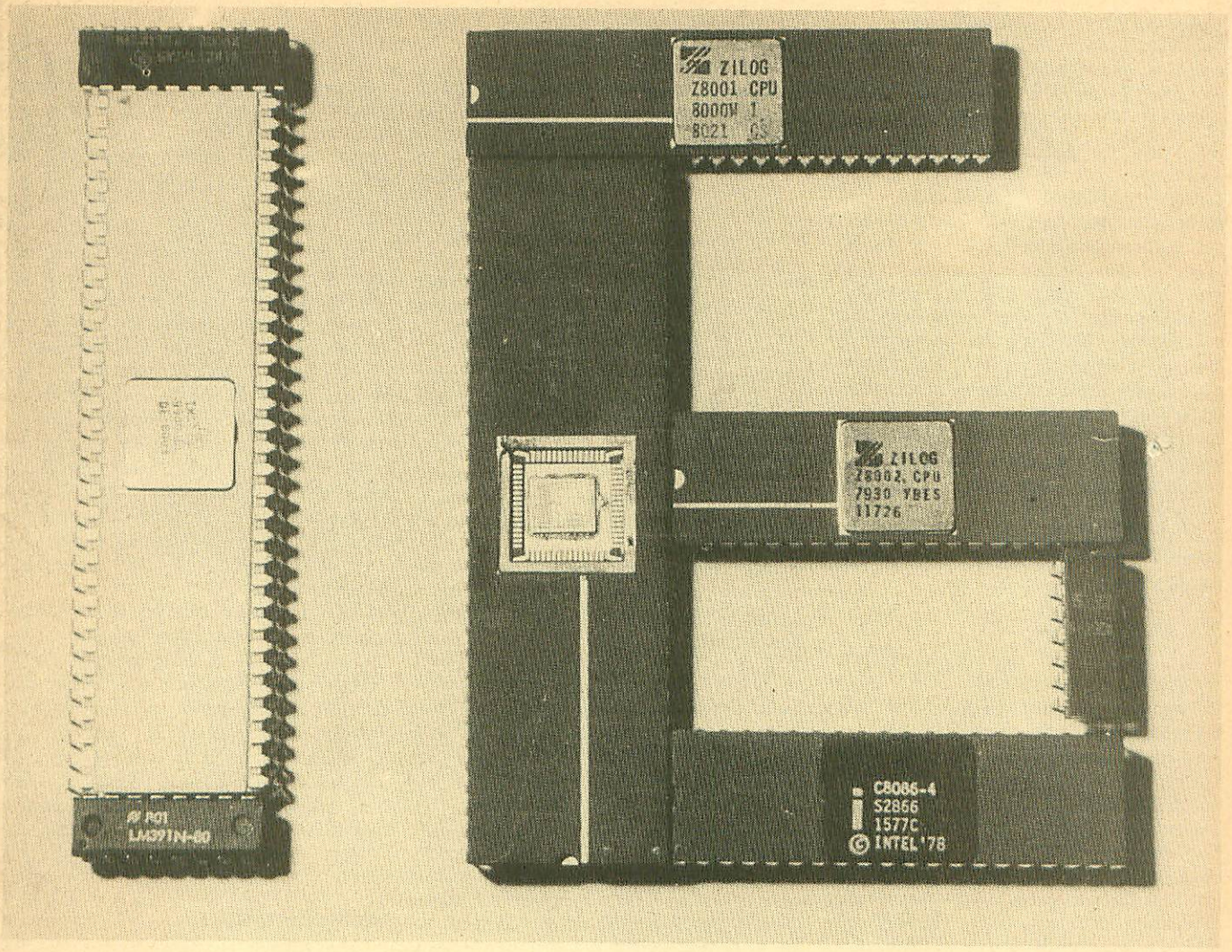


16-bit-µPs

Der Mikroprozessor in neuer Dimension

Erst wenige Jahre sind vergangen, seit die Mikrocomputertechnik auf breiter Basis ihren Einzug hielt. Bis jetzt reichte jedoch, trotz rascher Weiterentwicklung von Prozessoren und peripheren Bausteinen, die Leistungsfähigkeit der Mikrocomputer nicht an das Können der Minicomputer, ihrer älteren Kollegen, heran. Das soll sich nun ändern. Eine neue Mikroprozessor-Generation verhilft den "Mikros" dazu, die "Minis" zu überrunden.

Dieser Artikel will einen Überblick über die Fähigkeiten und die spezifischen Eigenheiten geben, mit denen die wichtigsten Vertreter der neuen 16-bit-µP-Generation ausgestattet sind. Er ist auch als Entscheidungshilfe bei der Auswahl eines bestimmten µP-Typs für einen vorgegebenen Zweck gedacht.

Von der Röhre zum Transistor . . .
vom Transistor zum TTL-IC . . .
von TTL nach CMOS . . .
von CMOS zum Mikroprozessor . . .
und jetzt 16-bit-Mikroprozessoren!

Dies sind die Schlaglichter einer Entwicklung, die in nur rund dreißig Jahren stattfand. Kein Wunder, daß mancher "Radiobastler", der während des Röhrenzeitalters zu seinem Hobby fand, in den letzten Jahren mehr und mehr die Orientierung verlor. An dieser Stelle können wir zwar nicht alles das aufarbeiten, was uns die vergangenen drei Jahrzehnte an Neuem und manchmal sogar Revolutionärem bescherten. Dafür soll aber das jüngste Kapitel um so eingehender beleuchtet werden. Wir wollen versuchen, einen grundlegenden Überblick über das neue Feld der "Super"-Mikroprozessoren zu geben; gleichzeitig sollen ihre Fähigkeiten eingehend betrachtet und miteinander verglichen werden.

Die 16-bit-Klasse

Nicht immer läßt sich ohne weiteres entscheiden, ob dieser oder jener Mikroprozessor der Gruppe der 16-bit-µPs zuzuordnen ist. Das hat bestimmte Gründe, auf die wir später noch zu sprechen kommen. Betrachten wir die "Mikros" zunächst einmal aus einem etwas allgemeineren Blickwinkel: Ein binäres 16-bit-Wort kann 65536 unterschiedliche Gestalten annehmen; mit ihm können daher beispielsweise die dezimalen Zahlen von –32768 bis +32767 dargestellt werden. Zweifellos ist dieser Zahlenbereich im Vergleich zu den 256 Zahlenwerten, die sich mit 8 bit darstellen lassen, bei weitem umfangreicher. Die Addition und Subtraktion von großen Zahlenwerten vereinfacht sich bedeutend, wenn der Prozessor nicht nur 8 bit, sondern 16 bit gleichzeitig verarbeitet. Die neuen 16-bit-µPs können jedoch nicht nur addieren und subtrahieren (wie die 8-bit-µPs), sie sind auch fähig, auf Befehl zu multiplizieren und zu dividieren.
Grundsätzlich besteht jedes Computersystem aus einigen wenigen Funktionsblöcken: den Input- und Outputeinheiten (Keyboard, Display, Steuerleitung usw.), dem Arbeitsspeicher (der sowohl das Programm als auch die eigentlichen Daten aufnimmt) und der "Zentraleinheit" (CPU). Die CPU (Central Processing Unit) sorgt für den notwendigen Datenfluß innerhalb des Systems und führt die arithmetischen und logischen Operationen aus: Addition und Subtraktion, Multiplikation und Division, AND, OR, EXOR usw. Ferner stellt sie sicher, daß sowohl der Datenaustausch als auch die Operationen in der vom Programm vorgeschriebenen Weise ausgeführt werden. Das ist schon eine ganze Reihe von Aufgaben, doch die 16-bit-Mikroprozessoren schaffen dies leicht – und darüber hinaus noch viel mehr. Bei einem Vergleich der einzelnen Vertreter dieser Klasse müssen deshalb die folgenden Kriterien unbedingt mit einbezogen werden:

- Welche Operationen (arithmetisch, logisch usw.) können unmittelbar ausgeführt werden, und für welche ist ein Unterprogramm erforderlich?
- Wie umfassend ist der Speicherbereich, der sich von der CPU adressieren läßt, und wie läuft der Datentransport zum Speicher und in umgekehrter Richtung ab?
- Welche Software-Techniken (Sprünge, Schleifen, Subroutinen usw.) stehen dem Programmierer zur Verfügung?

Ein weiteres wichtiges Kriterium für die Beurteilung der Brauchbarkeit ergibt sich aus der Tatsache, daß Mikrocomputersysteme fast immer starke Wachstumtendenzen zeigen. Wenn die Speicherkapazität erweitert wird und noch neue periphere Einheiten hinzukommen, gewinnen folgende Fragen zusätzlich an Bedeutung:

- Wie schnell können Interrupts erledigt werden, die zu ungünstigen Zeitpunkten von externen Geräten eintreffen?
- Wie sieht es mit der Verträglichkeit aus, wenn andere Mikroprozessoren Teile des Systems wie Speicher, Peripherie usw. mitbenutzen? Da hierauf meistens der größte Teil der Kosten entfällt, ist der Multi-Prozessor-Betrieb oft ökonomischer als die Lösung der gestellten Aufgaben mit Einzelsystemen.
- Wie schnell arbeitet der Mikroprozessor? Wächst das System, dann werden auch die Programme komplexer. Das Ausführen einer Division in 40 Mikrosekunden mag auf den ersten Blick schnell erscheinen, doch wenn innerhalb eines Programms mehrere tausend arithmetische Operationen und ebenso viele Datentransporte erledigt werden müssen, addieren sich Mikrosekunden zu Sekunden oder sogar zu Minuten. Einige Schachcomputer benötigen sogar Stunden, um komplizierte Züge zu berechnen!

Doch zurück zur Klassifizierung der 16-bit-Mikroprozessoren. Man sollte meinen, jeder Prozessor, der Datenworte mit einer Breite von 16 bit verarbeitet, gehört zu dieser Kategorie. Bei genauerer Betrachtung liegen die Dinge jedoch komplizierter: Verschiedene µPs arbeiten *intern* mit einem 16 bit breiten Datenbus, außerhalb des Chips aber setzt sich das 16-bit-Datenwort aus zwei aufeinanderfolgenden 8-bit-Bytes zusammen. Handelt es sich hier um einen 16-bit-Prozessor? Ja und Nein! Dieser Prozessor unterscheidet sich von einem echtem 16-bit-Prozessor im wesentlichen nur dadurch, daß er für den Datentransfer die doppelte Zeit benötigt. Ähnliche Definitionsschwierigkeiten ergeben sich bei Prozessoren, die intern Datenworte mit einer Breite von 32 bit verarbeiten, äußerlich jedoch 16-bit-Prozessoren sind. So wird beispielsweise der MC68000 von Motorola gelegentlich als "32-bit-CPU mit 16-bit-Gesicht" bezeichnet.
In die Tabelle 1 wurden ohne Unterschied alle Prozessoren aufgenommen, die äußerlich das Erscheinungsbild eines 16-bit-Prozessors zeigen. Zu finden sind dort elf Haupttypen, von denen die fünf Typen noch einmal in Tabelle 2 erscheinen, die auch für den Nicht-Profi von Interesse sind. Aus Tabelle 3 gehen die wichtigsten vergleichbaren Daten dieser fünf Prozessoren und der mit ihnen unmittelbar verwandten Typen hervor.

Erster Eindruck

In der Entwicklungsgeschichte der 16-bit-Mikroprozessoren zeichnen sich zwei verschiedene Wege ab, die jedoch meistens gleichzeitig beschritten werden: Zum einen ist der Stammvater oft ein kleinerer 8-bit-Mikroprozessor, zum anderen stand die Minicomputertechnik Pate bei der Gesamtkonzeption. Motorola und Zilog zum Beispiel haben die Befehlssätze ihrer µPs auf der

Tabelle 1 16-bit-Mikroprozessoren (Haupttypen)

Typ	Entwickler	Technologie	Anwendung
MN 601	Data General	NMOS	OEM-Minicomputer
9440	Fairchild	I^2L	OEM-Minicomputer
F100L	Ferranti	bipolar	militärische Anw.
CP1600	General Instr.	NMOS	elektron. Spiele
8086	Intel	HMOS	Universal-µP
MC 68000	Motorola	NMOS	Universal-µP
NS 16032	National Semiconductor	XMOS	Universal-µP
MN 1610	Panafacom	NMOS	?
TMS 9900	Texas Instr.	NMOS	Universal-µP
WD 16	Western Digital	NMOS	OEM-Minicomputer
Z 8001	Zilog	NMOS	Universal-µP

Tabelle 2 16-Bit-Mikroprozessoren, Universaltypen und Hersteller

Typ	Hersteller
8086	**Intel**, Mitsubishi, Mostek, Siemens
68000	**Motorola**, Hitachi, Rockwell, Thomson
16032	**National Semiconductor**, Fairchild
9900	**Texas Instruments**, AMI, ITT
8001	**Zilog**, AMD, SGS-Ates

Tabelle 3a

Haupttyp	Nebentyp*	Datenwortlänge intern/extern	Adressenbereich Bus/Speicher/mit Hilfsbausteinen	Multiplex-Daten-/Adreßbus	Abstammung (Entwicklung)
8086	8088	16/16 bit 16/18 bit	20 bit/1 MByte/1 MByte	ja	aufsteigend vom 8080; absteigend von Minicomputern
68000		32/16 bit	23 bit/16 MByte/64 MByte	nein	aufsteigend von 6800; absteigend von Minicomputern
16032		32/16 bit	24 bit/16 MByte		aufsteigend vom 8080; absteigend von Minicomputern
	16016	16/16 bit	16 bit/64 KByte	ja	
	16008	16/8 bit	16 bit/64 KByte		
9900		16/16 bit	15 bit/64 KByte	nein	absteigend von Minicomputern
	9940	kein externer Daten- und Adreßbus; 2 KByte RAM/ROM auf dem Chip			
	9980/9981	16/8 bit	14 bit/16 KByte	nein	
	9995	16/8 bit	15 bit/64 KByte		
8001		16/16 bit	23 bit/8 MByte/48 MByte	ja	aufsteigend vom Z80; absteigend von Minicomputern
	8002		16 bit/64 KByte/384 KByte		
	8003		wie 8001		
	8004		wie 8002		

* bei den Nebentypen sind nur die Abweichungen vom Haupttyp angegeben

Tabelle 3b

Haupttyp	Nebentyp	Universalregister	Spezial- und Steuerregister	Bytefolge im Speicher	Taktfrequenz	kürzeste Befehlsdauer**	längste Befehlsdauer**
8086		–	14 (16 bit)	low-high	8/5/4 MHz	0.25 µs	20 µs (①)
	8088				5 MHz	0.4 µs	32 µs (①)
68000			18 (32 bit), 1 (16 bit)	high-low	8/6/4 MHz	0.5 µs	20 µs (②)
16032		8 (32 bit)	6 (24 bit), 2 (16 bit)	low-high	10 MHz	0.3 µs	8 µs (②)
	16016	8 (16 bit)	8 (16 bit)				
	16008	8 (16 bit)	8 (16 bit)				
9900		16 (16 bit***)	3 (16 bit)	high-low	3.3/4 MHz	2 µs	31 µs (①)
	9980/9981				2.5 MHz	2.6 µs	41 µs (①)
	9995				6 MHz	1.1 µs	17 µs (①)
8001		16 (16 bit)	7 (16 bit)	high-low	6/4 MHz	0.5 µs	140 µs (③)
	8002	16 (16 bit)	4 (16 bit)		6/4 MHz	0.5 µs	(19 µs (②))
	8003				10 MHz	0.3 µs	80 µs (③)
	8004				10 MHz	0.3 µs	(11 µs (②))

* Low-High = niederwertiges Byte an niederwertiger Adresse, High-Low = umgekehrte Reihenfolge
** bei höchstzulässiger Taktfrequenz
*** diese Register befinden sich nicht in der CPU, sondern im RAM

① Division ohne Vorzeichen, (32 bit) : (16 bit) = (16 bit) + (16 bit) Rest
② Division mit Vorzeichen, (32 bit) : (16 bit) = (16 bit) + (16 bit) Rest
③ Division mit Vorzeichen, (64 bit) : (32 bit) = (32 bit) + (32 bit) Rest

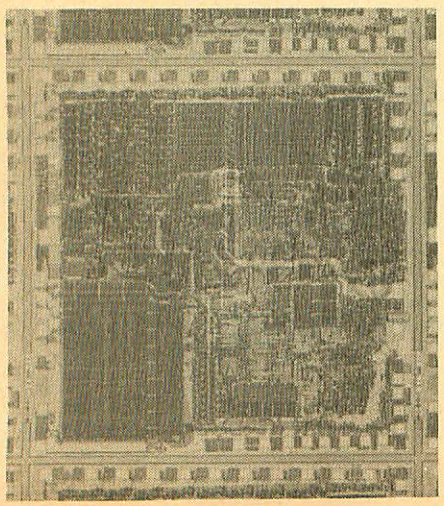
Tabelle 3c

Haupttyp	Nebentyp	Interruptarten: NMI	Traps	nicht vektoriell	vektoriell	I/O-Bereich	Befehlsfolge	ABORT für virtuellen Speicher
8086		1	4	–	251	64 KByte	6 Byte	nein
	8088						4 Byte	
68000		–	27	–	227	*	nein	nein
16032		1	9	1	240	*	8 Byte	ja
	16016							
	16008							
9900		2	16	–	15	4 KBit	nein	nein
	9980/9981							
	9995							
8001		1	4	1	128	64 KByte	nein	nein
	8002				255			nein
	8003				128			ja
	8004				255			ja

