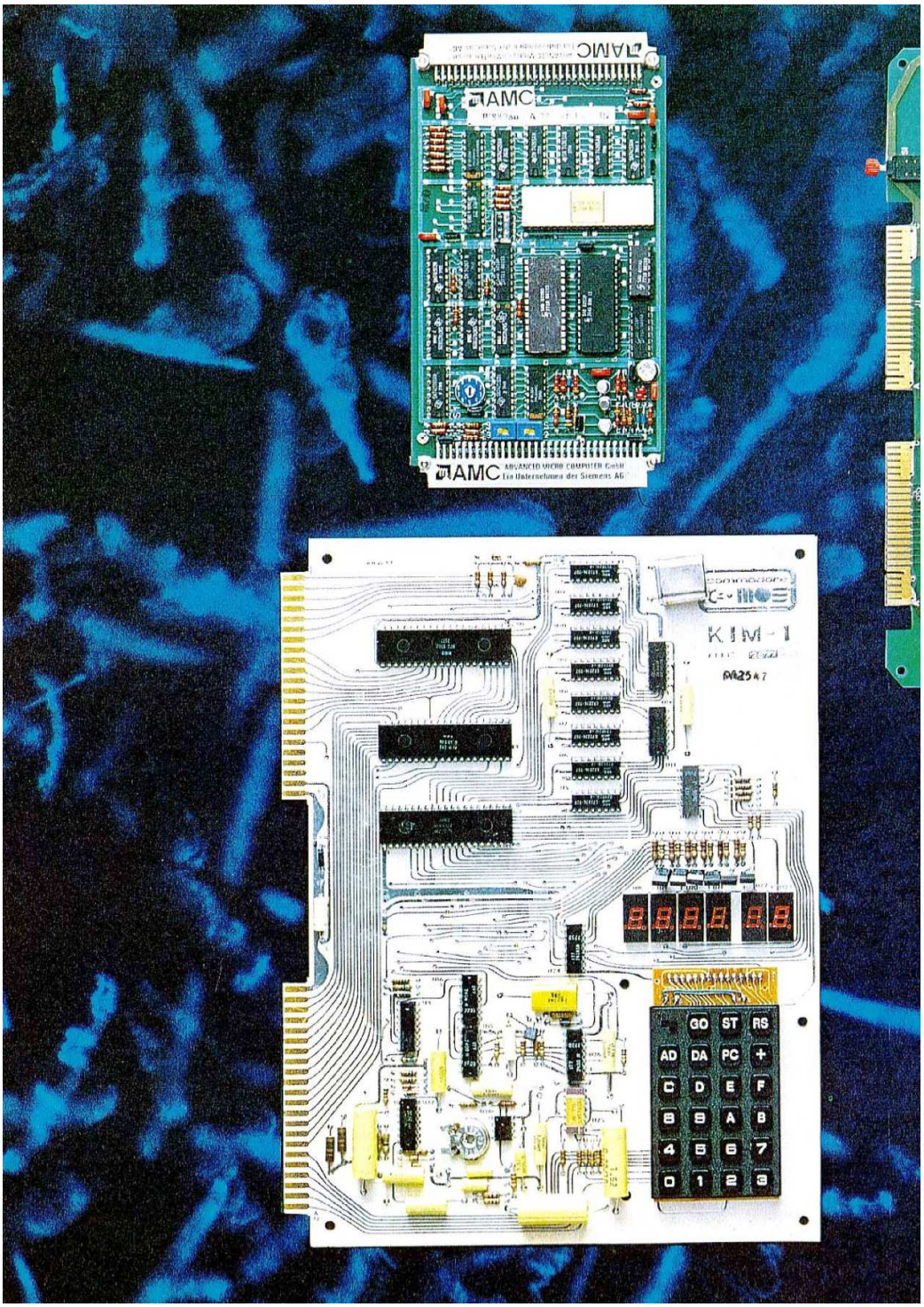
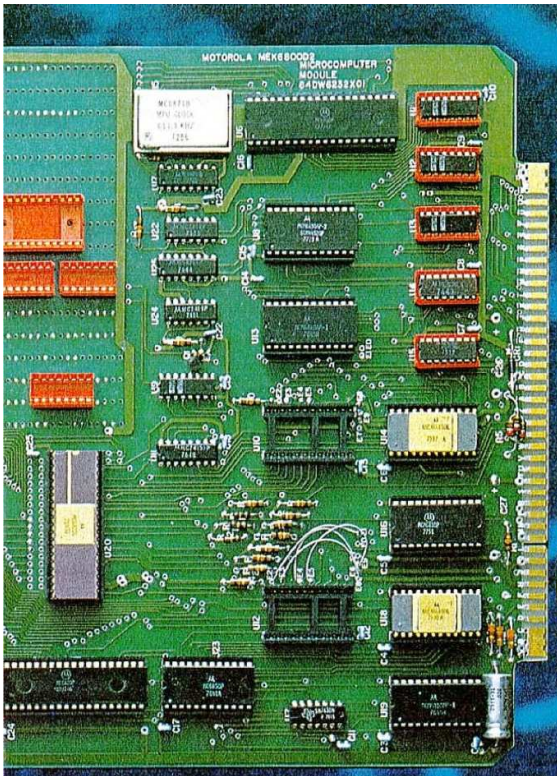
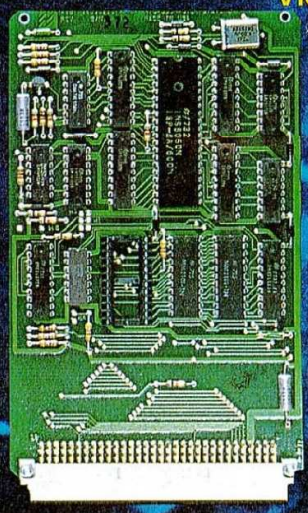

