

QU 68000 (V1)

Anwenderhandbuch

D922.025

Version : 1.1

Datum : 01. November 1982

Copyright by Peripherie Computer Systeme GmbH München

Seite -2-

PCS-0U68000 Prozessor

Anwenderhandbuch

Ergänzung vom ersetzt S.-S. Zusatz zu S. Bemerkungen

--	--	--	--

Copyright by PCS GmbH München.

Für diese Unterlagen behalten wir uns alle Rechte vor. Ohne unsere vorherige Zustimmung dürfen sie weder vervielfältigt noch Dritten zugänglich gemacht, oder durch den Empfänger oder Dritte in anderer Weise mißbräuchlich verwertet werden.

Änderungen sind vorbehalten.

INHALT

	Seite
1. Einleitung	4
2. Schaltungsbeschreibung.....	5
3. Adreßraumverteilung u. Systemzustände.....	9
3.1 Anlaufzustand.....	10
3.2 Betriebszustand.....	13
4. Anpassung an den Q-Bus.....	14
4.1 Lese-Operation.....	15
4.2 Schreib-Operation	16
4.3 Bus-Init-Signal	17
4.4 Interrupt	18
4.5 DMA.....	20
4.6 Prozessor Control Register PCR	24
4.7 Error Status Register ESR	26
5. Speicherverwaltung.....	28
5.1 Umsetzung von log. in virt. Adresse	30
5.2 Umsetzung von virt. in physik. Adresse	35
6. Speicher-Bus.....	41
6.1 Adreß- und Datenbus.....	42
6.2 Steuersignale	43
6.3 Spezielle Signale	44
6.4 Interface-Bausteine	44
6.5 Mechanisches Konzept	44
7. Multiprozessor-Systeme	45
8. Unterschiede von Version 1 zu Version 2	46
9. Anhang	47

1. EINLEITUNG

Die Baugruppe PCS-QU68000 ist eine Zentralbaugruppe im Quad slot - Format für Q-Bus-Systeme. Sie basiert auf dem zur Zeit wohl leistungsfähigsten 16-bit Mikroprozessor MC68000 von Motorola, der es in der bereits angekündigten 12 MHz-Variante in etwa mit der VAX 11/750 von DEC aufnehmen kann, wenn es sich nicht um Gleitkomma - intensive Aufgaben handelt.

Der Prozessor MC68000 wurde auf der Baugruppe optimal in die Q-Bus-Umgebung eingepaßt, sodaß ein Q-Bus-Zyklus wesentlich schneller als bei den DEC-Prozessoren LSI 11/2 oder LSI 11/23 abläuft. Die Karte PCS-QU68000 kann also in bestehenden LSI 11/2 oder LSI 11/23 - Systemen diese Zentralbaugruppe ersetzen, und somit die Systemleistung unter Beibehaltung der bestehenden restlichen Hardware ganz entscheidend steigern.

Zur Umsetzung der logischen Adressen in die physikalisch vorhandenen ist eine MMU - Schaltung (Memory Management Unit) implementiert, die auch alle Aktivitäten bezüglich Zugriffsrechten und der Page - Fault - Behandlung (beim MC 68010) durchführt.

Als Betriebssystem für die PCS-QU68000-Baugruppe dient das auf der UNIX Version 7 (von Bell Lab.) basierende MUNIX, zu dem standardmäßig schon C - Compiler, 68000 - Assembler sowie viele Utilities gehören, und wofür noch eine große Anzahl von anderen Sprachen- und Dienstprogrammen verfügbar ist.

Die Baugruppe ist in 2 Versionen verfügbar:

- QU68050 mit 2-stufiger MMU und S-Bus-Interface
- QU68030 mit 1-stufiger MMU ohne S-Bus-Interface

In der folgenden Dokumentation wird nur von QU68000 gesprochen.

2. SCHALTUNGSBESCHREIBUNG

Der Prozessor PCS-QU68000 arbeitet als Zentralbaugruppe in einem Q-Bus-System. Der logische Adreßraum von 4 x 16 MByte, die die CPU MC68000 adressieren kann, wird durch die 22 Adreßleitungen auf dem Q-Bus auf einen physikalischen Adreßraum von 4 MByte begrenzt. Die mit MSI- und LSI-Bausteinen aufgebaute MMU übersetzt den logischen Adreßraum in den virtuellen und anschließend in die physikalische Adresse (2 - stufige Adreßumsetzung), entdeckt Verletzungen von Zugriffsrechten und unterstützt beim MC68010 die Page-Fault-Behandlung.

Unabhängig von der MMU ist noch ein direkt adressierbarer Speicher (EPROM und RAM) für Boot, Initialisierung der MMU - Register, Diagnose -Programm und Online - Debugger auf der Karte vorhanden.

Um die maximal mögliche Geschwindigkeit der 68000-CPU auszunutzen, erfolgt der Zugriff auf den Speicher über einen separaten S-Bus (Speicher-Bus). Der Q-Bus wird vom Prozessor nur noch für Ein-/Ausgabe - Aktivitäten benutzt. Mit der Dualport - Struktur des lokalen Speichers können auch DMA - Zugriffe direkt vom Q-Bus durchgeführt werden.

Dieses Konzept erlaubt desweiteren Multiprozessor-Konfigurationen mit max. 16 PCS-OU68000-Baugruppen für zeitkritische Rechenaufgaben mit geringen Ein-/Ausgabe-Aktivitäten.

Bild 2.1 zeigt die Gesamtstruktur eines auf dem QU68000-Prozessor basierenden Rechnersystems. Am Q-Bus können Standard-Q-Bus-Speicher (22 bit Adressierung), programmgesteuerte Ein-Ausgabe-Controller und DMA-Controller angeschlossen sein. Die Rolle der LSI 11 (Prozessor, Bus-Arbiter) übernimmt der QU68000-Prozessor, der seinerseits über einen eigenen Speicher-Bus (S-Bus) mit speziellen Speicherkarten verbunden ist (256 kByte oder 512 kByte, mit zusätzlicher Verbindung zum Q-Bus) DMA-Transfers von der Peripherie sind in diese Speicher möglich, wobei ein 2. Port der Speicher verwendet wird (Q-Bus-Port); bei mehreren Speichermodulen ist auch ein völlig überlappter Betrieb (S-Bus und Q-Bus) möglich.

