

7. ReAI-Bytecodes

Bytecodes sind binär codierte maschinenunabhängige Darstellungen von ReAI-Programmen mit unbeschränktem Adressierungsvermögen. Die Auslegung von Bytecodes gehört zum allgemeinen Fachwissen der Informatik. Deshalb kann sich die folgende Darstellung auf ein typisches Ausführungsbeispiel beschränken.

ReAI-Programme bestehen aus Folgen von Bytes. Es gibt Steuerbytes (Abb. 7.1) und numerische Angaben.

7	6	5	4	3	2	1	0
Längenangabe (0...7)				Typangabe			

Abb. 7.1 Grundformat eines Steuerbytes

Numerische Angaben

Das Steuerbyte enthält eine Längenangabe im Bereich von 1 bis 7. Die Längenangabe kennzeichnet, wie lang die nachfolgende numerische Angabe ist. Die Typangabe kennzeichnet die Art der nachfolgenden numerischen Angabe (Tabelle 7.1).

Codierte Längenwerte: 1 Byte, 2 Bytes, 3 Bytes, 4 Bytes, 6 Bytes, 8 Bytes, reserviert.

Operatoren

Die Längenangabe ist 0. Die Typangabe kennzeichnet die Art des Operators (Tabelle 7.2).

Steuerbyte 00H: wirkungslos (NOP).

Typangabe	entsprechende Operatoren
Ressourcentyp	s
Ressourcennummer (im Ressourcenvorat)	s
Ressourcenanzahl	s
Variablenart	p, a
Variablennummer	p, a
Bitkette (Direktwert)	p
Quellressource	a, l, c, d
Quellparameter	a, l, c, d
Zielressource	p, l, c, d, y
Zielparameter	p, l, c, d
Funktionscode	h, m, u

Tabelle 7.1 Typangaben (1): numerische Angaben (Ordinalzahlen)

Operator	Wirkung	Operator	Wirkung
NOP	keine	c	ausgewählte Ressourcen über Parameter miteinander verketteten (connect)
s	Ressourcen auswählen (select)	d	Verkettung trennen (disconnect)
s_a	eine Ressource auswählen und Nummer oder Adresse zuweisen (select & assign)	l	Parameter zwischen ausgewählten Ressourcen transportieren (link)
p	Parameter aus Speichermitteln der Plattform in die ausgewählten Ressource transportieren	r	Ressourcen zu anderweitiger Verwendung freigeben (release)
p_imm	Direktwert (immediate) in die ausgewählte Ressource transportieren	h	zusätzliche Angaben (Hinweise) zur Unterstützung der Compilierung und zur Ablaufbeschleunigung. Reserviert
y	Verarbeitungsfunktionen in der ausgewählten Ressource aktivieren (yield)	m	zusätzliche metasprachliche Angaben. Reserviert
a	Parameter aus der ausgewählten Ressource in Speichermittel der Plattform transportieren (assign)	u	Dienstfunktionen (utilities). Hier werden alle maschinenspezifischen Codes eingeordnet, die zur Unterstützung der eigentlichen ReAI-Operatoren vorgesehen sind (z. B. zum Laden von Registern der Plattform)

Tabelle 7.2 Typangaben (2): Operatoren

Im Steuerbyte sind zu codieren:

- 11 Arten numerischer Angaben,
- 14 verschiedene Operatoren.

Das 5-Bit-Feld des hier beschriebenen Formats läßt somit noch eine Reserve von 21 bzw. 18 Codepositionen frei (es können noch 21 weitere Arten numerischer Angaben und 18 weitere Operatoren codiert werden).

Hinweise:

1. Manche Operatoren sind in mehreren Varianten zu codieren. Unser Beispiel beschränkt sich auf zwei derartige Operatoren, s_a und p_imm.
2. Die jeweilige Wirkung der Operatoren h, m und u wird in zusätzlichen numerischen Angaben codiert (Funktionscodes).