AMC
AMC
KIM-1
GO ST RS
AD DA PC +
C D E F
8 9 A B
4 5 6 7
0 1 2 3



Test:

Sieger-Kür

Vier Mikroprozessoren auf dem Prüfstand



M. STAHR E. SCHAEFER

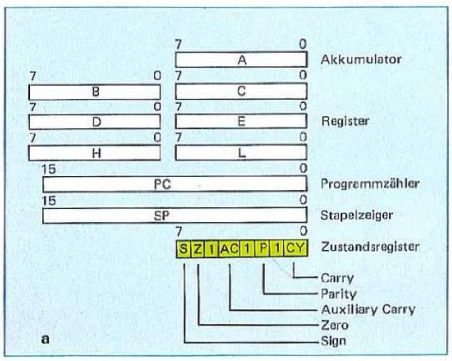
Wer kennt sie nicht — die Mikroprozessoren SAB 8080, 6502, MC 6800 und SC/MP? Sie alle stammen von bekannten Herstellern und bieten erstaunliche Leistungen. Doch wie kann man erkennen, wo die einzelnen Vorteile der unterschiedlichen Mikroprozessoren dann im System liegen? Da reichen Bücher nicht mehr aus. Da müssen Tests her, die aussagekräftig sind.

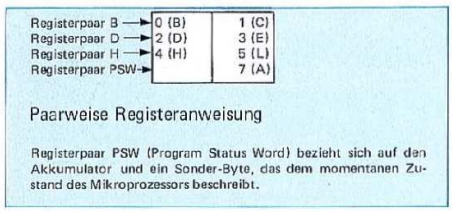
Sieger-Kür

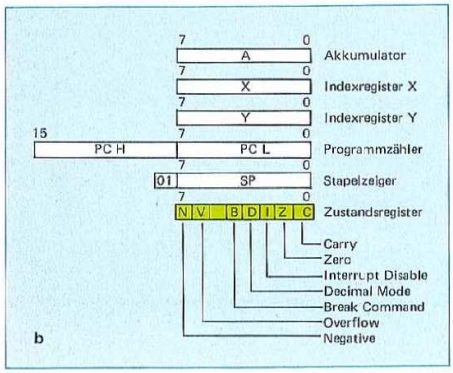
Tests, deren Ergebnisse eindeutig zu erkennen geben, welches System den jeweils gestellten Anforderungen weitestgehend entspricht. Das kann sich ja von Aufgabe zu Aufgabe ändern.
Hier helfen die sogenannten Benchmarkprogramme, also Prüfprogramme. Am Beispiel eines populären Testprogramms ergibt sich bereits eine Fülle von Erkenntnissen, die Sie immer wieder verwerten können. Außerdem erhalten Sie Hinweise über die Funktion der vier populären Mikroprozessorsysteme, die Sie sonst so leicht nicht erhalten.

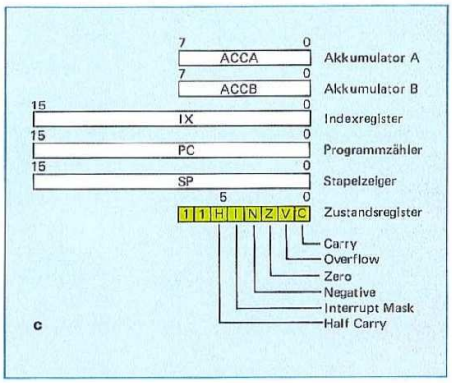
Interne Registerstruktur

Um dem Anwender eines Mikroprozessors eine Vorstellung von der unterschiedlichen Leistungsfähigkeit der verschiedenen Systeme zu geben, werden hier vier Systeme vorgestellt und durch dazu geeignete Programme sogenannte Benchmarkprogramme, verglichen (Benchmark = Richterbank). Benchmarkprogramme sind Programmsätze, die allgemein repräsentative Routinen darstellen. Mit solchen Programmen kann ein allgemeiner Vergleich von verschiedenen Mikroprozessorsystemen, also Mikrocomputern, durchgeführt werden.
Bevor diese Programme hier nun näher beschrieben werden, stellen wir die vier hier zu vergleichenden Systeme kurz vor. Verglichen werden die Systeme 8080 A von Siemens, KIM-1 von MOS-Technology, MC 6800 von Motorola und SC/MP von National Semiconductor. Die drei erstgenannten Systeme sind in N-MOS-Technologie hergestellt, während der Mikroprozessor SC/MP von National Semiconductor in P-MOS-