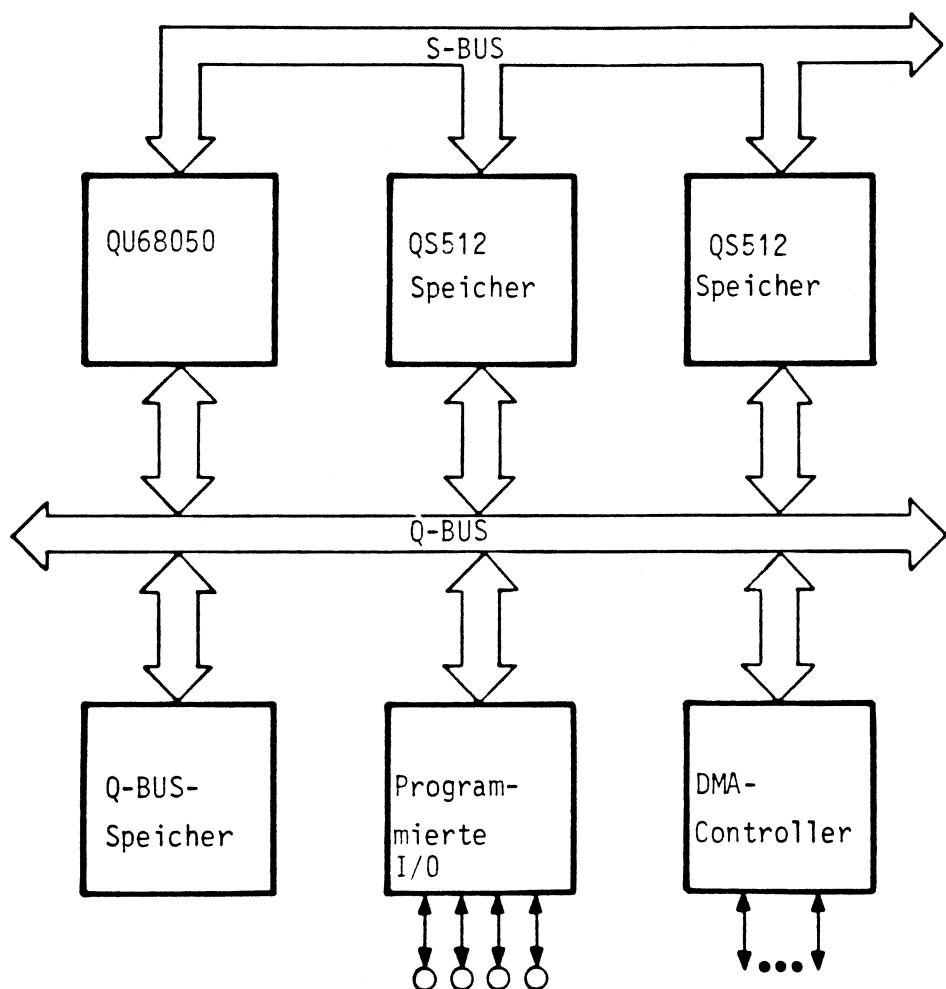


Bild 2.1 GESAMT-STRUKTUR DES RECHNERSYSTEMS

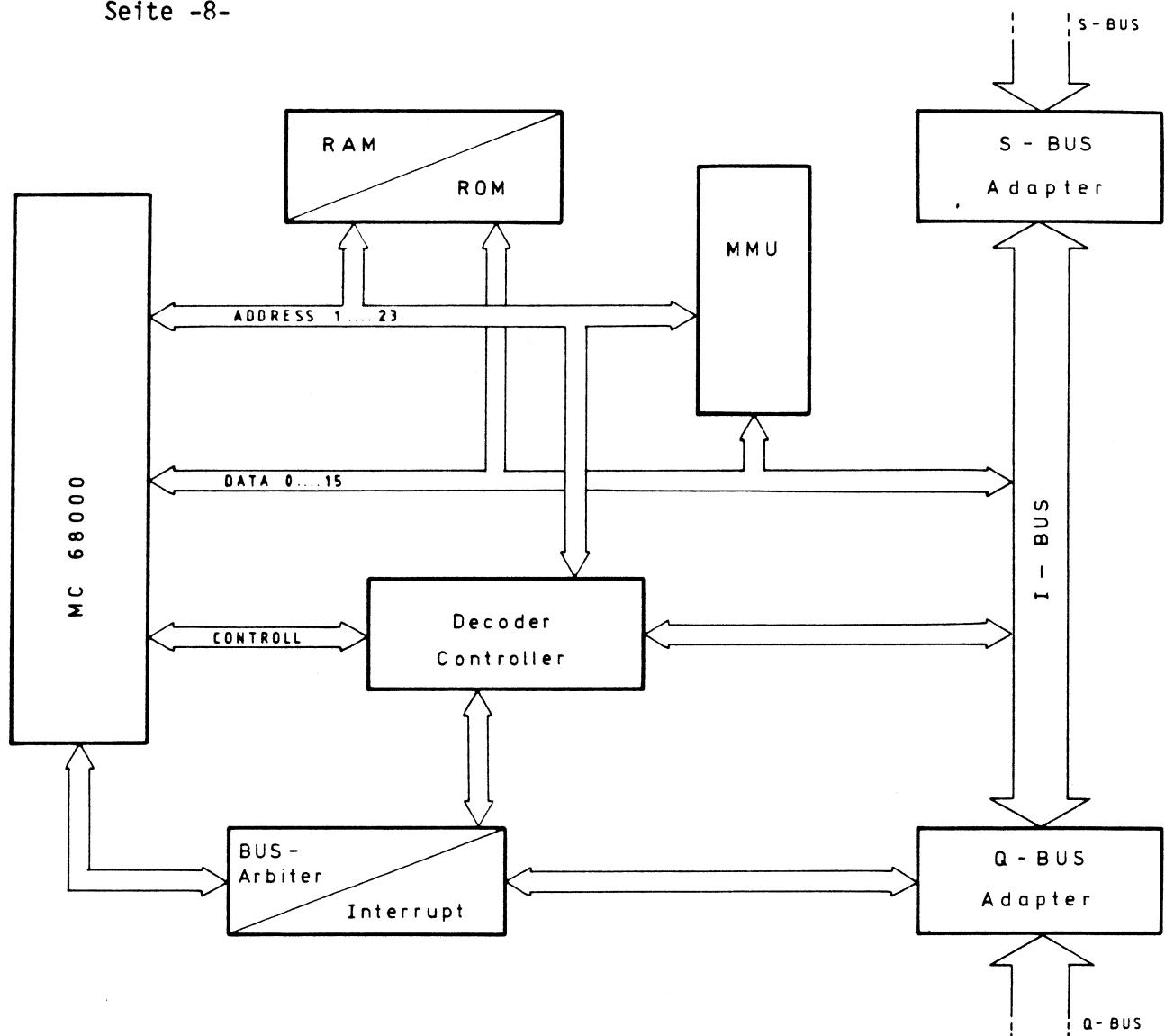
BLOCKSCHALTBILD

Bild 2.2 zeigt ein Blockschaltbild der Prozessorkarte QU68000. Der Prozessor MC68000 bildet das Herz der Einheit und wird um eine Timing-Controll-Einheit sowie die Decodierungseinrichtung ergänzt. Die Q-Bus-Anpassung erfolgt bezüglich der Datenwege zum einen über den Q-Bus-Adapter, die Anpassung an die Übertragungsprotokolle (Bus-übertragungsprotokoll, Interrupt-Protokoll, DMA-Protokoll) erfolgt durch eine Arbiter-Interrupt-Einheit. Die internen EPROM- und RAM-Speicher sind am internen Adreß- und Datenbus angeordnet; die vom Prozessor gelieferten Adressen werden von der MMU so umgesetzt, daß sie zur Adressierung auf dem Q-Bus bzw. auf dem S-Bus zur Verfügung stehen. Der S-Bus-Adapter ist mit besonderer Beachtung der Gatterlaufzeiten konzipiert, um die - bei weitem häufigsten - Zugriffe zum privaten Speicher so günstig wie möglich zu gestalten.

Der 16 Bit-breite bidirektionale Datenbus wird auf dem internen I-Bus zwischen Prozessor, S-Bus-Adapter und Q-Bus-Adapter verteilt (Tristate-Bus). Die vom MC68000 (MMU) gelieferten Adressen gelangen entweder zum Q-Bus-Adapter (auf welchem Adressen und Daten gemultiplext werden) oder zum S-Bus-Adapter.

Diese Betrachtung macht deutlich, daß auf der Prozessor-Karte zahlreiche Daten- und Adreßleitungen verwaltet werden müssen, wobei die auf dem Q-Bus gemultiplexten Adreß- und Datenleitungen die Verhältnisse noch etwas erschweren.

Besonders wichtig ist eine geeignete Adreßraumverteilung. Alle im System enthaltenen Speicher haben einen gemeinsamen, max. 4 MByte großen Adreßraum, die entweder am S-Bus und am Q-Bus oder nur am Q-Bus angeordnet sein können. Natürlich muß durch Verdrahten entsprechender Lötbrücken festgelegt werden, welche physikalischen Adreßbereiche zum S-Bus gehören. Insgesamt gibt es 8 Einheiten zu 512 KByte. Für die Anlaufphase gilt im Übrigen eine andere Abbildung der logischen Adressen auf die physikalischen Adreßräume als im normalen Betriebszustand, wo die unteren 5 MByte (0 bis 4 MByte) in den physikalischen Adreßraum des Q-Bus gemappt werden (siehe 3.2).



BLOCKDIAGRAMM PCS - QU 68000

Der Q-Bus-Adapter paßt das Handshake-Protokoll für die einzelne Datenübertragung an, sodaß der Prozessor wahlweise auf den lokalen Speicher oder den Q-Bus-Speicher zugreifen kann, oder ein DMA - Controller Zugriff auf den Q-Bus-Speicher hat.

