endet mit der letzten Datenausgabe. Aus Abbildung 14.6 entnehmen wir für das Beispiel den Beginn der Berechnung mit Eingabe des Datenelements b_{11} im Zeitschritt 0, das Ende der Berechnung nach Ausgabe des Ergebnisses c_{35} im Zeitschritt 14. Dies ergibt insgesamt 15 Zeitschritte — fünf mehr als bei reiner Betrachtung der eigentlichen Berechnungen.

14.3.4. Raum-Zeit-Abbildung induziert Datenraten

Die Ein-/Ausgabeschemata der Matrizen A und B in Abbildung 14.1(a) zeigen einen kompakten Aufbau: Zeichnen wir die Ränder der Matrizen in die Abbildung ein, liegen im Inneren keine ungenutzten Plätze.

Anders in Abbildung 14.6. In jedem beliebigen Eingabedatenstrom wird dort jedes Datenelement von jeweils zwei ungenutzten Plätzen gefolgt. Auf die einzugehenden Matrizen bedeutet dies: Die Randzellen des systolischen Feldes erhalten nur in jedem dritten Zeitschritt ein echtes Datenelement.

Diese Eigenschaft ist ein direktes Ergebnis der jeweils angewandten Raum-Zeit-Abbildung. In beiden Beispielen sind die abstrakten Datenstrukturen selbst kompakt. Wie dicht die einzelnen Elemente dann aber im Ein-/Ausgabeschema zu liegen kommen, hängt vom absoluten Betrag der Determinante der Transformationsmatrix T ab: In jedem Ein-/Ausgabedatenstrom folgen die echten Elemente genau im Abstand von $|\det(T)|$ Plätzen aufeinander. Während in der Tat $|\det(T)| = 1$ für Abbildung 14.1 gilt, kann für Abbildung 14.6 der nun praktisch interpretierbare Wert $|\det(T)| = 3$ festgestellt werden.

Was geschieht nun mit ungenutzten Plätzen, wie in Abbildung 14.6 zu sehen? Obwohl jede Zelle des systolischen Feldes in Abbildung 14.3 eigentlich nur in jedem dritten Zeitschritt sinnvolle Arbeit leistet, macht es wenig Sinn, sie nach jedem Arbeitsschritt zwei Zeitschritte pausieren zu lassen. Bei genauer Betrachtung lässt sich feststellen, dass Werte auf den in Abbildung 14.6 mit Punkten markierten Plätzen keinen Einfluss auf die Berechnung der dargestellten Elemente c_{ij} besitzen, weil sie niemals zum Zeitpunkt der Berechnung einer Variablen $c(i, j, k)$ in der betreffenden

Zelle anlangen. Wir können die ungenutzten Plätze also einfach mit beliebigen, auch zufälligen Werten belegen, ohne das Ergebnis zu verfälschen. Es ist sogar möglich, drei verschiedene Matrixmultiplikationen gleichzeitig auf dem systolischen Feld aus Abbildung 14.3 auszuführen, ohne dass diese sich stören. Dies wird in Abschnitt 14.3.7 noch genauer ausgeführt.

14.3.5. Ein-/Ausgabeexpansion und erweitertes Ein-/Ausgabeschema

Bei genauerem Studium von Abbildung 14.6 tritt ein weiteres Problem zu Tage. Wir betrachten beispielsweise den Weg von c_{22} durch die Zellen des systolischen Feldes. Laut der Raum-Zeit-Abbildung erfolgen hierfür Berechnungen in den Zellen $(-1, 0)$, $(0, 0)$, $(1, 0)$ und $(2, 0)$. Tatsächlich durchläuft c_{22} laut dem Ein-/Ausgabeschema aus Abbildung 14.6 zuvor auch die Zelle $(-2, 0)$ und anschließend noch die Zelle $(3, 0)$.

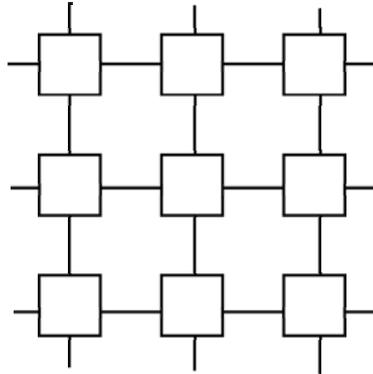
Dies kann so interpretiert werden, dass durch die gewählte Raum-Zeit-Abbildung fiktive Berechnungen in das System (14.3) eingebracht wurden, hier zum Beispiel an den neuen Trägerpunkten $(2, 2, 0)$ und $(2, 2, 5)$. Der Grund für dieses Phänomen ist die Tatsache, dass die Trägermengen der Ein-/Ausgabeoperationen nicht parallel zur gewählten Projektionsrichtung u verlaufen. Dadurch werden Ein-/Ausgabeoperationen auf Zellen projiziert, die nicht am Rand des systolischen Feldes liegen. Dort können diese Ein-/Ausgabeoperationen aber nicht direkt ausgeführt werden. Durch Fortsetzung der Transportwege, längs der beziehungsweise gegen die Datenflüsse, von diesen inneren Zellen bis zum Rand des systolischen Feldes, wird das Problem behoben. Es kommen dadurch aber neue Berechnungen und eventuell auch neue Trägerpunkte hinzu (*Ein-/Ausgabeexpansion*).

Es muss nun verhindert werden, dass die in den Zellen $(-2, 0)$ und $(3, 0)$ stattfindenden, zusätzlichen Berechnungen den eigentlichen Wert von c_{22} verfälschen. Im Falle der Matrixmultiplikation ist dies recht einfach zu bewerkstelligen (der allgemeine Fall ist dagegen schwieriger). Der generische Summenoperator besitzt ein neutrales Element, nämlich die Null. Können wir also erzwingen, dass in den ergänzten Berechnungen stets nur die Null addiert wird, entsteht kein Schaden. Wir brauchen also durch die Gestaltung neuer Eingaben nur dafür zu sorgen, dass stets mindestens ein Operand der zusätzlichen Multiplikationen Null ist.

Abbildung 14.7 zeigt ein Beispiel eines geeignet erweiterten Ein-/Ausgabeschemas. Vor und nach der Matrix A wurden die benötigten Nullelemente eingefügt. Da einzugebende Nullen wie Datenelemente gezählt werden müssen, wurde das Ein-/Ausgabeschema aus Abbildung 14.6 dazu nochmals um einen Platz aus dem systolischen Feld zurückgezogen. Die Berechnung beginnt nun schon mit dem Zeitschritt -1 , endet aber nach wie vor mit dem Zeitschritt 14. Wir erhalten somit 16 Zeitschritte Gesamtdauer.

14.3.6. Behandlung stationärer Variablen

Kommen wir zurück auf das Beispiel aus Abbildung 14.1(a). Für die Eingabe der Elemente von A und B ist keine Expansion nötig, da diese immer an Randzellen zuerst benötigt werden. Anders verhält es sich mit der Matrix C . Deren Elemente



14.7. Abbildung. Erweitertes Ein-/Ausgabeschema zu Abbildung 14.6.

werden in stationären Variablen berechnet, also immer in derselben Zelle. Damit fallen Ergebnisse c_{ij} im Inneren des systolischen Feldes an, von wo sie nach ihrer Berechnung in einem weiteren Vorgang zu Randzellen des systolischen Feldes transportiert werden müssen, da sie nur dort ausgegeben werden können.

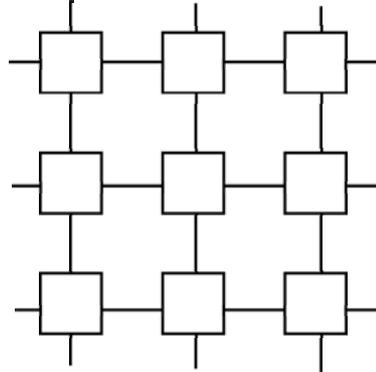
Obwohl die zu lösende Aufgabe, oberflächlich betrachtet, doch sehr der aus Abschnitt 14.3.5 ähnelt und damit zunächst recht einfach erscheint, liegt hier eine völlig andere Situation vor. Es geht nämlich nicht darum, bestehende Datenflüsse nach vorwärts und rückwärts bis zu den Rändern des systolischen Feldes zu verlängern. Bei stationären Variablen ist der Abhängigkeitsvektor ja der Nullvektor. Es kann somit gar keinen dadurch induzierten räumlichen Fluss geben. Einen solchen Fluss müssen wir erst konstruieren, wobei es viele Freiheitsgrade gibt. Auch ist dann grundsätzlich in den Zellen eine Steuerung nötig. Das Problem wird daher in Abschnitt 14.4 weiter verfolgt.