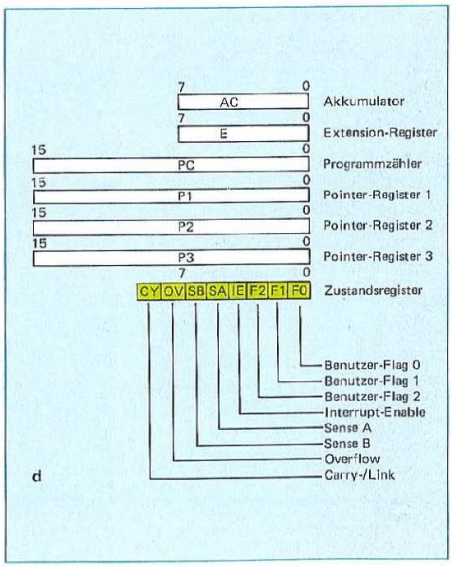


Bild 1: Interne Registerstruktur a SAB 8080, b KIM-1, c MC 6800, d SC/MP

SAB 8080	KIM-1	MC 6800	SC/MP
IN input			
LDA load accu direct	LDA load accu with memory	LDA load accu	LD load
MVI move immediate			LDI load immediate
LXI load reg. pair immediate			
LDAX load accu indirect			
			LDE load accu from extension
LHLD load HL direct	LDX load index X with memory	LDX load index	
	LDY load index Y with memory		
		LDS load stack	
MOV move	TAX tranf. accu to index X		
	TAY transf. accu to index Y		
	TSX transf. stack to index X	TSX transf. stack to index	
SPHL HL to stack	TXS transf. index X to stack	TXS transf. index to stack	
	TXA transf. index X to accu		
	TYA transf. index Y to accu		
		TAB transf. accus (A → B)	
		TBA transf. accus (B → A)	
		TAP transf. accu to status	CAS copy accu to status
		TPA transf. status to accu	CSA copy status to accu
OUT output			
POP pop reg. pair off stack	PLA pull accu from stack	PUL pull data	
	PLP pull status from stack		
PUSH push reg. pair on stack	PHA push accu on stack	PSH push data	
	PHP push status on stack		
SHLD store HL direct	STX store index X in memory	STX store index	
	STY store index Y in memory		
STA store accu direct	STA store accu in memory	STA store accu	ST store
STAX store accu indirect			
XCHG exchange DE, HL			
		STS store stack	XAE exchange accu and extension
XTHL exchange top of stack			XPAH exchange pointer high
			XPAL exchange pointer low
			XPPC exchange pointer with PC

Sieger-Kür

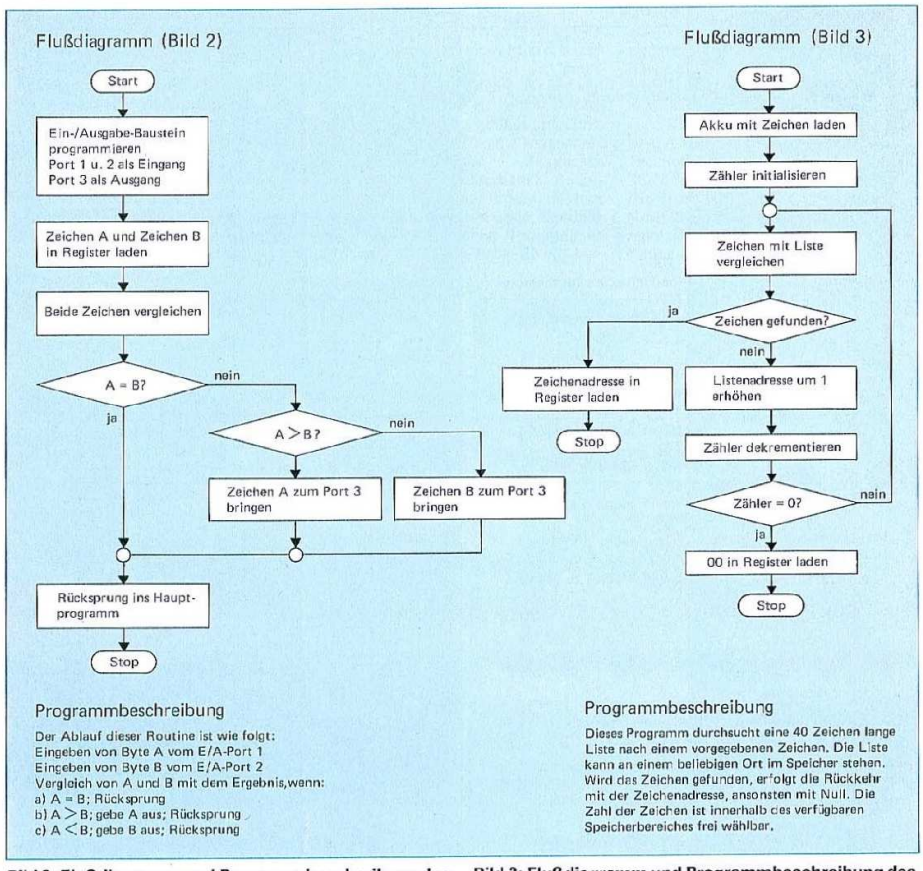


Bild 2: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 1: Ein-/Ausgaberoutine

Bild 3: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 2: Zeichensuche

Technologie aufgebaut und daher in der Schnelligkeit der Programmausführung natürlich wesentlich langsamer ist. Die für den Programmierer wichtige interne Registerstruktur ist bei den vier Systemen recht verschieden. Die Bilder 1a bis 1d zeigen die internen Registerstrukturen der einzelnen Mikroprozessoren grafisch sehr anschaulich.

Der Mikroprozessor **8080 A** besteht aus zehn internen Registern, wobei der Programmzähler und der Stapelzeiger, wie auch bei allen anderen Mikroprozessoren, 16 bit lang sind und sich somit 64 K-Speicherplätze adressieren lassen. Durch die paarweise Anordnung der Register B/C, D/E und H/L kann eine Reihe von Befehlen jeweils eins dieser Registerpaare benutzen. Darüber hinaus ist natürlich der Zugriff auf jedes einzelne Register möglich. Bild 1a zeigt die paarweise Registeranordnung. Dieses System hat also dann sieben 8-bit-Rechenregister.