3. ADRESSRAUMERTEILUNG UND SYSTEMZUSTÄNDE

Da die MMU-Register zu Beginn nicht mit sinnvollen Werten besetzt sind (ein Zugriff auf die Speicher jedoch nur mit definierten MMU-Register-Inhalten möglich ist) und der 68000-Prozessor hardwaremäßig so eingerichtet ist, daß er sich den Anfangszustand des Befehlszählers und des Prozessor-Status-Registers aus hardwaremäßig vorgegebenen Speicherstellen (0 bis 7) herausholt, müssen beim Einschalten des Prozessors diese Speicherzellen definierte, sinnvolle Werte enthalten. Auf der Prozessor-Karte besteht also die Notwendigkeit, zwei Systemzustände zu unterscheiden:

- Anlaufzustand
- Betriebszustand

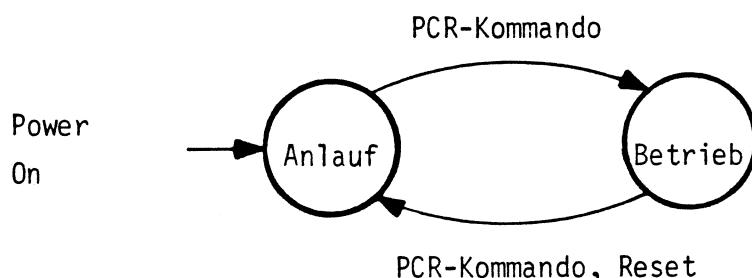


Bild 3.1 ZUSTANDSDIAGRAMM

Ein Zustand-Bit (Anlauf-Bit) legt fest, ob sich die Prozessorkarte im Anlaufzustand oder im Betriebszustand befindet (Bild 3.1). Mit dem Einschalten des Prozessors (Anlauf-Reset, Prozessor-Init) gelangt das Anlauf-Flip-Flop in jedem Fall in den Anlauf-Zustand. Ein definierter Befehl bringt es schließlich in den Betriebszustand, ein anderer Befehl ermöglicht die Rückkehr in den Anlaufzustand (vgl. PCR, Prozessor-Control Register).

3.1 ANLAUFZUSTAND

Im Anlaufzustand wird verhindert, daß Speicherzellen auf dem Q-Bus oder dem S-Bus erreicht werden. Es sind also nur die internen, auf der Prozessor-Karte vorhandenen Betriebsmittel (insbesondere das EPROM, das RAM und die MMU-Register) sowie die External Page des Q-Bus erreichbar.

Die höherwertigen neun Adreßbit des MC68000-Prozessors werden gewissermaßen nicht decodiert; dies bedeutet, daß der 32 K-Adreßraum, aus welchen mit den restlichen 15 Adreßbit jeweils eine Speicherzelle ausgewählt werden kann, insgesamt 512-mal in den 16 MByte Adreßraum des Prozessors hineingespiegelt wird. Damit erreicht man dieselbe Zelle sowohl mit der Adresse x als auch mit der Adresse $X + N \times 2^9$ ($0 = N = 511$) usw. Insbesondere bedeutet dies, daß man den EPROM Speicher, der an den Anfang des 32-K-Adreßbereichs gelegt wird, unter der Adresse "0" erreichen kann. Die ersten 8 Byte (4 Worte) des EPROM enthalten die für den Prozessoranlauf benötigte Startadresse des Programmzählers sowie das Prozessor-Status-Wort.

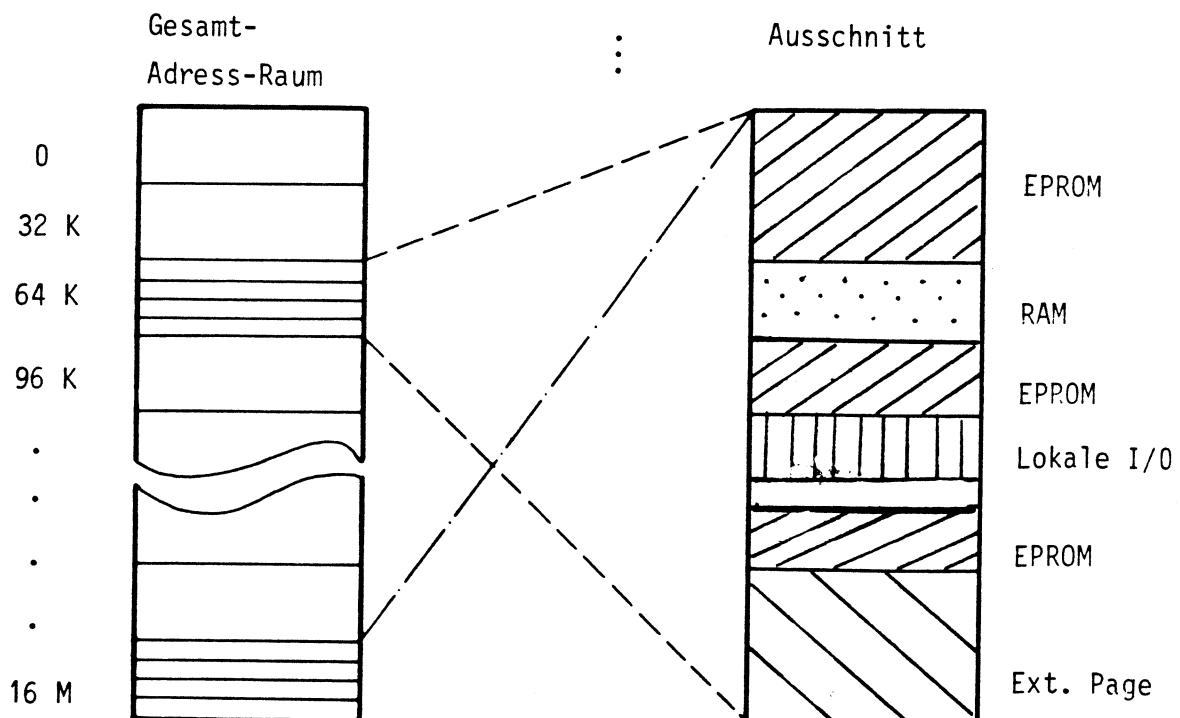


Bild 3.2 ADRESSRAUM-ZUORDNUNG IM ANLAUF-ZUSTAND

Die Adreßraumverteilung innerhalb des 32 KB-Adreßraums zeigt die Abb. 3.2.

- a) EPROM-Speicher, max. 8 K-Worte a 16 bit.
- b) RAM-Speicher, max. 2 K-Worte a 16 bit.
- c) Adressierung der Prozessor-internen Register, insbesondere MMU-Register. Von den hier verfügbaren 8 KByte werden nur wenige Adressen für die Versorgung der MMU-Segment-Descriptoren sowie der Seiten/Kachel-Umsetzung benötigt. Zusätzlich hier decodierte Ein-/Ausgabebefehle betreffen beispielsweise die Veränderung des Prozessor-Control-Registers PCR (u.a. Anlauf-Bit oder die Steuerung der 20 ms-Echtzeituhr) oder das Einlesen des BUS-Error-Statusworts.
- d) Der letzte 8 KByte-Bereich entspricht der External-Page der PDP11-Architektur. Es sei darauf hingewiesen, daß diese External-Page auch im Anlaufzustand nur dann sinnvoll benutzt werden kann, wenn der Segment-Deskriptor 15 korrekt besetzt ist.

Folgende Funktionen werden durch Programme im EPROM unterstützt:

- Boot, in welchem ein konsistenter Anfangszustand (insbesondere bezüglich der MMU-Register hergestellt wird).
- Diagnose-Programm, welches beim Anlauf die prinzipielle Funktionsfähigkeit der Hardware -Komponenten überprüft.
- Ein ODT (Online-Debug-Tool), welches die Bedienung des Prozessors von einem Terminal (angeschlossen über eine am Q-Bus angebrachte DLV 11-Karte) ermöglicht. Hierüber können - ähnlich wie bei der LSI 11 - alle internen Register des Prozessors und der Prozessor-Karte gelesen und modifiziert werden, außerdem ist in den Grundzügen ein überwachter Ablauf von Programmen möglich.