14.3.7. Verschränkung von Berechnungen

Wie leicht erkennbar, ist die *Auslastung* des systolischen Feldes in Abbildung 14.3 mit Ein-/Ausgabeschema aus Abbildung 14.7 recht dürftig. Ohne die Anlaufphase und die Schlussphase näher zu studieren, sehen wir schon, dass die durchschnittliche Auslastung des Feldes weniger als ein Drittel beträgt — schließlich leistet jede Zelle höchstens in jedem dritten Zeitschritt einen echten Berechnungsbeitrag.

Eine einfache Technik, dieses Verhalten zu verbessern, ist die *Verschränkung* von Berechnungen. Haben wir drei unabhängige Matrixmultiplikationen mit übereinstimmenden Problemparametern durchzuführen, können wir deren Daten im Abstand von nur einem Zeitschritt nacheinander eingeben, ohne an dem systolischen Feld oder seinen Zellen irgendetwas zu ändern. Abbildung 14.8 zeigt als Momentaufnahme einen Ausschnitt des systolischen Feldes mit Teilen des entsprechenden Ein-/Ausgabeschemas.

Wir wollen uns nun durch eine formale Herleitung davon überzeugen, dass diese Idee tatsächlich funktioniert. Wir modifizieren hierzu das System (14.3), indem wir die Variablen und die Trägermengen um eine vierte Dimension erweitern, die lediglich



14.8. Abbildung. Verschränkte Berechnung dreier Matrixmultiplikationen auf dem systolischen Feld aus Abbildung 14.3 (Ausschnitt).

der Unterscheidung der drei Matrixmultiplikationen dient:

Eingabeoperationen

$$\begin{aligned} a(i, j, k, n) &= a_{ik}^n & 1 \leq i \leq N_1, j = 0, 1 \leq k \leq N_3, 1 \leq n \leq 3 \\ b(i, j, k, n) &= b_{kj}^n & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3, 1 \leq n \leq 3 \\ c(i, j, k, n) &= 0 & 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0, 1 \leq n \leq 3 \end{aligned}$$

Berechnungen

$$\begin{aligned} a(i, j, k, n) &= a(i, j - 1, k, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3, 1 \leq n \leq 3 \\ b(i, j, k, n) &= b(i - 1, j, k, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3, 1 \leq n \leq 3 \\ c(i, j, k, n) &= c(i, j, k - 1, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3, 1 \leq n \leq 3 \\ &+ a(i, j - 1, k, n) * b(i - 1, j, k, n) \end{aligned}$$

Ausgabeoperationen

$$c_{ij}^n = c(i, j, k, n) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = N_3, 1 \leq n \leq 3 \quad (14.31)$$

Offensichtlich haben die Probleme für verschiedenes n in System (14.31) nichts miteinander zu tun. Dies muss nun auch im systolischen Feld so bleiben. Eine diesbezüglich für die Abbildung geeignete Raum-Zeit-Matrix ist

$$T = \begin{pmatrix} 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (14.32)$$

Die Matrix T ist nun nicht mehr quadratisch, aber das macht nichts aus. Für die Berechnung der Raumkoordinaten ist die vierte Dimension der Iterationsvektoren völlig irrelevant; sie kann deshalb durch entsprechende Null-Einträge in der letzten Spalte der ersten beiden Zeilen von T einfach ausgeblendet werden.

Die letzte Zeile von T bildet wieder der Zeitvektor π . Durch geeignete Wahl von π werden die drei zu lösenden Probleme überschneidungsfrei in das Raum-Zeit-Kontinuum eingebettet. Sich entsprechende Instanzen der Iterationsvektoren der drei Probleme werden im Abstand von je einer Zeiteinheit auf dieselben Zellen projiziert, weil der vierte Eintrag von π gleich 1 ist.

Wir berechnen nun noch die durchschnittliche Auslastung mit beziehungsweise ohne Verschränkung für die konkreten Problemparameter $N_1 = 3$, $N_2 = 5$ und $N_3 = 4$. Für eine einzelne Matrixmultiplikation sind $N_1 * N_2 * N_3 = 60$ Berechnungen auszuführen. Dabei zählen eine Multiplikation und eine zugehörige Addition als Verbundoperation, das heißt zusammen als nur eine Berechnung; Ein-/Ausgabeoperationen werden nicht mitgezählt. Das systolische Feld verfügt über 36 Zellen.

Ohne Anwendung von Verschränkung benötigt unser systolisches Feld insgesamt 16 Zeitschritte zur Berechnung einer Matrixmultiplikation. Dies ergibt eine durchschnittliche Auslastung der Zellen von

$$60/(16 * 36) = 0,104 \text{ Berechnungen pro Zeitschritt und Zelle.}$$

Bei Anwendung der beschriebenen Verschränkungstechnik dauert die Berechnung aller drei Matrixmultiplikationen nur zwei Zeitschritte länger, also 18 Zeitschritte. Die Anzahl der durchgeführten Berechnungen hat sich dadurch jedoch verdreifacht, wir erhalten also eine durchschnittliche Auslastung der Zellen von

$$3 * 60/(18 * 36) = 0,278 \text{ Berechnungen pro Zeitschritt und Zelle.}$$

Wir konnten somit die Auslastung durch Verschränkung um 167 Prozent steigern!

Übungen

14.3-1. Leiten Sie die räumlichen Differenzvektoren der Matrizen B und C für das in Abbildung 14.6 gezeigte Ein-/Ausgabeschema formal aus der Beziehung (14.30) her.

14.3-2. Stellen Sie das erweiterte Ein-/Ausgabeschema zu Abbildung 14.6 für den Fall dar, dass jeweils beide Operanden der überschüssigen Multiplikationen Null gesetzt werden sollen.

14.3-3. Wenden Sie die in Abschnitt 14.3 vorgestellten Techniken auf das systolische Feld aus Abbildung 14.1 an.

14.3-4.* Beweisen Sie die im Abschnitt 14.3.7 behaupteten Eigenschaften der speziellen Raum-Zeit-Abbildung (14.32) bezüglich des Systems (14.31).

14.4. Steuerungsaspekte

Bisher sind wir davon ausgegangen, dass sich die Zellen eines systolischen Feldes in jedem Zeitschritt völlig gleich verhalten. Zwar gibt es einige interessante Beispiele solcher systolischer Felder. Im Allgemeinen müssen die Zellen jedoch durch

eine Steuerung in verschiedene *Betriebsmodi* gebracht werden können. Wir studieren hierzu nachfolgend einige typische Situationen.

14.4.1. Steuerungslose Zellen

Die Zelle in Abbildung 14.3(b) enthält die Register RA, RB, RC, die bei Aktivierung mit dem globalen Taktsignal die jeweils an ihren Eingängen anliegenden Signale übernehmen und diese anschließend an ihren Ausgängen wiedergeben. Abgesehen von dieser generellen Aktivierung ist die Funktionalität der Zelle aber in allen Zeitschritten immer die gleiche: Es wird auf den drei eingehenden Operanden A , B , C eine *multiply-add*-Operation durchgeführt, deren Ergebnis einer Nachbarzelle zugeführt wird; parallel dazu werden die Operanden A und B an zwei andere Nachbarzellen weitergereicht. Somit weist diese Zelle keinerlei Steuerung auf.

Die Initialwerte $c(i, j, 0)$ für die Ausführung des generischen Summenoperators, die hier nicht unbedingt Null sein müssen, werden dem systolischen Feld laut Abbildung 14.6 in Form von Eingabeströmen zugeführt, die Endergebnisse $c(i, j, N_3)$ fließen in derselben Richtung über den Rand des Feldes ab. Die Ein-/Ausgabe ist also impliziter Bestandteil der Funktionalität der Zellen nach Abbildung 14.3(b). Der Preis für diese äußerst einfache, steuerungslose Zellfunktion ist die Beschränkung aller drei Matrixdimensionen: Eine $(M_1 \times M_3)$ -Matrix A kann nur dann auf dem systolischen Feld der Abbildung 14.3 mit einer $(M_3 \times M_2)$ -Matrix B multipliziert werden, wenn bei festgelegten Feldparametern N_1, N_2, N_3 die Beziehungen $M_1 \leq N_1$, $M_2 \leq N_2$ und $M_3 \leq N_3$ gelten.