* nur "Memory Mapped"

Grundlage einer Analyse der meistgebrauchten Befehle aufgebaut. Da auf die verwandtschaftliche Beziehung nach unten (8-bit-µPs) und nach oben (Minicomputer) unterschiedliches Gewicht gelegt wurde, sind auch die Endprodukte *recht verschieden*:

- Intel (8086, 8088) lehnte seine 16-bit-µPs überwiegend an die bekannte 8080-Familie an. Die Register des 8080 sind im Registersatz des 8086 enthalten, so daß 8080-Programme ohne große Änderungen auch auf 8086-Systemen laufen können. Auch hier sind die Register oft bestimmten Befehlen zugeordnet, was einerseits die Länge der Befehle (in Maschinensprache) verkürzen kann, andererseits aber die Freizügigkeit des Programmierers einengt.
- Motorola (MC68000) orientierte sich an Zukunftsperspektiven: 32-bit-Register und ein Befehlssatz mit hoher Effektivität, der teilweise aus der Minicomputerpraxis stammt, sind hier die herausragenden Merkmale. Gleichzeitig erreichte Motorola eine weitgehende Kompatibilität mit der 6800-Familie, so daß die hier bereits vorhandenen peripheren Chips (I/O usw.) paarweise verwendet werden können.
- National Semiconductor (NS16032, 16016, 16008) hatte ebenfalls die Zukunft im Auge, doch auch die Vergangenheit blieb nicht ohne Einfluß. Das Ergebnis ist eine Verschmelzung von alten und brandneuen Konzeptionen: Einerseits sind hier verschiedene, für den 8080 typischen Merkmale wie die Low-High-Speicherbelegung (mehr darüber später) zu finden, andererseits warten diese 16-bit-µPs mit einem 16-MByte-Adressenbereich, dem Slave-Prozessor-Konzept und den Voraussetzungen für ein Virtual-Memory-System auf (auch zu letzterem später mehr).
- Texas Instruments (TMS9900-Familie) hatte sich zum Ziel gesetzt, die Zentraleinheit eines Minicomputers auf einem einzigen Chip zu integrieren. Der dabei entstandene 16-bit-µP ist deutlich langsamer als seine Mitbewerber, sein Adressenbereich hat weniger Umfang, die Interrupt-Möglichkeiten sind eingeschränkter, und auch der Befehlssatz ist begrenzter. Der Grund



1a

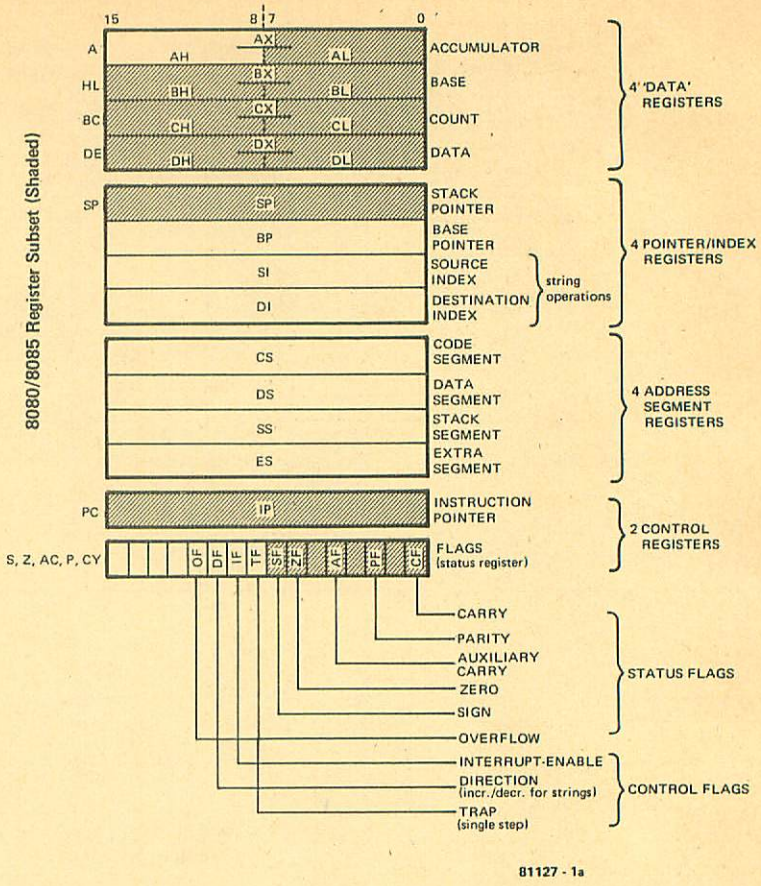


Bild 1a. Register des 8086-Prozessors. Die schraffiert gezeichneten Register sind auch im 8080/8085-Prozessor enthalten.

1b

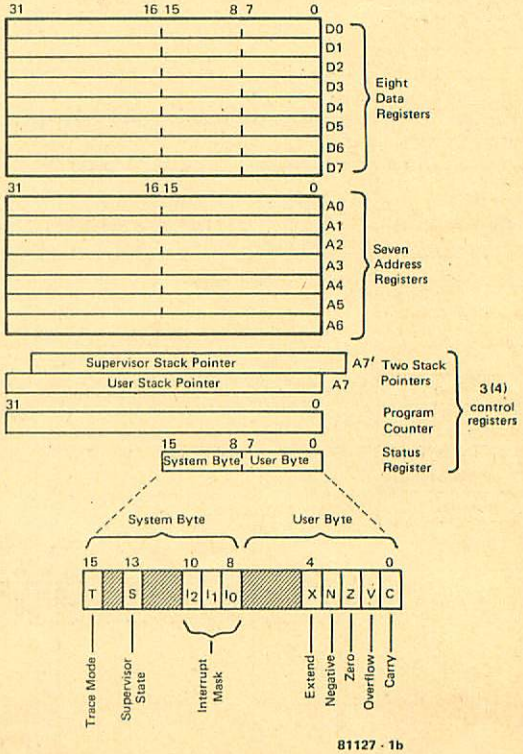


Bild 1b. Die Register des 68000-Prozessors sind 32 bit breit. Ist das noch ein 16-bit-µP?

1c

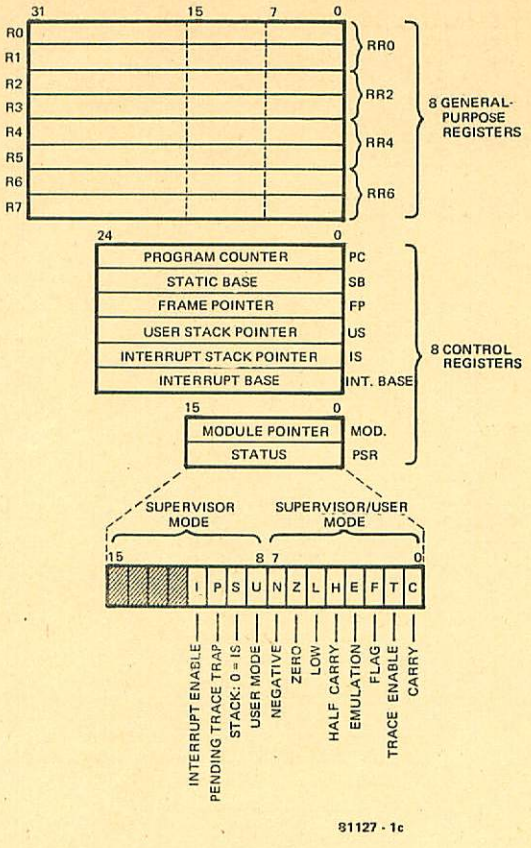


Bild 1c. Auch beim 16000-Prozessor sind die Arbeitsregister 32 bit breit.

1d

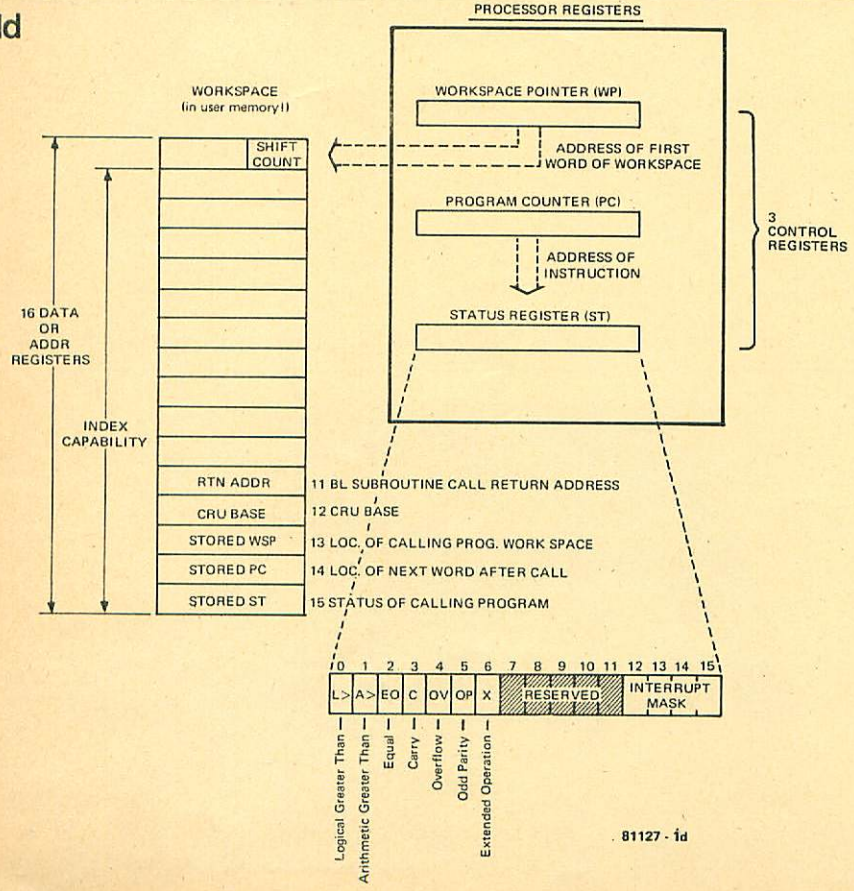


Bild 1d. Beim 9900-Prozessor sind die Register im RAM enthalten. Das kann ein großer Vorteil sein.

dafür liegt in dem höheren Alter dieser Entwicklung. Zu der Zeit waren Speicher und periphere Bausteine wesentlich teurer als heute, so daß die Systeme damals schon aus Kostengründen kleiner ausfielen. Schade, denn eine andere Eigenschaft des 9900 ist einzigartig: Dieser µP ermöglicht das Setzen von kompletten, frei verwendbaren Registersätzen in das RAM, wodurch Interrupt- und Subroutinen-Prozeduren enorm vereinfacht werden.

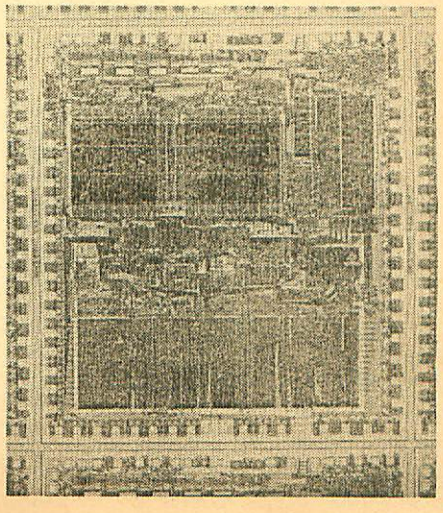
- Zilog (Z8001, 8002, 8003 8004) wollte offensichtlich einen Mikroprozessor höchster Leistungsfähigkeit schaffen, der sich so universell wie möglich verwenden läßt. Dies scheint gelungen zu sein, denn das Ergebnis ist eine geglückte Kombination aus den Vorzügen der besten bereits vorher existierenden Mikroprozessoren und den Erfahrungen aus der Minicomputerpraxis.

Register

Jeder Mikroprozessor erfüllt bestimmte Aufgaben mit Hilfe von Registern:

- In ein Register werden Daten geladen, wenn mit diesen Daten arithmetische oder logische Operationen ausgeführt werden sollen.
- Bestimmte Speicheradressen werden in Registern aufbewahrt (z.B. die Startadresse eines Datenblocks, eines Stacks oder eines Programmabschnitts).
- Andere Register erfüllen verschiedene für die Programmbearbeitung notwendige Steuerfunktionen. Dazu gehören der Programmzähler, der die Adresse der jeweils nächsten Instruktion angibt, das Statusregister mit seinen Flags und ähnliche Register.

Die Handhabung der Register kann unterschiedlich aussehen: Bei vielen älteren 8-bit-Prozessoren ist jedem Register eine bestimmte Aufgabe zugeordnet. Man findet dort einen "Akkumulator" für die Datenoperationen, einen "Stackpointer", der die erste Adresse eines Stack aufnimmt, sowie weitere spezielle Register. Eine flexiblere Lösung sind die bei neueren µP-Entwicklungen vorhandenen "Universalregister", die jede ihnen vom Programmierer zugewiesene Aufgabe übernehmen können. Ein Nachteil der



Universalregister ist die notwendigerweise größere Länge der Instruktionen. Der Befehl "Addiere 1" ist unvollständig, denn der Prozessor muß außerdem noch wissen, zu welchem Registerinhalt die Zahl 1 addiert werden soll.
Bei 16-bit-Mikroprozessoren bevorzugt man eindeutig das Universalregister-System. Sehr deutlich geht dies aus Bild 1 hervor; hier sind die Registersätze der verschiedenen Prozessoren dargestellt.
Der 8086 (Bild 1a) besitzt insgesamt vierzehn 16-bit-Register. Im Prinzip sind diese, wie im Bild angegeben, für unterschiedliche Zwecke vorgesehen. Intel betont jedoch, daß die ersten acht Register als Universalregister zu betrachten sind: "Die Datenregister können ebenso wie das Pointer- und das Indexregister ohne Einschränkung für die meisten arithmetischen und logischen Operationen benutzt werden. Die Funktion des Akkumulators bei Mikroprozessoren der ersten und der zweiten Generation kann jedes der acht Universalregister übernehmen."
Ähnliches wie für den 8086 gilt auch für den 68000 (Bild 1b). Hier sind die ersten acht 32-bit-Register für Datenoperationen vorgesehen, während die zweite, ebenfalls aus acht Registern bestehende Gruppe, zur Stack- und Base-Adressierung dient. Alle sechzehn Register können zur Indizierung benutzt werden.
Der 16000 (Bild 1c) besitzt acht 32-bit-Universalregister sowie eine größere Gruppe von Steuerregistern.
Ein ganz anderer Weg wurde beim 9900 (Bild 1d) beschritten. Der Prozessor selbst enthält die beiden üblichen Steuerregister (Programmzähler und Statusregister) und außerdem einen "Workspace Pointer". Dieser zeigt die Adresse des ersten Registers an, das zusammen mit weiteren Registern *im RAM* untergebracht ist. Insgesamt besteht der Registersatz hier aus sechzehn Universalregistern. Wenn ein weiterer Satz von sechzehn Registern für eine Subroutine oder einen Interrupt benötigt wird, braucht nur die Adresse im Workspace Pointer geändert zu werden!
Die 8000-Familie schließlich (Bild 1e) verfügt über sechzehn Universalregister, von denen eine bzw. zwei doppelt vorhanden sind: einmal für den System- und einmal für den Normal-Modus.
In diesem Zusammenhang muß man auf die Möglichkeiten der Teilung und der Kombination von einzelnen 16-bit-Registern mit größerer oder kleinerer Breite hinweisen. Die punktierten Linien in den Zeichnungen aus Bild 1 deuten bereits darauf hin. Auch in dieser Hinsicht unterscheiden sich die einzelnen Typen voneinander:

- 8086: Die ersten vier Register können in sechzehn 8-bit-Abschnitte aufgeteilt und einzelnen adressiert werden. Hier stehen also vier 16-bit-Register oder acht 8-bit-Register oder eine beliebige Kombination aus beiden Registerarten zur Verfügung.