3.2 BETRIEBSZUSTAND

Nachdem im Anlaufzustand die Initialisierung der MMU-Segment-Desribtoren und anderer Prozessor-Register abgeschlossen wurde, wird in dem Anlaufprogramm auf den Betriebszustand umgeschaltet. Dadurch wird zum einen die MMU eingeschaltet, zum anderen eine andere Abbildung der logischen Adressen auf die physikalischen Adressen bewirkt. Bild 3.4 zeigt, daß mit Ausnahme des obersten 32 K-Blocks alle 68000-Adressen über die MMU an den S-Bus-oder Q-Bus-Speicher übergeben werden. Der in Bild 3.3 ebenfalls gezeigte Detailausschnitt macht deutlich, daß auch die obersten 8 K des 32 K-Blocks (External Page) zum Auslösen eines Q-Bus-Zyklus führen: Gleichzeitig ist das Bus-Signal BBS7 aktiviert, welches den am Q-Bus anliegenden Peripherie-Controllern anzeigt, daß jetzt eine Ein-Ausgabe-Operation stattfindet. Es sei jedoch darauf hingewiesen, daß die Realisierung dieses 32 K-Ausschnitts nur für den Supervisor-Mode des 68000-Prozessors gilt: (insbesondere ist damit die Bedienung der Peripherie nur im Supervisor-Mode möglich): Im User-Mode erfolgt eine Abbildung des gesamten 16 MByte, Adreßraums auf Q-Bus bzw. S-Bus.

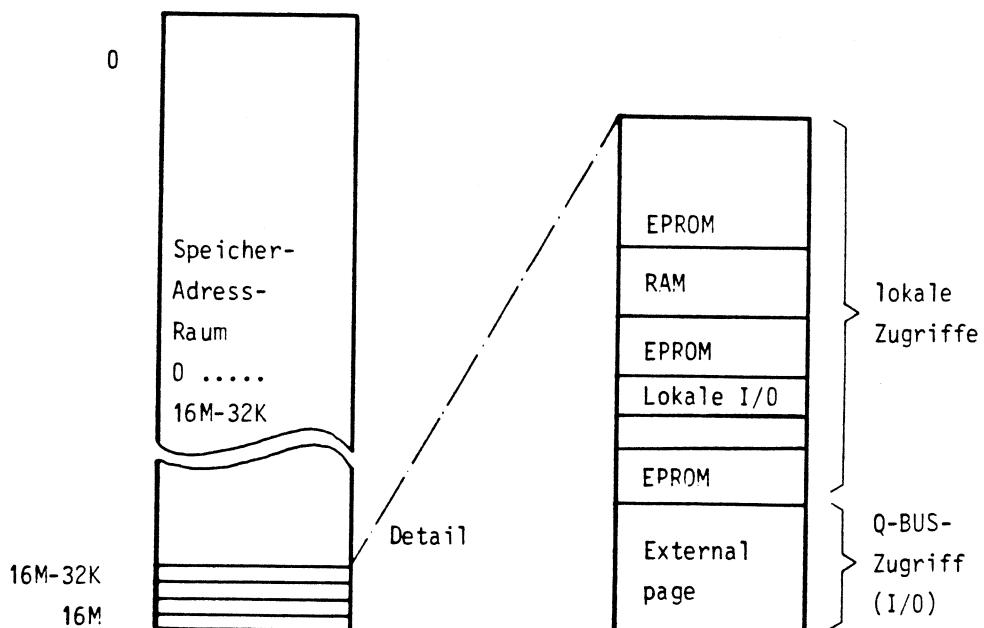


Bild 3.3 ADRESSRAUMAUFTHEILUNG IM
SUPERVISOR-MODE DES BETRIEBSZUSTANDES

4. ANPASSUNG AN DEN Q-BUS

Die Protokollabbildung zwischen dem MC68000 und Q-Bus ist nahezu vollständig durch die Timing-Restriktionen des Q-Bus bestimmt/1/. Auch die Zeitverzögerung, die sich aus der Tatsache ergibt, daß am MC68000 keine AO-Adressleitung existiert, kann in der Zeit "versteckt" werden, die von der MMU zur Erzeugung des physikalischen Adresse benötigt wird (AO wird aus dem Signal UDS abgeleitet).

Die Anpaßschaltung hat folgende Aufgaben:

- Das Hand-Shake-Protokoll für die einzelne Datenübertragung muß adaptiert werden. Insbesondere müssen aus den Signalen LDS und UDS sowohl das niederwertigste Adreßbit Ao als auch - in Verknüpfung mit den Übertragungsrichtungssignal - die Signale DIN und DOUT sowie WTBTR erzeugt werden. Das von 68000-Prozessor erwartete Signal DTACK muß wiederum aus dem Rückmeldesignal RPLY abgeleitet werden. Hierbei ist insbesondere zu beachten, daß das niederwertigste Bit erst mit dem Anliegen von UDS/LDS bekannt ist. Bei schreibenden Transaktionen kann erst zu diesem Zeitpunkt festgestellt werden, ob es sich um eine Wort- und Byte-Operation handelt; Hieraus ergeben sich leider zusätzliche Verzögerungen, die es - zusammen mit den Zeitbedingungen im Q-Bus - verhindern den Adreß-Strobe AS des 68000 direkt in das Q-Bus-Signal "Sync" abzubilden.
- Das Signal BBS7, das eine Adressierung der External Page angibt (Ein-Ausgabe-Register- Sätze), wird aus der 68000-Adresse abgeleitet. Da diese Adresse (niederwertige 13-Bit) ebenfalls über die MMU geleitet wird, muß auf die Besetzung des entsprechenden MMU-Segment-Registers geachtet werden (Segment 15).
 - die Datenpfade können so durchgeschaltet werden, daß
 - o der Prozessor auf den privaten Speicher zugreifen kann
 - o der Prozessor auf den Q-Bus-Speicher zugreifen kann und
 - o ein DMA-Controller auf den Q-Bus-Speicher zugreifen kann.
- Bei der letztgenannten Datenübertragung ist der Prozessor nur insoweit betroffen, als er die Funktion des Bus-Arbiter bereitstellen muß, Prozessor-Zugriffe über den S-Bus können parallel dazu erfolgen.

4.1 LESE-OPERATIONEN (Zeitdiagramm)

Bild 4.1 Zeigt die Zeitverhältnisse beim Lesen von Datenworten aus dem Q-Bus-Speicher. Zunächst müssen die MMU-Adressen (sowie UDS) stabil anstehen, danach folgt am O-Bus eine minimale Setup-Zeit von 150 ns. Dann wird BSYNC erzeugt, und der Rest des Lesezyklus am Q-Bus läuft - beim 10 MHz-Prozessor - mit minimal zulässigen Zeiten ab. Dennoch wird der 68000-Bus-Zyklus um 4 Wait-States auf insgesamt 8 Takte verlängert - eine Verdopplung also, die die Verarbeitungsleistung des MC68000 quasi halbiert. Der Q-Bus-Zyklus allerdings läuft wesentlich schneller ab als bei den DEC-Prozessoren LSI11/2 oder LSI11/23 (ca. 0,8... 1us).