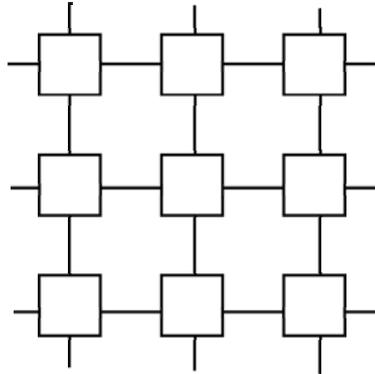
14.4.2. Zellen mit Globalsteuerung

Die Vorgaben für das systolische Feld in Abbildung 14.1 sind diesbezüglich weniger streng: Zwar sind auch hier M_1 und M_2 durch $M_1 \leq N_1$ und $M_2 \leq N_2$ eingeschränkt, es gibt jedoch keine Beschränkung für M_3 . Problemparameter, die trotz vorgegebener Feldparameter unbeschränkt sind, können sich nur in der Zeit, nicht jedoch im Raum manifestieren. Dies erzwingt die Verwendung *stationärer Variablen*.

Jedes einer stationären Variablen zugeordnete Register muss zu Beginn einer neuen Berechnung in einen von der vorhergehenden Berechnung unabhängigen Grundzustand gebracht werden. Im Falle der systolischen Zelle aus Abbildung 14.3(b) ist dies das Register RC. Durch ein globales Signal, vergleichbar dem Taktsignal, kann das Register RC in allen Zellen gleichzeitig gelöscht, das heißt auf den Wert Null gebracht werden. Abbildung 14.9 zeigt einen Feldaufbau und eine Zellstruktur für diese Idee.

14.4.3. Lokale Steuerung

Leider funktioniert das Prinzip der globalen Steuerung alleine bei der Matrixmultiplikation nicht ausreichend. Dem in Abbildung 14.1 präsentierten systolischen Feld fehlen nämlich sogar zwei wesentliche Eigenschaften: Es werden einerseits die Variablen $c(i, j, 0)$ nicht initialisiert, andererseits aber auch die Ergebnisse c_{ij} nicht über den Rand ausgegeben.



14.9. **Abbildung.** Rücksetzen von Registern durch Globalsteuerung: (a) Feldaufbau; (b) Zellstruktur.

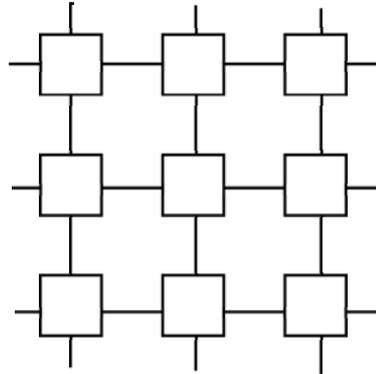
Zunächst erscheint die zusätzliche Aufgabe recht einfach, die Ergebnisse zum Rand zu führen: Nach Abschluss der Berechnung eines Elementes c_{ij} werden die von Zelle (i, j) zu den Nachbarzellen $(i, j+1)$ und $(i+1, j)$ führenden Verbindungen nicht mehr für das Weiterreichen von Elementen der Matrizen A und B benötigt. Sie können dann also für andere Zwecke benutzt werden. Zum Beispiel könnten wir alle Elemente von C über die nach unten führenden Verbindungen zum unteren Rand des systolischen Feldes führen.

Leider stellt sich heraus, dass noch nicht abgeschlossene Berechnungsvorgänge im unteren Teil des Feldes die Durchleitung der Ergebnisse aus dem oberen Teil behindern. Würde das im Zeitschritt $i+j+N_3$ berechnete Ergebnis c_{ij} im nächsten Zeitschritt an die Zelle $(i+1, j)$ weitergeleitet, käme es zu einer Wertekollision: Da von Zelle $(i+1, j)$ nur ein Wert pro Zeitschritt über den unteren Kanal ausgegeben werden kann, müsste dann entweder c_{ij} warten oder das in Zelle $(i+1, j)$ berechnete Ergebnis. Dieses Problem würde sich nach unten über alle Zellen fortsetzen.

Zur Abhilfe können wir die Weitergabe von c_{ij} verzögern. Benötigt c_{ij} für das Durchlaufen einer Zelle nicht einen, sondern jeweils zwei Zeitschritte, bleiben Kollisionen aus. Die Ergebnisse laufen dann im Abstand eines Zeitschritts über dieselbe Verbindung hintereinander her. An der unteren Randzelle einer Spalte erscheint zunächst das Element der letzten Zeile dieser Spalte, danach das vorletzte, zuletzt das erste. Wir erhalten also das in Abbildung 14.10 gezeigte Ein-/Ausgabeschema.

Woher weiß eine Zelle, wann sie über den unteren Kanal statt Elementen der Matrix B auf einmal Elemente der Matrix C weitergeben soll? Wir können diese Aufgabe durch eine Kombination aus Globalsteuerung und lokaler Steuerung der Zelle durch Automaten lösen:

Wenn wir bei Eingabe der letzten Werte von A und B in die Zelle $(1,1)$ allen Zellen ein globales Signal senden, das diese Tatsache mitteilt, kann in jeder Zelle ein Zähler gestartet werden, der die Anzahl der noch auszuführenden Berechnungsschritte angibt. In Zelle (i, j) sind dabei jeweils noch $i+j-1$ Schritte auszuführen, bevor zur reinen Durchleitung umgeschaltet wird. Das bereits besprochene globale Rücksetzsignal schaltet später wieder in den Berechnungsmodus zurück.

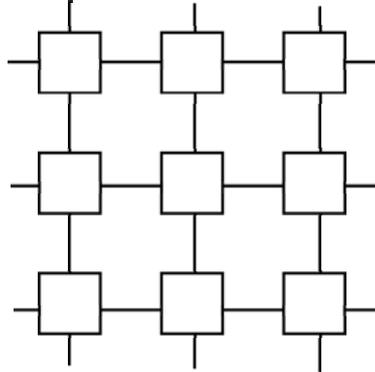


14.10. **Abbildung.** Ein-/Ausgabeschema bei verzögerter Ergebnisausgabe.

Ein nach diesem Prinzip funktionierendes systolisches Feld ist in Abbildung 14.11 zu sehen. Feldaufbau und Verbindungsstruktur sind im wesentlichen gleich geblieben. Jede Zelle kann nun in zwei Zuständen betrieben werden, zwischen denen eine Steuerlogik umschaltet:

- (a) Im *Berechnungsmodus* wird (wie bisher) das Ergebnis der Addition in Register RC geschrieben. Gleichzeitig wird der für die Multiplikation verwendete Operand aus Register RB über den unteren Kanal der Zelle durchgereicht.
- (b) Im *Propagierungsmodus* werden die Register RB und RC in Reihe geschaltet. Die einzige Funktion der Zelle in diesem Modus ist es, jeden über den oberen Kanal eingelesenen Wert um zwei Zeitschritte verzögert über den unteren Kanal weiterzureichen.

Der erste im Propagierungsmodus von der Zelle (i, j) ausgegebene Wert ist der letzte berechnete und dann in Register RC abgelegte Wert, also das Ergebnis c_{ij} . Alle weiteren ausgegebenen Werte sind durchgeleitete Ergebnisse weiter oben liegender Zellen. Eine formale Beschreibung des in Abbildung 14.11 implementierten Algorithmus liefert das zuweisungsfreie System (14.33).



14.11. **Abbildung.** Kombination von Globalsteuerung und lokaler Steuerung: (a) Feldaufbau; (b) Zellstruktur.

Eingabeoperationen

$$\begin{aligned} a(i, j, k) &= a_{ik} & 1 \leq i \leq N_1, j = 0, 1 \leq k \leq N_3 \\ b(i, j, k) &= b_{kj} & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= 0 & 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0 \end{aligned}$$

Berechnungen

$$\begin{aligned} a(i, j, k) &= a(i, j - 1, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ b(i, j, k) &= b(i - 1, j, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= c(i, j, k - 1) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ &+ a(i, j - 1, k) * b(i - 1, j, k) \end{aligned}$$

Durchleitung

$$\begin{aligned} b(i, j, k) &= c(i, j, k - 1) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq i + N_3 \\ c(i, j, k) &= b(i - 1, j, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq i - 1 + N_3 \end{aligned}$$

Ausgabeoperationen

$$c_{1+N_1+N_3-k,j} = b(i, j, k) \quad i = N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq N_1 + N_3 \quad (14.33)$$

Es bleibt noch zu klären, wie die Steuersignale einer Zelle in diesem Modell genau erzeugt werden. Zunächst muss die Zelle ein Zustands-Flipflop besitzen, das den gerade geschalteten Modus angibt. Der Ausgang dieses Flipflops wird mit den Steuereingängen der beiden Multiplexer aus Abbildung 14.11(b) verbunden. Durch das globale Rücksetzsignal wird nicht nur das Register RC der Zelle zurückgesetzt, sondern auch das Zustands-Flipflop: Die Zelle arbeitet nun im Berechnungsmodus.

