
Kapitel 10 CISC-Rechner

10.1. Allgemeine Charakterisierung

Speicher als absolut knappe Ressource, sehr teure Hardware für umfangreiche Schaltnetze und verbreitete Programmierung von Hand in Assembler prägten die Entwicklung der Rechner bis Ende der Achtzigerjahre.

Sie führten zu Rechnern mit komplexen Befehlssätzen (complex instruction set computer, CISC).

Typische Merkmale waren

- Befehle unterschiedlicher Länge von 1 - 6 Byte. Da der Zugriff auf den Speicher byteweise erfolgte und Speicher knapp war, war es sinnvoll, bei Programmen Speicherverchnitt zu vermeiden. Der Kaufpreis war ein komplexer Befehlszugriff: Je nach Befehlsbyte mußten weitere Bytes nachgeladen werden, um den Befehl zu komplettieren.
- Komplexer Befehlssatz
Um die Arbeit des Assemblerprogrammierers zu erleichtern, wurden viele Maschinenbefehle eingeführt. Sie erlaubten das Schreiben von kompaktem Code (Speicherplatz sparen!). Der Kaufpreis waren Schaltwerke, um die Befehle abzuarbeiten, die viele Taktzyklen brauchten. Sie kamen aber mit einfachen rechnenden Schaltnetzen aus.
- Direkte Operationen im Speicher
Da die Zahl der Register eng begrenzt war (teure Hardware!) (8 - 16 Register), mußten viele Operationen in den Speicher ausgelagert werden. Maschinenbefehle unterstützten das:
Speicher <- Speicherplatz 1 op Speicherplatz 2 waren geläufige Maschinenbefehle.
- Komplexe Adressierung des Speichers
Bedingt durch kurze Wortlängen in der CPU (Hardware!) (z. B. 16-Bit Adressregister) mußte Aufwand getrieben werden, wenn mehr als 64 k Byte adressiert werden sollten.

10.2. Beispiele

Es werden 4 Beispiele von Maschinen gezeigt;

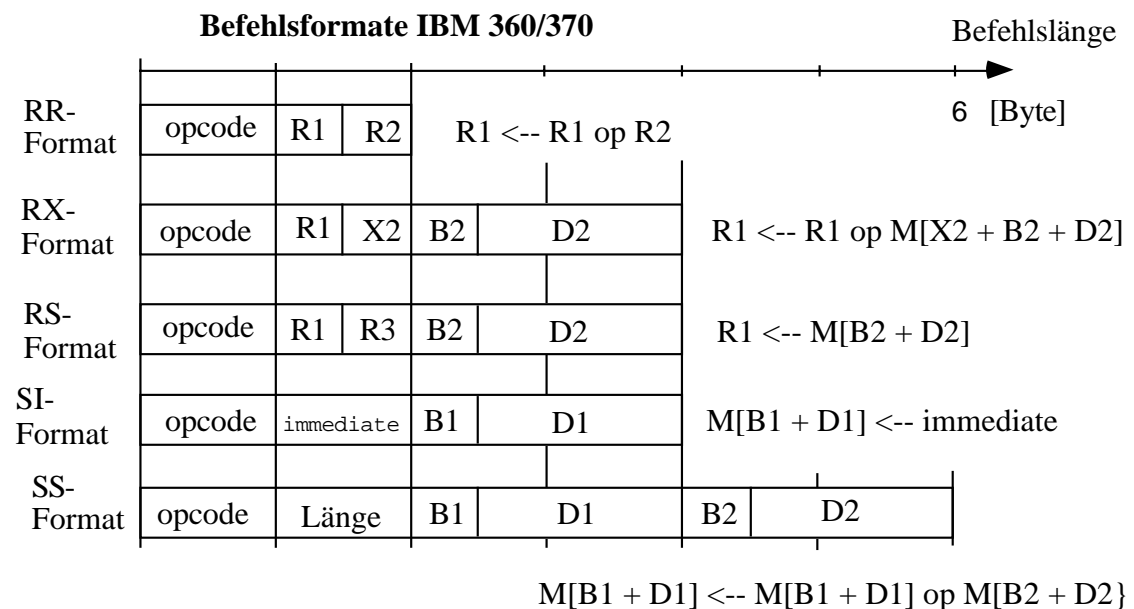
- IBM 360/370
- Intel 8086
- Intel 80386
- Motorola 68030 .

10.2.1. IBM 360/370

Es ist dies die Maschine, die bis weit in die Achtzigerjahre hinein die Szene der Großrechner prägte.

Charakteristika:

- 16 Register R0 - R15 à 32 Bit mit $R0 \equiv 0$. Paare von Registern sind als doppelt genaue GK-Register verwendbar.
- Registerinhalte können Daten und Adressen sein (general purpose register).
- Adressraum anfangs begrenzt auf 24 Bit (16 M Byte waren in den Siebzigerjahren ein **sehr** großer Speicher)
- Programmstatuswort $PSW = PC + \text{flags} + \text{Bedingungscode}$
- Es gab 5 verschiedene Befehlsformate



Die Adressierung auf Maschinenebene erfolgte einheitlich in der Form:

effektive Adresse = < Basisregister > + < Indexregister > + 12-Bit Displacement .

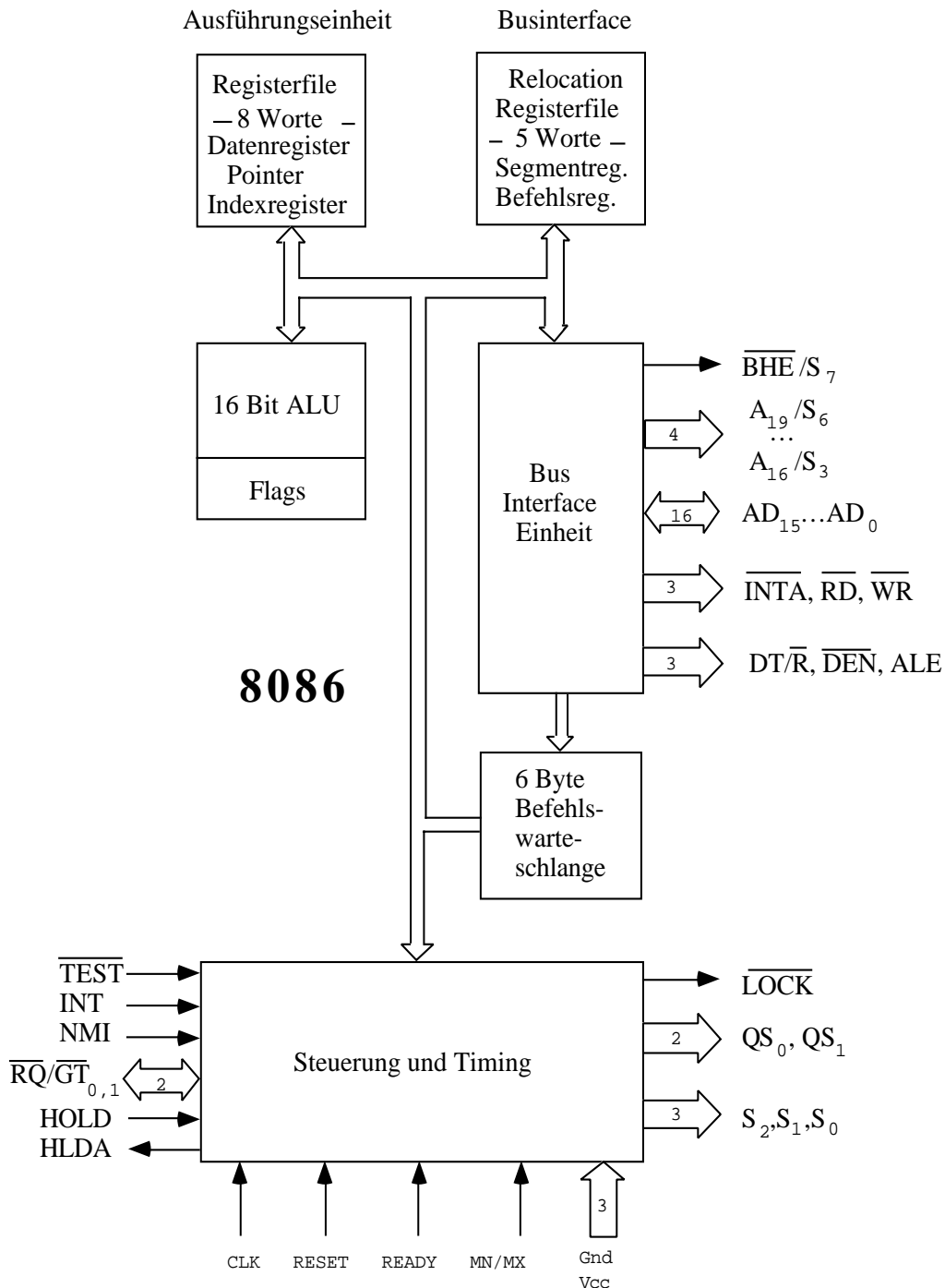
Die Angabe des Registers "0" bewirkte, daß Null für den Registerinhalt geschrieben wurde.