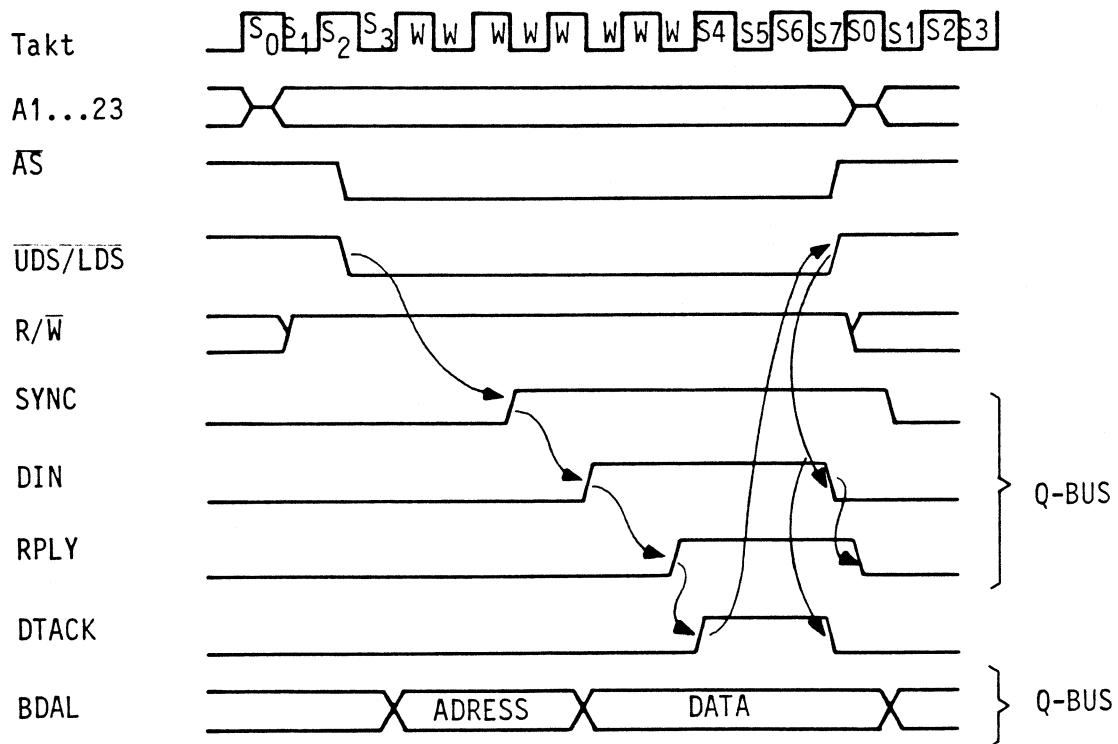


Bild 4.1 ADAPTION 68000-Q-BUS

FOR LESE-OPERATIONEN

4.2 SCHREIB-OPERATIONEN (Zeitdiagramm)

Das Zeitdiagramm 4.2 zeigt die entsprechenden Verhältnisse beim Schreiben; wegen der sequentiellen Übertragung von Adresse und Daten auf dem Q-Bus werden hier - optimale Speicher vorausgesetzt - zusätzliche Wartezustände benötigt, die allerdings statistisch (ca. 10 % Schreibzugriffe) nicht ins Gewicht fallen.

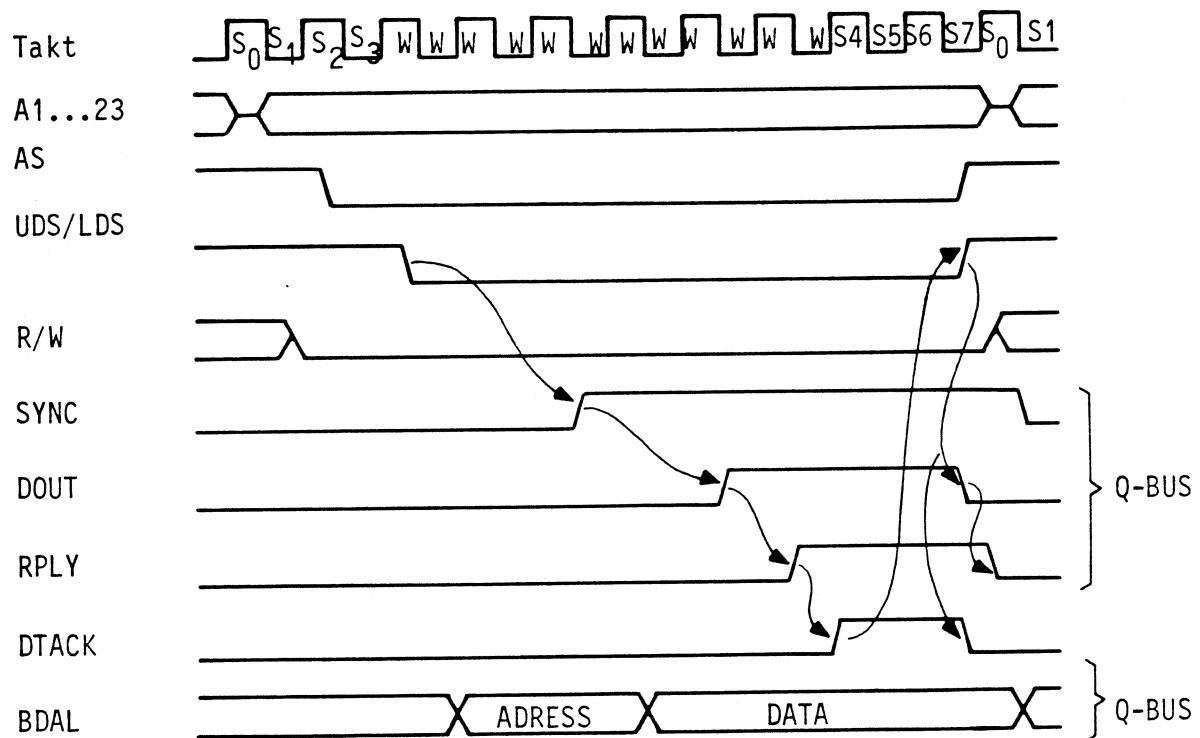


Bild 4.2 ADAPTION 68000-Q-BUS

FÜR SCHREIBOPERATIONEN

4.3 BUS-INIT-SIGNAL

Am Q-Bus ist ein BUS - Init-Signal definiert, das beim Power-on-Reset oder durch einen speziellen LSI 11-Befehl aktiviert wird. Dieser Befehl muß zum Rücksetzen von Peripherie-Controllern am Q-Bus durch ein Ausgabe-Kommando an das PCR nachgebildet werden; zusätzlich hat der RESET-Befehl des MC68000 dieselbe Wirkung.

4.4 INTERRUPT

Bei der Protokollabbildung des Interrupt ist es zum einen notwendig, die 4 am Q-Bus implementierten Interrupt-Ebenen auf vier der sieben MC68000-Ebenen abzubilden, zum anderen müssen auch einige Prozessorkarten-interne Ereignisse zur Programmunterbrechungen führen.

Bezüglich der Q-Bus-Interrupts sind die Konzepte des MC68000 und des Q-Bus außerordentlich ähnlich: Vier Interrupt-Request-Leitungen führen zur Aktivierung der MC68000-Interrupt-Leitungen, ein Interrupt-Acknowledge-Zyklus (eigener Funktionscode) dient dazu, die Auswahl der jeweils nächsten Interrupt-Ursache festzustellen. Auch das Interrupt-Vektor-Konzept des MC68000 ist dem des Q-Bus sehr ähnlich: Während allerdings bei der LSI 11 der 8-bit-breite Interrupt-Vektor (die beiden LSB-Bit sind 0) direkt zur Adressierung des Interrupt-Vektors (dort: Befehlszähler und Prozessor-Status-Wort) verwendet werden, wird das selbe Byte beim MC68000 zunächst um 2 Bit-Positionen verschoben, so daß insgesamt bis zu max. 256 Interruptvektoren möglich sind. In den vier jedem Interrupt-Vektor zugeordneten Bytes steht im übrigen der 24 (32) breite Befehlszähler (vgl. Beschreibung des MC 68000). Die Zuordnung der Interruptebenen erfolgt im einzelnen wie folgt:

- Q-Bus-Level 4: 68000-Level 1
- Q-Bus-Level 5: 68000-Level 2
- Q-Bus-Level 6: 68000-Level 4
- Q-Bus-Level 7: 68000-Level 5

Zusätzlich werden drei Prozessorkarten-interne Signale für Programmunterbrechungen herangezogen. Um dabei nicht eigene Schaltkreise zur Bereitstellung von Interrupt-Vektoren einsetzen zu müssen, wird für diese drei Ereignisse der Autovektor-Mode des MC 68000 verwendet. Im einzelnen handelt es sich um folgende Ereignisse:

- Power Fail:

Das Q-Bus-Signal BPOKH zeigt das Ereignis "Power-Fail" an und löst eine MC68000-Programmunterbrechung auf der höchsten verfügbaren Interruptebene (Ebene 7) aus. Dies entspricht der Vektor-Nr.31 (dezimale Adresse 124).

- Q-Bus-Halt:

Das Q-Bus-Signal BHALT zeigt die Anforderung nach dem Anhalten des Q-Bus-Betriebs an und soll den QU68000 in den Monitor-Zustand überführen. Diesem Ereignis ist der Interrupt auf der Ebene 6 mit dem Autovektor Nr. 30 (Dezimaladresse 120) zugewiesen.

- Bus-Event:

Als drittes Ereignis dient das am Q-Bus vorhandene Signal BEVNT, welches ein aus dem Netzclock abgeleitetes Echtzeit-Interruptsignal darstellt. Im MC68000 führt dies zu einer Unterbrechung auf Ebene 3, was einem Autovektor der Nummer 27 (Dezimaladresse des Vektors 108) entspricht. Über diese Interruptebene kann der Aufbau der Echtzeituhr erfolgen. Über ein Ausgabe-Kommando an das PCR ist es im Übrigen möglich, diese Ereignisleitung ein- bzw. auszuschalten.