Trifft das globale Endesignal ein, wird ein Abwärts-Zähler in der Zelle gestartet und in jedem Zeitschritt um 1 vermindert. Als Startwert des Zählers ist — zellen-

abhängig — der Wert $i + j - 1$ eingestellt. Mit Erreichen des Zählerwerts Null wird das Flipflop gesetzt: Die Zelle geht in den Propagierungsmodus über.

Der letzte vor dem Rücksetzen vom Register RB an das Register RC einer Zelle durchgeleitete Wert kann nun auch als frei wählbarer Initialwert für das nächste in der Zelle zu berechnende Skalarprodukt benutzt werden, wenn auf das direkte Rücksetzen des Registers RC verzichtet wird. Wir berechnen dann, wie schon beim systolische Feld in Abbildung 14.3, sogar das allgemeinere Problem

$$C = A \cdot B + D, \quad (14.34)$$

nach den Formeln

Eingabeoperationen

$$\begin{aligned} a(i, j, k) &= a_{ik} & 1 \leq i \leq N_1, j = 0, 1 \leq k \leq N_3 \\ b(i, j, k) &= b_{kj} & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= d_{ij} & 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0 \end{aligned}$$

Berechnungen

$$\begin{aligned} a(i, j, k) &= a(i, j - 1, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ b(i, j, k) &= b(i - 1, j, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ c(i, j, k) &= c(i, j, k - 1) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ &+ a(i, j - 1, k) * b(i - 1, j, k) \end{aligned}$$

Durchleitung

$$\begin{aligned} b(i, j, k) &= c(i, j, k - 1) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq i + N_3 \\ c(i, j, k) &= b(i - 1, j, k) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq i - 1 + N_3 \end{aligned}$$

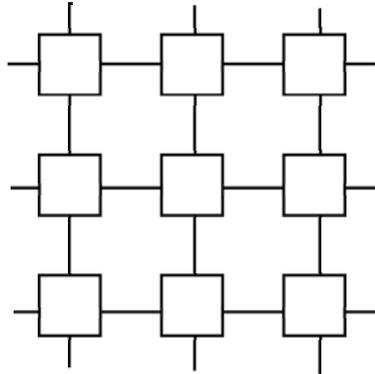
Ausgabeoperationen

$$c_{1+N_1+N_3-k,j} = b(i, j, k) \quad i = N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq N_1 + N_3 \quad (14.35)$$

14.4.4. Verteilte Steuerung

Die in Abbildung 14.11 gezeigte Methode hat noch die folgenden Nachteile:

- Es werden globale Steuersignale verwendet. Dies erfordert eine hohe technische Genauigkeit.
- Jede Zelle benötigt einen Zähler. Der Aufwand für einen Zähler mit Zählregister ist erheblich.
- Der Startwert des Zählers ist nicht für alle Zellen gleich. Damit muss jede Zelle individuell entworfen und implementiert werden.
- Mit der Eingabe eines neuen Problems muss gewartet werden, bis das letzte Ergebnis das systolische Feld verlassen hat.



14.12. Abbildung. Matrixmultiplikation auf rechteckigem systolischem Feld, mit Ergebnisausgabe und verteilter Steuerung: (a) Feldaufbau; (b) Zellstruktur.

Diese Nachteile lassen sich vermeiden, wenn Steuersignale wie Daten weitergereicht werden, also eine verteilte Steuerung benutzt wird. Wir behalten hierzu die Verschaltung der Register **RB** und **RC** mit den Multiplexern aus Abbildung 14.11(b) bei, erzeugen jedoch keine Steuersignale in den Zellen selbst; auch das globale Rücksetzsignal entfällt. Stattdessen werden den Zellen die nötigen Steuersignale von außen zugeführt, in einem zusätzlichen Register **RS** (mit Breite nur 1 Bit!) gespeichert und von dort an Nachbarzellen geeignet weiterverteilt. Die Erzeugung der eigentlichen Steuersignale übernimmt der Wirtsrechner, die Einspeisung erfolgt ausschließlich über Randzellen. Abbildung 14.12(a) zeigt den dazu erforderlichen Feldaufbau, Abbildung 14.12(b) die modifizierte Zellstruktur.

Die Umschaltung in den Propagierungsmodus erfolgt für die Zellen einer Spalte nach unten in jeweils einem Zeitschritt Abstand. Daher genügt die durch das Register **RS** alleine bewirkte Verzögerung.

Das Rücksetzen in den Berechnungsmodus erfolgt über dieselbe Steuerleitung und wird daher ebenfalls mit einem Zeitschritt Verzögerung pro Zelle vorgenommen. Da die Resultate c_{ij} jedoch nur mit halber Geschwindigkeit nach unten abfließen, muss mit dem Rücksetzen der Zellen ausreichend lange gewartet werden: Wird eine Zelle im Zeitschritt t in den Berechnungsmodus geschaltet, so geht sie im Zeitschritt $t + N_3$ in den Propagierungsmodus über und schaltet im Zeitschritt $t + N_1 + N_3$ wieder zurück in den Berechnungsmodus.

Wie sich also zeigt, unterscheidet sich die verteilt gesteuerte Variante des systolischen Feldes von der lokal/globalen Steuerung im makroskopischen Zeitverhalten. Während das systolische Feld aus Abbildung 14.12 alle $N_1 + N_3$ Zeitschritte mit der Berechnung eines neuen Problems (14.34) beginnen kann, ist dies für das systolische Feld aus Abbildung 14.11 nur alle $2N_1 + N_2 + N_3 - 2$ Zeitschritte möglich. Die Zeitdifferenz $N_1 + N_3$ beziehungsweise $2N_1 + N_2 + N_3 - 2$ heißt *Periode*, ihr Kehrwert der *Durchsatz*.

System (14.36) beschreibt die formalen Zusammenhänge zwischen verteilter Steuerung und Berechnung. Es wird dabei von einer beliebig langen Folge dichtestmöglich aufeinander folgender Matrixmultiplikationen ausgegangen, die zusätzliche Iterationsvariable n ist deshalb unbeschränkt.

Steuerung

$$\begin{aligned} s(i, j, k, n) &= 0 & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ s(i, j, k, n) &= 1 & i = 0, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq N_1 + N_3 \\ s(i, j, k, n) &= s(i - 1, j, k, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3 \end{aligned}$$

Dateneingabe

$$\begin{aligned} a(i, j, k, n) &= a_{ik}^n & 1 \leq i \leq N_1, j = 0, 1 \leq k \leq N_3 \\ b(i, j, k, n) &= b_{kj}^n & i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \\ b(i, j, k, n) &= d_{N_1+N_3+1-k, j}^{n+1} & i = 0, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq N_1 + N_3 \end{aligned}$$

Variablen mit Alias

$$c(i, j, k, n) = c(i, j, N_1 + N_3, n - 1) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, k = 0$$

Datenoperationen

$$\begin{aligned} a(i, j, k, n) &= a(i, j - 1, k, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3 \\ b(i, j, k, n) &= \begin{cases} b(i - 1, j, k, n) & : s(i - 1, j, k, n) = 0 \\ c(i, j, k - 1, n) & : s(i - 1, j, k, n) = 1 \end{cases} & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3 \\ c(i, j, k, n) &= \begin{cases} c(i, j, k - 1, n) \\ + a(i, j - 1, k, n) \\ * b(i - 1, j, k, n) & : s(i - 1, j, k, n) = 0 \\ b(i - 1, j, k, n) & : s(i - 1, j, k, n) = 1 \end{cases} & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3 \end{aligned}$$

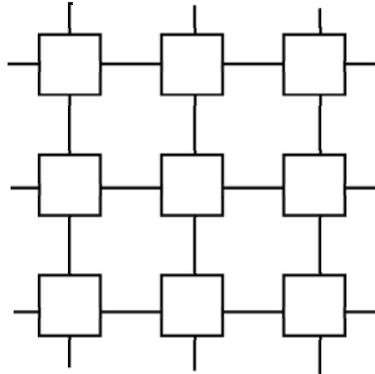
Datenausgabe

$$c_{1+N_1+N_3-k, j}^n = b(i, j, k, n) \quad i = N_1, 1 \leq j \leq N_2, 1 + N_3 \leq k \leq N_1 + N_3 \quad (14.36)$$

Formel (14.37) zeigt die zugehörige Raum-Zeit-Matrix, bei der einer der Einträge nicht konstant ist, sondern von Problemparametern abhängt.