Das **KIM-1**-System besteht aus sechs internen Registern, wobei der Stapelzeiger hier nicht, wie bei den anderen drei Mikroprozessoren, an beliebiger Stelle im Speicherplatz angeordnet werden kann, sondern an Page 1 gebunden ist. An Rechenregistern stehen hier gegenüber dem System 8080 nur drei Register zur Verfügung.

Der Mikroprozessor **6800** besteht ebenfalls aus sechs internen Registern. Allerdings hat dieses System zwei zentrale Register: die Akkumulatoren A und B. Das Indexregister ist gegenüber den beiden 8-bit-Indexregistern des KIM-1-Systems 16 bit breit.

Das System **SC/MP** verfügt über sieben interne Register. Die Pointer-Register 1 bis 3 dienen der Adressierung von Speicherplätzen, peripheren Geräten und werden außerdem als Stapelzeiger oder Indexregister verwendet.

Jeder Mikroprozessor hat ein Zustandsregister, das bestimmte Zustände innerhalb der CPU (CPU = Central Proces-

Arithmetische Operationen

SAB 8080	KIM-1	MC 6800	SC/MP
ADD add		ADD add	
		ABA add accus	ADE add extension
ADC add with carry	ADC add memory to accu with carry	ADC add with carry	ADD add
ACI add immediate to accu with carry			ADI add immediate
ADI add immediate to accu			
DAA decimal adjust accu		DAA decimal adjust	DAD decimal add
			DAE decimal add extension
			DAI decimal add immediate
DCR decrement reg. or memory	DEC decrement memory by one	DEC decrement	DLD decrement and load
DCX decrement reg. pair	DEX decrement index X by one	DEX decrement index	
	DEY decrement index Y by one		
		DES decrement stack	
JNR increment reg. or memory	INC increment memory by one	INC Increment	ILD increment and load
INX increment reg. pair	INX increment index X by one	INX increment index	
	JNY increment index Y by one		
		INS increment stack	
SBB subtract with borrow	SBC subtract memory from accu with borrow	SBC subtract with borrow	
		SBA substract accus	
SBI subtract immediate from accu with borrow		SUB	
SUI subtract immediate from accu			
SUB subtract		SUB subtract	

Ergebnisvergleich der Benchmarkprogramme

Benchmark-programme	Anzahl der Assembleranweisungen				Benötigte Programmspeicher-Bytes				Dauer der Ausführungszeiten (µs)			
	8080 A	KIM-1	MC 6800	SC/MP	8080 A	KIM-1	MC 6800	SC/MP	8080 A	KIM-1	MC 6800	SC/MP
Ein-/Ausgaberoutine	13	12	15	23	27	32	39	42	45,5	35	48	469
Zeichensuche	9	17	9	16	20	29	19	28	760	556	772	6 996
Berechneter Sprung	11	12	13	18	17	26	25	26	145	114	165	1 286
Shift-Routine	12	5	5	11	19	11	11	20	137	87	92	1 256
Vektorenaddition 8 bit	14	10	21	20	23	19	47	34	736,5	446	1246	3 988
Vektorenaddition 16 bit	20	14	33	23	29	29	68	40	1096,5	746	2346	6 188
16-bit-Multiplikation	53	25	20	39	79	61	44	71	1019	936	589	11 204
64-Wort-Block-verschieben	9	7	15	14	16	14	33	25	1261,5	1092	2770	9 028
Gesamt	141	102	131	164	230	221	286	286	5205	4042	8028	40 415

Sieger-Kür

sing Unit = Zentraleinheit) signalisieren kann. Über ein solches Zustandsregister werden von dem Computer Entscheidungen getroffen, die in Abhängigkeit bestimmter Zustände der Flags stehen und bestimmte Reaktionen (einen bedingten Sprung) nach sich ziehen. Dieses Zustandsregister ist in jedem Mikroprozessor verschieden aufgebaut und soll hier nicht näher beschrieben werden. Überhaupt soll die Vorstellung dieser Systeme nur einen sehr groben Überblick geben und der Anwender dieser kommt nicht umhin, sich näher über die einzelnen Systeme zu informieren. Die einzelnen Adressierungsarten der verschiedenen Systeme werden hier nicht beschrieben.

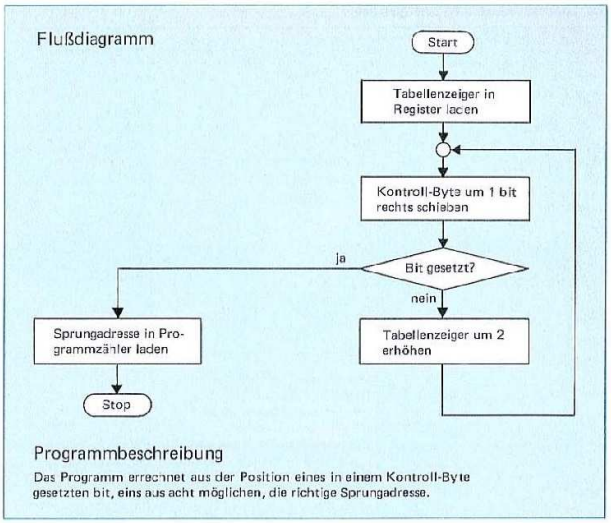


Bild 4: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 3: Berechneter Sprung

