Registermodelle und Befehlsformate

1. Keine Register. Nur Speicheroperanden

a) Dreiadreßmaschine <S1> := <2> OP <3>

b) Zweiadreßmaschine <S1> := <S1> OP >S2>

Lange Speicheradressen führen zu langen Befehlen. Beispiel: RCA 301 (Zweiadreßmaschine):

1 Befehl = 60 Bits = 10 Zeichen zu 6 Bits

6 Bits	6 Bits	4 Zeichen zu 6 Bits = 24 Bits	4 Zeichen zu 6 Bits = 24 Bits
Op-code	Erweite-rung	1. Speicheradresse	2. Speicheradresse

Das sind fast 8 Bytes für einen einzigen vergleichsweise harmlosen Befehl!

Dreiadreßmaschine ist dann optimal, wenn überwiegend unabhängige Verknüpfungen ausgeführt werden (A := B + C; D = E * F usw.).

Solche Maschinen werden umso mehr unzweckmäßig, je häufiger folgende Verarbeitungsfälle vorkommen:

1. verschachtelte Operationen (A := ((C + D) * (E/F)) + (X * Y). Zwingen zum Einführen von Hilfsvariablen für die Zwischenergebnisse. Diese wenigen Variablen brauchen aber auch die langen Adressen.

2. Operationen, an denen dieselben Variablen immer wieder beteiligt sind. Jeder Zugriff zur gleichen Variablen erfordert immer wieder die lange Adresse.

Speicher-Mehradreßmaschinen hatten ihre Rechtfertigung nur durch Einsparungen in der Hardware (Register viel teurer als Speicher, also Vermeiden aufwendiger Registeranordnungen).

Baut heute keiner mehr.

2. Einadreßmaschinen

Die Einadreßmaschine braucht einen impliziten Bezug für den zweiten Operanden:

a) Akkumulator

b) Stack

Akkumulatormaschine = einfachste Hardware. Erfordert aber viele Befehle (u. teils exzessive Trickprogrammierung - einschließlich Befehlsmodifikation)

3. Mehrregister-Einadreßmaschinen
Zusätzliche Register zur Adreßrechnung usw. Akkumulator ist nach wie vor Bezugspunkt der Verarbeitungsoperationen.

Varianten:

- Register als feste Zellen im Arbeitsspeicher
- Akkumulator im Arbeitsspeicher (z. B. NCR 300)
- Akkumulator im Arbeitsspeicher verschieblich und in der Länge einstellbar (Akkumulatoradreßregister, Akkumulatorlängenregister). Z. B. GE 400.

4. Universalregister-Zweiadreßmaschinen
Mehrere Universalregister statt eines einzigen Akkumulators. Paradebeispiel: IBM S/360.

Befehlswirkungen:

<Ra> := <Ra> OP <Rb>

<Ra> := <Ra> OP <Adresse> (Speicheroperand)

Befehlsformate:

- RR - Register-Register. Zwei Registeradressen
- RX - Register-Speicher mit Indexadressierung. 3 Registeradressen (Operandenregister, Basisadreßregister, Indexregister), ein Speicheradreß-Displacement.
- RS - Register-Speicher. 3 Registeradressen (Operandenregister, Basisadreßregister, 3. register befehlsspezifischr), ein Speicheradreß-Displacement.
- SI - Speicher-Direktwert. Direktwert (1 Byte), Basisadresse, Speicheradreß-Displacement.
- S - Speicher. Nur ein Speicheroperand. Basisadresse, Speicheradreß-Displacement.
- SS - Speicher-Speicher. zwei Basisadressen, zwei Speicheradreß-Displacements, Operandenlängenangaben (2 * 4 oder 1 * 8 Bits).

Effektivität steht und fällt mit der Möglichkeit, den Registersatz tatsächlich 1:1 in Hardware zu bauen (als wirklichen Schnellspeicher mit entsprechend breiten Zugriffswegen). War bei S/360 typischerweise nicht der Fall (nur bei den teuersten Modellen). Deshalb war auch die Konkurrenz mit ihren Akkumulatormaschinen nicht gänzlich chancenlos.

5. Load-Store-Architekturen
Operandentransporte und Verknüpfungen sind befehlsmäßig voll voneinander entkoppelt.

a) Zweiregistermaschine (Zuse Z3/Z4)

b) Stackmaschine

Vorteil beider Auslegungen: implizite Registeradressierung - es gibt keine Registeradressen

c) Mehrregister-Dreiadreßmaschine

<R1> := >R2> OP <R3>

Die modische RISC-Auslegung

Rechenbeispiel:

X := A + B; Y := (A * B) / C

Dreioperandenmaschine	Registermaschine (RISC)	Stackmaschine

ADD A, B, X
 	
MUL A, B, Hilf
 	
DIV Hilf, C, Y
	LD R1, A
	LD R2, B
	LD R3,C
	
	ADD R1, R2, R4
	ST R4, X
	
	MUL R1, R2, R1
	DIV R1,R3,R1
	
	ST R1,Y
		PUSH A
		PUSH B
		ADD
		POP X
		
		PUSH A
		PUSH B
		MUL
		PUSH C
		DIV
		POP Y

Jede Adreßangabe: 14 Bits (= ein typischer Offset)

Dreioperandenmaschine
Der einzelne Befehl: 3 * 14 Bits + 8 = 50 Bits. ggf. 48, realistisch 64 je Befehl (6...8 Bytes). 3 Befehle. Insgesamt 18 oder 24 Bytes.

Registermaschine
Jeder Befehl wie üblich 4 Bytes. 8 Befehle. Insgesamt 32 Bytes. Adressierung aber flexibler (Auswahl an Basisregistern).

Stackmaschine
Jeder Befehl 2 Bytes. 10 Befehle. Insgesamt 20 Bytes. 3 Bytes weniger (nur 17 Bytes), wenn man die Operationsbefehle in einem Byte codiert.