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & N_1 + N_3 \end{pmatrix} \quad (14.37)$$

Es fällt auf, dass auch die Zellen einer Zeile nach rechts im Abstand eines Zeitschritts umgeschaltet werden müssen. Unter Aufgabe der vollständigen Regularität des systolischen Feldes könnte dieser Umstand genutzt werden, um Steuersignale nur über Zelle (1, 1) einzuspeisen und damit den Wirtsrechner zu entlasten. Wir würden dann die Steuerung folgendermaßen abändern:



14.13. Abbildung. Matrixmultiplikation auf rechteckigem systolischem Feld, mit Ergebnisausgabe und verteilter Steuerung: (a) Felddaufbau; (b) Zelle des oberen Randes; (c) Zelle des übrigen Gebietes.

Steuerung

$$\begin{aligned}
 s(i, j, k, n) &= 0 & i = 1, j = 0, 1 \leq k \leq N_3 \\
 s(i, j, k, n) &= 1 & i = 1, j = 0, 1 + N_3 \leq k \leq N_1 + N_3 \\
 s(i, j, k, n) &= s(i - 1, j, k, n) & 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3
 \end{aligned}$$

...

Variablen mit Alias

$$s(i, j, k, n) = s(i + 1, j - 1, k, n) \quad i = 0, 1 \leq j \leq N_2, 1 \leq k \leq N_1 + N_3$$

...

(14.38)

Abbildung 14.13 zeigt das Ergebnis dieser Modifikation. Es gibt nun zwei Zelltypen: die Zellen am oberen Rand des systolischen Feldes (Abbildung 14.13(b)) und alle übrigen Zellen (Abbildung 14.13(c)). Auch die Verbindungsstruktur am oberen Rand des systolischen Feldes ist geringfügig anders als im regulären Bereich.

14.4.5. Das Zellprogramm als lokale Sichtweise

Wir können die Funktionalität einer Zelle auch durch ein *Zellprogramm* ausdrücken. Dies ist insbesondere dann interessant, wenn wir ein programmierbares systolisches Feld vorliegen haben, dessen Zellen in der Tat durch ein repetitiv ablaufendes Programm gesteuert werden.

Wie die globale Sicht, also die Architektur des systolischen Feldes, wird die lokale Sicht, das heißt das Zellprogramm, ebenfalls bereits durch die *Raum-Zeit-Abbildung* festgelegt. Sie ergibt sich aber hierdurch nur in impliziter Form und muss daher erst durch eine mathematische Transformation in eine explizite Form überführt werden,

die dann als Zellprogramm tauglich ist.

Die Instanzen der Programmvariablen sind in allgemeiner Form durch *Indexausdrücke* beschrieben, die Bezug auf die Iterationsvariablen nehmen. Nehmen wir beispielsweise die Gleichung

$$c(i, j, k) = c(i, j, k-1) + a(i, j-1, k) * b(i-1, j, k) \quad 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3$$

aus System (14.3). Die Instanz $c(i, j, k-1)$ der Programmvariablen c besitzt die Indexausdrücke i , j und $k-1$, die als Funktionen der Iterationsvariablen i, j, k aufgefasst werden können.

Wie wir gesehen haben, geht durch Anwendung der Raum-Zeit-Abbildung (14.11) mit Transformationsmatrix T aus (14.13) die Menge der Iterationsvektoren (i, j, k) der Quantifizierung in eine Menge von Raum-Zeit-Koordinaten (x, y, t) über,

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} = T \cdot \begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} i \\ j \\ k \end{pmatrix}. \quad (14.39)$$

Da jede Zelle durch ihre Raumkoordinaten (x, y) beschrieben wird und ihr Zellprogramm Bezug auf die laufende Zeit t nehmen muss, sind die in den Indexausdrücken der Programmvariablen vorkommenden Iterationsvariablen i, j, k nicht mehr brauchbar und müssen in die neuen Koordinaten x, y, t umgeschrieben werden. Wir drücken dazu die Iterationsvariablen i, j, k mittels der Umkehrabbildung der Raum-Zeit-Abbildung in (14.39) als Funktionen der Raum-Zeit-Koordinaten (x, y, t) aus,

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = T^{-1} \cdot \begin{pmatrix} x \\ y \\ t \end{pmatrix} = \frac{1}{3} \cdot \begin{pmatrix} -1 & -2 & 1 \\ -1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ t \end{pmatrix}. \quad (14.40)$$

Eine solche Umkehrabbildung existiert, wenn die Raum-Zeit-Abbildung auf der Trägermenge injektiv ist — und das sollte sie stets sein, da sonst mehrere Berechnungsinstanzen in einer Zelle im gleichen Zeitschritt bearbeitet werden müssten. Die Umkehrbarkeit wird im Beispiel auch ohne Bezug auf die Trägermenge durch die quadratische, nicht singuläre Matrix T garantiert. Bezogen auf Zeitvektor π und Projektionsrichtung u genügt die Eigenschaft $\pi \cdot u \neq 0$.

Nach dem Ersetzen der Iterationsvariablen durch Raum-Zeit-Koordinaten, interpretierbar als eine *Transformation der Trägermengen*, ergeben sich im Allgemeinen unschöne neue Indexausdrücke. So wird aus $c(i, j, k-1)$ hier

$$c((-x - 2 * y + t)/3, (-x + y + t)/3, (2 * x + y + t)/3).$$

Wir können aber durch eine nachgeschaltete *Transformation der Indexmengen* die Instanzen der Programmvariablen so umbenennen, dass der Bezug zu Zelle und Zeit klarer erscheint. Insbesondere ist es erstrebenswert, das Gleichungssystem wieder in *Ausgabenormalform* zu bringen, also die im Zeitschritt t in der Zelle (x, y) berechneten Ergebnisse durch Instanzen (x, y, t) der Programmvariablen zu bezeichnen.

Das Verständnis für diese Vorgänge erlangen wir am besten durch eine abstrakte mathematische Formulierung, die wir anschließend auf unsere spezielle Situation anpassen. Sei eine quantifizierte Gleichung mit Programmvariablen r und s sowie

Trägermenge S durch

$$r(\psi_r(v)) = \mathcal{F}(\dots, s(\psi_s(v)), \dots) \quad v \in S \quad (14.41)$$

gegeben. Die *Indexfunktionen* ψ_r beziehungsweise ψ_s generieren die Instanzen der Programmvariablen als Tupel von Indexausdrücken.

Durch Transformation der Trägermenge mittels einer bezüglich S injektiven Funktion φ geht (14.41) über in

$$r(\psi_r(\varphi^{-1}(e))) = \mathcal{F}(\dots, s(\psi_s(\varphi^{-1}(e))), \dots) \quad e \in \varphi(S), \quad (14.42)$$

wobei φ^{-1} eine Funktion ist, die auf $\varphi(S)$ die Umkehrfunktion von φ bildet. Die neuen Indexfunktionen sind $\psi_r \circ \varphi^{-1}$ und $\psi_s \circ \varphi^{-1}$.

Transformationen von Indexmengen haben mit der Trägermenge nichts zu tun und können für jede Programmvariable separat ausgeführt werden, da sie nur die Instanzen dieser einen Programmvariablen konsistent umbenennen. Mit solchen Umbenennungen ϑ_r und ϑ_s wird (14.42) zu

$$r(\vartheta_r(\psi_r(\varphi^{-1}(e)))) = \mathcal{F}(\dots, s(\vartheta_s(\psi_s(\varphi^{-1}(e))))), \dots) \quad e \in \varphi(S). \quad (14.43)$$

Soll sich die Ausgabenormalform ergeben, muss $\vartheta_r \circ \psi_r \circ \varphi^{-1}$ die Identität sein.

Im einfachsten Fall (im Beispiel vorliegend) ist ψ_r immer die Identität und ψ_s eine affine Abbildung der Form $\psi_s(v) = v - d$ mit konstantem d , dem bereits bekannten Abhängigkeitsvektor. ψ_r kann auf dieselbe Weise dargestellt werden, mit $d = \vec{0}$. Die Transformation der Trägermengen erfolgt durch die Raum-Zeit-Abbildung $\varphi(v) = T \cdot v$ mit einer invertierbaren Matrix T . Für alle Indextransformationen wird übereinstimmend $\vartheta = \varphi$ gewählt. Damit gestaltet sich (14.43) als

$$r(e) = \mathcal{F}(\dots, s(e - T \cdot d), \dots) \quad e \in T(S). \quad (14.44)$$

Für die Generierung eines *Zellprogramms* müssen nun in jedem Zeitschritt klar sein: die auszuführenden Operationen, die Herkunft der Daten und die Bestimmungsorte der Resultate (aus Assemblerprogrammen bekannt als `opc`, `src`, `dst`).

Die jeweils auszuführende Operation (`opc`) ergibt sich direkt aus der Funktion \mathcal{F} . Für Zellen mit Steuerung muss noch der Zeitraum ermittelt werden, in dem dieses spezielle \mathcal{F} ausgeführt wird. Dieser kann in Abhängigkeit von den Raumkoordinaten durch Projektion von $T(S)$ auf die Zeitachse bestimmt werden, für allgemeines polyedrisches S zum Beispiel mit einer „Fourier-Motzkin-Elimination“.