Logische Operationen

SAB 8080	KIM-1	MC 6800	SC/MP
ANA and	AND and memory with accu	AND logical and	AND and ANE and extension
ANI and immediate with accu			ANI and immediate
	BIT testbits in memory with accu	BIT bit test	
CMA complement accu		COM complement	
			CAD complement and add
			CAE complement and add extension
			CAI complement and add immediate
		NEG negate	
CMP compare	CMP compare memory and accu	CMP compare	
		CBA compare accus	
CPI compare immediate with accu			
	CPX compary memory and index X	CPX compary index	
	CPY compare memory and index Y		
ORA or	ORA or memory with accu	ORA inclusiv or accu	OR or
			ORE or extension
ORI or immediate with accu			ORI or immediate
XRA exclusive or	EOR exclusive or memory with accu	EOR exclusive or	XOR exclusive or
			XRE exclusive or extension
XRI exclusive or immediate with accu			XRI exclusive or immediate

Registeranweisungen

SAB 8080	KIM-1	MC 6800	SC/MP
RLC rotate accu left	ASL shift left one bit	ASL arithmetik shift left	
		ASR arithmethik shift right	
RRC rotate accu right	LSR shift right one bit	LSR logical shift right	SR shift rigth
RAL rotate accu left through carry	ROL rotate one bit left	ROL rotate left	SRL shift right with link
RAR rotate accu right through carry	ROR rotate one bit right	ROR rotate right	RRL rotate right with link
			RR rotate right
	CLC clear carryflag	CLC clear carry	CCL clear carry/link
CMC complement carry			
STC set carry	SEC set carryflag	SEC set carry	SCL set carry/link
	CLD clear decimal mode		
	SED set decimal mode		
	CLI clear interrupt disable	SLI clear interruptmask	
	SEI set interrupt disable	EI set interruptmask	
		CLR clear	
	CLV clear overflowflag	CLV clear overflow	
		SEV set overflow	

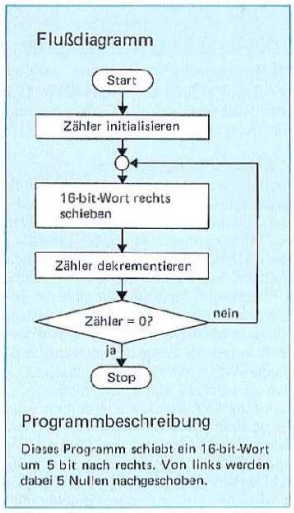


Bild 5: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 4: Shift-Routine

Assemblerprogramm SAB 8080

Zeichensuche

Addr.	1. 2. 3.	Label	Op.-Code	Operand	Comment
1050		LIA			Listenanfang
1077		LIE			Listenende
1078		ZEICH			Speicherplatz für Zeichen
1000	21 50 10	BEGIN	LXI	H, LIA	lade Rp HL mit Listenanfang
1003	3A 78 10		LDA	ZEICH	lade Akku mit Zeichen
1006	0E 28		MVI	C, 28	Zähler initialisieren
1008	BE	LOOP	CMP	M	vergleiche Zahlen mit Liste
1009	CA 30 10		JZ	FOUND	springe nach FOUND, wenn Zeichen gefunden
100C	23		INX	H	inkrementiere Rp HL
100D	OD		DCR	C	dekrementiere Zähler
100E	C2 08 10		JNZ	LOOP	springe nach LOOP, wenn Zähler ≠ 0
1011	21 00 00		LXI	H, 00	lade Rp HL mit 00, wenn Zeichen nicht gefunden
1030		FOUND	...		

Assembleranweisungen: 9
Programmspeicher-Bytes: 20
Max. Ausführungszeit: 760 µs

Sieger-Kür

Sprungbefehle

SAB 8080	KIM-1	MC 6800	SC/MP
JNC jump on no carry	BCC branch on carry clear	BCC branch if carry clear	
JC jump on carry	BCS branch on carry set	BCS branch if carry set	
JZ jump on zero	BEQ branch on result zero	BEQ branch if equal to zero	JZ jump if zero
JNZ jump on no zero	BNE branch on result not zero	BNE branch if not equal to zero	JNZ jump if not zero
JM jump on minus	BMI branch on result minus	BMI branch if minus	
JP jump on positive	BPL branch on result plus	BPL branch if plus	JP jump if positive
	BVC branch on overflow clear	BVC branch if overflow clear	
	BVS branch on overflow set	BVS branch if overflow set	
		BGE branch if greater or equal zero	
		BGT branch if greater than zero	
		BHI branch if higher	
		BLE branch if less or equal	
		BLS branch if lower or same	
		BLT branch if less than zero	
		BRA branch always	
JMP jump unconditional	JMP jump to new location	JMP jump	JMP jump
JPE jump on parity even			
JPO jump on parity odd			
PCHL HL to programcounter			

Assemblerprogramm MC 6800

Zeichensuche

Addr.	1. 2. 3.	Label	Op.-Code	Operand	Comment
1050		LIA			Listenanfang
1077		LIE			Listenende
1078		Zeich			Speicherplatz für Zeichen
1000	CE 10 50	BEGIN	LDX	LIA	lade Listenanfang ins Indexregister
1003	B6 10 78		LDA A	ZEICH	lade Zeichen im Akku A
1006	C6 28		LDA B	# 28	initialisiere Zähler
1008	A1 00	LOOP	CMP A	X	vergleiche Zeichen
100A	27 24		BEQ	FOUND	springe nach FOUND, wenn Zeichen gefunden
100C	08		INX		erhöhe Listenadresse um 1
100D	5A		DEC B		dekrementiere Zähler
100E	26 F8		BNE	LOOP	springe nach LOOP, wenn zähler + 0
1010	CE 00 00		LDX	# 00	lade Indexregister mit 00, wenn Kennzeichen gefunden
1030		FOUND	...		