- Die Anzahl der Befehle lag bei 106:

- 28 Befehle im RR-Format (Festkomma-, logische - und GK-Befehle)
- 7 Befehle zum Verzweigen und Setzen des Status im RR-Format
- 19 Befehle im RX-Format (Festkomma-, logische - und GK-Befehle)
- 9 Verzweige- Lade- und Speichere-Befehle im RX-Format
- 23 Befehle im RS- und SI-Format
- 20 Befehle im SS-Format.

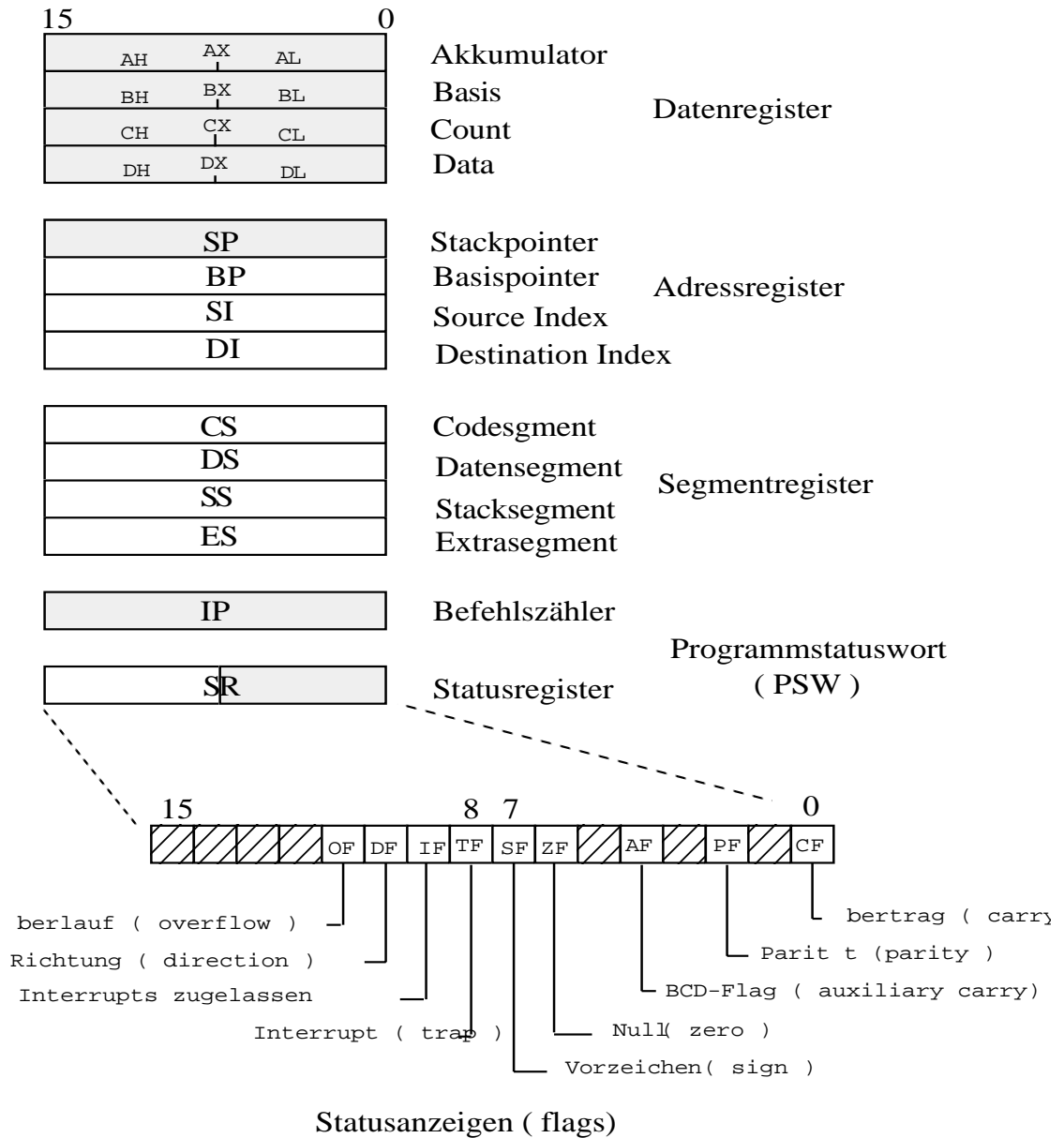
10.2.2. Intel 8086

Es war dies ein 16-Bit-Rechner, Vorläufer der 80386/80486 und Pentium-Prozessoren.



Er hatte seinerseits einen Vorgänger, der 8080, zu dem er programmkompatibel sein mußte. Dieser Zwang zur Kompatibilität spiegelte sich wieder im Programmiermodell der Maschine.

Registerstruktur des 8086



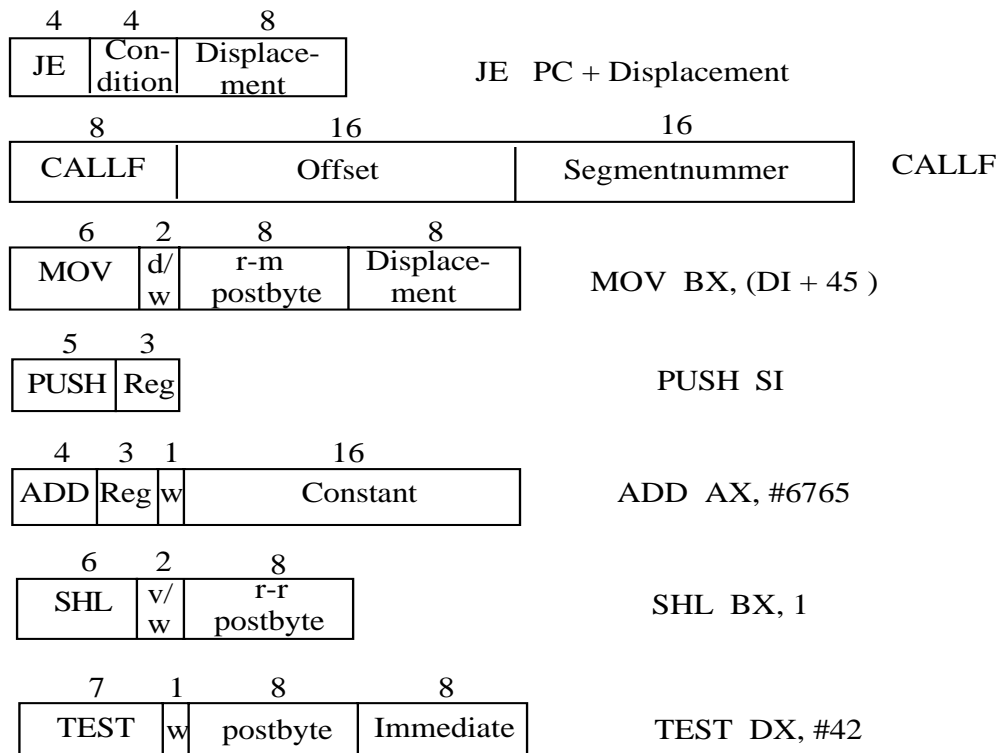
Die Register des 8080 sind die 8-Bit-Register in A, B, C, D, der 16-Bit-Stackpointer, der 16-Bit PC und die untersten 8 Bit des Statusregisters.

Damit konnte der 8086 die Befehle des 8080 ebenfalls abarbeiten. Sie waren eine Unter-
menge des Befehlssatzes.

Die Maschine kennt 2-Operandenbefehle:

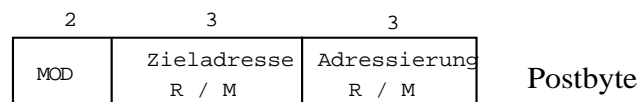
<u>Quell/Zieloperand</u>	<u>zweiter Operand</u>	<u>Beispiel</u>
Register	Register	AND BX, DX
Register	Immediate	ADD AX, # 6765
Register	Speicher	SUB BX, (DI + 45)
Speicher	Register	MOVW (DI + 17), AX
Speicher	Immediate	MOV (DI + 33), # 31415

Der 8086 verfügte über mehrere Befehlsformate.



w : Byte- oder Wortoperation d : Richtung der Übertragung
v : variable Länge

Das Postbyte charakterisiert die Adressierungsart im Befehl.