4.5 DIRECT MEMORY ACCESS DMA

Beim DMA-Betrieb ist es sinnvoll, daß auf dem Q-Bus Datenübertragungen zwischen einer DMA-Einheit und einem Q-Bus-Speicher gleichzeitig mit Speicherzugriffen des MC68000-Prozessors auf dem privaten Speicher ablaufen können. Dazu werden folgende Festlegungen getroffen:

- Der MC68000-interne Arbiter wird nicht zur Arbitrierung am Q-Bus herangezogen; Vorgänge im QU68000 über den S-Bus und auf dem Q-Bus können damit parallel ablaufen.
- Ein spezieller Arbiter bedient den Q-Bus und teilt ihn - wenn nicht gerade ein interner Q-Bus-Zyklus abläuft - dem anfordernden DMA-Gerät zu.

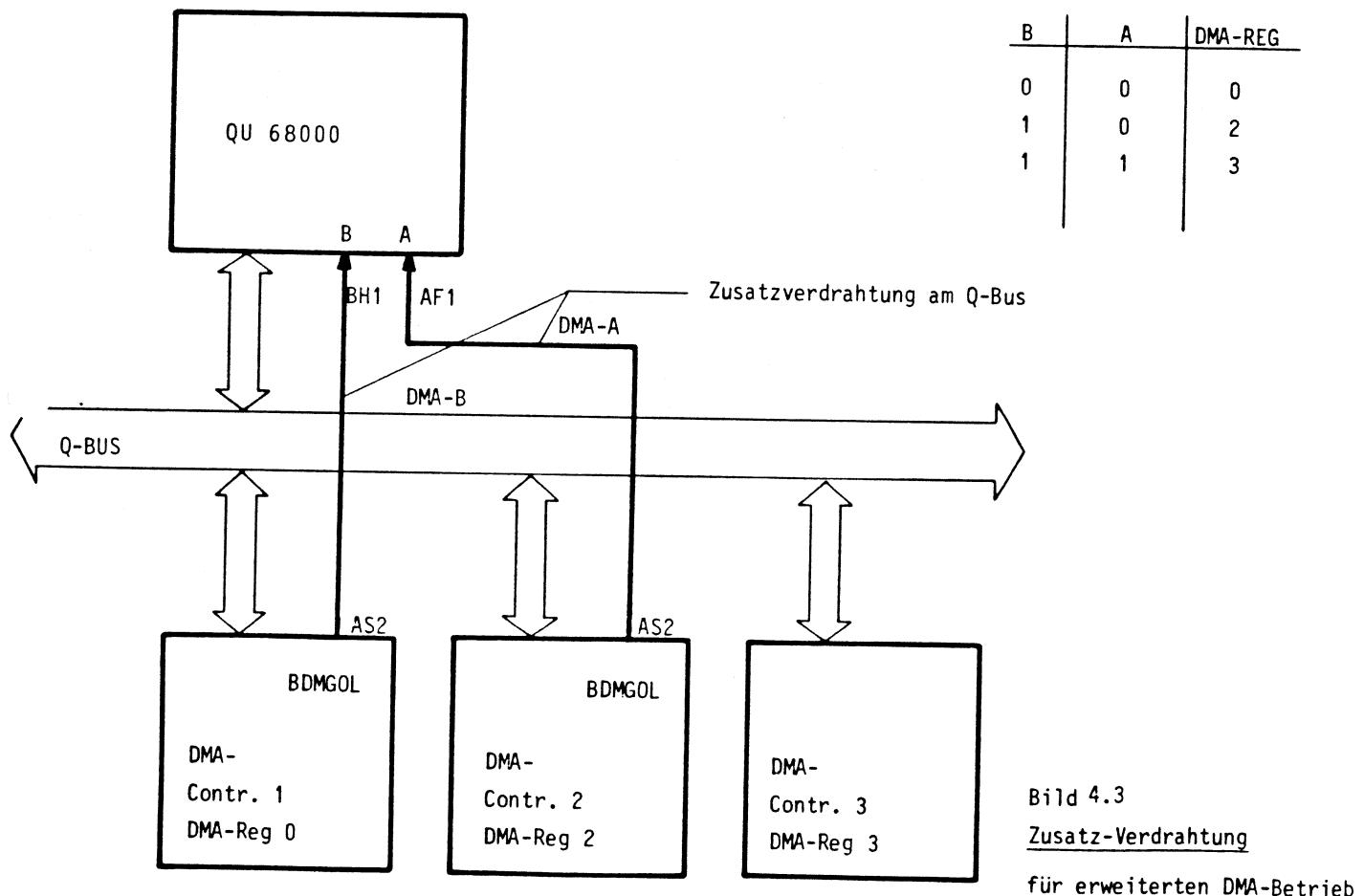
Diese Lösung optimiert den Systemdurchsatz (Parallelbetrieb von Bussen und Speichern) und weist in Richtung "Multiprozessing".

Die meisten Standard-DMA-Controller stellen am Q-Bus nur 18 Adreßbit zur Verfügung. Daher können nur die untersten 256 K des 4 MByte-Adreßraums erreichen. Sie kann durch Verfügbarmachung eines Adreß-Extension-Registers für die Bit-Positionen 18 bis 21 entschärft werden. Dabei sind zwei Varianten denkbar:

- Bei jedem DMA-Zugriff wird ein einzelnes 4 bit-Register auf die erweiterten Adreßleitungen 18 bis 21 angelegt. Damit kann ein einzelnes DMA-Gerät auf jeden beliebigen Speicherbereich im Q-Bus-Speicher und privaten Speicher zugreifen. Probleme entstehen dann, wenn mehrere DMA-Controller quasi gleichzeitig in unterschiedlichen 256 K-Segmenten arbeiten sollen.
- Alternativ können mehrere dieser 4-Bit-Register (Registersatz) verwendet werden, wobei in Abhängigkeit vom aktiven DMA-Controller jeweils ein anderes Register eingeschaltet wird. Leider ist diese Variante nicht ohne Modifikationen am Q-Bus realisierbar: Eine zusätzliche Verdrahtung auf der Q-Bus-Grundplatine (Verbindungsleitung von den Daisy-Chain-Gliedern zu der Prozessorkarte) ermöglicht es jedoch, den gerade aktiven Master zu identifizieren und das korrekte Erweiterungsregister auszuwählen. Diese 4-bit-Register werden in der Anlauf-Routine mit "0" vorbesetzt, sodaß sie nur dann wirksam werden, wenn sie in einer Treiberroutine entsprechend (0) besetzt werden (wired OR am Q-Bus).

Bild 4.3 zeigt schematisch die Anordnung von Prozessor und 3 DMA-Controllern am Q-Bus zusammen mit den zusätzlich benötigten Verbindungen: Die gewählte direkte Verwendung der Daisy-Chain-Signale DMG erlaubt die Auswahl von 3 Extension-Registern DER 0, 2 und 3 (die bit-Kombination "01" tritt nicht auf, DER 1 ist also nicht nutzbar).

Werden die Zusatzleitungen nicht verdrahtet, dann degeneriert die Lösung zu der oben beschriebenen 1-Register-Variante(DERO). Die Auswahl der Varianten liegt also beim Anwender.



Zusatzverdrahtung für erweiterten DMA-Betrieb

Bild 4.4 zeigt die Pins am Backplane.

Signal QU 68000 DMA-Dev 1 DMA-Dev 2

DMA-A	AF 1	-----	AS 2
DMA-B	BH 1	AS 2	-----

VORSICHT:

Wenn die DMA-Controller an andere Steckplätze im Q-Bus gesteckt werden, muß die Zusatzverdrahtung der neuen Konfiguration angepasst werden.

Sind in einem System mehr als 3 DMA-Controller, dann können mehrere Controller das gleiche DMA-Extension-Register benutzen, d. h., alle Controller die vor DMA-Dev 1 gesteckt werden benützen das DMA-Reg. 0.

Alle Controller die zwischen DMA-Dev 1 und DMA-Dev 2 sind benützen das DMA-Reg 2.

Alle Controller die nach DMA-Dev 3 sind benützen das DMA-Reg 3.