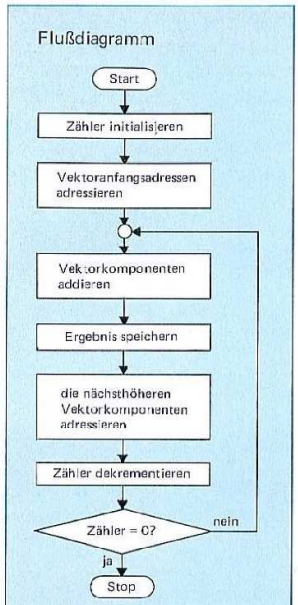
Assembleranweisungen: 9 Programmspeicher-Bytes: 19 Max. Ausführungszeit: 772 µs

Erläuterung der Befehlsarten

Transferbefehle
Die Transferbefehle transportieren Daten zwischen Speicher, Akkumulator und Ein-/Ausgabeeinheit.

Arithmetische Operationen
Arithmetische Operationen sind bei den meisten Mikroprozessoren die Additions- und Subtraktionsbefehle. Der SC/MP kennt keinen Subtraktionsbefehl. Mit diesen Befehlen lassen sich „höhere" Funktionen aufbauen. Weitere arithmetische Operationen sind die Inkrement- und Dekrementbefehle (Aufwärts- und Abwärtszählen) sowie Befehle zum bequemen Rechnen im Dezimalmodus.

Logische Operationen
Diese Befehlskategorie umfaßt die logischen Verknüpfungen wie UND, ODER und bei einigen Mikroprozessoren EXKLUSIV-ODER und NEGATION. Es wird dabei bitweise eine Speicherzelle mit dem Inhalt des Akkumulators oder eines Registers verknüpft. Andere logische Operationen sind die Vergleichsbe-



Programmbeschreibung

Dieses Programm addiert zwei zwanzigdimensionale Vektoren, die an beliebigen Speicherplätzen stehen können, zu einem dritten zwanzigdimensionalen Vektor. Negative Vektorkomponenten müssen im Zweierkomplement eingegeben werden. Der darstellbare Zahlenbereich liegt bei dieser Routine zwischen −128 und +127.

Assemblerprogramm KIM-1

Zeichensuche

Addr.	1. 2. 3.	Label	Op.-Code	Operand	Comment
0250		LIA			Listenanfang
0277		LIE			Listenende
0278		ZEICH			Speicherplatz für Zeichen
0200	A2 00	BEGIN	LDX	#00	lade Indexregister X mit 00
0202	A0 28		LDY	#28	Zähler initialisieren
0204	AD 78 02		LDA	ZEICH	lade Akku mit Zeichen
0207	DD 50 02	LOOP	CMP	X	vergleiche Zeichen
020A	F0 24		BEQ	FOUND	springe nach FOUND, wenn Zeichen gefunden
020C	E8		INX		inkrementiere Indexregister X
020D	88		DEY		dekrementiere Zähler
020E	D0 F7		BNE	LOOP	springe nach LOOP, wenn Zähler ≠ 0
0210	A2 00		LDX	#00	lade Indexregister X mit 00
0230	8A	FOUND	TXA		lade Akku mit Indexregister X
0231	D8		CLD		lösche Decimal-Flag
0232	18		CLC		lösche Carry-Flag
0233	69 50		ADC	LIA, L	berechne Zeichenadresse low Byte
0235	AA		TAX		lade Indexregister X mit Zeichenadresse low Byte
0236	A9 02		LDA	LIA, H	lade Akku mit Zeichenadresse high Byte
0238	69 00		ADC	#00	berechne Zeichenadresse high Byte
023A	A8		TAY		lade Indexregister Y mit Zeichenadresse high Byte

Assembleranweisungen: 17
Programmspeicher-Bytes: 29
Max. Ausführungszeit: 586 µs

Bild 6: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 5: Vektoren-Addition 8 bit

fehle (compare). Durch diese Operationen werden entsprechend dem Vergleichsergebnis verschiedene Bits im Zustandsregister gesetzt, ohne die Operanden zu verändern.

Registeranweisungen
Registeranweisungen bewirken die Löschung oder Setzung der Registerinhalte. Andere Registeranweisungen sind die Schiebebefehle, die zum Verschieben der Information innerhalb eines Bits im Akkumulator, Register oder Speicherzelle dienen. Bei den Mikroprozessoren wird jeweils um eine Stelle nach rechts, links oder im Kreis geschoben.

Sprungbefehle
Durch die Sprungbefehle kann man bestimmte Programmteile wiederholt durchführen und in verschiedene Programmteile verzweigen, und zwar abhängig von Entscheidungen. Sprungbefehle laden den Programmzähler mit einer neuen Adresse, wodurch das Programm an eben dieser Stelle fortfährt. Man unterscheidet bedingte und unbedingte Sprungbefehle. Die unbedingten Sprungbefehle führen in jedem Falle zur Fortsetzung des Programms an der im Adreßteil angegebenen Stelle. Die bedingten Sprungbefehle verzweigen nur, wenn bestimmte Bedingungen erfüllt sind.

Unterprogrammbehandlung
Unterprogramme (Subroutine) sind in sich geschlossene Programmteile, die von beliebigen Stellen des Hauptprogramms aus aufgerufen werden können. Diese Unterprogramme werden durch spezielle Sprungbefehle, dem Unterprogrammsprung, adressiert. Bei dieser Art Sprungbefehl wird der Inhalt des Programmzählers zur Zeit des Sprungs festgehalten. Dadurch kann das Hauptprogramm fortgesetzt werden, wenn der letzte Befehl des Unterprogramms ausgeführt ist.