Adressierungsarten des 8086 oder 80386 im 16-Bit-Modus

MOD = 11			Berechnung der effektiven Adresse			
R / M	W = 0	W = 1	R / M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	AH	SP	100	(SI)	(SI) + D8	(SI) + D16
101	CH	BP	101	(DI)	(DI) + D8	(DI) + D16
110	DH	SI	110	direkte Adresse	(BP) + D8	(BP) + D16
111	BH	DI	111	(BX)	(BX) + D8	(BX) + D16

Die Speicher-Adressierungsarten sind

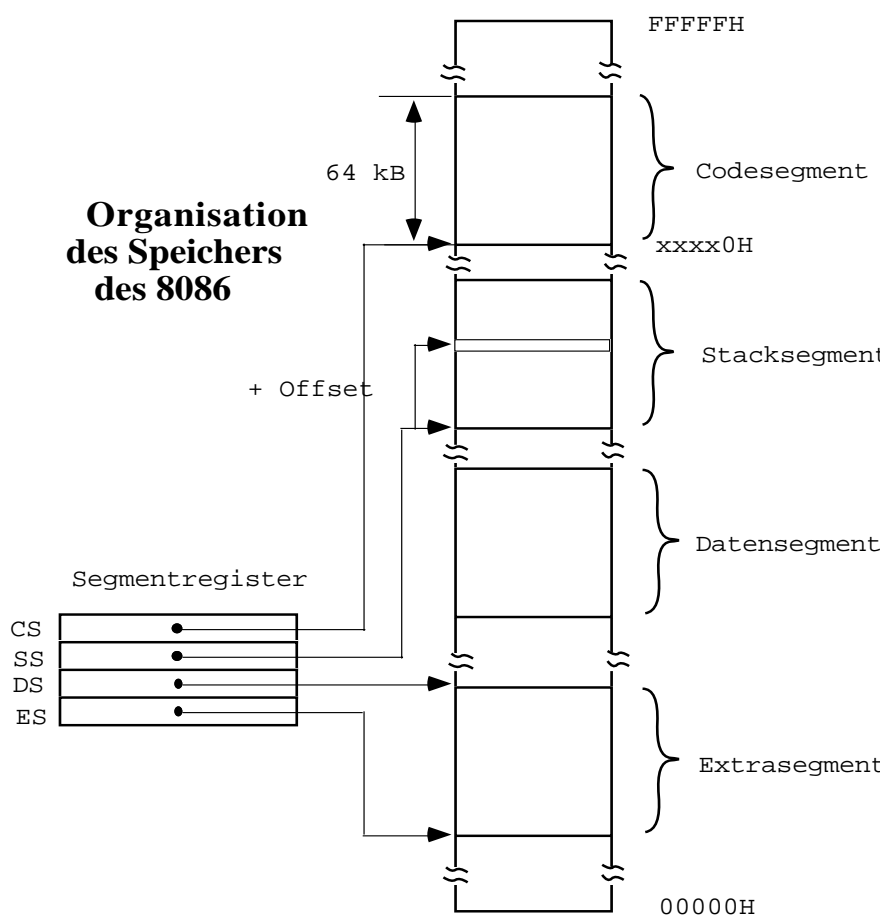
- Absolute Adresse
- Register indirekt
- Basismodus mit 8- oder 16-Bit Displacement
- Indiziert (based index), Summe von zwei Registern
- Indiziert mit Indexregister + 8- oder 16-Bit Displacement.

Insgesamt kennt der 8086

- 40 arithmetische und logische Befehle
- 43 Steuerbefehle
- 19 Datenübertragungsbefehle
- 25 Zeichenkettenbefehle

127 Summe der Befehle

Die Adressierung des Speichers geschieht über die Segmentregister, zu denen die logische 16-Bit-Adresse im Befehl - der Offset - implizit hinzuaddiert wird.



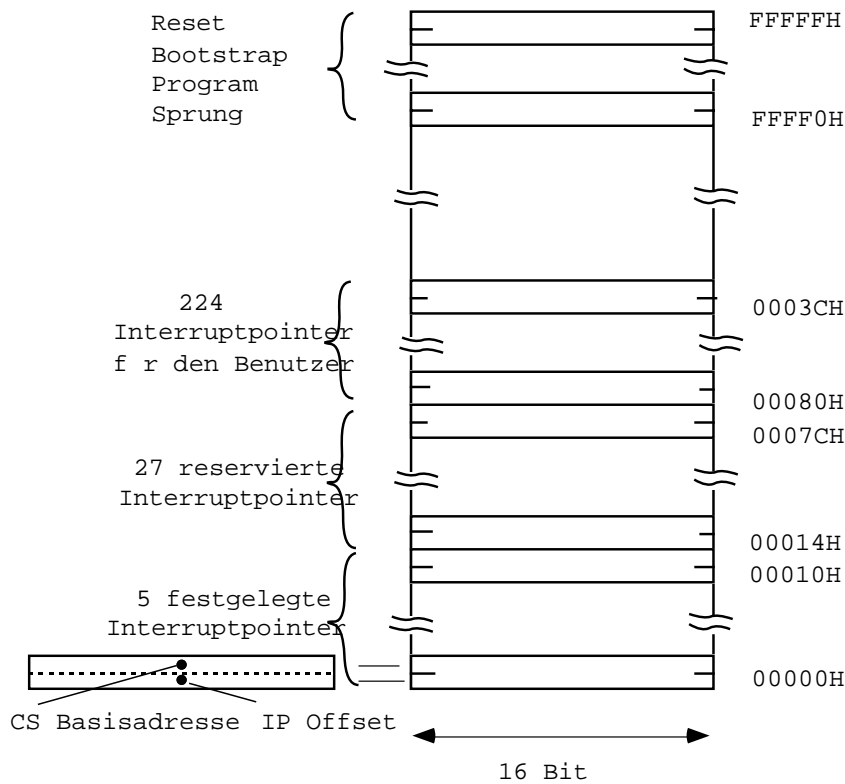
Der Adressraum sind 20 Bit, aufgeteilt in 64 k B Segmente adressiert mit 16 Bit-Offset.

Der Inhalt der Segmentregister wird um 4 Bit links verschoben und dann zum Offset addiert.

Die 14 Register sind in 4 Gruppen aufgeteilt:

- Datenregister (AX, BX, CX, DX)
- Adressregister (Stackpointer SP, Basisregister BP, 2 Indexregister SI und DI)
- Segmentregister (Code CS, Stack SS, Daten DS und Extra ES)
- Steuerregister (Befehlszähler BP und Statusregister FLAGS) .

Im Speicher selber sind zwei Bereiche reserviert, in denen die Trapvektoradressen bei Interrupts stehen und dazu ein Bootstrapbereich:



reservierte Speicherplätze beim 8086

Die reservierten Plätze am unteren Ende des Speichers sind für die Interrupts bestimmt. Es sind jeweils zwei Worte zu 16 Bit reserviert: der Inhalt des Code-Segmentregisters des zugehörigen Programms und der Offset in das Programm, bei einem Interrupt geladen in das CS-Segmentregister und den PC. Vom Hersteller sind vorgesehen Interrupts für

- Division durch Null,
- Einzelschrittmodus,
- NMI (non maskable interrupt),
- interner Interrupt
- Überlauf OV

Die reservierten Plätze am oberen Ende des Speichers sind für die Bootstrap-Routine bei einem Reset vorgesehen.