1e

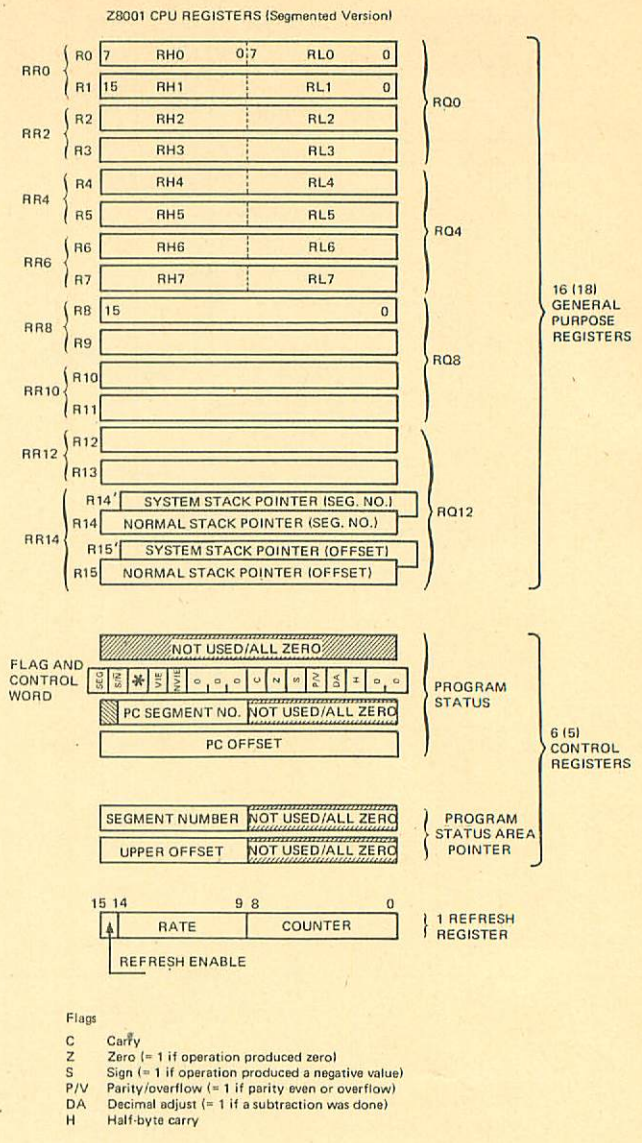
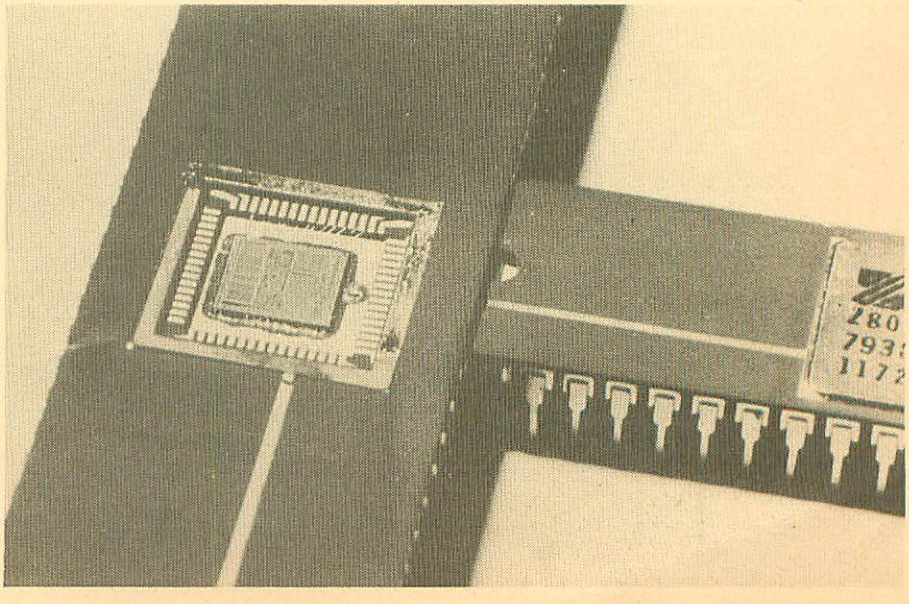
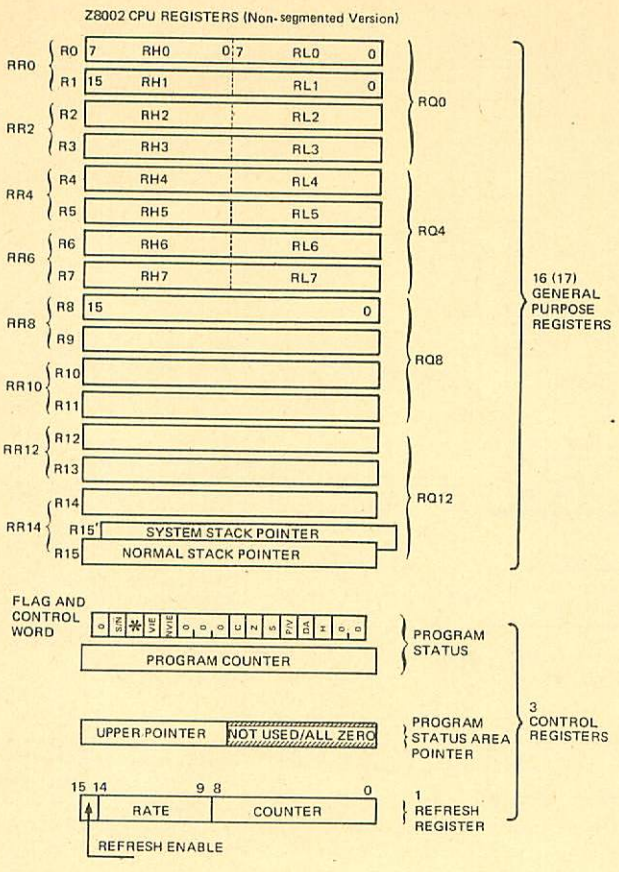
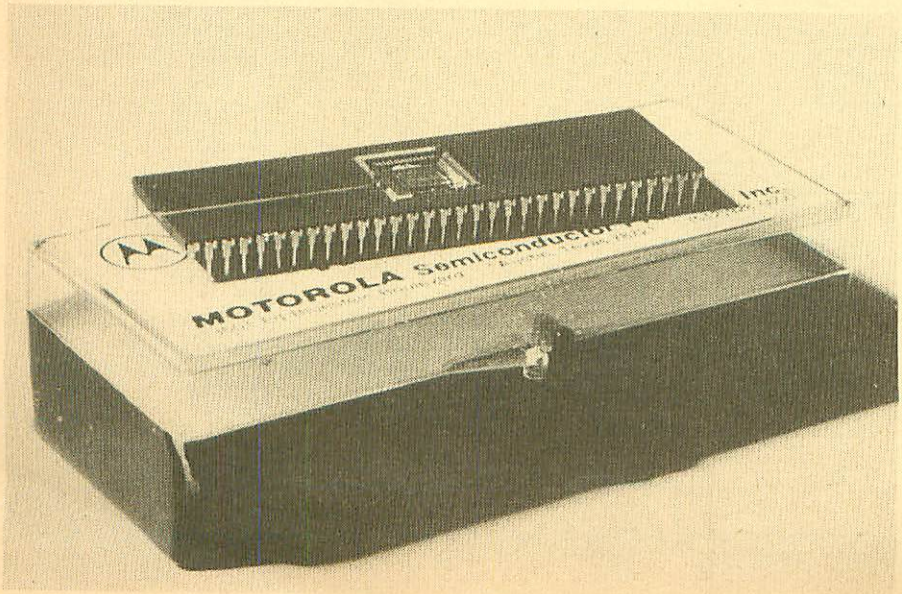


Bild 1e. Register des 8001- und 8002-Prozessors. Sie können unterschiedlich kombiniert und





sogar zu einem 64-bit-Register zusammengeschaltet werden!



- 68000: 8-bit- und 16-bit-Sektionen der ersten acht 32-bit-Register lassen sich wie im Bild gezeigt beliebig verwenden; die übrigen Register können nur in 16-bit-Blöcke unterteilt werden.
- 16000: Für Datenformate von 8 bit oder 16 bit wird der niederwertige Teil eines Registers benutzt. Hier ist es auch möglich, zwei Register zu kombinieren und dieses Registerpaar wie ein einzelnes 64-bit-Register zu verwenden.
- Z8000: Die ersten acht Register können halbiert werden. Außerdem lassen sich 16-bit-Register-Paare als 32-bit-Register und 16-bit-Register-Quartette als 64-bit-Register verwenden.

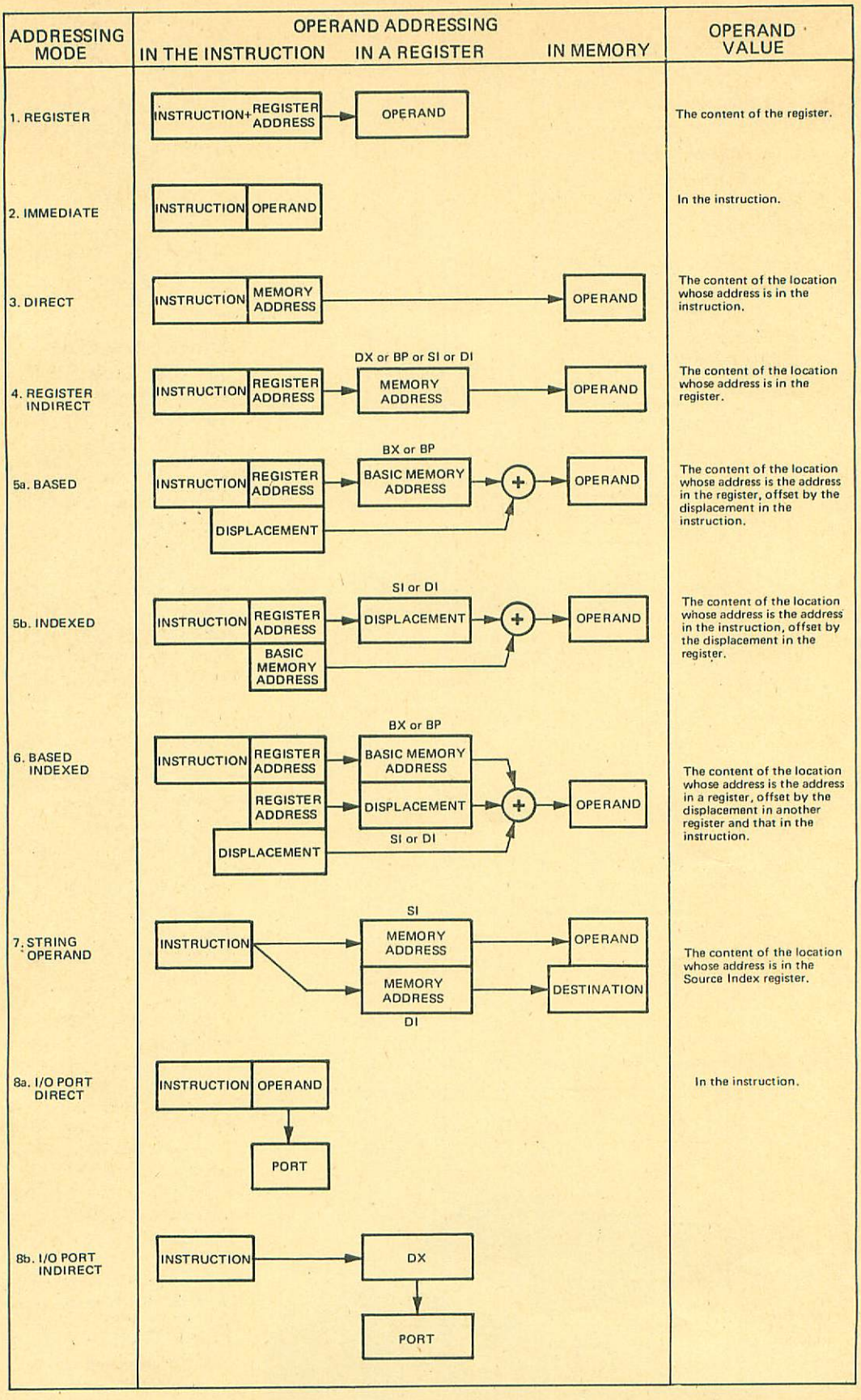
Adressierungsarten

Die Anweisungen an den Prozessor, aus denen sich bekanntlich ein Programm zusammensetzt, müssen grundsätzlich zwei Informationen enthalten: Zum einen muß der Prozessor wissen, welche Operation er ausführen soll, zum anderen, mit welchen Daten diese Operation vorzunehmen ist. Um dem Prozessor mitzuteilen, wo die betreffenden Daten stehen, kann der Programmierer unter verschiedenen Möglichkeiten wählen. Man nennt sie die Adressierungsarten des Prozessors. Die folgenden Adressierungsarten sind bei praktisch allen Prozessoren vorhanden:

- "Register": Der Befehl bezieht sich auf ein bestimmtes Register; dieses Register enthält das Datenwort.
- "Immediate": Das Datenwort ist Bestandteil der Instruktion.
- "Direct": Der Befehl ist durch eine Speicheradresse ergänzt, unter der das Datenwort im Speicher steht.
- "Indirect": Der Befehl bezieht sich auf ein Register oder ist durch eine Speicheradresse ergänzt; dieses Register bzw. dieser Speicherplatz enthält die Adresse, unter der das Datenwort zu finden ist.
- "Relative": Das Datenwort steht unter einer Adresse im Speicher, die vom augenblicklichen Stand des Programmzählers um eine bestimmte Schrittzahl entfernt liegt; diese Schrittzahl ist Bestandteil der Instruktion.
- "Indexed": Das Datenwort steht unter einer Adresse im Speicher, die von der in der Instruktion genannten Adresse um eine bestimmte Schrittzahl entfernt liegt; diese Schrittzahl steht in einem Indexregister.

Zusätzlich zu diesen Adressierungsarten bieten die einzelnen Prozessoren weitere spezielle Möglichkeiten der Adressierung. Aus Bild 2 geht hervor, daß sich auch hier die einzelnen 16-bit-µP-Typen voneinander unterscheiden. Dazu muß noch folgendes angemerkt werden: Leider ist die Bezeichnung von einzelnen Adressierungsarten nicht einheitlich; sie kann in Abhängigkeit vom Hersteller variieren. Die Adressierungsart "Direct" bedeutet normalerweise, daß die Instruktion die Speicheradresse enthält, unter der das Datenwort im Speicher steht. Motorola nennt diese Adressierungsart jedoch "Absolute"; bei "(Register) Direct"

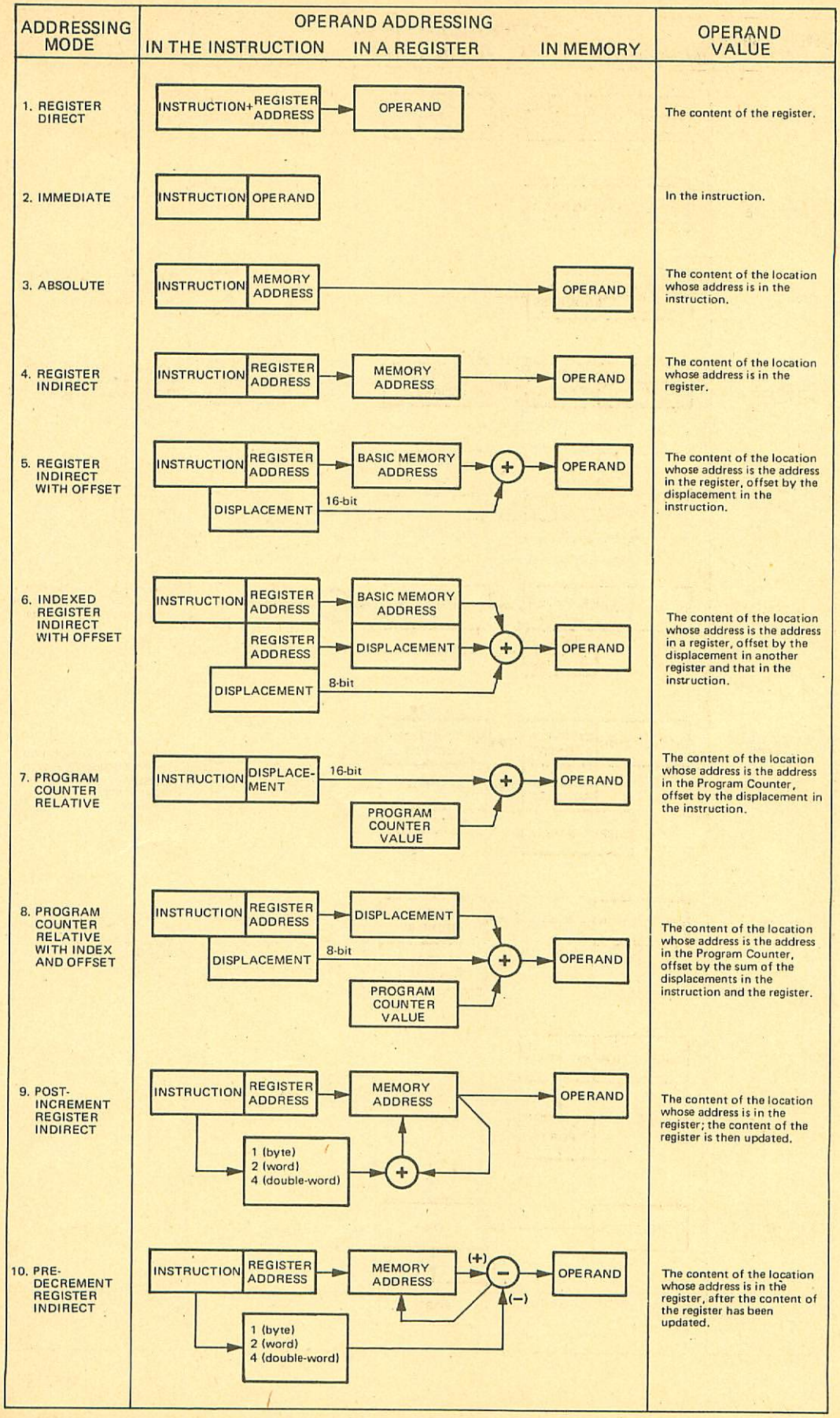
2a



81127 - 2a

Bild 2a. Die Adressierungsmöglichkeiten des 8086- (und 8088-) Prozessors. Auffällig ist, daß in den meisten Adressierungsarten ein bestimmtes Register benutzt werden *muß* – zum Beispiel SI oder DI für INDEXED.