Die Operatoren müssen durch numerische Angaben ergänzt werden. Es gibt zwei Möglichkeiten der Reihenfolge:

1. Postfixnotation (Abb. 7.2). Erst kommen die numerischen Angaben, dann folgt der Operator. Das interpretierende System hält eine Art Zustandspuffer, der den aktuellen Wert einer jeden Angabe aufnimmt. Beim Fortschreiten von einem Operator zum nächsten sind nur jene Angaben neu einzutragen, die sich jeweils geändert haben.
2. Präfixnotation (Abb. 7.3). Erst kommt der Operator, dann folgen die numerischen Angaben. Deren Anzahl muß der jeweiligen Syntax entsprechen. Das interpretierende System muß eine Akzeptorautomaten haben, der gültige Reihen erkennt. Wurde eine solche Reihe erkannt, so wird die jeweilige Wirkung ausgelöst.

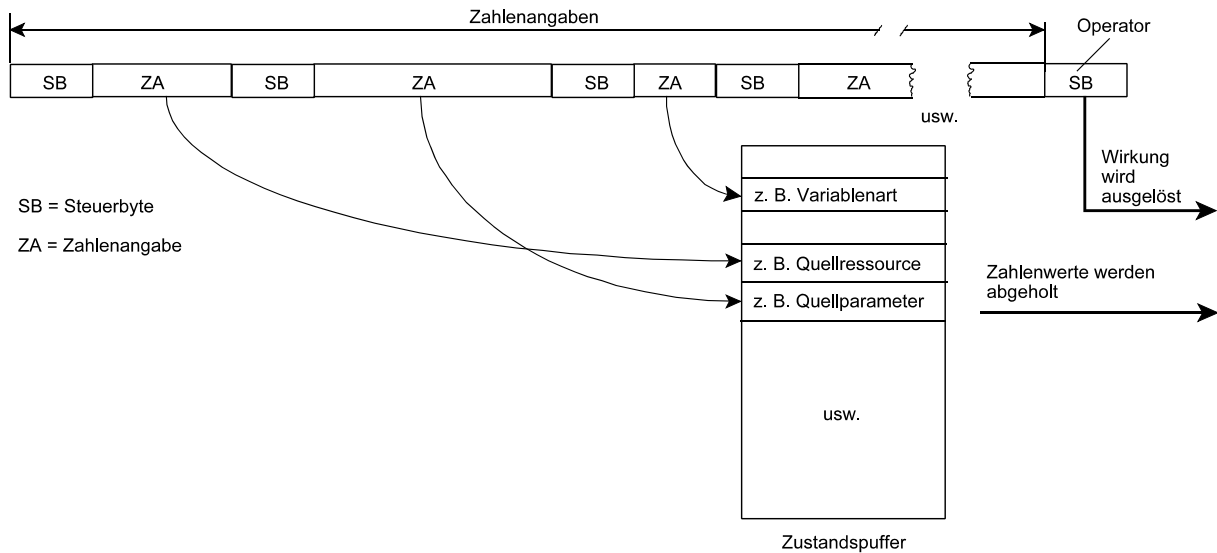


Abb. 7.2 Bytecode in Postfixnotation

Gemäß den Steuerbytes werden die Zahlenangaben in die jeweiligen Positionen des Zustandspuffers eingetragen. Tritt ein Operator auf, so wird die betreffende Wirkung ausgelöst. Dabei werden die jeweils benötigten Angaben aus dem Zustandspuffer geholt.

Im Beispiel hat der Zustandspuffer 11 Einträge, einen für jede Art der numerischen Angaben gemäß Tabelle 7.1. Werden beispielsweise bis zu 8 Bytes lange Parameter zugelassen, so muß jeder Eintrag die 8 Datenbytes aufnehmen können.

In einer ersten Ausführung wird zusätzlich die jeweilige aktuelle Längenangabe (aus dem Steuerbyte) mitgespeichert.

In einer alternativen Ausführung sind alle Parameter im Zustandspuffer beispielsweise 8 Bytes lang. Kürzere Angaben werden rechtsbündig eingefügt. Ein Parameterwert von einem Byte Länge wird in die Bitpositionen 7..0 eingetragen, einer von zwei Bytes Länge in die Bitpositionen 15...0 usw.