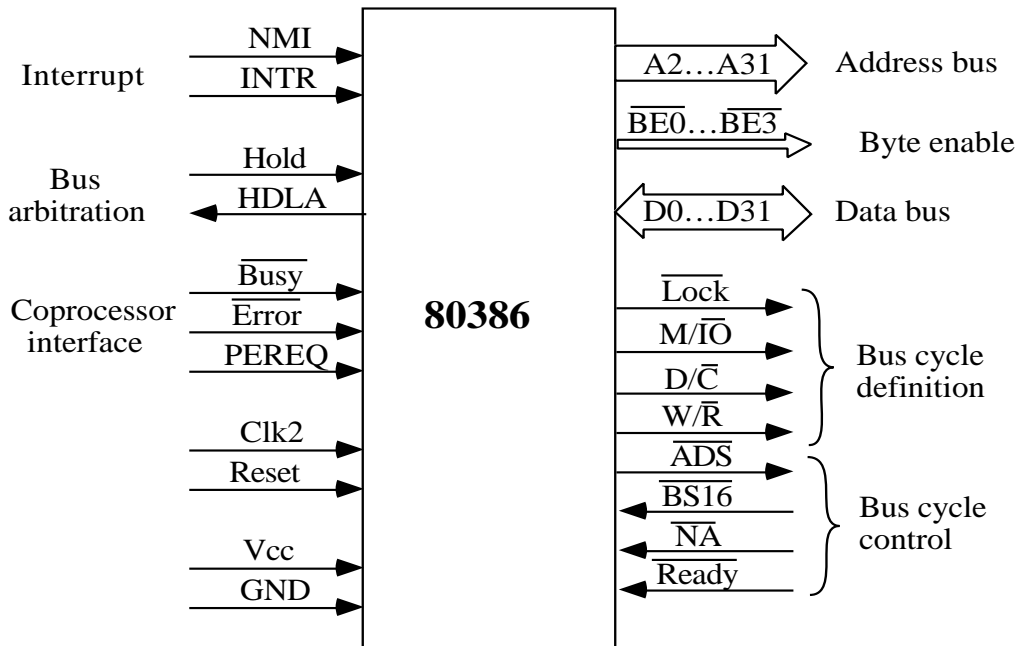
10.2.3. Intel 80386

Als Beispiel einer 32-Bit-Cisc-Maschine soll hier der Prozessor Intel 80386 vorgestellt werden.

10.2.3.1. Aufbau

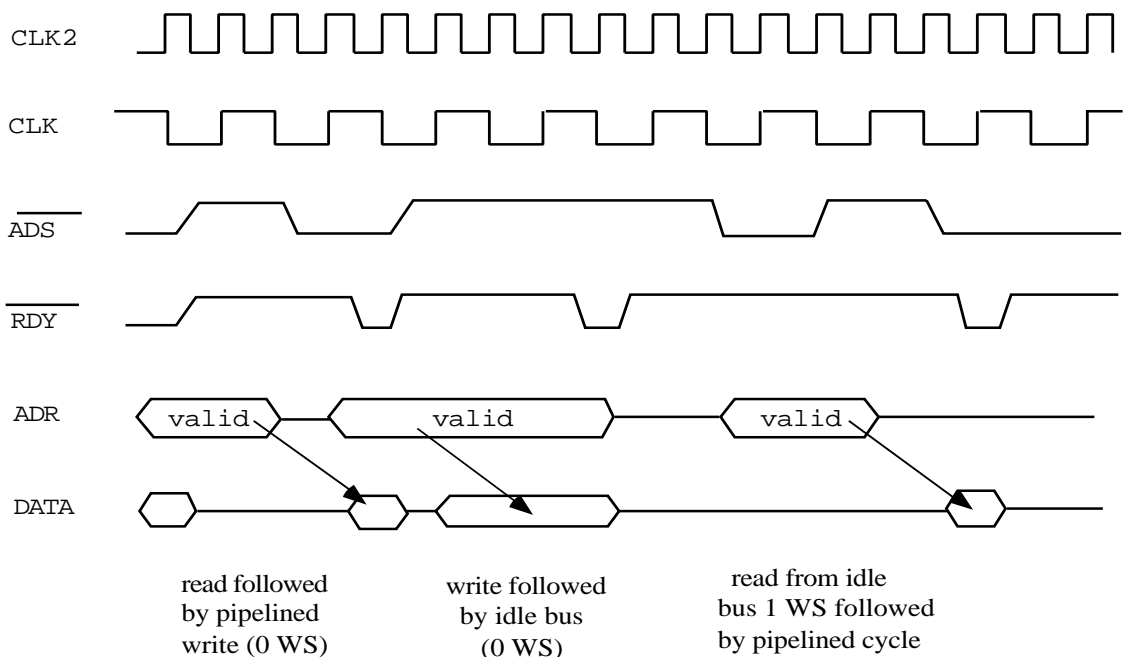
Nach außen hin ein Mikroprozessor, der an einem Bus mit 32 Daten und 32 Adressleitungen hängt. Der Bus wird mit 4 Steuerleitungen gesteuert und der Buszyklus mit weiteren 4 Leitungen. Er kann einen arithmetischen Coprozessor 80387 bedienen.

Die Anschlüsse des 80386 und das Timing auf dem Bus zeigen die nachfolgenden Bilder.



Das Timing wird bestimmt durch die externe Clock Clk2, die intern durch zwei geteilt wird. Mit ADS (address strobe) 1 --> 0 wird die Gültigkeit der Adresse nach außen signalisiert und mit RDY (ready) 1 --> 0 zeigt das externe Gerät die Datenübergabe an.

Bustiming des 80386

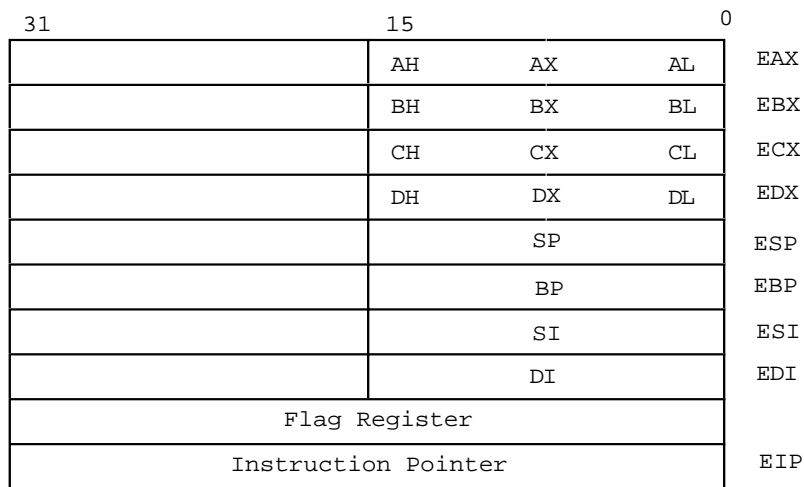


Das Nachfolgemodell 80486 enthält zusätzlich eine MMU mit virtuellem Paging und einen 8 kByte Cache mit Cache-Controller und eine Gleitkommarecheneinheit (floating point unit, FPU).

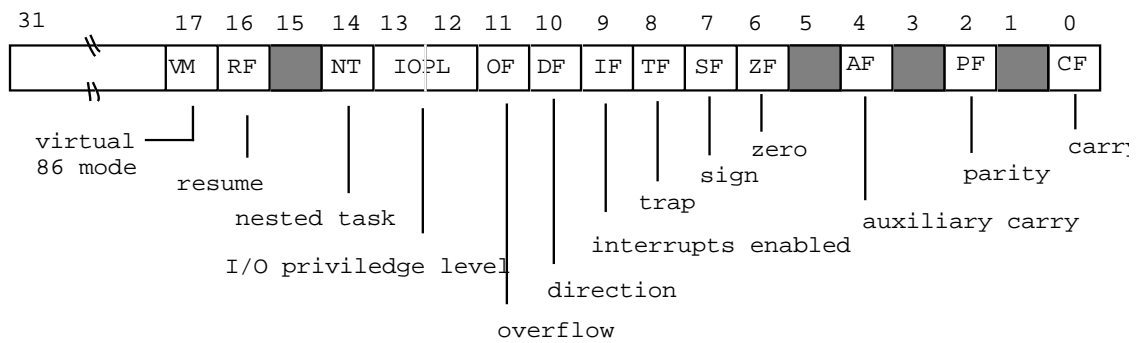
Das Programmiermodell entspricht - bis auf die Registerlänge von 32 Bit - dem Aufbau des 8086, zu dem der 80386/7 80486 programmkompatibel ist.