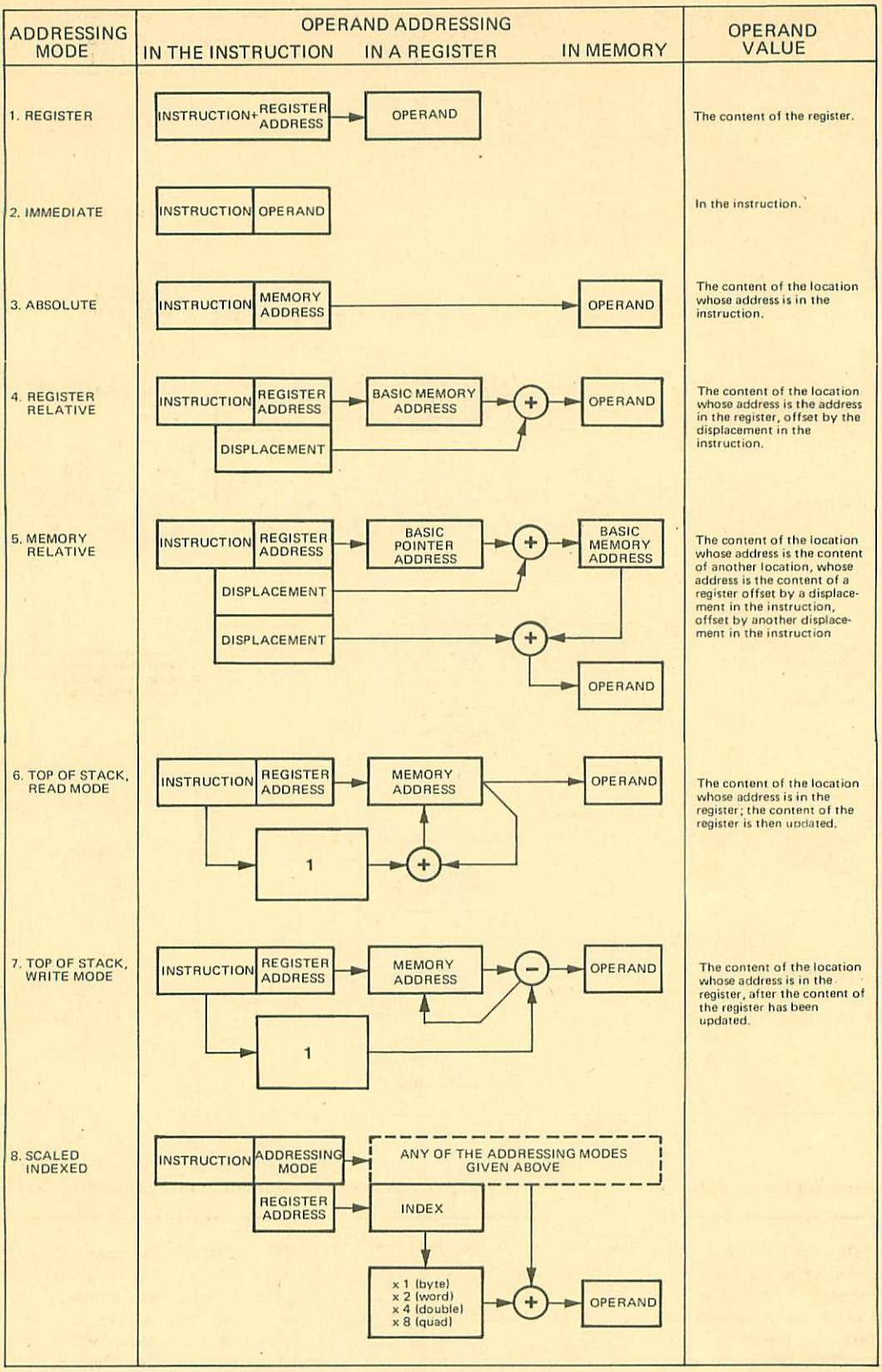
2b



81127 - 2b

Bild 2b. Der 68000-Prozessor kennt auch die Adressierung mit POST-INCREMENT und PRE-INCREMENT. Sehr schön, wenn große Datenblöcke abgearbeitet werden müssen!

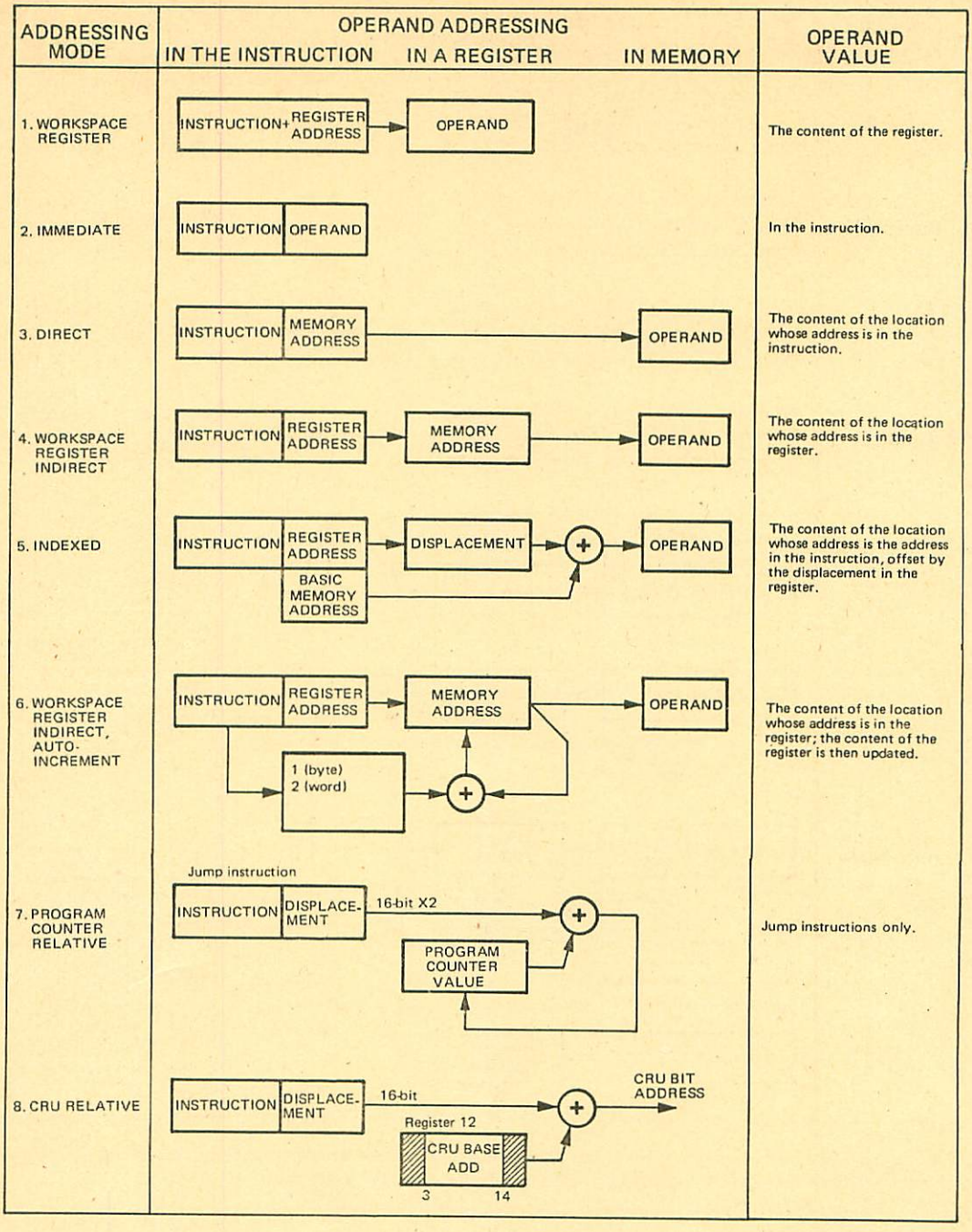
2c



81127 - 2c

Bild 2c. Der 16000-Prozessor bietet wieder andere Möglichkeiten. MEMORY RELATIVE und SCALED INDEXED sind hier vor allem erwähnenswert. Mit dieser letzten Adressierungsart ist eine umfangreiche indizierte Adressierung möglich!

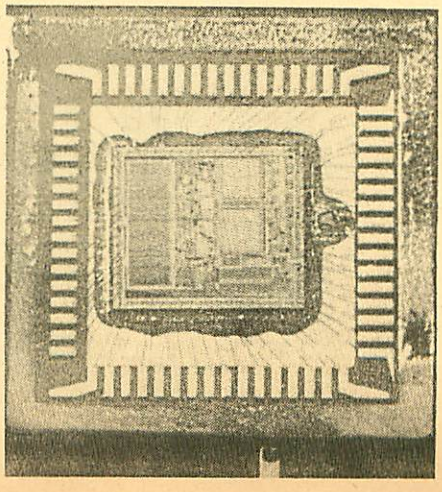
2d



81127 - 2d

Bild 2d. Der 9900-Prozessor hat ebenfalls die bekannten Adressierungsarten und außerdem einige Prozessor-spezifische Varianten.

steht hier das Datenwort in dem Register, auf das sich der Befehl bezieht.
Noch eine weitere Anmerkung zu Bild 2: Der Unterschied zwischen den Adressierungsarten "Based" und "Indexed" beim 8086 ist nicht groß, aber trotzdem von Bedeutung. Das soll an einem Beispiel deutlich werden: Nehmen wir an, daß die Daten aller in einem Betrieb beschäftigten Mitarbeiter in Form von Tabellen im Speicher stehen. Zu jedem Mitarbeiter gehört eine solche personenbezogene Tabelle. Will man nun sämtliche Daten eines bestimmten Mitarbeiters ausdrucken lassen, dann kann man sich der Adressierungsart "Indexed" bedienen. Dem Computer wird die erste Adresse der



auszudruckenden Tabelle eingegeben; er durchläuft diese, indem er den Inhalt des Indexregisters schrittweise um eins erhöht. Ein anderes Vorgehen ist notwendig, wenn beispielsweise die Summe der Personalkosten berechnet werden soll. Der Computer erhält die Angabe, an welcher Stelle in jeder Tabelle die Gehaltsangabe zu finden ist. Um aus jeder Tabelle nur diese eine Information zu entnehmen, ist die Adressierungsart "Based" vorgesehen. Mit ihr wird der Inhalt des "Base adress"-Registers in gleichbleibenden Intervallen erhöht. Einige Prozessoren bieten diese Möglichkeiten als Erweiterung der Adressierungsart "Indexed" an ("Increment" und "Decrement"),

2e

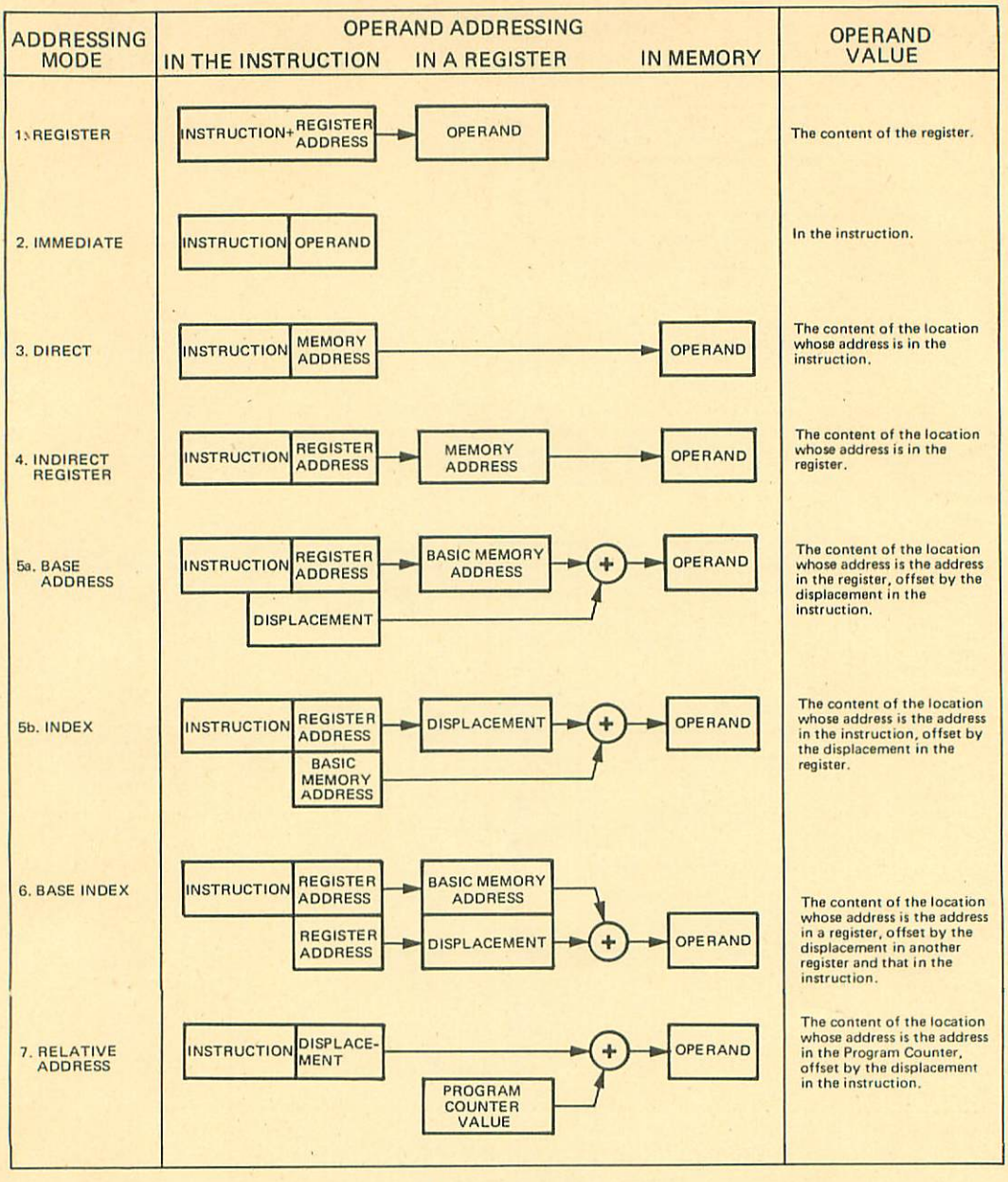


Bild 2e. Die 8001- und 8002-Prozessoren bieten auf den ersten Blick weniger Möglichkeiten als die anderen. Einige der Adressierungsarten, z.B. auch INCREMENT und DECREMENT, tauchen jedoch bei diesen Prozessoren im Befehlssatz auf.

während andere über separate Inkrement- und Dekrement-Befehle verfügen. Die Werte der Intervalle können dabei 1, 2, 4 oder sogar beliebig (Z8000) sein.