Programmunterbrechung
Programmunterbrechungen (Interrupts) verursachen einen vorübergehenden Stopp des gerade laufenden Hauptprogramms, um ein anderes Programm bearbeiten zu können. Nach Beendigung dieses Programms wird wieder in das vorhergehende Hauptprogramm zurückgesprungen. Interrupts werden nur erkannt, wenn das entsprechende Bit im Zustandsregister (enable interrupt) vom Programmierer gesetzt worden ist, anderenfalls wird ein Interrupt ignoriert.
→

Sieger-Kür

Unterprogrammbehandlung

SAB 8080	KIM-1	MC 6800	SC/MP
CALL call unconditional CC call on carry CNC call on no carry CM call on minus CP call on positive CNZ call on no zero CZ call on zero CPE call on parity even CPO call on parity odd	JSR jump to new location saving returnaddress	JSR jump to subroutine BSR branch to subroutine	
RET return RM return on minus RP return on positive RC return on carry RNC return on no carry RZ return on zero RNZ return on no zero RPE return on parity even RPO return on parity odd RST restart	RTS return from subroutine RTI return from interrupt	RTS return from subroutine RTI return from interrupt	

Programmunterbrechung

SAB 8080	KIM-1	MC 6800	SC/MP
DI disable interrupt EI enable interrupt	BRK force break	WAI wait for interrupt SWI software interrupt	DINT disable interrupt IEN enable interrupt

Sonstige Befehle

SAB 8080	KIM-1	MC 6800	SC/MP
HLT halt NOP no operation	NOP no operation	NOP no operation TST test	HALT puls H-flag NOP no operation DLY delay SIO serial input/output

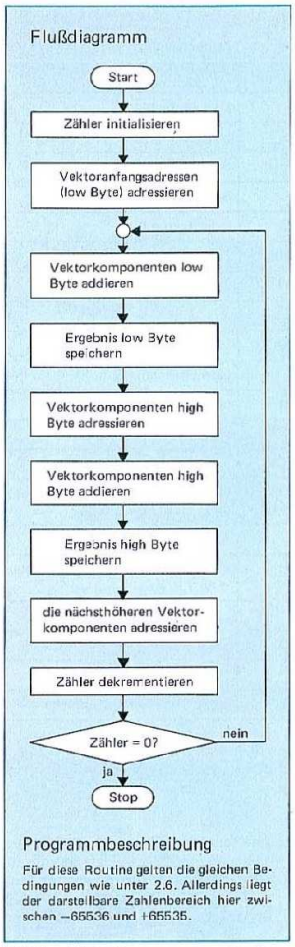


Bild 7: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 6: Vektoren-Addition 16 bit

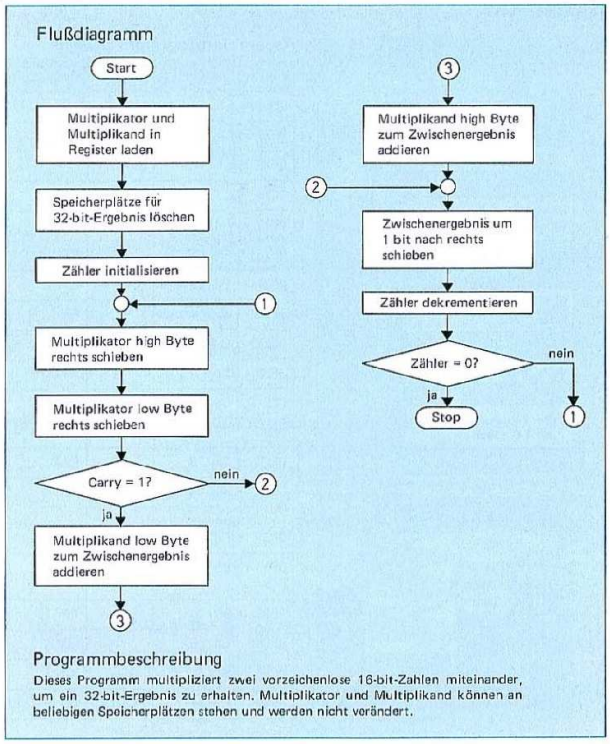


Bild 8: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 7: 16-bit-Multiplikation

Weitere Befehle

Sonstige Befehle sind der Leerbefehl NOP (no operation) und der Haltbefehl. Beim Haltbefehl wird der Programmzähler (PC) auf die Adresse des nächstfolgenden Befehls erhöht, dann erfolgt bis zu einem Interrupt keine Aktivität des Mikroprozessors. Wurde das Interruptsystem ausgeschaltet, muß der Mikroprozessor zum weiteren Betrieb wieder neu gestartet werden.

Der Befehlsvergleich in den Tabellen — aufgeteilt in acht Befehlskategorien — zeigt, welche Befehle einander entsprechen und welche Befehle vorhanden oder nicht vorhanden sind. Bereits das sorgfältige Studium dieser Tabellen läßt einen ersten Systemvergleich ohne weiteres zu.

So sind die Benchmarkprogramme

Wie zu Anfang erwähnt, soll der Leistungsvergleich mit Benchmarkprogrammen durchgeführt werden. Ein Kriterium zum Vergleich von verschiedenen Mikroprozessoren ist die relative Leistungsfähigkeit an einem solchen ausgewählten Satz von verschiedenen Benchmarkprogrammen. Für den Vergleich wurden drei Kriterien herangezogen:

1. Anzahl der Assembleranweisungen
2. Anzahl der Programmspeicher-Bytes und
3. Dauer der Ausführungszeit

Das erste Kriterium, die Anzahl der zu schreibenden Befehle, reflektiert die Leistungsfähigkeit eines Befehlssatzes und damit die einmaligen Softwareentwicklungskosten.