Programmiermodell des 80386

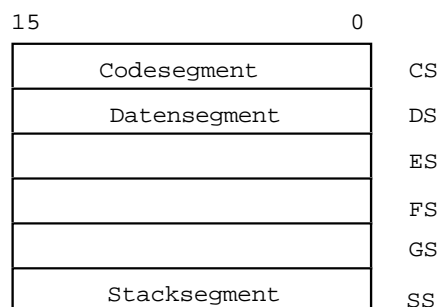
Registerblock



Flag Register



Segmentregister

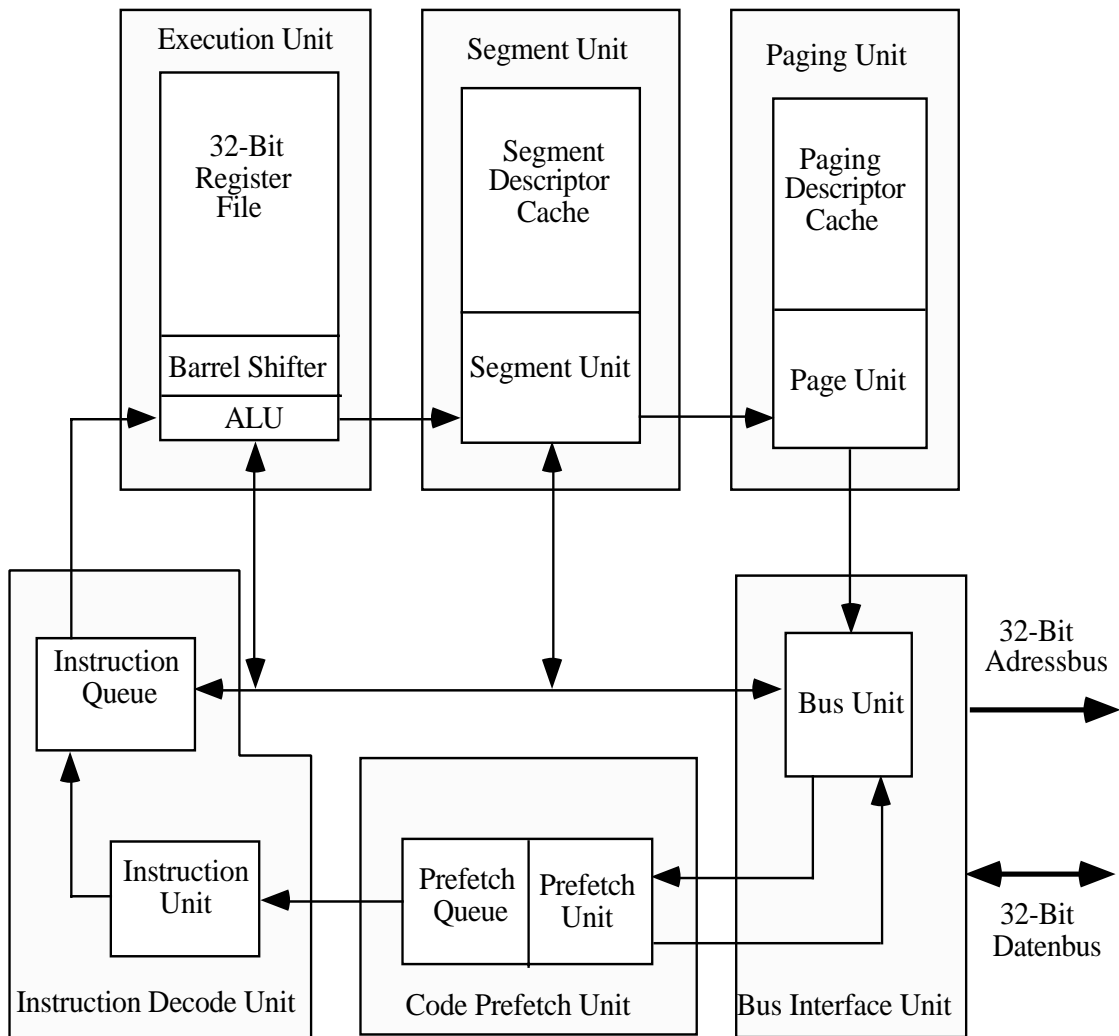


10.2.3.2. Speicheradressierung

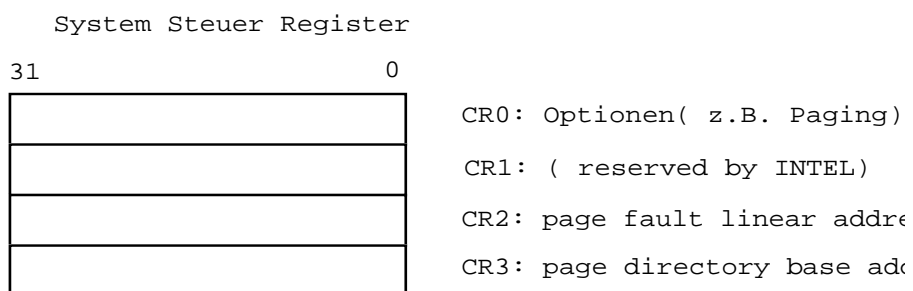
Der große Unterschied ist die wesentlich komplexere Adressierung des Speichers.

Sie ist schon sichtbar im Blockdiagramm der Maschine, wo die Adressumrechnung über zwei Cache-Speicher vorgenommen wird, die die Deskriptoren für Segmente und Seiten vorhalten.

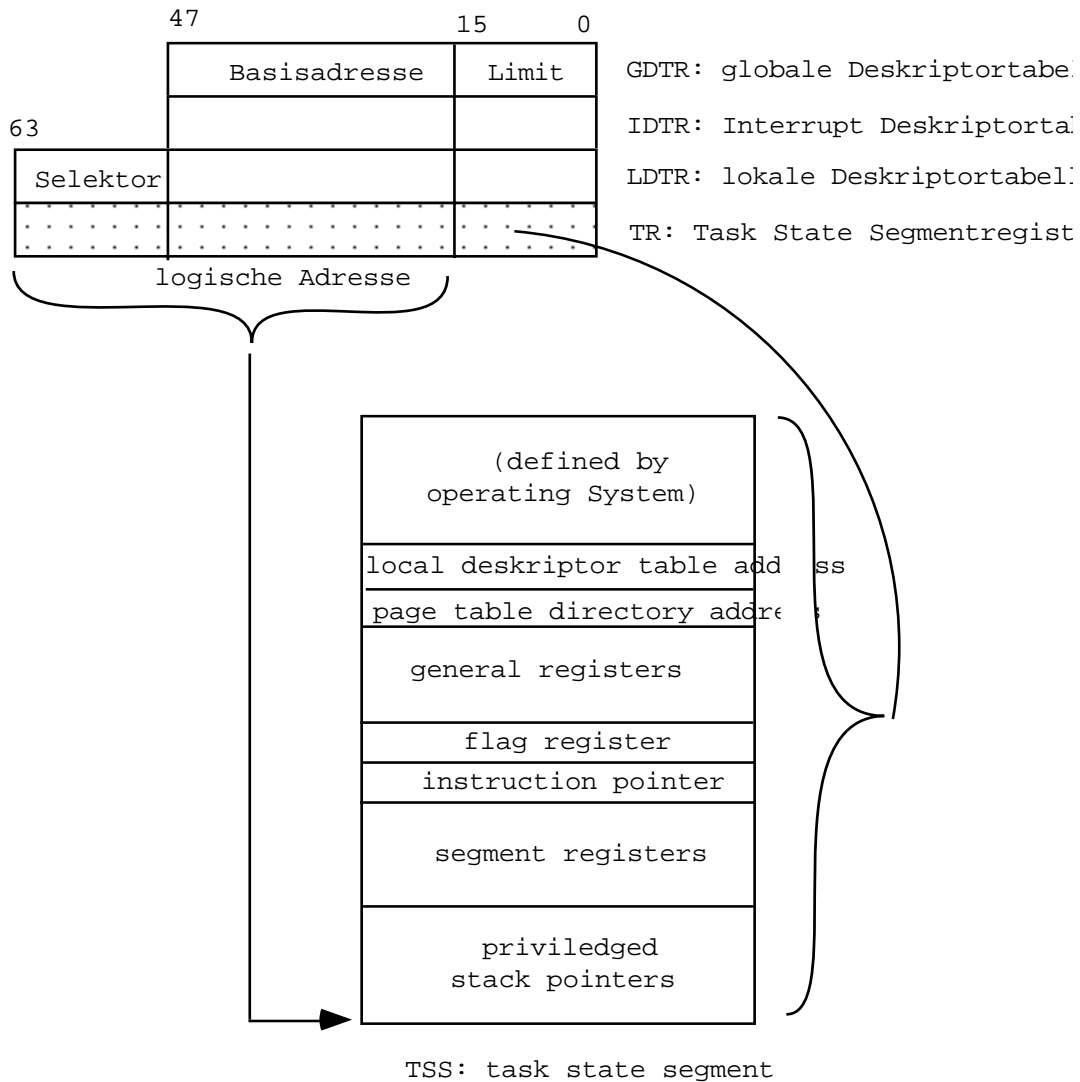
Blockdiagramm des Intel 80386 Mikroprozessors



Intern, vom normalen Programm her nicht zugänglich, nur vom Betriebssystem, gibt es zusätzlich vier System-Steuerregister, in denen die Basisadressen stehen, insbesondere in CR3 die des Directories der Seitentabellen.