Die Art der Unterbringung von Datenworten im Speicher ist ein weiterer Punkt, zu dem einige Anmerkungen notwendig sind. Die auf dem Markt befindlichen Speicher-ICs wurden für Prozessoren mit 8-bit-Datenbus konzipiert. Wie aber speichert man nun ein 16-bit-Datenwort? Sicherlich in zwei Blöcken zu jeweils 8 bit. Das bedeutet aber, daß zu jedem 16-bit-Wort zwei Speicheradressen gehören. Die einzelnen Hersteller verfahren damit unterschiedlich: Intel und National entschieden sich, das niederwertige Byte unter der niederwertigen Adresse abzulegen; sie schreiben beispielsweise die Zahl "1981" als "8119". Bei den übrigen Herstellern stehen die beiden (8-bit-) Bytes eines 16-bit-Datenworts in um-

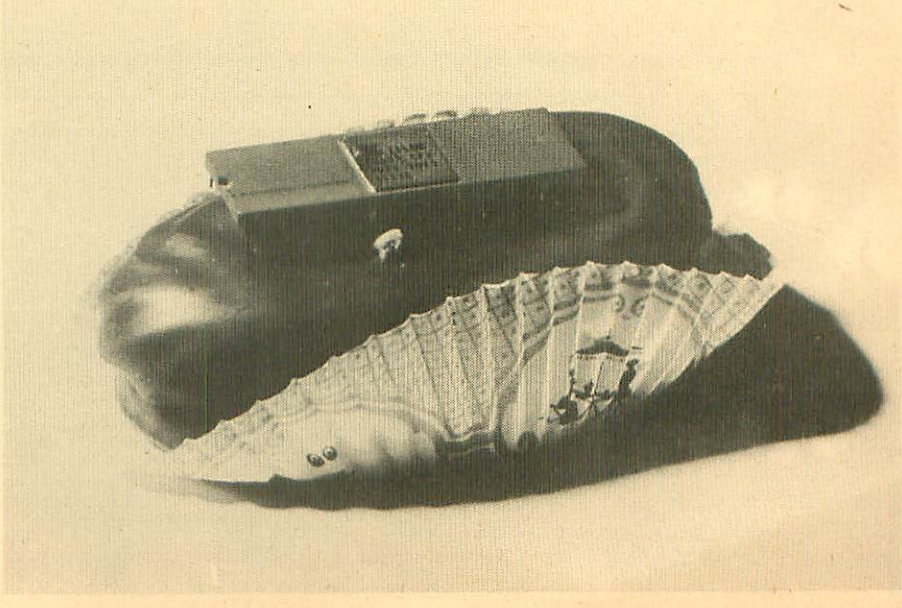


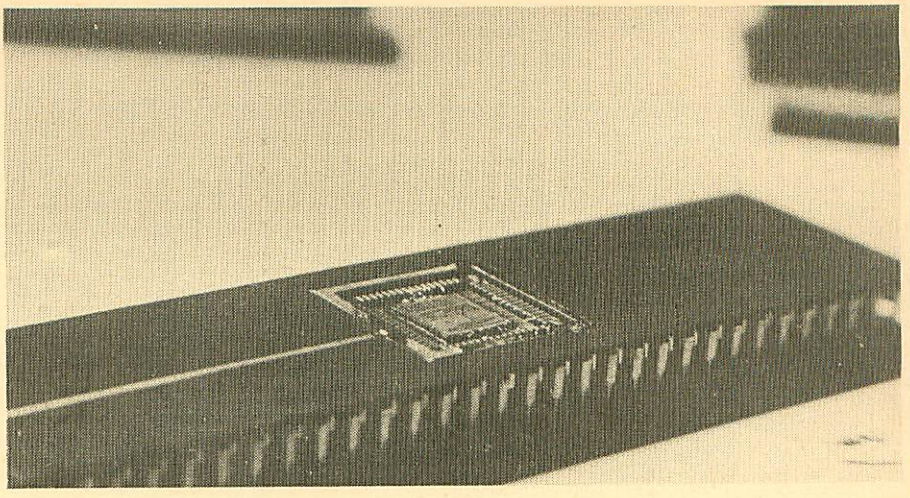
Tabelle 4.

	INTEL 8086	MOTOROLA 68000	NATIONAL 16032	TEXAS 9900	ZILOG 8001
DATA TRANSFER					
• **move:** general purpose	X	X	X	XX	13 (B/W/DW)
immediate to register	X	X	X	X	
immediate to memory	X				XX
to/from dedicated registers (accu, addr. reg, program status etc.)	XX (accu) XX (segm. reg.) XX (EA/pointers) XX (flags)	X (CCR) XX (SR) X (USP) XX (An) XX (SP/An)		(XX CRU = I/O) XXX (workspace pointer) X (SR) X (int. mask)	
move multiple registers		X			
• **stack:** push	XXXX				XX
pop	XXXX				XX
save registers			X		XX
restore registers			X		
• **other:** exchange data	XX	X			
clear		X			XX
swap bytes		X		X	XX
load address	X		X	X	
translate byte	X				XX
BLOCK TRANSFER AND STRING MANIPULATION					
• repeat	X				
• move	X		XX		(n.a.)
load	X				XXXX
store	X				
move and repeat			(X)		
• compare	X		XX		XXXX
compare and repeat			(X)		8
scan	X				8
translate	X				
translate and repeat					XX
translate, test					XX
translate, test and repeat					XX
• skip string			X		XX
INPUT/OUTPUT					
• **input**	XX	m	m	X (CRU)	XX
input and incr./decr.	see 8089	e	e		XXXX
input, incr./decr. and repeat		m	m		XXXX
special input		o	o		XX
special input and incr./decr.		r	r		XXXX
special input/incr./decr. and repeat		y	y		XXXX
• **output**	XX	m	m	X (CRU)	XX
output and incr./decr.	see 8089	a	a		XXXX
output, incr./decr. and repeat		p	p		XXXX
special output		p	p		XX
special output and incr./decr.		e	e		XXXX
special out, incr./decr. and repeat		d	d		XXXX
• **more** peripheral data (8-bit)		X			
• **communication register:**					
test CRU bit				X	
set CRU bit				X	
clear CRU bit				X	
ARITHMETIC					
• **add**	XXX	XXXX	XX	XXX	XXX
add with carry	XXX		X	(X: 9940)	XX
add decimal		X	X		
decimal adjust for add	X	(n.a.)			X
ASCII adjust for add	X				
increment by one	XX			X	
increment by two				X	
increment by 'n'					XX
add address		X	X		
• **subtract**	XXX	XXXX	X	XX	XXX
subtract with borrow	XXX		X		XX
subtract decimal		X	X	(X:9940)	
decimal adjust for sub.	X	(n.a.)			(X)
ASCII adjust for sub.	X				
decrement by one	XX			X	
decrement by two				X	
decrement by 'n'					XX
change sign	X	X	X	X	XX
change sign, decimal		X			
subtract address		X			
• **multiply**, unsigned	X	X	X	X	
multiply, signed	X	X	X		XX
ASCII adjust for mult.	X				
• **divide**, unsigned	X	X	X	X	
divide, signed	X	X	X		XX
ASCII adjust for divide	X				
extend sign	XX	X	XX		XXX
evaluate periodic function			X		
modulus of periodic function			X		
remainder			X		
• **compare**	XXX	XXX	XX	XXX	XXXXX
check R against bounds		X	X		
compare address		X	X		
• **absolute value**			X	X	

	INTEL 8086	MOTOROLA 68000	NATIONAL 16032	TEXAS 9900	ZILOG 8001
LOGIC					
• AND	XXX	XX	X	X	XX
• OR	XXX	XX	X	X	XX
• EX`R	XXX	XX	X	X	XX
• NOT	X	X	X	X	XX
• test flag(s)/CC	XXX	(n.a.)		(X CRU = I/O)	XX
test operand		X			XXX
test and set		X			
ROTATE AND SHIFT					
• shift logical left	} X	X	X	} X	XXX
shift arithmetic left		X	X		XXX
shift logical right	X	(X) ≈ SLL	(X) ≈ SLL	X	XXX
shift arithmetic right	X	(X) ≈ SAL	(X) ≈ SAL	X	XXX
shift dynamic logical			(X) ≈ SLL		XXX
shift dynamic arithmetic			(X) ≈ SAL		XXX
• rotate right	X	X	X	X	XX
rotate right through carry/extend	X	X			XX
rotate left	X	X	(X) ≈ RR		XX
rotate left through carry/extend	X	X			XX
rotate digit left					X
rotate digit right					X
BIT MANIPULATION					
• bit test		X	X		XXXX
bit test and change		X			
bit test and clear		X			
bit test and set		X			XX
compare ones corresponding				X	
compare zeroes corresponding				X	
find first set bit			X		
• set ones				X	
set bits corresponding			X	XX	XXXX
set CRU bit				(X = I/O)	
set bit			XX		
• reset bits corresponding			X	XX	XXXX
reset CRU bit				(X = I/O)	
reset bit			XX		
• invert bit			X		
• extract bit field			XX		
insert bit field			XX		
convert bit field pointer			X		
PROGRAM CONTROL					
• call subroutine	XXXX	XX	XX	XX	XX
return from call	XXXX	X	X		X
extended operation (user-def.)			XX	X	
execute (variable instruction)				X	
system call			X		X
• interrupt call	XXX				
return from interrupt	X	XX	XX		X
• jump/branch, unconditional	XXXXX	XX	XX	XX	
jump/branch, conditional	16	X	14	12	4
multiway branch			X		
• loop, conditional	XXX	X	X		
jump from loop	X				
PROCESSOR CONTROL					
• control bits, clear	XXX		X		XX
control bits, set	XXX	X	XX		XX
control bits, invert	X				X
control bits, move			XX		XXXXX
multi-micro request					X
multi-micro set					X
multi-micro reset					X
multi-micro test					X
• halt, wait	XX	X	X	X	X
NOP		X	X		X
reset (external devices)		X		X	
escape (to external device)	X				
restart				X	
clock bus	X				
segment override	X				
trap		X	X		
trap on overflow		X			
clock off				X	
clock on				X	
breakpoint			X		

Tabelle 4. Vergleichende Übersicht aller Befehlssätze. Die Zahl der Kreuze (oder eine Dezimalzahl) steht für die möglichen Varianten des betreffenden Befehls. Diese Übersicht gibt natürlich nur einen allgemeinen Eindruck. Exakte Angaben sind den "offiziellen" Datenblättern der Hersteller zu entnehmen.

gekehrter Reihenfolge. Ferner müssen die Daten in den meisten Fällen so in den Speicher eingelesen werden, daß die erste Adresse jedes 16-bit-Worts eine gerade Zahl ist.
Dadurch wird eine Leitung des Adreßbusses eingespart und gleichzeitig ein größerer maximaler Offset bei der Adressierungsart "Relative" erreicht. Andererseits können Daten nicht beliebig in den Speicher gebracht werden, was Intel (8086/8088) dazu veranlaßte, seinen Prozessor mit beiden Möglichkeiten auszustatten: Die Datenwörter können entweder unter Beachtung der geradzahligen Adressen oder aber beliebig im Speicher stehen; das erste Verfahren ist jedoch das schnellere.



Befehlssätze

Man sollte meinen, daß eine möglichst hohe Anzahl unterschiedlicher Befehle ein selbstverständliches Ziel bei der Konstruktion eines Mikroprozessors ist. Dies stimmt jedoch nur mit Einschränkungen. Wichtig ist an erster Stelle, daß die Effektivität der einzelnen Befehle möglichst hoch ist. Für den Transfer von Datenblöcken bietet zum Beispiel der 8086 die drei Befehle "Repeat", "Compare" und "Decrement" an, während diese Befehle beim Z8000 zu einem einzigen Befehl "Compare, Decrement and Repeat" zusammengefaßt sind. Jeder Prozessor hat seine Stärken und Schwächen; der 8086, um ein weiteres Beispiel zu nennen, ist dafür als einziger mit dem Befehl "ASCII Adjust for Add and Subtract" ausgestattet.
In Tabelle 4 werden die Befehlssätze der einzelnen 16-bit-µPs, soweit dies überhaupt möglich ist, einem Vergleich unterzogen. Das aus dieser Tabelle resultierende Gesamtbild muß zwangsläufig unvollständig bleiben; ein exakter Vergleich ist nur durch eingehendes Studium der Datenbücher möglich. Die Befehlscodes in Maschinensprache lassen sich bei einigen Prozessoren relativ leicht im Gedächtnis behalten, was die Praxis des Hobby-Programmierers erleichtert. Einige Assembler sind effektiver als andere; dies ist mehr für die professionellen Anwender von Bedeutung. Die Befehlssätze orientieren sich nach Art und Umfang zum Teil mehr, zum Teil weniger an höheren Programmiersprachen wie zum Beispiel PASCAL. Alle diese sicher nicht unwichtigen Details gegeneinander abzuwägen, würde jedoch den Rahmen dieses Artikels sprengen.

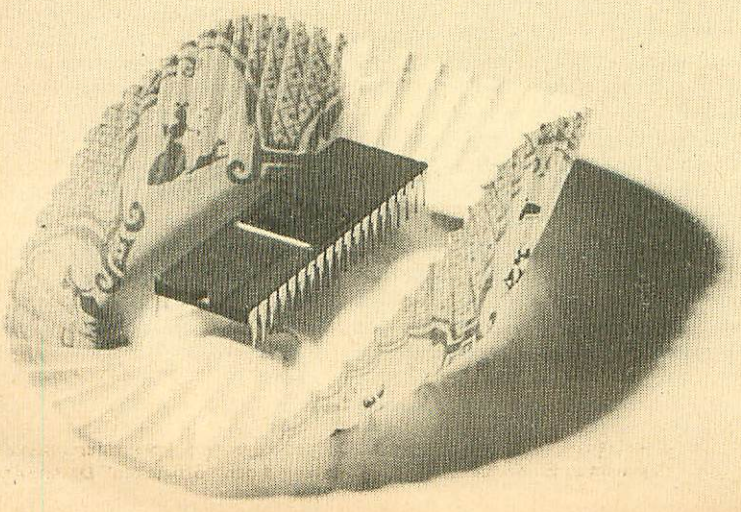
Interrupts

Ein Interrupt ist eine Unterbrechung des laufenden Programms zu einem Zeitpunkt, an dem der Prozessor eine dringendere Aufgabe vorab erledigen muß. Wenn dies geschehen ist, kehrt der Prozessor zum ursprünglichen Programm zurück und setzt die Abarbeitung dieses Programms an gleicher Stelle fort. Zum Beispiel nutzen manche Schachcomputer die Zeit, während der ihr menschlicher Gegner am Zug ist, zum eigenen "Nachdenken". Sobald der Stand der Figuren vom Menschen verändert wird, muß der Computer seine Berechnungen unterbrechen und diese Veränderung registrieren, bevor er seine Berechnungen fortsetzen kann.
Interrupts können bei einem Computersystem von den verschiedensten externen Quellen stammen. Entsprechend zahlreich sind auch die Interruptroutinen, die jeweils durchlaufen werden müssen. Damit keine Zeit verlorengeht, muß der Prozessor möglichst schnell eine Information darüber erhalten, welche Routine er ansteuern muß. Alle 16-bit-Prozessoren arbeiten hier mit sogenannten Interruptvektoren: Die Interruptquelle gibt die Position in einer Adressentabelle an, an der die Startadresse der benötigten Interruptroutine steht.
Die Adressentabelle ist abhängig vom Prozessortyp an einer bestimmten Stelle im Speicher untergebracht. Wie aus Bild 3 hervorgeht, benutzen die meisten Prozessoren hierzu einen Bereich, der mit der Adresse 0000 beginnt; manche verwenden zusätzlich noch einen Bereich in der Nähe der letzten Speicheradresse. Der Z8000 und der NS16000 machen hier eine Ausnahme: Beim Z8000 kann die "Program Status Area" ebenso wie beim 16000 die "Interrupt and Trap Vector Table" beliebig in den Speicherbereich gelegt werden.
Außerordentlich nützlich ist ferner eine Einteilung der Interrupts in Dringlichkeitsstufen, gemessen an dem zum Zeitpunkt ihres Auftretens laufenden Abschnitt des Hauptprogramms. Dies führt zur Unterscheidung folgender Interruptarten:

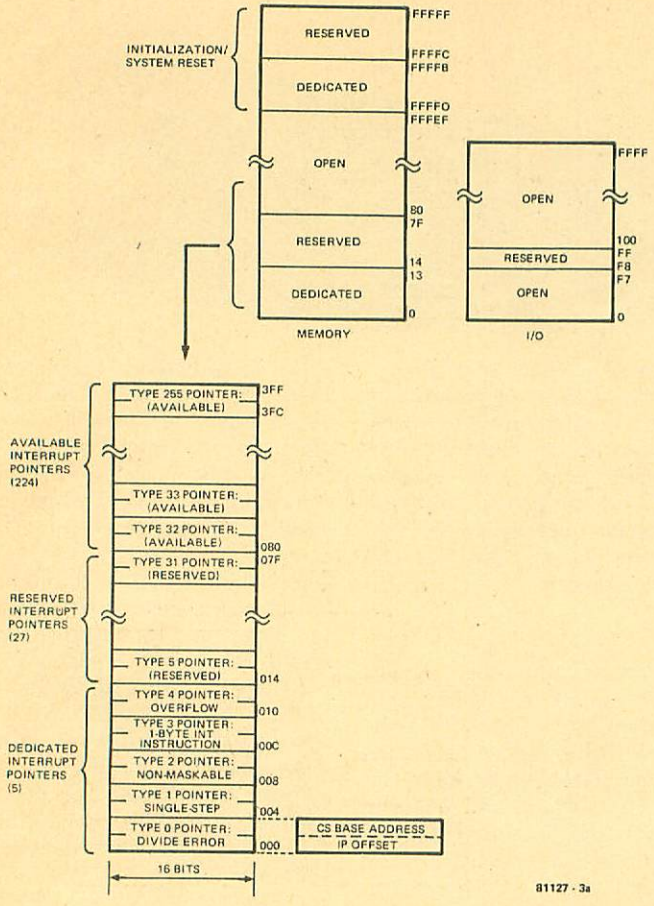
- Nicht maskierbarer Interrupt: Wenn dieser auftritt, muß die zugehörige Interruptroutine ohne Verzögerung begonnen werden. Sie hat unbedingten Vorrang vor allen anderen Aufgaben des Prozessors.
- Prioritätskodierte Interrupts: Hier liefert die Interruptquelle eine Information über die Dringlichkeit des Interrupt. Ein solcher Interrupt wird nur erledigt, wenn seine Dringlichkeit höher als der gerade ausgeführte Abschnitt des Hauptprogramms ist.

Schließlich unterscheidet man noch

- normale Interrupts: Diese werden wie oben beschrieben von außen durch irgendein unvorhersehbares Ereignis ausgelöst.
- Software-Interrupts ("Traps"): Ihr Ursprung ist beispielsweise ein Overflow während einer normalen Abarbeitung eines Programms. Bei einigen Prozessoren sind Software-Interrupts auch durch Befehl innerhalb des Programms möglich; beim 8086 können sogar Hardware-Interrupts vom Programm ausgelöst werden.