In System (14.44) ergibt sich der neue Abhängigkeitsvektor $T \cdot d$, der aus zwei Komponenten besteht, einem (vektoriellen) räumlichen und einem (skalaren) zeitlichen Anteil. Der räumliche Anteil Δz beschreibt als Differenzvektor, von welcher Nachbarzelle der Operand berechnet wurde. Wir können diese Information für die Eingabe des Operanden in die Zelle z direkt in einen Kanalbezeichner mit Kanalposition $-\Delta z$ übersetzen (`src`). Entsprechend muss die den Operanden berechnende Zelle $z - \Delta z$ diesen Wert auf einen Kanal mit Kanalposition Δz ausgeben (`dst`).

Der zeitliche Anteil von $T \cdot d$ gibt als Zeitdifferenz Δt an, wann die Berechnung des Operanden geschah. Diese Information ist für die lesende Zelle z bedeutungslos, da von Nachbarzellen immer Ausgaben des unmittelbar vorhergehenden Zeitschritts

gelesen werden. Jedoch muss für $\Delta t > 1$ der Operand von der ihn berechnenden Zelle $z - \Delta z$ über $\Delta t - 1$ Zeitschritte zwischengespeichert werden. Dies kann zum Beispiel durch $\Delta t - 1$ Kopieranweisungen im Zellprogramm der berechnenden Zelle $z - \Delta z$ wiedergegeben werden, mittels derer der Wert des Operanden über $\Delta t - 1$ Register bis zu seiner endgültigen Ausgabe aus der Zelle fortgeschrieben wird.

Die Anwendung dieser Methode auf (14.36) mit T aus (14.37) ergibt

$$\begin{aligned}
 s(x, y, t) &= s(x-1, y, t-1) \\
 a(x, y, t) &= a(x, y-1, t-1) \\
 b(x, y, t) &= \begin{cases} b(x-1, y, t-1) & : s(x-1, y, t-1) = 0 \\ c(x, y, t-1) & : s(x-1, y, t-1) = 1 \end{cases} \\
 c(x, y, t) &= \begin{cases} c(x, y, t-1) \\ + a(x, y-1, t-1) \\ * b(x-1, y, t-1) & : s(x-1, y, t-1) = 0 \\ b(x-1, y, t-1) & : s(x-1, y, t-1) = 1 \end{cases}
 \end{aligned} \tag{14.45}$$

Die Iterationsvariable n ist hier nur für das Ein-/Ausgabeschema von Belang und kann für die Transformation auf einen festen Wert gesetzt werden. Das zugehörige Zellprogramm, das in jedem Zeitschritt abläuft, lautet:

ZELLPROGRAMM ZU ABBILDUNG 14.12

```

1  RS ← C(-1,0)(0)
2  RA ← C(0,-1)
3  RB ← C(-1,0)(1 : N)
4  C(1,0)(0) ← RS
5  C(0,1) ← RA
6  if RS = 1
7    then C(1,0)(1 : N) ← RC
8         RC ← RB
9  else C(1,0)(1 : N) ← RB
10    RC ← RC + RA * RB

```

Die Kanalbezeichner stehen für lokale Ein-/Ausgaben der Zelle. Ihre Form erhalten sie von der Lage des Kanals relativ zur Zellmitte. $C(0, -1)$ liegt am linken Rand der Zelle, $C(0, 1)$ am rechten Rand, $C(-1, 0)$ ist oben, $C(1, 0)$ unten. Hinter dem Kanalbezeichner kann noch ein Bitbereich angegeben werden: $C(-1, 0)(0)$ bedeutet ausschließlich Bit 0 des Kanals, $C(-1, 0)(1 : N)$ die Bits 1 bis N desselben. Die Bezeichner RA, RB, ... stehen für Register der Zelle.

Entsprechende Anwendung auf (14.35) mit T aus (14.12) ergibt

$$\begin{aligned}
 a(x, y, t) &= a(x, y - 1, t - 1) & 1 + x + y \leq t \leq x + y + N_3 \\
 b(x, y, t) &= b(x - 1, y, t - 1) & 1 + x + y \leq t \leq x + y + N_3 \\
 c(x, y, t) &= c(x, y, t - 1) & 1 + x + y \leq t \leq x + y + N_3 \\
 &+ a(x, y - 1, t - 1) * b(x - 1, y, t - 1) \\
 b(x, y, t) &= c(x, y, t - 1) & x + y + 1 + N_3 \leq t \leq 2 * x + y + N_3 \\
 c(x, y, t) &= b(x - 1, y, t - 1) & x + y + 1 + N_3 \leq t \leq 2 * x + y - 1 + N_3
 \end{aligned}
 \tag{14.46}$$

Wir sehen nun sehr schön die Vorteile der verteilten Steuerung. Das Zellprogramm zu (14.45) ist relativ zu einem beliebigen Zeitschritt t formuliert, benötigt also keine Reaktion auf globale Steuersignale, keine Zählregister, keine Zähloperationen und keine Kodierung der lokalen Zellkoordinaten.

Übungen

14.4-1. Geben Sie Ein-/Ausgabeschemata zur Durchführung zweier möglichst dicht aufeinander folgender Berechnungen laut System (14.35) für die in den Abbildungen 14.11 beziehungsweise 14.12 vorgestellten systolischen Felder an.

14.4-2. Wie müsste das systolische Feld aus Abbildung 14.12 abgeändert werden, um die Berechnung von Matrixprodukten mit Parametern $M_1 < N_1$ oder $M_2 < N_2$ effizient zu unterstützen?

14.4-3. Wie lautet das Zellprogramm für das systolische Feld in Abbildung 14.3?

14.4-4.* Welchen Durchsatz erreicht das systolische Feld aus Abbildung 14.3 für die konkreten Werte von N_1, N_2, N_3 ? Welchen für allgemeines N_1, N_2, N_3 ?

14.4-5.* Modifizieren Sie das systolische Feld aus Abbildung 14.1 so, dass die stationären Variablen nach Abschluss der Berechnung über zusätzliche Verbindungen nach rechts unten (also von Zelle (i, j) zu Zelle $(i + 1, j + 1)$) ausgegeben werden. Geben Sie ein zu (14.35) korrespondierendes zuweisungsfreies System an, das eine Beschreibung des zugehörigen Verhaltens liefert. Wie sieht das Ein-/Ausgabeschema aus? Welche Periode wird erreicht?

14.5. Lineare systolische Felder

Die Überlegungen aus den oben stehenden Abschnitten sind auf zweidimensionale systolische Felder zugeschnitten. Sie lassen sich jedoch auch auf eindimensionale systolische Felder übertragen.

Der wesentlichste Unterschied zwischen beiden Formen betrifft den *Rand* des systolischen Feldes. Eindimensionale systolische Felder können einerseits als ausschließlich aus Randzellen bestehend aufgefasst werden, Eingaben vom und Ausgaben zum Wirtsrechner sind dann ohne zusätzliche Maßnahmen möglich. Andererseits verfügen

sie über eine volle Dimension und eine nur formale Dimension. Bei Kommunikation längs der Laufrichtung des linearen systolischen Feldes stellen sich damit unter Umständen doch ähnliche Fragen wie in Abschnitt 14.3.5. Schließlich kann der Rand des linearen systolischen Feldes auch noch ganz anders definiert werden, nämlich als nur aus den beiden Endzellen bestehend.

14.5.1. Matrix-Vektor-Multiplikation

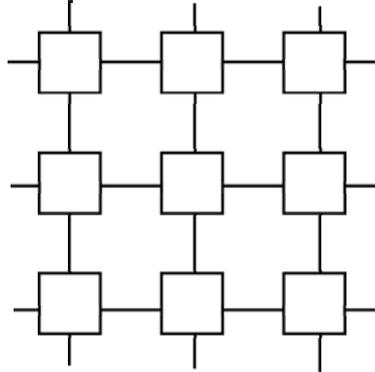
Erhält in Abbildung 14.1 einer der Problemparameter N_1 oder N_2 den Wert 1, wird die Matrixmultiplikation zur Multiplikation einer Matrix mit einem Vektor: von links für $N_1 = 1$ beziehungsweise von rechts für $N_2 = 1$. Das zweidimensionale systolische Feld *degeneriert* dann zum eindimensionalen systolischen Feld. Der zu multiplizierende Vektor wird als Eingabedatenstrom über eine Endzelle des linearen systolischen Feldes zugeführt. Die Matrixelemente werden simultan über die gesamte Längsseite in das Feld eingebracht.