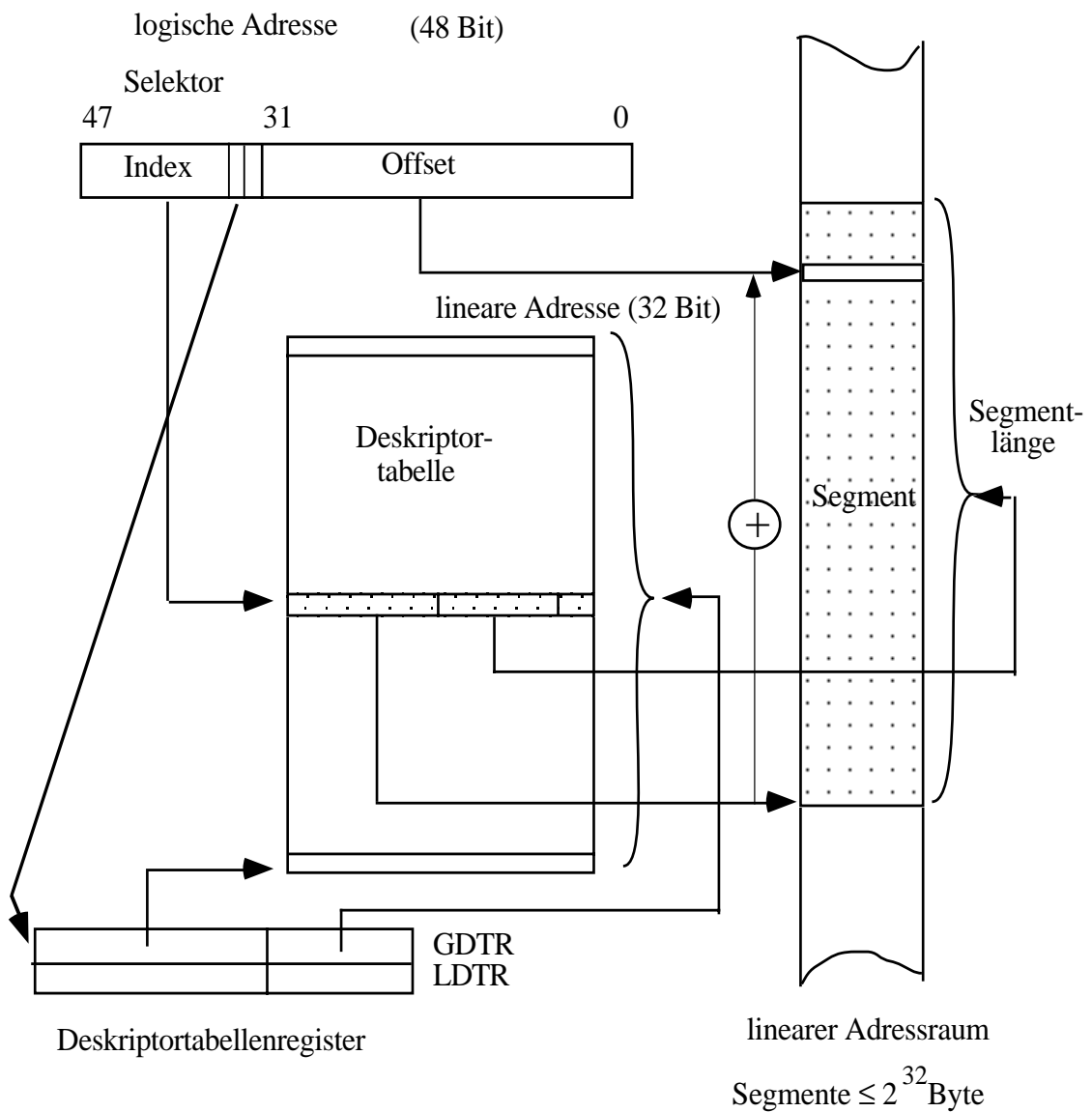


Bei der Adressierung sind eine Segmentaufteilung und eine Seitrenadressierung überlagert. Die Segmentadressierung ist ein Relikt aus der 8086-Programmierung, die Seiteneinteilung bei großen Speichern sinnvoll. Die Segmenteinteilung des Speichers wird in System-Adress-Registern verwaltet, die Basisadressen auf Segmentdeskriptortabellen enthalten.



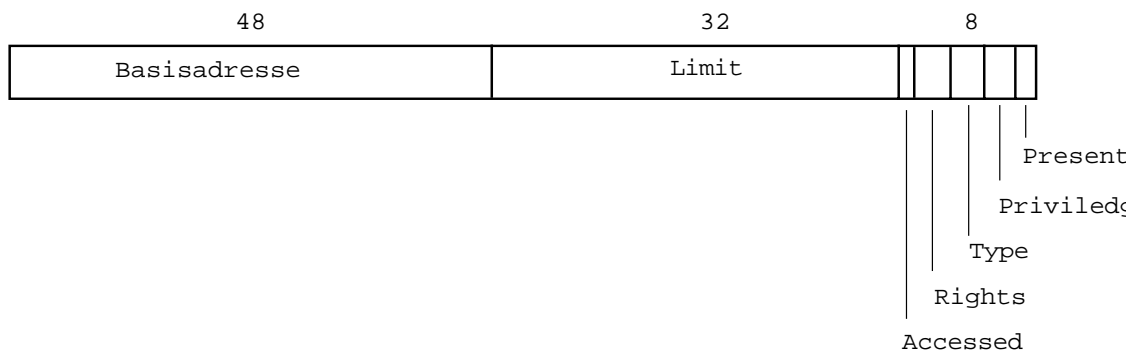
Die Adressierung erfolgt in zwei Stufen: Die logische 48-Bit-Adresse entsteht aus der Konkatenation der normalen Adresse (32 Bit) und dem Segmentregister (16 Bit).

Da in der 8086-Programmierung der Inhalt der Segmentregister um zwei nach links geschoben wird, kann hier über die untersten zwei Bit der Segmentregister anderweitig verfügt werden. Der Inhalt des Segmentregisters adressiert in diesen zwei Bit ein System-Adressregister und bildet mit 14 Bit den Index in eine Deskriptortabelle. Ihre Einträge sind Basisadressen und Längenangaben für bis zu 2^{32} Byte große Segmente im linearen Adressraum.



Die Deskriptorfelder enthalten zugleich auch die Zugriffsprivilegien auf ein Segment.

Aufbau eines Deskriptorfeldes



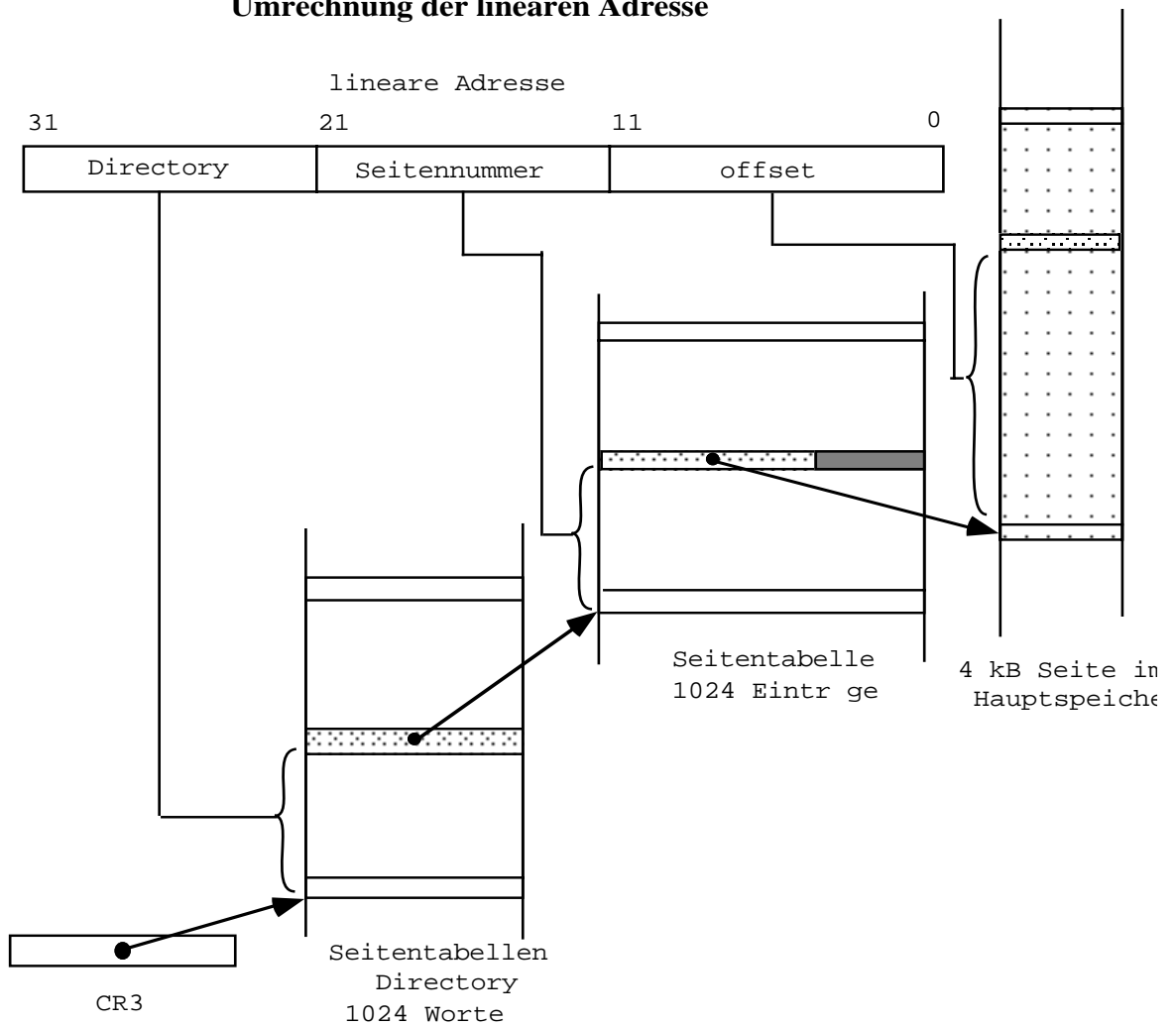
Der 32-Bit Offset in der logischen Adresse bildet die **lineare Adresse** im angesprochenen Segment. Das Segment seinerseits ist in Seiten aufgeteilt.