3a



3b

ADDRESS HEX	VECTOR NUMBER(S)	ASSIGNMENT
000	0	RESET: INITIAL SSP
004	–	RESET: INITIAL PC
008	2	BUS ERROR
00C	3	ADDRESS ERROR
010	4	ILLEGAL INSTRUCTION
014	5	ZERO DIVIDE
018	6	CHK INSTRUCTION
01C	7	TRAPV INSTRUCTION
020	8	PRIVILEGE VIOLATION
024	9	TRACE
028	10	LINE 1010 EMULATOR
02C	11	LINE 1111 EMULATOR
030	12*	(UNASSIGNED, RESERVED)
034	13*	(UNASSIGNED, RESERVED)
038	14*	(UNASSIGNED, RESERVED)
03C	15	UNINITIALIZED INTERRUPT VECTOR
040	16-23*	(UNASSIGNED, RESERVED)
05F	23	•
060	24	SPURIOUS INTERRUPT
064	25	LEVEL 1 INTERRUPT AUTOVECTOR
068	26	LEVEL 2 INTERRUPT AUTOVECTOR
06C	27	LEVEL 3 INTERRUPT AUTOVECTOR
070	28	LEVEL 4 INTERRUPT AUTOVECTOR
074	29	LEVEL 5 INTERRUPT AUTOVECTOR
078	30	LEVEL 6 INTERRUPT AUTOVECTOR
07C	31	LEVEL 7 INTERRUPT AUTOVECTOR
080	32-47	TRAP INSTRUCTION VECTORS
0BF	47	•
0C0	48-63*	(UNASSIGNED, RESERVED)
0FF	63	•
100	64-255	USER INTERRUPT VECTORS

81127 - 3b

Bild 3. Alle Prozessoren arbeiten mit mehr oder weniger RAM-Speichern. Unter anderem werden die Adressen von Interrupt-Routinen dort gespeichert. Bei dem 16000- (Bild 3c) und dem 8001- (Bild 3e) Prozessor kann das in einem beliebigen RAM-Bereich geschehen; bei allen anderen liegt dieser Bereich mindestens teilweise fest.

Systemausbau

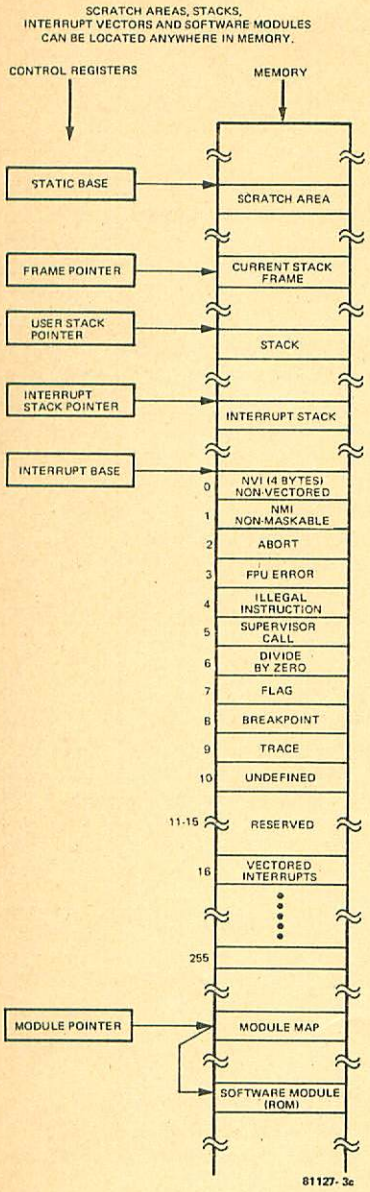
Es wurde bereits erwähnt: Computersysteme neigen dazu, stetig oder auch sprunghaft zu wachsen. Bild 4 gibt einen Eindruck davon, wohin dies in der Praxis führen kann. Die Grenzen der Ausbaufähigkeit sind in diesen Skizzen allerdings noch längst nicht sichtbar, denn fast täglich kündigen Hersteller neue Chips an, die das System auf die eine oder andere Weise ergänzen. Einige dieser "Erweiterungen" sind jedoch manchmal überflüssig, soweit es um die Lösung von typischen Mikrocomputer-Aufgaben geht. So ist zum Beispiel das Demultiplexen des Daten- und Adreßbusses, wie in den Bildern angegeben, nicht in allen Fällen notwendig.

Bei einigen peripheren Chips handelt es sich um autonome (Slave-)Mikroprozessoren. Als Beispiel sei der Input/Output-Prozessor 8089 (Bild 4a) genannt, der in gerader Linie der 8080-Familie entstammt. Ähnliches gilt für die Speichermanagement-Bausteine (MMU = Memory Management Unit) in Bild 4b, 4c und 4e. Hier gibt gleichzeitig ein neues Speicherkonzept sein Debüt; "virtuelle" Speicher lösen die "realen" Speicher ab. Wenn ein Prozessor imstande ist, einen Speicherbereich von 48 oder sogar 64 Megabyte zu adressieren, so läßt sich ein derart umfangreicher Speicher nur schwerlich als RAM realisieren. Aus diesem Grund geht man allgemein dazu über, das RAM wesentlich kleiner zu halten und die fehlende Speicherkapazität durch einen Floppy-Disk- oder einen ähnlichen Low-Cost-Massenspeicher zu ersetzen. Sobald es das Programm verlangt, werden die benötigten Daten von der Platte in das RAM geladen und damit dem Prozessor zugänglich gemacht.

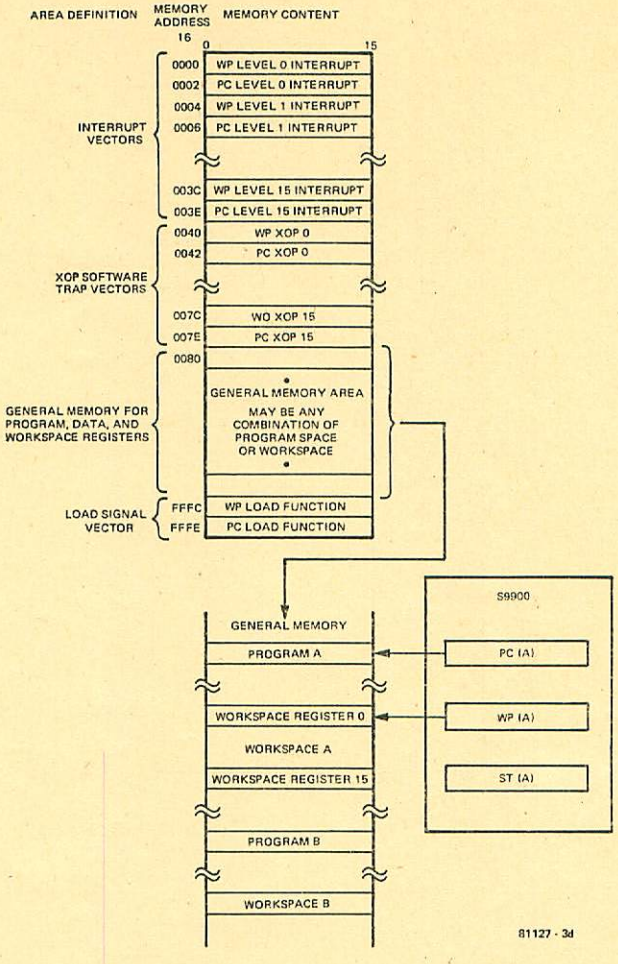
Um den Prozessor (und den Programmierer!) mit diesem ständigen Transport von großen Datenmengen nicht zu überlasten, leistet der Speichermanagement-Baustein Hilfestellung. Er prüft die vom Prozessor ausgegebenen logischen Adressen und stellt fest, ob sich das betreffende Datensegment zur Zeit im RAM oder außerhalb befindet. Ist ersteres der Fall, dann setzt der MMU die physikalische Adresse (RAM-Adresse) auf den Adreßbus. Anderenfalls stoppt er den Mikroprozessor und tauscht den Inhalt des RAM gegen das benötigte Datensegment aus; erst dann erhält der Prozessor ein neues Startzeichen. Damit hierbei die im Prozessor stehenden Daten nicht verändert werden oder verlorengehen, sind natürlich geeignete Vorkehrungen notwendig.

An dieser Stelle soll die "Abort"-Möglichkeit eingeführt werden. Zilog definierte das bei der Vorstellung der Z8003- und Z8004-Prozessoren folgendermaßen: "'Abort' erlaubt die Unterbrechung von Befehlen oder des Zugriffs zu Daten, die nicht im Hauptspeicher enthalten sind. Oder, allgemeiner gesagt, wenn der Z8003/4-Prozessor versucht, nicht-existente Daten aus dem Speicher

3c



3d



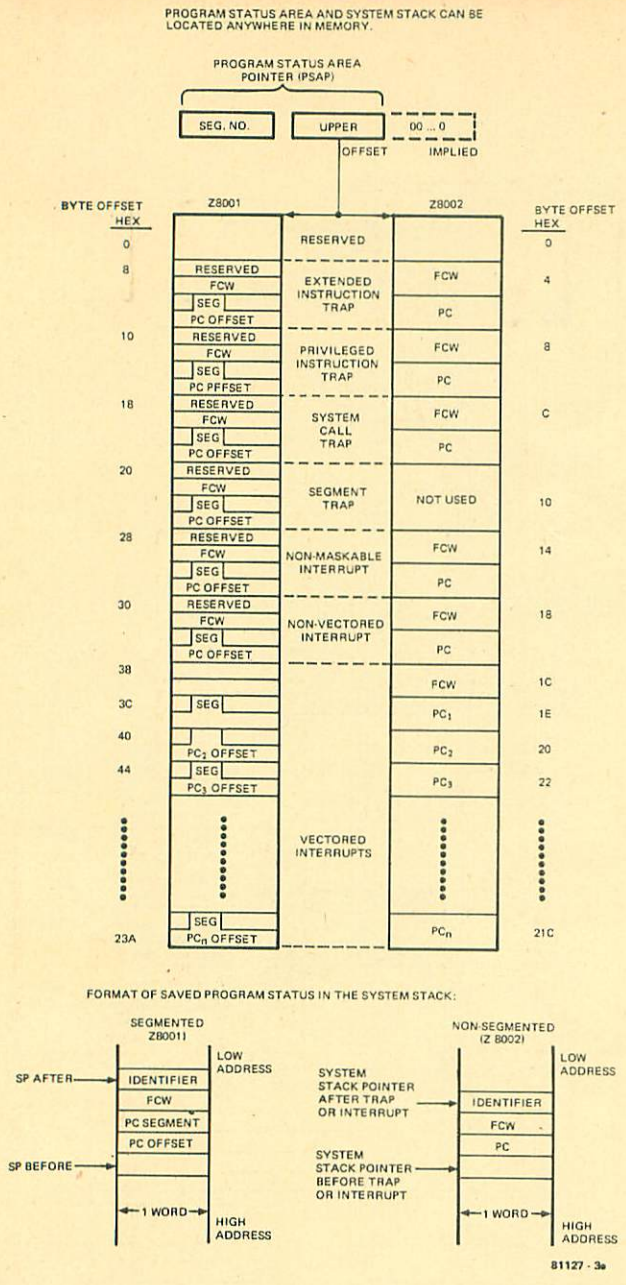
zu holen, wird dieser Versuch 'höflich' untersagt".
Für den Systemausbau sind ferner Leistungsmerkmale wie der direkte Speicherzugriff (DMA), Multi-Prozessor-Betrieb usw. von großer Wichtigkeit. Da jedoch alle in diesem Artikel genannten Prozessoren diese Leistungsmerkmale besitzen, soll auf die Darstellung von Details verzichtet werden. Auch was die Software betrifft, so stehen sich die einzelnen Bewerber in nichts nach. Verfügbar ist eine Fülle von Literatur, die Software aller Art bereithält.

Schlußfolgerungen

Jeder der fünf Prozessoren hat zweifellos spezielle Stärken vorzuweisen. Das heißt aber nicht, daß der eine Prozessor eher vor einem gestellten Problem kapitulieren müßte als der andere; sie sind alle nahezu jeder Aufgabe gewachsen. Die Frage nach dem "besten" Mikroprozessor wurde kürzlich von kompetenter Seite so beantwortet: "Wenn es tatsächlich einen 'besten' µP gibt, dann wird dieser Höhenflug wahrscheinlich nur von kurzer Dauer sein. Viele Faktoren, die bei der Bildung dieses globalen Urteils mitspielen, sind schnellen Wandlungen unterworfen. Niemand sollte enttäuscht sein, wenn der Mikroprozessor seiner Wahl nicht als Sieger aus irgendeinem Test hervorgeht. Die Entthronung des Spitzenreiters läßt meistens nicht lange auf sich warten!"
Wer heute mit dem Aufbau eines 16-bit-Mikrocomputersystems beginnen will, für den werden bei der Wahl des Prozessortyps noch weitere, bisher unberücksichtigte Faktoren eine Rolle spielen:

- Der Preis und die Beschaffbarkeit können sich von Woche zu Woche ändern. Da Preise in dieser Branche manchmal gleich einem Erdrutsch fallen, empfiehlt es sich, den aktuellen Stand bei Distributoren und Händlern zu erfragen. Allerdings dürfte das beim NS16000 zur Zeit noch nicht möglich sein, denn dieser Prozessor ist so brandneu, daß die für den Artikel benötigten Unterlagen aus den USA beschafft werden mußten. Die ersten Exemplare des 16000 erscheinen voraussichtlich Ende des Jahres auf dem Markt.
- Ein handlicher und effektiver Befehlssatz ist für den Hobbyprogrammierer meistens wertvoller als ein besonders leistungsfähiger Assembler. Wo hier letzten Endes die Schwerpunkte zu setzen sind, hängt von den jeweiligen Intentionen ab. Im Auge behalten sollte man jedenfalls die bei manchen Herstellern gelegentlich beobachtbare Kluft zwischen Geschriebenem und Wirklichkeit. Zum Beispiel betont Motorola, daß sein µP "über einen effektiven, für jeden Anwendungszweck gleich gut geeigneten Befehlssatz verfügt, der das Gedächtnis des Programmierers entlastet." Das ist sicher richtig. Zilog nimmt dagegen für sich in Anspruch, den umfassendsten und ausgedehntesten Befehlssatz zu

3e



haben. Auch das stimmt zweifellos. Wenn man jedoch den Blick von den Mnemonics abwendet und die Bits der einzelnen Befehlskodes betrachtet, sind Überraschungen nicht ausgeschlossen. Ein Beispiel dafür: die Schiebebefehle. Motorola gibt für den 68000 vier Schiebebefehle an ("Arithmetic Shift", links und rechts, und "Logical Shift", links und rechts), während Zilogs Befehlsliste für den Z8000 sechs Schiebebefehle enthält ("Shift Dynamic", arithmetisch und logisch, "Shift Left", arithmetisch und logisch, und "Shift Right", arithmetisch und logisch). Motorola ergänzt, daß für dynamische und statische Shifts die gleichen Befehle gelten, denn weniger sei besser! (Dynamisch bedeutet in diesem Zusammenhang, daß die Anzahl der Positionen, um die die Bits verschoben werden, in einem Register steht; bei statischen Shifts ist sie dagegen Bestandteil der Instruktion.) Was ist der Kern des ganzen? Beide Prozessoren verwenden für alle Schiebeoperationen einen einzigen Basisbefehl! Zwei Bit drücken aus, ob es sich um ein Byte, ein 16-bit-Wort oder um ein 32-bit-Wort handelt, während ein weiteres Bit die Schiebeart (arithmetisch oder logisch) markiert. Der 68000 benutzt ein Bit zur Unterscheidung von linken und rechten Shifts, der Z8000 verwendet hierzu positive und negative Zahlen. Bei letzterem ist dadurch der Schiebebereich auf 32 Positionen festgelegt, beim 68000 sind dagegen 64 Positionen möglich. Andererseits benutzt der Z8000 ein Bit, um zwischen statischen und dynamischen Schiebeoperationen zu unterscheiden. Hier sind deshalb bei statischen Shifts 32 Schiebepositionen möglich, während die Anzahl der Positionen bei Motorolas 68000 auf 8 begrenzt ist. Wer will da entscheiden, welches der bessere Prozessor ist?

Zum Verwirrspiel trägt neben vielem anderem auch bei, daß einige Prozessoren einen größeren Befehlssatz, dafür aber weniger Adressierungsarten bieten. Ein Beispiel dafür gibt der Z8000: "Indirect Register with Increment or Decrement" fehlt in der Liste der Adressierungsarten, während andererseits der Befehlssatz die Befehle "Load", "Load and Decrement", "Load Decrement and Repeat" usw. enthält.

Zukunftsmusik

Ein Ende ist in der Entwicklungsgeschichte der Mikroprozessoren noch längst nicht abzusehen. So ließ Motorola unlängst verlauten: "Stringoperationen sind mit der heutigen Version des 68000 noch nicht möglich; die nächste Version wird nicht nur hierzu, sondern auch zu Fließkommaoperationen in der Lage sein." Texas Instruments, so hörte man in Fachkreisen, arbeitet geschäftig hinter den Kulissen – es fragt sich nur: woran?