Das zweite Kriterium, die Anzahl der benötigten Speicher-Bytes, vergleicht die effektive Speicherausnutzung, was für speicherplatzbegrenzte Mikrocomputeranwendungen und Großstückzahlen wegen der geringeren Speicherkosten besonders wichtig ist.

Das dritte Kriterium ist die Dauer der Ausführungszeit, die wesentlich ist — z.B. für schnelle Prozeßsteuerungen.

Um diese Beurteilungskriterien bestimmen zu können, sind folgende acht Programme auf jedem System geschrieben worden:

1. Ein-/Ausgaberoutine
2. Zeichensuche
3. Berechneter Sprung
4. Shift-Routine
5. Vektoren-Addition 8 bit
6. Vektoren-Addition 16 bit
7. 16-bit-Multiplikation
8. 64-Block-Wort verschieben →

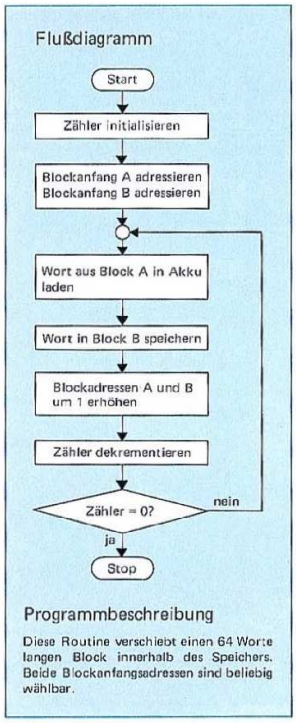


Bild 9: Flußdiagramm und Programmbeschreibung des Benchmarkprogrammes 8: 64-Wort-Block verschieben

Assemblerprogramm SC/MP — Zeichensuche

Addr.	1. 2. 3.	Label	Op.-Code	Operand	Comment
0F50		LIA			Listenanfang
0F77		LIE			Listenende
0F1E		ZEICH			Speicherplatz für Zeichen
0F1F		ZÄHL			Schleifenzähler
0F20	C4 28	BEGIN	LDI	#28	lade Akku mit 28
0F22	C8 FC		ST	ZÄHL	Zähler initialisieren
0F24	C4 CF		LDI	#0 F	
0F26	35		PAH	P1	
0F27	C4 50		LDI	#50	
0F29	31		XPAL	P 1	lade Listenanfang in P1
0F2A	C5 C1	LOOP	LD	(P 1)	lade Akku mit Zeichen
0F2C	03		SCL		setze Carry-Flag
0F2D	F8 F0		CAD	ZEICH	vergleiche Zeichen
0F2F	98 CF		JZ	FOUND	springe nach FOUND, wenn Zeichen gefunden
0F31	B8 FD		DLD		dekrementiere Zähler
0F33	9C F5		JNZ	LOOP	springe nach LOOP, wenn Zähler ≠ 0
0F35	35		XPAH	P 1	
0F36	C4 00		LDI	# 00	
0F38	31		XPAL	P1	lade P1 mit 00
0F40	C5 FF	FOUND	LD	(P1)	lade P1 mit Zeichenadresse

Assembleranweisungen: 16
Programmspeicher-Bytes: 28
Max. Ausführungszeit: 6996 µs

Aus Platzgründen wird nur das zweite Programm vollständig dargestellt. Für die anderen Programme erfolgt eine kurze, aber doch aussagekräftige Programmbeschreibung. Am Schluß werden die Ergebnisse natürlich diskutiert.

Auswertung der Ergebnisse

Das KIM-1-System schneidet beim Gesamtergebnis in allen drei Beurteilungskriterien gegenüber den anderen Systemen am besten ab. Dieses Ergebnis läßt sich auf einen leistungsfähigen Befehlssatz und auf die vielen sinnvollen Adressierungsarten zurückführen.

Das KIM-System benötigt:

gegenüber dem 8080 A:
a) 72,3% der zu schreibenden Befehle
b) 96,1% der Speicherplätze
c) 77,7% der Ausführungszeit

gegenüber dem MC 6800:
a) 77,9% der zu schreibenden Befehle
b) 77,3% der Speicherplätze
c) 50,3% der Ausführungszeit

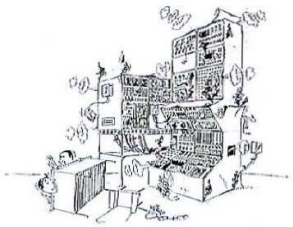
gegenüber dem SC/MP:
a) 62,2% der zu schreibenden Befehle
b) 77,3% der Speicherplätze
c) 10,0% der Ausführungszeit

Dieses Ergebnis bedeutet für das KIM-1-System:

a) Die weniger zu schreibenden Befehle bewirken eine schnellere und direktere Programmierung. Dies wiederum senkt die einmaligen Software-Entwicklungskosten.

b) Der Bedarf an weniger Speicherplätzen bedeutet einen kleineren Speicher und damit geringere Hardwarekosten. Dies ist besonders wichtig bei speicherplatzbegrenzten Mikrocomputeranwendungen und bei Einsätzen mit großen Stückzahlen.

c) Die schnellere Programmausführung bedeutet größeren Datendurchsatz (throughput-rate) pro gewählter Zeiteinheit. □



„Hundertmal habe ich dir gesagt: Keine widersprüchlichen Fragen stellen!"
Aus maul + computer-Späße