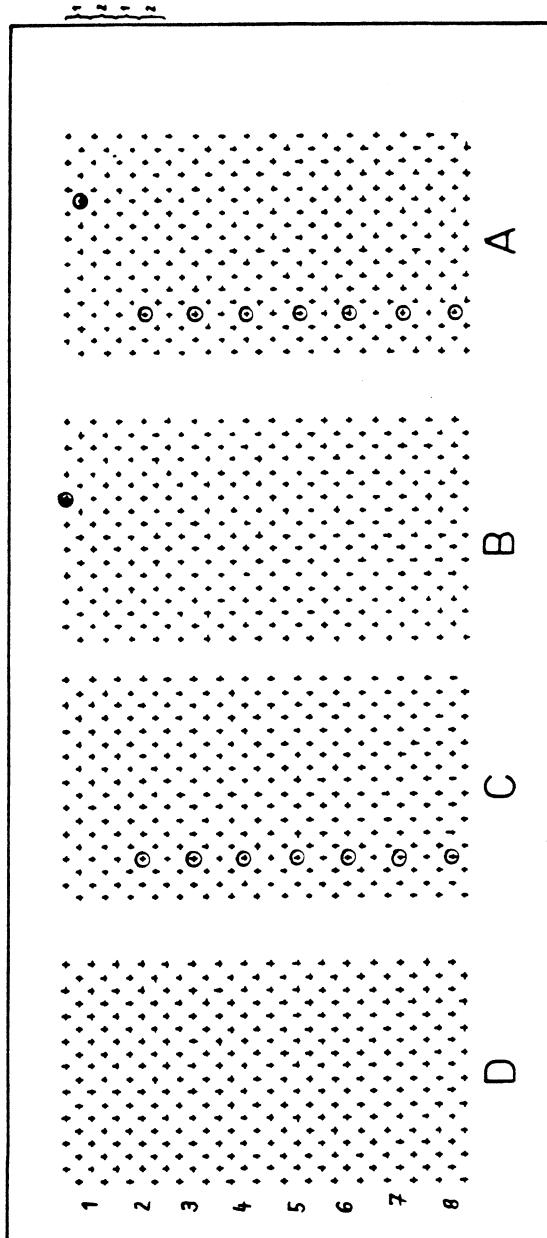


Bild 4.4

4.6 PROZESSOR CONTROL REGISTER PCR

Das Prozessor Control Register PCR dient der Steuerung der Prozessors sowie einiger für Maintenance-Programme relevanter Funktionen:

- Anlaufbit (bit 0). Dieses bit steht - wie alle PCR-bit - nach dem Prozessor-Anlauf auf "0" und steuert damit den in Kap. 3 beschriebenen Anlaufzustand. Durch Beschreiben dieser bit-Position mit "1" gelangt der QU68000 in den normalen Betriebszustand (Ready Write-bit).
- BUS-INIT (bit 1). Die Aktivierung dieses Bits (= 1) bewirkt das programmgesteuerte Anschalten des BINIT-L-Signals am Q-BUS. Dieses Bit wird statisch auf BINIT-L abgebildet (Read/Write).
- BUS-EVENT-ENABLE (bit 2). Wird dieses Bit eingeschaltet (=1), dann wird der 20ms-Netz-Clock aktiviert: alle 20 ms wird auf Interrupt-Ebene 3 eine Programmunterbrechung ausgelöst (Netz-Synchron) (Read/Write).
- INTERRUPT SYSTEMENABLE (bit 3). Das gesamte Interrupt-System des QU68000 wird erst eingeschaltet, wenn dieses bit=1 gesetzt wird (Read/Write).
- FORCE PARITY ERROR (bit 4). Bei Aktivierung dieses Bits wird in den Speichermodulen (S-Bus und Q-Bus) bei Schreib-Zyklen ein Parity-Fehler erzwungen: Diese Eigenschaft (die im Q-Bus-Protokoll definiert ist) wird für Diagnose-Zwecke verwendet (Überprüfung der Parity-Erzeugung und -Prüfung in den Speichern) (Read/Write).
- DMR/DMG (bit 5). Auch dieses Bit dient Wartungszwecken, mit dem Setzen (= 1) wird am Q-Bus ein DMA-Request erzwungen. Die Reaktion kann durch Einlesen des bit 5 (Read-Funktion: DMA-Grant) überprüft werden; auch die Funktion der DMA-Synchronisierung kann damit geprüft werden - bei DMA-Request müssen Q-Bus-Cycles zu Time-Out führen (Read: DMG/Write:DMR).

- HALT (bit 7). Das HALT-Signal am Q-Bus führt zu einer Programmunterbrechung (Ebene 6); die Überprüfung dieses bit 7 erlaubt es dem ODT-Programm, bei Verschwinden des HALT-Signals am Q-Bus in den normalen Programm-Mode zurückzukehren (Read only).
- ENS/POWERFAIL (bit 6). Dieses Bit hat 2 verschiedene Bedeutungen: (write/read) ENS (write). Durch aktivieren dieses Bits kann erzwungen werden, daß alle externen Speicherzugriffe (S-Bus u. Q-Bus) über den Q-Bus ausgeführt werden.

POWERFAIL (read): dieses Readonly-bit zeigt an, daß vom Prozessor die Powerfail-Bedingung entdeckt wurde. Das Ereignis führt zu einem NON-Maskable Interrupt (NMI, Ebene 7), jedoch kann die Fortsetzung des normalen Programmablaufs erst erfolgen, wenn das Bit wieder verschwunden ist (Negieren der Powerfail condition = Restart) (Read only).

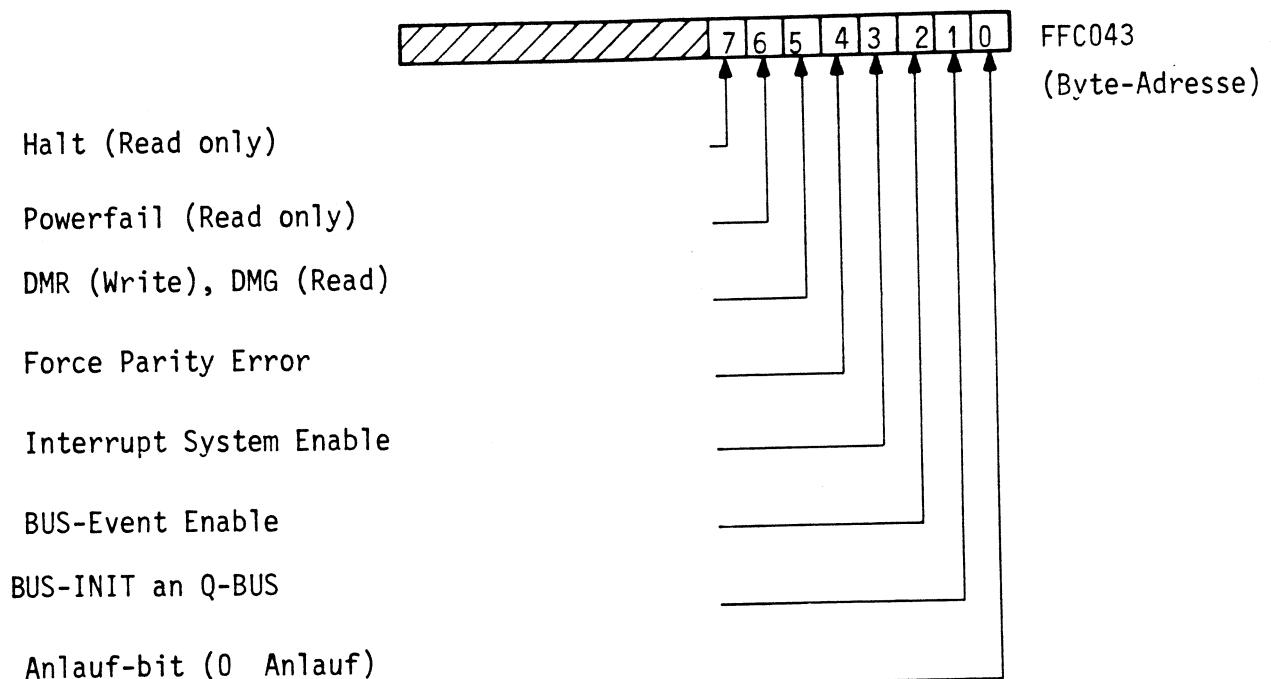


Bild 4.5 PROCESSOR CONTROL REGISTER PCR

4.7 ERROR STATUS REGISTER ESR

In die Kategorie der Interrupts gehört der Bus-Error des QU68000, welcher durch drei prinzipielle Ereignisse ausgelöst werden kann:

- Einen Timeout-Fehler, wenn ein adressiertes Speicherwort (Peripherie-Register) nicht innerhalb einer vorgegebenen Zeit reagiert.
- Den Parity-Fehler, welcher bei Entdecken einer Parity-Fehlers in einem Speicherwort ausgelöst wird und
- Verletzung von Zugriffsprivilegien, welche im Speicherverwaltungssystem entdeckt werden.