In einer weiteren Abwandlung ist der Zustandspuffer als Stack ausgebildet. Numerische Angaben werden auf den Stack gelegt, Operatoren entnehmen ihre Parameter vom Stack. Steuerbytes mit nachfolgenden numerischen Angaben stellen praktisch Push-Befehle dar, Steuerbytes die Operatoren codieren, Operationsbefehle (nach deren Ausführung die Operanden vom Stack entfernt werden).

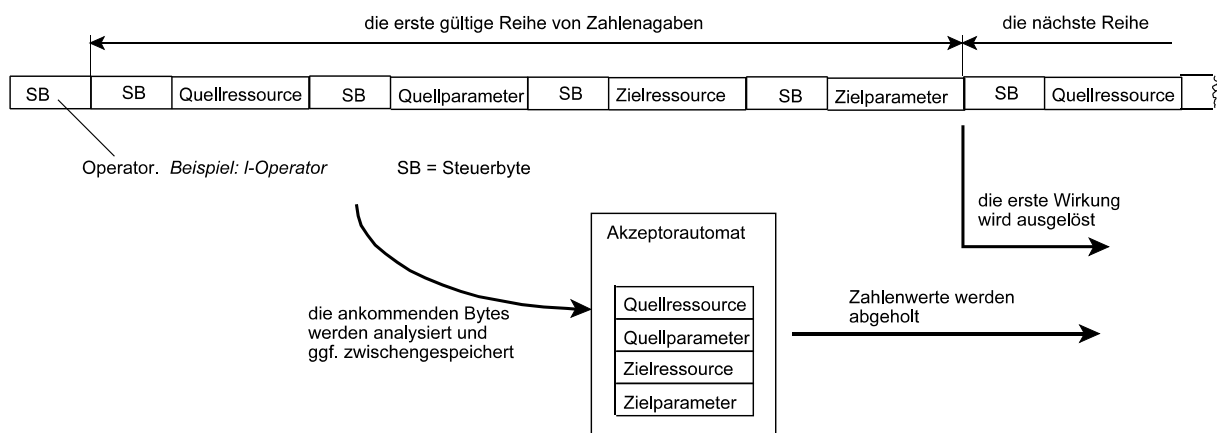


Abb. 7.3 Bytecode in Präfixnotation

Der Akzeptorautomat analysiert den Bytestrom. Jeder Operator ist durch bestimmte zulässige Folgen von Zahlenangaben gekennzeichnet (Tabellen 7.3, 7.4). Die erkannten Zahlenangaben werden zwischen-

gespeichert. Nach Erkennung einer vollständigen zulässigen Folge wird die entsprechende Wirkung ausgelöst. An eine solche Folge darf sich entweder eine weitere zulässige Folge oder ein Operator anschließen.

Ordinalzahl- oder Adreßangaben

Die soeben beschriebenen Auslegungen des Bytecodes können mit beiden Arten von Angaben arbeiten; es ist lediglich eine Frage der Interpretation.

Operator	Syntax der numerischen Angaben
s	Ressourcenanzahl - Ressourcentyp
s_a	Ressourcentyp - Ressourcennummer - Zielressource
p	Variablenart - Variablennummer - Zielressource - Zielparameter
p_imm	Direktwert - Zielressource - Zielparameter
y	Zielressource
a	Quellressource - Quellparameter - Variablenart - Variablennummer
l	Quellressource - Quellparameter - Zielressource - Zielparameter
c	Quellressource - Quellparameter - Zielressource - Zielparameter
d	Quellressource - Quellparameter - Zielressource - Zielparameter
r	Zielressource

Tabelle 7.3 Zulässige Folgen numerischer Angaben (1). Ordinalzahlangaben oder geteilter Ressourcenadreßraum

Operator	Syntax der numerischen Angaben
s	Ressourcentypadresse
s_a	Ressourcentypadresse - Ressourcenadresse
p	Variablenadresse - Zielressourcenadresse
p_imm	Direktwert - Zielressourcenadresse
y	Zielressourcenadresse
a	Quellressourcenadresse - Variablenadresse
l	Quellressourcenadresse - Zielressourcenadresse
c	Quellressourcenadresse - Zielressourcenadresse
d	Quellressourcenadresse - Zielressourcenadresse
r	Zielressourcenadresse

Tabelle 7.4 Zulässige Folgen numerischer Angaben (2). Flacher Ressourcenadreßraum