Die lineare Adresse wird interpretiert als Konkatination eines Directory, einer Seitennummer und eines 12-Bit Offset.

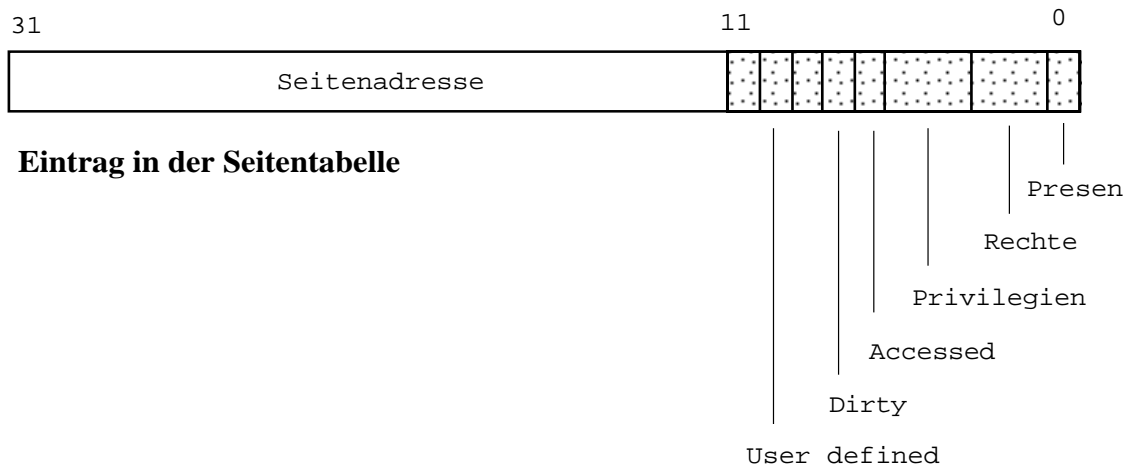
Das gültige Steuerregister enthält die Basisadresse des Directory, dessen Index das entsprechende 10-Bit-Directory-Feld der linearen Adresse bildet.

Das Seitentabellen-Directory enthält die Basisadressen von Seitentabellen. Die ausgewählte Seitentabelle wird mit der Seitennummer der linearen Adresse indiziert und die angesprochene Seitenadresse bildet mit dem Offset die eigentliche physikalische Adresse im Speicher.

Umrechnung der linearen Adresse



Die Einträge in der Seitentabelle enthalten ebenfalls die Zugriffsrechte auf die angesprochene Seite.



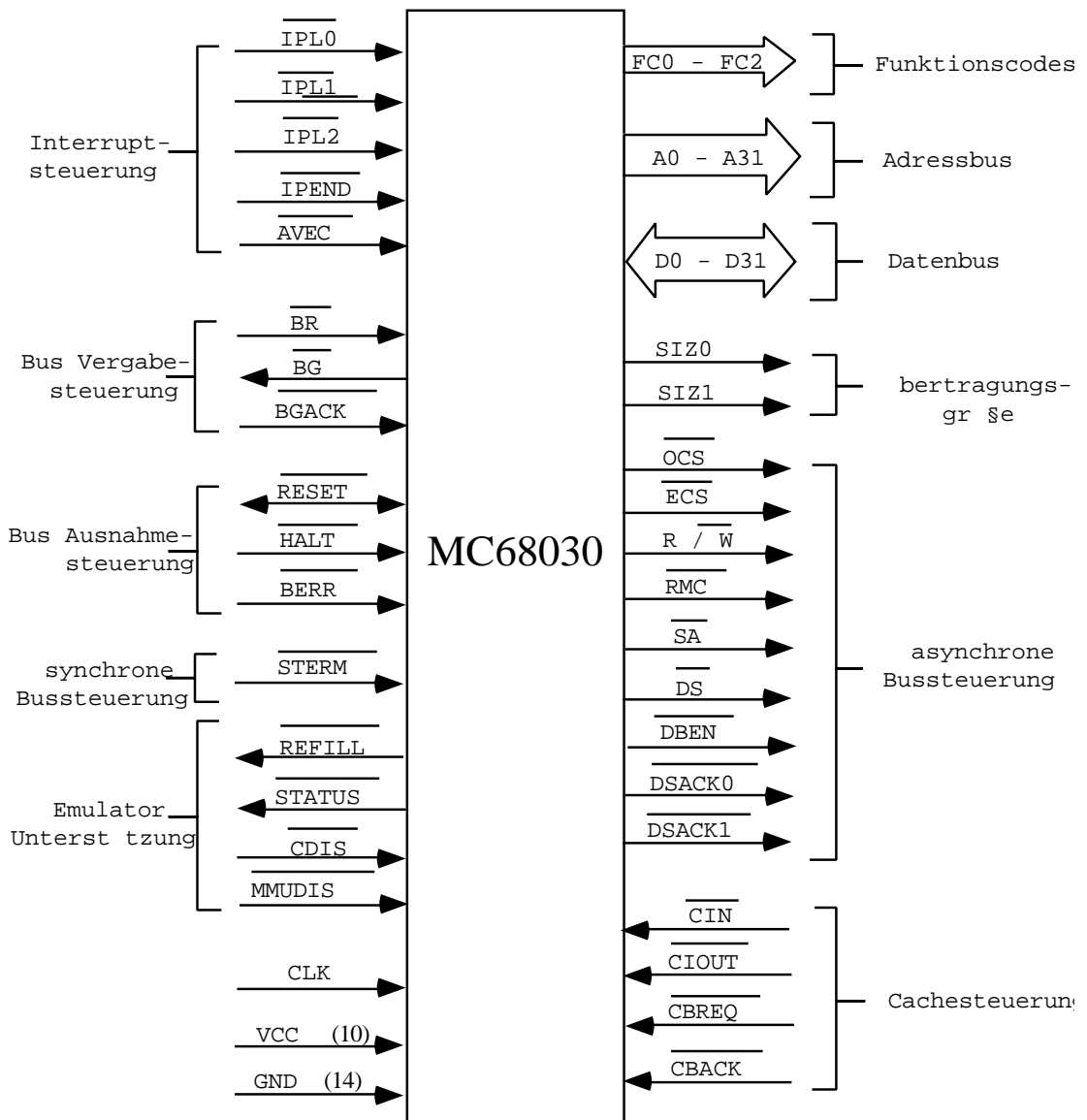
Die physikalische Adresse adressiert ein Segment und kann max. 2^{32} Byte adressieren. Sie hat eine Länge von 32 Bit: 20 Bit Kacheladresse und 12 Bit Offset in einer Kachel. Seiten- und Kacheladresse sind beim 80386 identisch- es gibt kein virtuelles paging.

Der Mechanismus der Adressierung wird durch Caches an Bord des Mikroprozessors unterstützt, die die gültigen Segmentdeskriptoren bei Eingabe des Segmentregisters in einen Assoziativspeicher enthalten und die Kacheladresse herauslesen bei Eingabe von Directory und Seitennummer in einen weiteren Assoziativspeicher. Damit wird der Zugriff schnell.