Wie bei der vollen Matrixmultiplikation fallen die Ergebnisse stationär an. Sie können nun aber entweder längs des Feldes über eine der Endzellen entnommen werden oder direkt von den sie berechnenden Zellen an den Wirtsrechner ausgegeben werden. Dies ergibt unterschiedliche Steuerungsmechanismen, Zeitschemata und Gesamtausführungszeiten.

Wäre es nun auch möglich gewesen, *alle* Eingaben über Endzellen vorzunehmen? Jedenfalls nicht, wenn die Gesamtausführungszeit die Komplexität $O(N)$ besitzen soll. Die einzugebende Matrix besitzt $O(N^2)$ Elemente, es müssen also $O(N)$ Elemente pro Zeitschritt eingegeben werden. Die Anzahl der Eingaben in einem Zeitschritt über eine Endzelle ist jedoch beschränkt. Die *Ein-/Ausgabedatenrate*, hier in der Größenordnung $O(N)$, nimmt damit unter Umständen schon gewisse Entscheidungen vorweg.

14.5.2. Sortieren

Beim Sortieren besteht die Aufgabe darin, eine Menge $\{x_1, \dots, x_N\}$ von Elementen aus einer vollständig geordneten Grundmenge G in aufsteigende Reihenfolge $\{m_i\}_{i=1, \dots, N}$ zu bringen, so dass also $m_i \leq m_k$ für $i < k$ gilt. Dieses Problem kann zuweisungsfrei folgendermaßen formuliert werden, wobei MAX das Maximum von G bezeichnet:



14.14. **Abbildung.** „Bubble Sort“ auf linearem systolischem Feld: (a) Feldaufbau mit Ein-/Ausgabeschema; (b) Zellstruktur.

Eingabeoperationen

$$\begin{aligned} x(i, j) &= x_i & 1 \leq i \leq N, j = 0 \\ m(i, j) &= MAX & 1 \leq j \leq N, i = j - 1 \end{aligned}$$

Berechnungen

$$\begin{aligned} m(i, j) &= \min\{x(i, j - 1), m(i - 1, j)\} & 1 \leq i \leq N, 1 \leq j \leq i \\ x(i, j) &= \max\{x(i, j - 1), m(i - 1, j)\} & 1 \leq i \leq N, 1 \leq j \leq i \end{aligned} \tag{14.47}$$

Ausgabeoperationen

$$m(i, j) = m_j \quad 1 \leq j \leq N, i = N$$

Mittels einer Projektion in Richtung $u = (1, 1)$, fortgesetzt zur Raum-Zeit-Abbildung

$$\begin{pmatrix} x \\ t \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} i \\ j \end{pmatrix} \tag{14.48}$$

erhalten wir das eindimensionale systolische Feld in Abbildung 14.14, als eine Implementierung des „Bubble-Sort“-Verfahrens.

Entsprechend würde die Raum-Zeit-Matrix

$$T = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \tag{14.49}$$

zu einem linearen systolischen Feld führen, das „Insertion Sort“ implementiert, die Raum-Zeit-Matrix

$$T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{14.50}$$

schließlich zu „Selection Sort“.

Wir haben beim Sortierproblem $O(N)$ Eingabedaten, $O(N)$ Ausgabedaten und $O(N)$ Zeitschritte. Dies ergibt eine Ein-/Ausgabedatenrate von $O(1)$. Anders als bei der Matrix-Vektor-Multiplikation in Abschnitt 14.5.1 lässt die Ein-/Ausgabedatenrate hier also prinzipiell noch die Kommunikation ausschließlich über Endzellen eines linearen systolischen Feldes zu.

In allen drei beschriebenen Varianten des Sortierens erfolgen jedoch Eingaben über alle Zellen: Bei Bubble Sort nur die zu sortierenden Elemente, bei Selection Sort zusätzlich die konstanten Werte MAX , bei Insertion Sort nur die konstanten Werte. Letztere bräuchten allerdings nicht unbedingt als Eingaben zur Verfügung gestellt werden, sondern könnten direkt in den Zellen erzeugt oder aus Festwertspeichern ausgelesen werden.

Alle drei Varianten benötigen eine Zellsteuerung: Insertion Sort und Selection Sort, weil sie stationäre Variablen besitzen; Bubble Sort, weil zwischen der Verarbeitung von Eingabedaten und von berechneten Werten umgeschaltet werden muss.

14.5.3. Lineares Gleichungssystem mit unterer Dreiecksmatrix

Die Formeln in (14.51) beschreiben einen lokalisierten Algorithmus zur Lösung des linearen Gleichungssystems $A \cdot x = b$, wobei die $(N \times N)$ -Matrix A eine untere Dreiecksform besitzt.

Eingabeoperationen

$$\begin{aligned} a(i, j) &= a_{i, j+1} & 1 \leq i \leq N, 0 \leq j \leq i-1 \\ u(i, j) &= b_i & 1 \leq i \leq N, j = 0 \end{aligned}$$

Berechnungen

$$\begin{aligned} u(i, j) &= u(i, j-1) - a(i, j-1) * x(i-1, j) & 2 \leq i \leq N, 1 \leq j \leq i-1 & (14.51) \\ x(i, j) &= u(i, j-1) / a(i, j-1) & 1 \leq i \leq N, j = i \\ x(i, j) &= x(i-1, j) & 2 \leq i \leq N-1, 1 \leq j \leq i-1 \end{aligned}$$

Ausgabeoperationen

$$x_i = x(i, j) \quad 1 \leq i \leq N, j = i$$

Bei allen vorher studierten Beispielen war, abgesehen von Kopieroperationen, an einem Trägerpunkt immer derselbe Satz von Rechenoperatoren durchzuführen: Multiplikation und Addition in Hintereinanderausführung bei den Multiplikationsalgorithmen, Minimum und Maximum in Parallelausführung bei den Sortieralgorithmen. In System (14.51) gibt es dagegen Trägerpunkte, an denen multipliziert und subtrahiert wird, und andere Trägerpunkte, an denen nur dividiert wird.

Bei Projektion von (14.51) auf ein lineares systolisches Feld ergeben sich, abhängig von der gewählten Projektionsrichtung, gleiche oder unterschiedliche Zellfunktionen. Mit der Projektionsrichtung $u = (1, 1)$ (und nur mit dieser) erhalten wir eine einzige Zelle mit Dividierwerk, alle anderen Zellen besitzen ein Multiplizier- und Subtrahierwerk. Bei Projektion längs $u = (1, 0)$ oder $u = (0, 1)$ ergeben sich lauter

gleiche Zellen, die sowohl ein Dividierwerk als auch ein Multiplizier- und Subtrahierwerk enthalten. Aus der Projektionsrichtung $u = (1, -1)$ resultiert ein lineares systolisches Feld mit drei verschiedenen Zelltypen: In den beiden Endzellen wird nur ein Dividierwerk benötigt. Alle übrigen Zellen enthalten ein Multiplizier- und Subtrahierwerk, wobei dort abwechselnd eine Zelle mit Dividierwerk und eine Zelle ohne Dividierwerk vorkommt. Eine bestimmte Art der Projektion kann also zu *Inhomogenitäten* in einem systolischem Feld führen (die wünschenswert sein können — oder auch nicht).

Übungen

14.5-1. Geben Sie für die in Abschnitt 14.5.1 angesprochenen Varianten der Matrix-Vektor-Multiplikation (Ausgabe der Ergebnisse über eine Endzelle beziehungsweise über alle Zellen) jeweils einen geeigneten Feldaufbau mit Ein-/Ausgabeschema und Zellstruktur einschließlich Steuerungsmechanismus an.

14.5-2. Studieren Sie die Effekte weiterer Projektionsrichtungen für System (14.51).

14.5-3. Geben Sie zu den in Abschnitt 14.5.2 beschriebenen Verfahren Insertion Sort und Selection Sort die zugehörigen systolischen Felder einschließlich Zellstruktur an.

14.5-4.* Wie kann das systolische Feld für Bubble Sort aus Abbildung 14.14 durch geschickte Gestaltung der Eingabeströme auch ohne Steuerung betrieben werden?

14.5-5.* Welche Funktion besitzt die Verwendung des Wertes *MAX* in System (14.47)? Wie könnte (14.47) ohne Verwendung dieses konstanten Wertes formuliert werden? Welche Konsequenzen hätte dies für die beschriebenen systolischen Felder?