Wir werden die Entwicklung mit großer Aufmerksamkeit verfolgen. Für spätere Elektor-Ausgaben sind Beiträge geplant, die sich mit den einzelnen Prozessorfamilien gesondert und ausführlich beschäftigen.

In diese Beiträge wird alles das einfließen, was bis dahin an Informationen verfügbar ist. Und schließlich enthält dieses Heft schon eine Anwendung des 16-bit-Mikroprozessors 8088, im Schachcomputer "Intelektor".

Literatur:

8086/8088:
- *The 8086 Family User's Manual (Intel 1979)*
- *16-bit Microprocessor Benchmark Report (Intel 1980)*

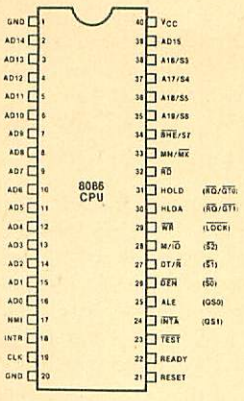
MC68000:
- *User information/preliminary descriptions, herausgegeben von: Motorola (MC68000), Rockwell (R68000), Thomson (EF68000), Hitachi (HD68000).*
- *MC68000 article reprints (Motorola 1980)*

NS 16000:
- *NS16000 family overview (National Semiconductor 1980)*
- *NS 16000 technical marketing brief (Nat. Sem. 1980)*

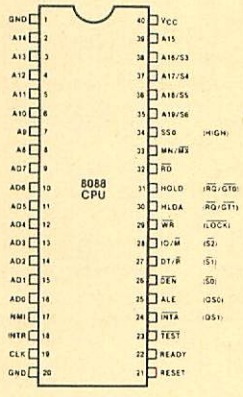
TMS 9900:
- *9900 family systems design (Texas Instruments 1978)*
- *data sheets/product information, herausgegeben von: Texas Instruments (TMS 900), AMI (S9900), ITT (ITT9900)*
- *16-bit µP Technical articles (AMI 1979)*

Z8000:
- *Z8001 and Z8002 programming manual (SGS/Ates 1980)*
- *Am Z8000 family data book (AMD 1980)*
- *data sheets/Application notes, herausgegeben von: Zilog (Z8000), AMD (Am Z8000)*
- *Programming the Z8000 (Sybex 1980)*

4a



MAXIMUM MODE PIN FUNCTIONS (e.g., LOCK) ARE SHOWN IN PARENTHESES



Common Signals

8086	Function	8088
AD15-AD0	Address/Data Bus	AD7-AD0
–	Address Bus	A15-A8
A19/S6-A16/S3	Address/Status	A19/S6-A16/S3
$\overline{BHE}$/S7	Bus High Enable/Status	–
MN/$\overline{MX}$	Minimum/Maximum Mode Control	MN/$\overline{MX}$
$\overline{RD}$	Read Control	$\overline{RD}$
$\overline{TEST}$	Wait On Test Control	$\overline{TEST}$
READY	Wait State Control	READY
RESET	System Reset	RESET
NMI	Non-Maskable Interrupt Request	NMI
INTR	Interrupt Request	INTR
CLK	System Clock	CLK
V_{CC}	+5 V	V_{CC}
GND	Ground	GND

Minimum Mode Signals (MN/MX = V_{CC})

HOLD	Hold Request	HOLD
HLDA	Hold Acknowledge	HLDA
$\overline{WR}$	Write Control	$\overline{WR}$
M/$\overline{IO}$	Memory I/O Control	IO/$\overline{M}$
DT/$\overline{R}$	Data Transmit/Receive	DT/$\overline{R}$
$\overline{DEN}$	Data Enable	$\overline{DEN}$
ALE	Address Latch Enable	ALE
$\overline{INTA}$	Interrupt Acknowledge	$\overline{INTA}$
–	S0 Status	SS0

Maximum Mode Signals (MN/MX = GND)

$\overline{RQ}$/GT1, 0	Request/Grant Bus Access Control	$\overline{RQ}$/GT1, 0
$\overline{LOCK}$	Bus Priority Lock Control	$\overline{LOCK}$
$\overline{S2}$-$\overline{S0}$	Bus Cycle Status	$\overline{S2}$-$\overline{S0}$
QS1, QS0	Instruction Queue Status	QS1, QS0

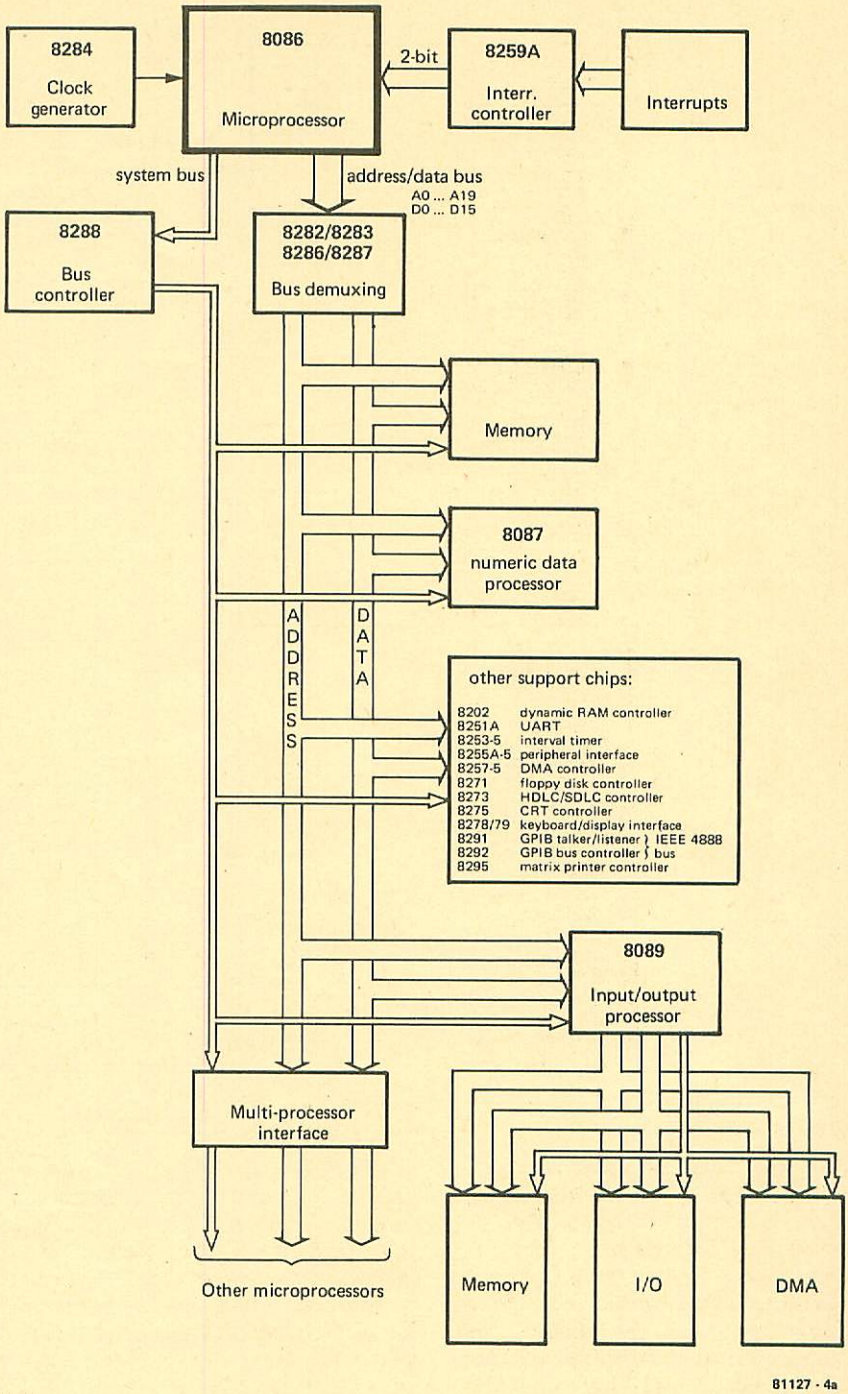
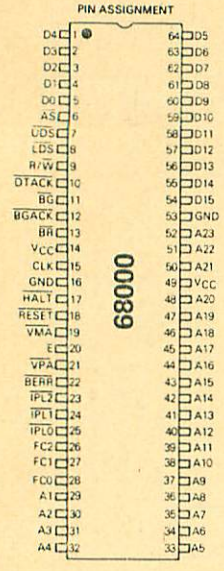
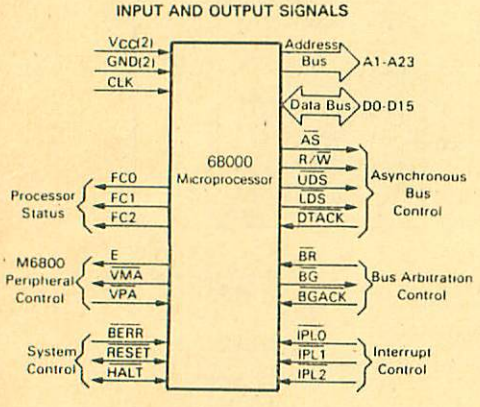
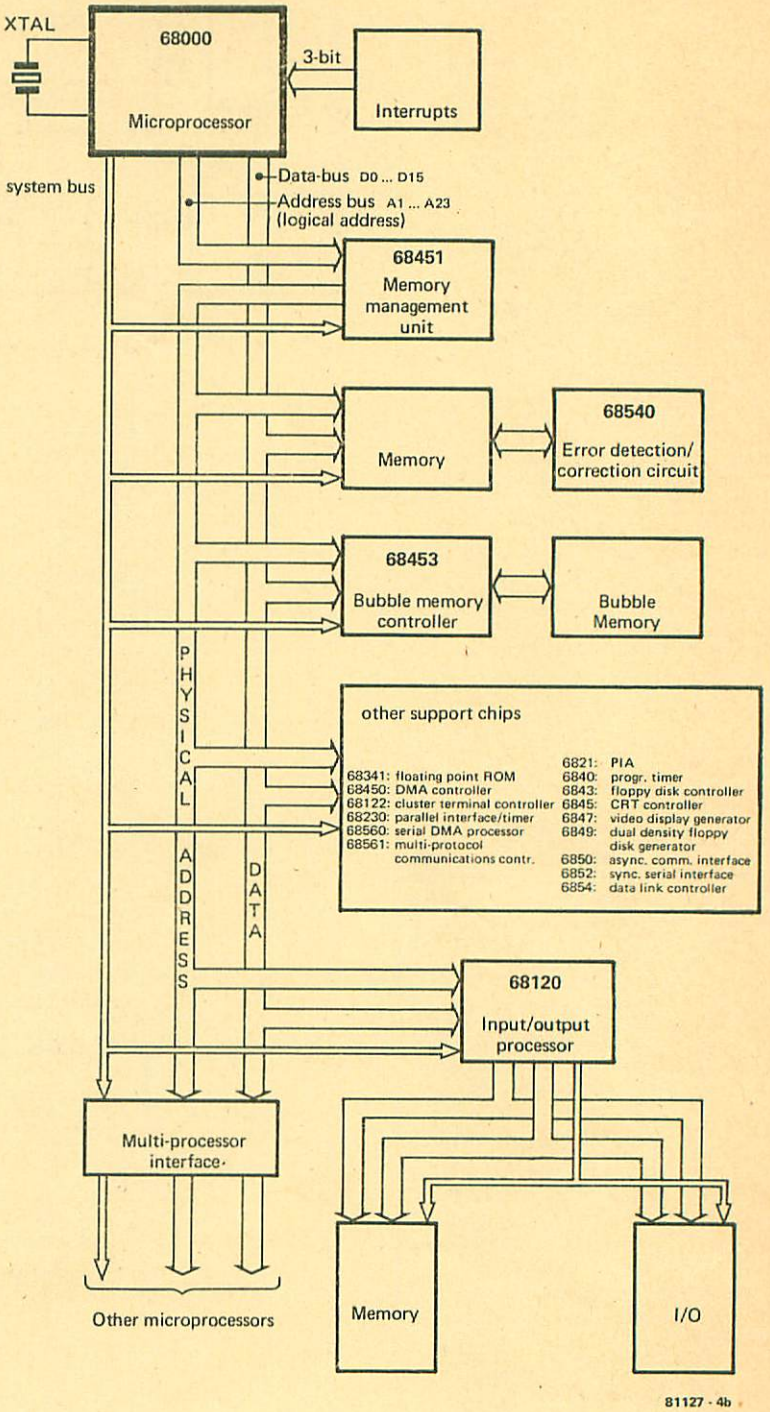


Bild 4a. Die 8086- und 8088-Mikroprozessoren gehören zur von Intel so genannten iAPX-86-Familie. Das System verwendet "Slave-Prozessoren": Mikroprozessor-artige Chips, die Funktionen erfüllen, die die CPU selbst nicht (oder nur teilweise) erfüllen kann. Solche Hilfs-Chips sind beispielsweise die "Numeric data processors" und der "Input/output processor". Im Minimal-Betrieb liefert der Prozessor selbst die Steuersignale für den "Control bus", während im Maximal-Betrieb ein zusätzlicher "Bus controller" diese Aufgabe übernimmt.

4b



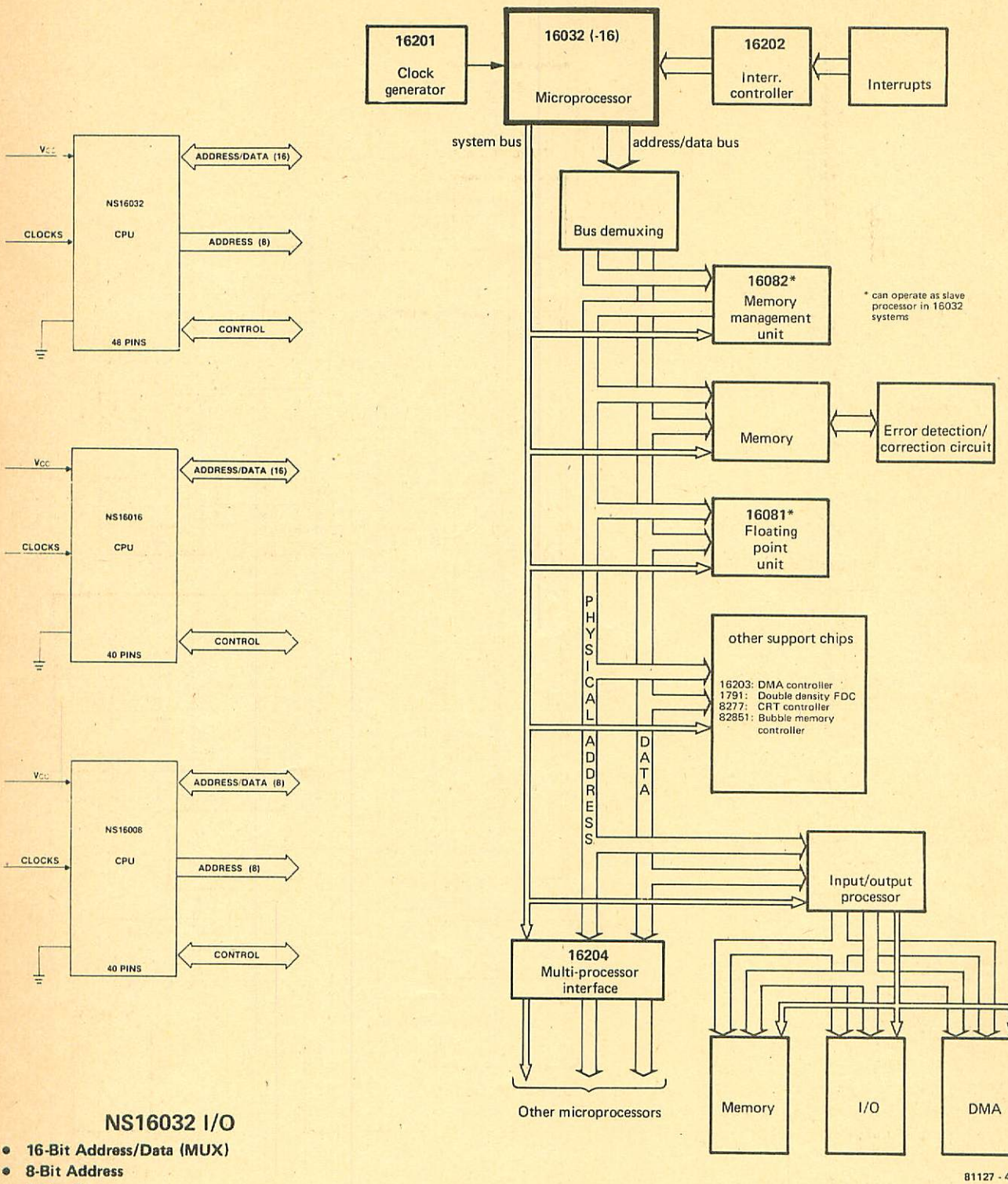




Pin	Function
A1-A23	Address Bus
D0-D15	Data Bus
$\overline{AS}$	Address Strobe
R/$\overline{W}$	Read/Write
$\overline{UDS}$, $\overline{LDS}$	Upper and Lower Data Strobes
$\overline{DTACK}$	Data Transfer Acknowledge
$\overline{BR}$	Bus Request
$\overline{BG}$	Bus Grant
$\overline{BGACK}$	Bus Grant Acknowledge
$\overline{IPL0 \ldots 2}$	Interrupt Priority Level
$\overline{BERR}$	Bus Error
$\overline{RESET}$	Reset
$\overline{HALT}$	Halt
E	Enable
$\overline{VMA}$	Valid Memory Address
$\overline{VPA}$	Valid Peripheral Address
FC0,FC1,FC2	Function Code Output
CLK	Clock
VCC	Power Input
GND	Ground

Bild 4b. Auch der 68000-Prozessor ist Mitglied einer großen Familie. Dazu gehören sowohl "intelligente", Mikroprozessor-artige Chips für spezielle Aufgaben, wie z.B. MEMORY MANAGEMENT und INPUT/OUTPUT CONTROL, als auch die schon erwähnten Hilfs-ICs. Motorola hat übrigens bewußt darauf geachtet, daß die eventuell vorhandene Peripherie eines 6800-Systems auch beim größeren Bruder zu gebrauchen ist. Außerdem fällt die vollständige Trennung von Adreß- und Daten-Bus auf.