10.2.4. Motorola 68030

Eine ähnliche Entwicklung hin zu immer komplexeren Maschinen mit Zwang zur Kompatibilität zu Vorgängermodellen gab es bei den Motorola-Rechnern der 68.000-Serie.

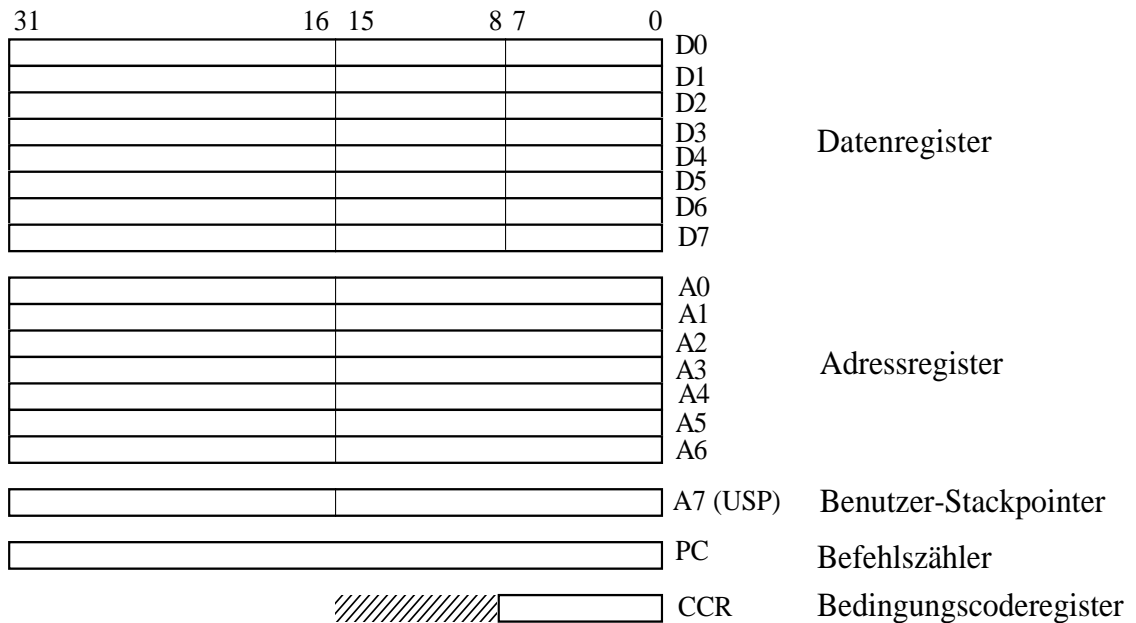
Den Endpunkt bilden die 68030/68040-Rechner.



Der Microprozessor hat 128 Anschlußstifte in einem PGA (pin grid array) - Gehäuse.

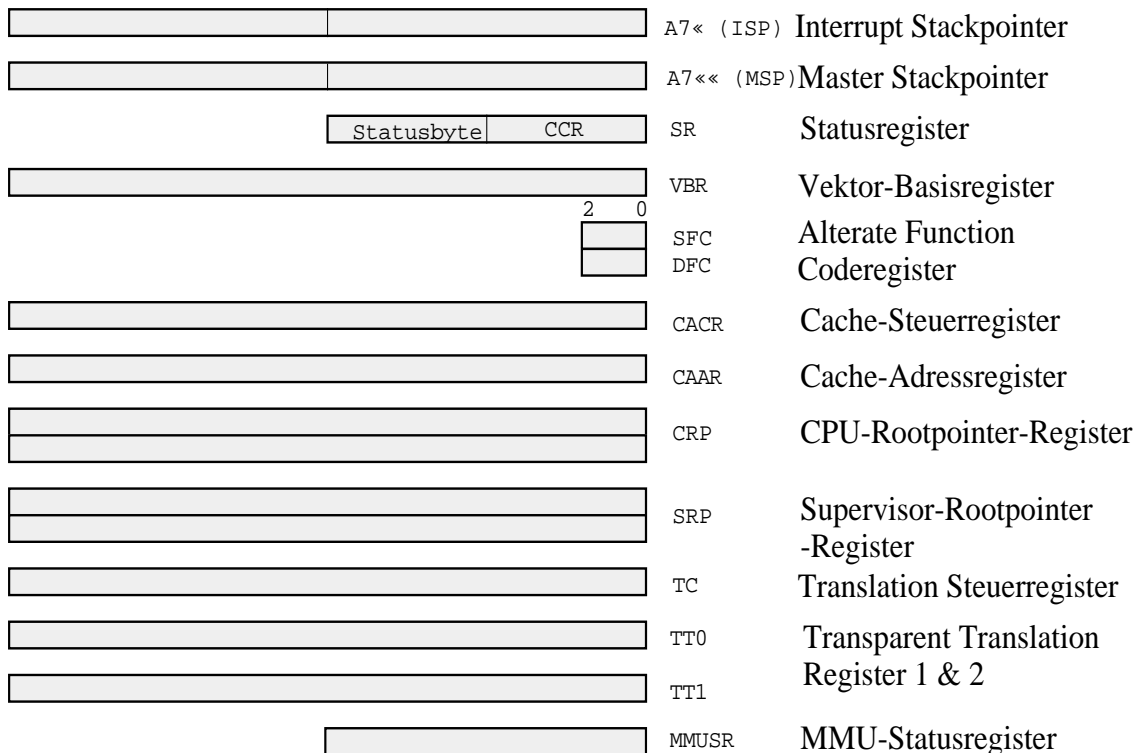
Das Programmiermodell der 68030 zeigen die folgenden Bilder; zunächst den Register-satz des 68000-Rechners, erweitert auf 32 Bit.

Programmiermodell der MC 68030

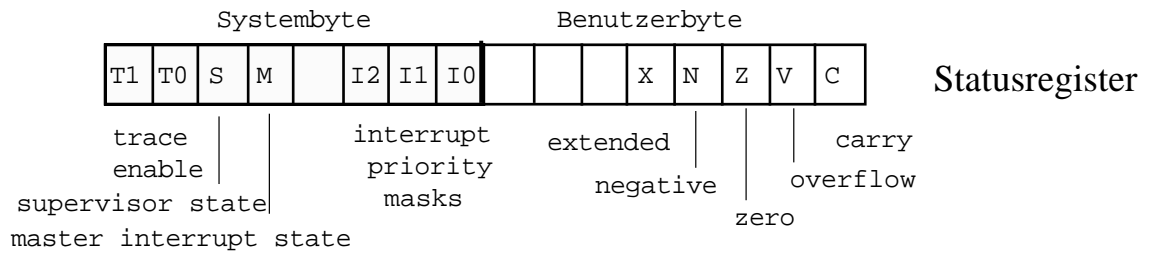


Die 8 Datenregister und 8 Adressregister, der Befehlszähler und das Statusregister (CCR) sind aus der 68.000 entnommen.

Hinzu kommen eine Vielzahl von Registern für das Betriebssystem zum Steuern der Caches, für eigene Stacks, für Interrupts und das B.S., für die Basisadresse von CPU und B.S. und Statusregister der MMU, die mit auf dem Chip integriert ist.



Das Statusregister ist auf 16 Bit aufgeweitet vom 8-Bit-Register der 68.000 Maschine:



Es besteht aus einem Benutzerbyte, dem Bedingungscode-Register (condition code register, CCR), das kurzlebige Informationen hält und einem Systembyte, das langfristig wirksame Informationen enthält.

Die CPU besteht aus

- einer vierstufigen Befehlspipeline mit einer parallelen ersten Stufe, verwaltet mit drei Registern TC, TT0 und TT1
- einem Mikroabwickler mit Kontrollspeicher für das Schaltwerk der Befehlsabarbeitung
- einer Ausführungseinheit, in der parallel mit getrennten Schaltwerken und Registern der Programmzähler fortgeschaltet wird, Adressrechnungen ablaufen und die Rechnungen auf Daten durchgeführt werden.
- einer Bussteuerung, die zwei Adresspuffer verwaltet für den schreibenden Zugriff auf den Speicher und das Lesen vom Speicher
- einer MMU mit einem Assoziativspeicher (address translation cache, ATC), s. Kap.8, mit den Root-Pointer-Registern CRP, SRP und DRP, den 3-Bit Registern SFC und DFC und einem Statusregister MMUSR
- einem Daten- und Programmcache mit Steuerregistern CACR und CAAR. Die Maschine realisiert intern eine Harvard-Architektur.

Das Nachfolgemodell 68040 hat zusätzlich eine Gleitkommaeinheit mit an Bord.

Das Blockdiagramm auf der folgenden Seite zeigt den komplexen Aufbau der Maschine.