Aufgaben

14-1. Bandmatrixalgorithmen nach Kung/Leiserson [3]

In den Abschnitten 14.1, 14.2, 14.5.1 und 14.5.3 sind wir immer von *vollbesetzten* Matrizen ausgegangen, das heißt der Möglichkeit, dass jedes dort angesprochene Matrixelement a_{ij} einen von Null verschiedenen Wert annehmen kann (bei der unteren Dreiecksmatrix sind die Elemente oberhalb der Hauptdiagonale zwar alle Null, stellen aber für den beschriebenen Algorithmus auch keine Eingabe dar).

In der Praxis kommen dagegen auch häufig *Bandmatrizen* vor. Bei diesen sind, bis auf ein schmales Band um die Hauptdiagonale herum, die meisten Diagonalen nur mit Nullen besetzt. Formal gilt also $a_{ij} = 0$ für alle i, j mit $i - j \geq K$ oder $j - i \geq L$, wobei K und L positive ganze Zahlen sind. Die *Bandbreite*, also die Anzahl der Diagonalen, in denen von Null verschiedene Elemente zugelassen sind, beträgt damit $K + L - 1$.

Es ist nun die Frage, ob die Bandstruktur einer oder mehrerer Eingabematrizen zur Optimierung der systolischen Berechnung ausgenutzt werden kann. Es besteht

einerseits die Möglichkeit, dass Zellen weggelassen werden können, weil sie niemals nützliche Arbeit verrichten. Andere Möglichkeiten der Verbesserung bestehen in einer Verkürzung der Ein-/Ausgabedatenströme, in einer Reduzierung der Gesamtausführungszeit oder in einer Erhöhung des Durchsatzes.

Studieren Sie, wie sich die in diesem Kapitel vorgestellten systolischen Felder diesbezüglich optimieren lassen.

Bemerkungen zum Kapitel

Der Begriff „systolisches Feld“ wurde von Kung und Leiserson in ihrem für die Entwicklung dieses Gebietes bahnbrechenden Artikel [3] geprägt. Karp, Miller und Winograd leisteten Pionierarbeit [2] im Bereich uniformer Rekurrenzgleichungen. Wesentliche Impulse für den systematischen Entwurf systolischer Felder gingen von Rao's Dissertation [5] und den Arbeiten von Quinton [4] aus. Teich und Thiele [6] zeigen in ihrem Beitrag, dass die formale Herleitung einer Zellsteuerung mit ganz ähnlichen Methoden wie die Bestimmung des allgemeinen Feldaufbaus und der normalen Zellfunktionalität erfolgen kann. Das moderne Buch von Darte, Robert und Vivien [1] führt ausgefeilte Methoden der Compilertechnik und der systolischen Felder zusammen und beschäftigt sich daher ausgiebig auch mit der Analyse von Datenabhängigkeiten. Den wohl auch heute noch umfangreichsten Gesamtüberblick zum Gebiet der systolischen Systeme gibt die Monografie [7].

Literaturverzeichnis

- [1] A. Darte, Y. Robert, and F. Vivien. *Scheduling and Automatic Parallelization*. Birkhäuser Boston, 2000. [748](#)
- [2] R. M. Karp, R. E. Miller, and S. Winograd. The organization of computations for uniform recurrence equations. *Journal of the ACM*, 14:563–590, 1967. [748](#)
- [3] H. T. Kung and C. E. Leiserson. Systolic arrays (for VLSI). In *SIAM Sparse Matrix Proceedings*, pages 256–282, 1978. [747](#), [748](#)
- [4] P. Quinton. Automatic synthesis of systolic arrays from uniform recurrent equations. In *Proc. 11th Ann. Int. Symp. Computer Architecture*, pages 208–214, 1984. [748](#)
- [5] S. K. Rao. *Regular iterative algorithms and their implementations on processor arrays*. PhD thesis, Stanford University, 1985. [748](#)
- [6] J. Teich and L. Thiele. Control generation in the design of processor arrays. *Journal of VLSI Signal Processing*, 3:77–92, 1991. [748](#)
- [7] E. Zehendner. *Entwurf systolischer Systeme (Design of Systolic Systems)*. B. G. Teubner Verlagsgesellschaft, 1996. [748](#)

Nameverzeichnis

Darte, Alain, 748
Karp, Richard M., 748
Kung, H. T., 747, 748
Leiserson, Charles E., 747, 748
Miller, Raymond E., 748
Quinton, Patrice, 748

Rao, Sailesh K., 748
Robert, Yves, 748
Teich, Jürgen, 748
Thiele, Lothar, 748
Vivien, Frédéric, 748
Winograd, Shmuel, 748
Zehendner, Eberhard, 748

Index

- MATRIXPRODUKT, 707
- Abhängigkeitsvektor, 720
- Addierwerk, 705
- Architektur, 717
- Architektur, systolische, 711
- Außenwelt, 705, 710
- Ausgabenormalform, 740
- Ausgangskanal, 705
- Auslastung, 713, 729, 731
- Bandbreite, 747
- Bandmatrizen, 747
- Bubble Sort, 745, **745**
- Datenabhängigkeit, 720, 720
- Dateneingabeschema, **704**
- Datenfluss, 720
- Datenrate, 727
- Datenspeicher, 710
- Datenstrukturindex, 724
- Datenstrukturvektor, 724
- Dividierwerk, 746
- Dreiecksmatrix, untere, 746
- Durchsatz, 737
- Ein-/Ausgabedatenrate, 744, 746
- Ein-/Ausgabeexpansion, 728, 728
- Ein-/Ausgabeschema, 712, 723, **724**
- Ein-/Ausgabeschema, erweitertes, 728, **729**
- Eingabestrom, 712
- Eingangskanal, 705
- Feldparameter, 706, 706
- Fließbandverarbeitung, 713, 713
- Flussabhängigkeit, 720
- Flussrichtung, 723
- Fourier-Motzkin-Elimination, 741
- generischer Operator, 707, 707, 722, 728
- gerichtete Verbindung, 711
- Gesamtausführungsdauer, 710, 719, 720
- Gleichungskalkül, 708
- Gleichzeitigkeit, 710
- Hardware-Algorithmus, 703
- Indexausdruck, 740
- Indexfunktion, 741
- Inhomogenität, 747
- Insertion Sort, 745
- Instanz, 708
- Iterationsvariable, 708
- Iterationsvektor, 708
- Kanal, 705
- Kommunikation in der Zeit, 721
- Matrix, vollbesetzte, 747
- Matrix-Vektor-Multiplikation, 744
- Matrixmultiplikation, 704, **704**, 705, 714
- Momentaufnahme, **714**, 725, 725
- Multiplizierwerk, 705
- multiply-add, 709
- Periode, 737
- Pipeline-Register, 713
- Polytop, 717
- Problemparameter, 706, 706
- Projektion, 716
- Projektionsmatrix, 716
- Projektionsrichtung, 716, 728
- Projektionsvektor, 716
- Quantifizierung, 708
- Rücksetzen, **733**
- Rand des systolischen Feldes, 728, 743
- Randzelle, 705
- Raum-Zeit-Abbildung, 714, 715, 715, 716, 739
- Raum-Zeit-Matrix, 730
- Raumkoordinaten, 706, 717
- Register, 705, 722
- regulärer Datenfluss, 705
- Rekurrenzgleichung, 708
- Scherung, 712
- Selection Sort, 745
- Serialisierung, 707, 708
- Skalarprodukt, 707
- Sortieren, 744
- stationär, 712, 723
- stationäre Variable, 712, 728, 732
- Steuerung, globale, 732, **733**
- Steuerung, lokale, 732
- Steuerung, verteilte, 736
- symbolische Berechnung, 717
- synchrone Arbeitsweise, 710
- Systolik, 703
- systolisch, 703
- systolischer Algorithmus, 704

- systolischer Algorithmus, uniformer, 714
- systolisches Feld, 703, 704, 704
- systolisches Feld, degeneriertes, 744
- systolisches Feld, hexagonales, 714, **715**, 719
- systolisches Feld, lineares, **745**
- systolisches Feld, mehrdimensionales, 713
- systolisches Feld, rechteckiges, **704**, 706
- systolisches System, 703, 704
- Taktsignal, 710
- Trägermenge, 708
- Trägermenge, dichte konvexe, 717
- Transformation einer Indexmenge, 740
- Transformation einer Trägermenge, 740
- unimodulare Matrix, 717, 718
- Verbindung, 705, 711
- Verbindungsstruktur, 717, 720
- Verbindungsstruktur, hexagonale, 721
- Verbundoperation, 709
- Verschränkung, 729, 729
- Verzögerung, 711
- Wirtsrechner, 704, 710
- Zeitschritt, diskreter, 710, 710
- Zeitvektor, 715
- Zelle, 704, 705
- Zelle, steuerungslose, 732
- Zellprogramm, 739, 741
- Zellstruktur, **704**, **715**, 717, 721
- zuweisungsfreie Notation, 708