4c

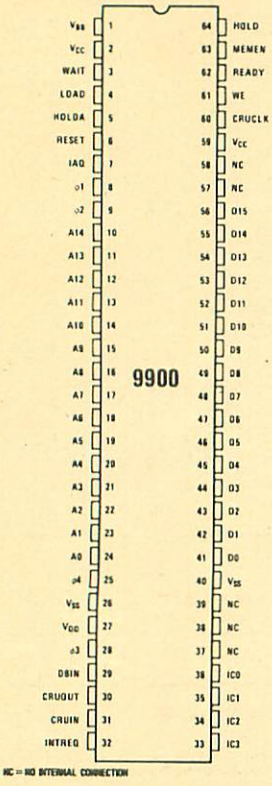


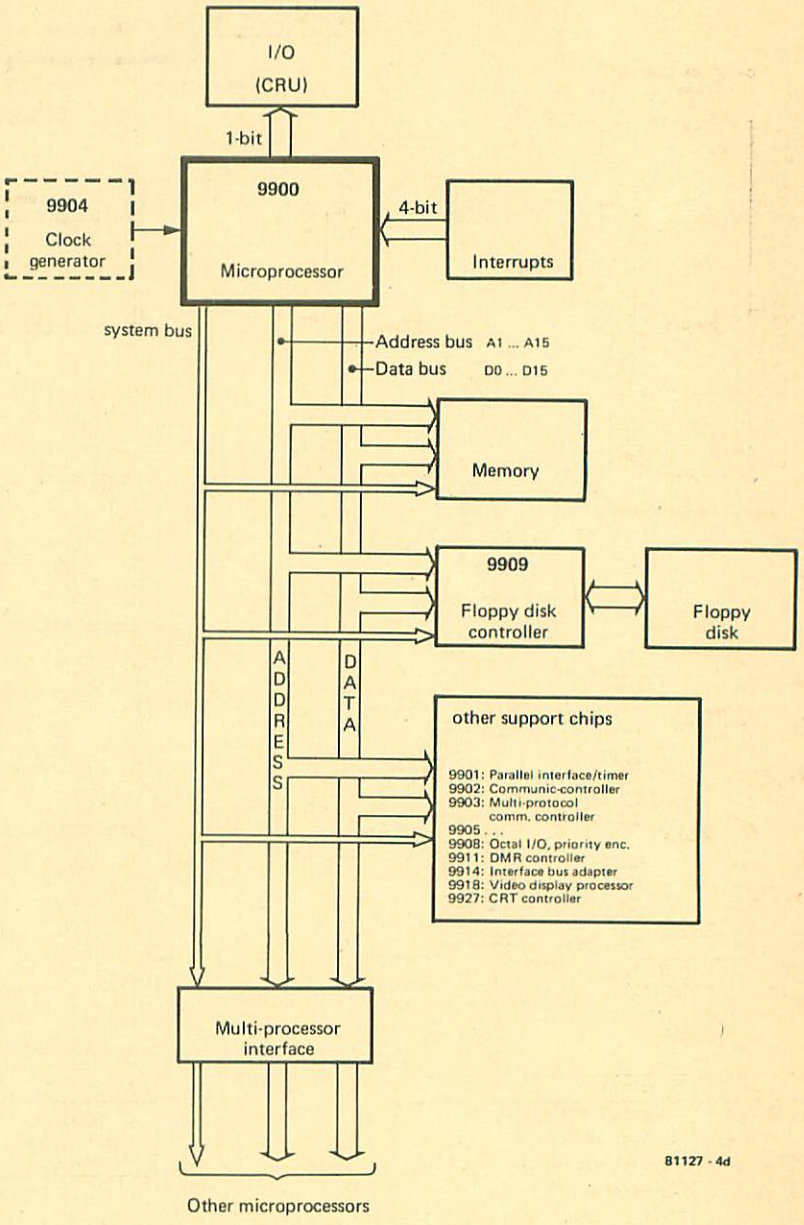
NS16032 I/O

- 16-Bit Address/Data (MUX)
- 8-Bit Address
- 4 Bits Status
- $\overline{ADS}$
- $\overline{DDIN}$
- $\overline{HBE}$
- RDY
- $\overline{HOLD}$, $\overline{HLDA}$, $\overline{ILO}$
- $\overline{NMI}$, $\overline{INT}$
- $\overline{ABT}$
- $\overline{FLT}$, U/$\overline{S}$, $\overline{PFS}$
- PH1, PH2
- $\overline{RST}$
- $\overline{SPC}$
- 2 GNDS And V_{CC}

Bild 4c. Der 16000-Prozessor ist so neu, daß exakte Daten über das Gehäuse und die Pin-Belegung noch gar nicht vorliegen. Trotzdem kann man aus der "Preliminary information" einige Angaben ablesen. Auch hier handelt es sich um eine ganze Familie, in der die Hilfs-Chips eine wichtige Rolle spielen. National Semiconductor geht sogar so weit, daß auch die Register der FLOATING POINT UNIT und MEMORY MANAGEMENT "mitzählen". Das sollte man in einem Vergleich mit anderen Prozessoren berücksichtigen. In diesem Bild kommt eine Stärke des 16000-Systems gar nicht zum Ausdruck: die Möglichkeit, sogenannte Software Modules (also Programmteile in einem ROM) in die Speicher einzubauen. Befehlssatz und Adressierung sind schon darauf abgestimmt. Und National Semiconductor hat anscheinend auch die Absicht, eine solche "Software-Bibliothek" einzurichten. Das wäre schön!

4d

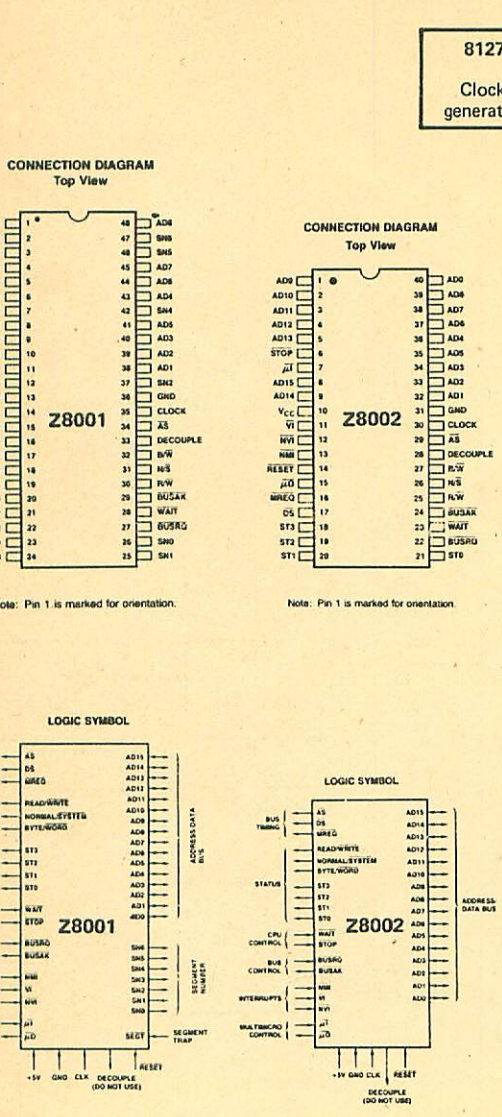




Pin	Function
A0-A14	Address bus
D0-D15	Data bus
ø1-ø4	clock
V_{BB}	–5 V
V_{CC}	+ 5 V
V_{DD}	+12 V
V_{SS}	GND
$\overline{INTREQ}$	Interrupt request
IC0-IC3	Interrupt codes
CRUIN	CRU data in
CRUOUT	CRU data out
CRUCLK	CRU clock
DBIN	Data bus in
$\overline{MEMEN}$	Memory enable
$\overline{WE}$	Write enable
READY	Memory ready
$\overline{HOLD}$	Hold request
HOLDA	Hold acknowledge
WAIT	Wait indication
$\overline{RESET}$	Reset
IAQ	Instruction acquisition
$\overline{LOAD}$	Load WP and PC

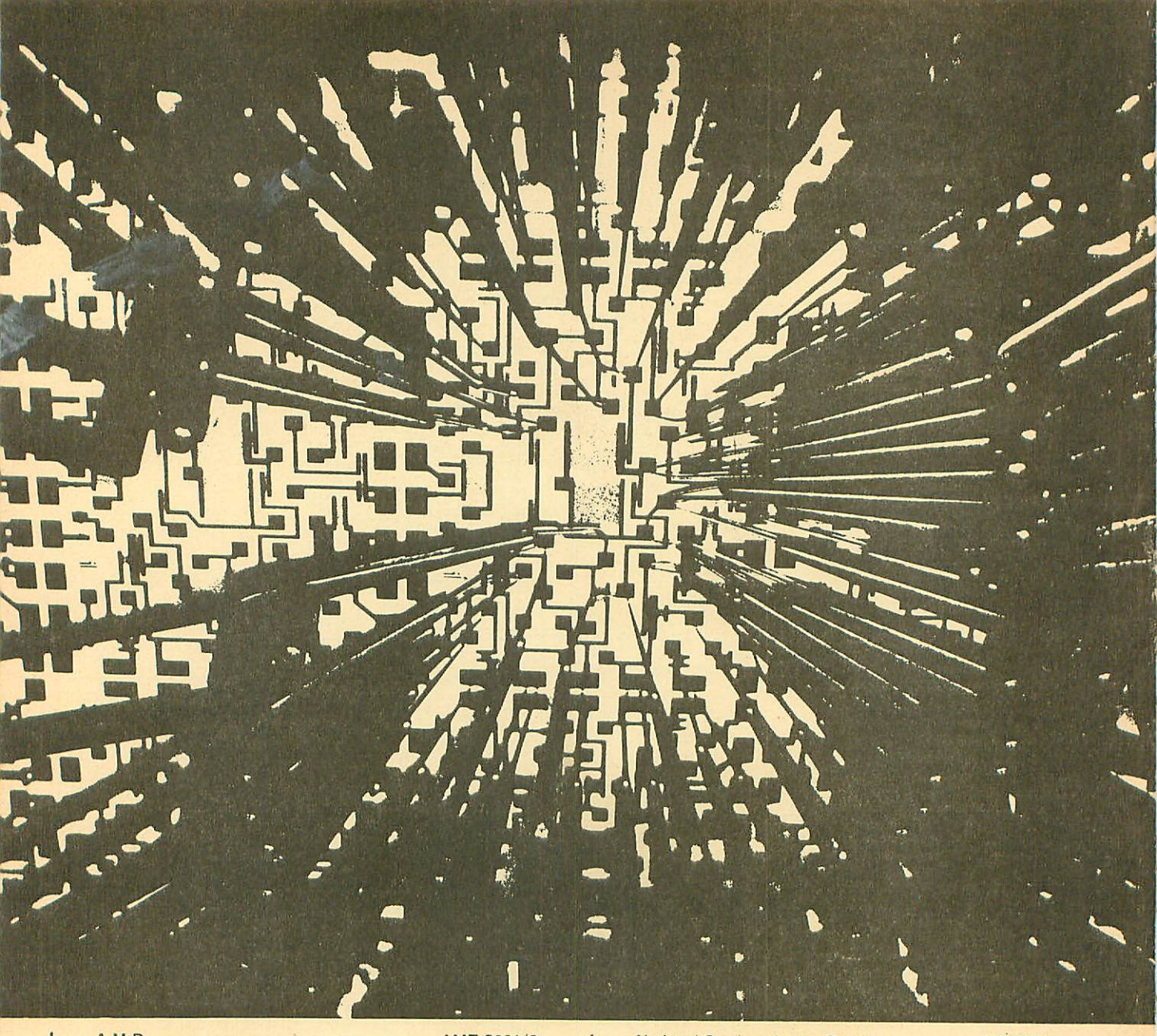
Bild 4d. Diese Übersicht gibt eigentlich kein vollständiges Bild der 9900-Familie. Texas Instruments hat nämlich nicht nur viele Hilfs-Chips sondern auch eine ganze Reihe Mikroprozessor-ähnlicher Familien-Mitglieder entwickelt. Mit oder ohne RAM und/oder ROM auf dem Chip, mit verschiedenen Arten der Informations-Ein- und -Ausgabe für die vielfältigsten Anwendungen. TI sagt selbst dazu: "Die 9900-Familie ist eine kompatible Gruppe von Mikroprozessoren, Mikrocomputern, Mikrocomputer-Modulen und Minicomputern." Ein weites Feld also. In nächster Zeit wird außerdem Nachwuchs in der 9900-Familie erwartet.

4e



Pin	Function
AD_0-AD_{15}	Address/Data
$\overline{AS}$	Address Strobe
$\overline{BUSAK}$	Bus Acknowledge
$\overline{BUSRQ}$	Bus Request
$\overline{DS}$	Data Strobe
$\overline{MREQ}$	Memory Request
$\overline{\mu I}$	Multi-Micro In
$\overline{\mu O}$	Multi-Micro Out
$\overline{NMI}$	Non-maskable Interrupt
$\overline{NVI}$	Non-vectored Interrupt
CLK	System Clock
$\overline{RESET}$	Reset
$R/\overline{W}$	Read/Write
SN_0-SN_6	Segment Number
$\overline{SEGT}$	Segmentation Trap
ST_0-ST_3	Status
$\overline{STOP}$	Stop
$\overline{VI}$	Vector Interrupt
$\overline{WAIT}$	Wait
$B/\overline{W}$	Byte/Word reference
$N/\overline{S}$	Normal/System Mode
Decouple	Output from on-chip negative substrate-bias generator. Presently not connected.

Bild 4e. Auch der Z8001-Prozessor hat einige Brüder, Schwestern, Nichten und Neffen. Die 8002-, 8003- und 8004-Prozessoren sind nahe Verwandte des 8001-Prozessors. Außerdem gibt es wieder eine MEMORY MANAGEMENT UNIT, einen INPUT/OUTPUT PROCESSOR usw. Zilog hat übrigens großen Wert auf die Möglichkeit des Multi-Prozessor-Betriebs gelegt – also z.B. die Verwendung mehrerer 8001 in einem System. Neben den dafür notwendigen Befehlen enthält der Prozessor sogar "Multi/micro input/output"-Pins.



A.M.D. Advanced Micro Devices GmbH Rosenheimer Str. 139 8000 München 80	AMZ 8001/2
A.M.I. Microsystems GmbH Rosenheimer Str. 30 Suite 237 8000 München 80	S 9900
Hitachi Electronic Components Europe GmbH Königsallee 6 4000 Düsseldorf	HD 68000
Intel Alfred Neye Enatechnik GmbH Schillerstr. 14 2085 Quickborn	8086/88
Deutsche ITT Industries GmbH Hans-Bunte-Str. 19 7800 Freiburg i. Br.	ITT 9900
Mitsubishi Electric Europe GmbH Karl-Rudolf-Str. 178 4000 Düsseldorf	M5L 8086
Mostek Alfred Neye Enatechnik GmbH Schillerstr. 14 2085 Quickborn	MK 8086
Motorola Geschäftsbereich Halbleiter Münchner Str. 18 8043 Unterföhring	MC 68000
National Semiconductor GmbH Industriestr. 10 8080 Fürstenfeldbruck	NS 16000
Rockwell International GmbH Microelectronic Devices Fraunhoferstr. 11a 8033 München-Martinsried	R 68000
SGS-Ates Halbleiter-Bauelemente GmbH Postfach 1269 8090 Wasserburg/Inn	Z8001/2
Siemens AG Postfach 202109 8000 München 2	SAD 8086
Texas Instruments Deutschland GmbH Haggertystr. 1 8050 Freising	TMS 9900
Thomson CSF GmbH Fallstr. 42 8000 München 70	EF 6800
Zilog Kontron Elektronik GmbH Oskar-von-Miller-Str. 1 8051 Eching	Z 8001/2/3/4