In der Beschreibung des MC68000 sind die im Falle eines Bus-Errors auf dem Stack abgelegten Informationen ausführlich beschrieben; zusätzlich wird auf der Prozessorkarte QU68000 ein Fehlerstatuswort (ESR = Error status Register; vgl. Bild A6) bereitgestellt, in welchem folgende Bit-Positionen definiert sind:

- Bit 0: Timeout-Fehler. Die Zeitüberwachung zeigt im allgemeinen den Versuch der Adressierung einer nicht vorhandenen Speicher- oder E/A-Registerzelle an.
- Bit 1: Memory-Paritätsfehler. Die Zeitverhältnisse beim QU68000 erlauben es - bei optimaler Nutzung der Zugriffszeiten - leider nicht, den Paritätsfehler in dem Zyklus anzuzeigen, in dem er beobachtet wurde: Dieser BUS-Error wird vielmehr im nächsten BUS-Zyklus angezeigt, so daß zwar keine Fehler-Auswirkungen auftreten, die Software-Analyse der fehlerhaften Speicherzelle aber erschwert wird.

- Bit 2: Überschreiten der max. zulässigen Segmentlänge, Fehlerbedingungen aus der MMU.
- Bit 3: Zugriff auf ein Segment im User-Mode, für welches dieser Zugriff nicht erlaubt ist (aus MMU).
- Bit 4: Befehlsfetch auf ein Segment, welches nicht als Code-Segment definiert ist (aus MMU).
- Bit 5: Datenzugriff auf ein Code-Segment (Kopierschutz, aus MMU).
- Bit 6: Schreibzugriff auf ein Segment, welches als Nur-Lese-Segment definiert ist (aus MMU).
- Bit 7: page-fault-Anzeige. In der 2. MMU-Stufe wird auf eine Seite zugegriffen, der keine Kachel zugewiesen wurde.

Die letzten sechs Status-Bit sind aus der MMU abgeleitet. Das Statuswort zeigt im übrigen die Fehlerbedingungen des jeweils letzten BUS-Errors an.

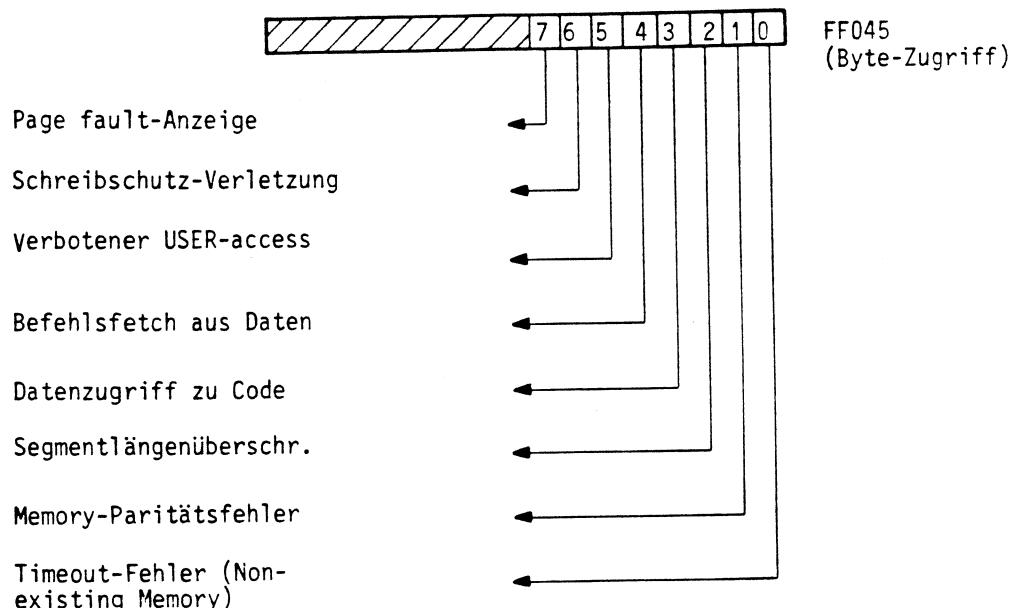


Bild 4.6 BUS-ERROR STATUS REGISTER ESR

5. SPEICHERVERWALTUNG MMU

Das UNIX-Betriebssystem verlangt eine Speicherverwaltungseinheit MMU (Memory-Management-Unit), welche in der Lage ist, den prozeßspezifischen logischen Adreßraum in den physikalischen Adreßraum des Speichers zu übersetzen und außerdem bestimmte Schutzmechanismen bereit zu stellen. Die bei der Firma Motorola in Entwicklung befindliche MMU für den M68000/3/ ist für diese Aufgabenstellung nicht optimal geeignet. Es war deshalb notwendig, eine eigene Speicherverwaltungseinheit zu entwerfen und durch MSI- und LSI-Schaltkreise zu implementieren. Diese MMU übernimmt folgende Aufgaben:

- a) Die Übersetzung der logischen Adressen in den physikalischen Adreßraum. Für die Implementierung von virtuellen Systemen besteht zusätzlich die Notwendigkeit, dazwischen einen virtuellen Adreßraum zu definieren, was zu einer 2-stufigen Adreßumsetzung führt.
- b) Der Zugriff auf bestimmte Speicherbereiche (Segmente) ist mit Zugriffsrechten ausgestattet, deren Verletzung von der MMU entdeckt und als Abbruchkriterium für den Prozeß an den Prozessor weitergegeben werden muß.
- c) Für die Implementierung des virtuellen Speichers besteht die Notwendigkeit einer Page-Fault-Behandlung; der Prozeß, der diesen Page-Fault ausgelöst hat, muß also solange suspendiert werden bis die fehlende Seite in einem reellen Speicher zur Verfügung steht. Mit der heutigen Variante des MC68000 ist eine konsistente Behandlung dieses Zustands noch nicht möglich, eine der nächsten Versionen (MC 68010) wird jedoch den dazu notwendigen Abbruch mitten in der Befehlausführung unterstützen.

Bild 5.1 zeigt schematisch diesen 2-stufigen Umsetzungsprozeß:

- Zunächst erfolgt die Umsetzung der vom MC68000-Prozessor angelieferten logischen Adresse, welche für jeden Prozeß identisch ist, in einen gemeinsamen virtuellen Adreßraum (mit unterschiedlichen virtuellen Adressen für jeden logischen Prozeß). Diese Abbildung erfolgt über 16 Segmente, wobei die Segment-Auswahl über die höchstwertigen 4 Adreß-Bit (20 bis 23) getroffen wird. Die maximale Segment-Größe beträgt 1 MByte. Jedem Segment kann dabei eine Reihe von Zugriffsrechten eingeräumt werden, wobei die Verletzung dieser Zugriffsrechte zu einem Abbruch des Bus-Zyklus führt (Bus-Error).
- Die zweite Abbildung entspricht der klassischen Seiten-Kachel-Tabelle. Sie wird im Wesentlichen über einen schnellen Halbleiterspeicher realisiert, in welchem für jede der 4 K-großen Seiten die zugehörige Kachel eingetragen werden kann. Bei fehlendem Bezug (nil) erfolgt in der ersten Implementierung ebenfalls ein Abbruch des Bus-Zyklus (Bus-Error). Bei der 68010 - Version des MC68000-Prozessors wird die dann verfügbare Page-Fault-Eigenschaft herangezogen. Zusätzlich existieren für jede Seite zwei Status-Bit (accessed, modified), welche bei entsprechenden Zugriffen auf diesen Seiten verändert werden. Mit dem Neubesetzen einer Seiten-Kachel-Zuweisung werden diese Status-Bit auf 0 gesetzt.

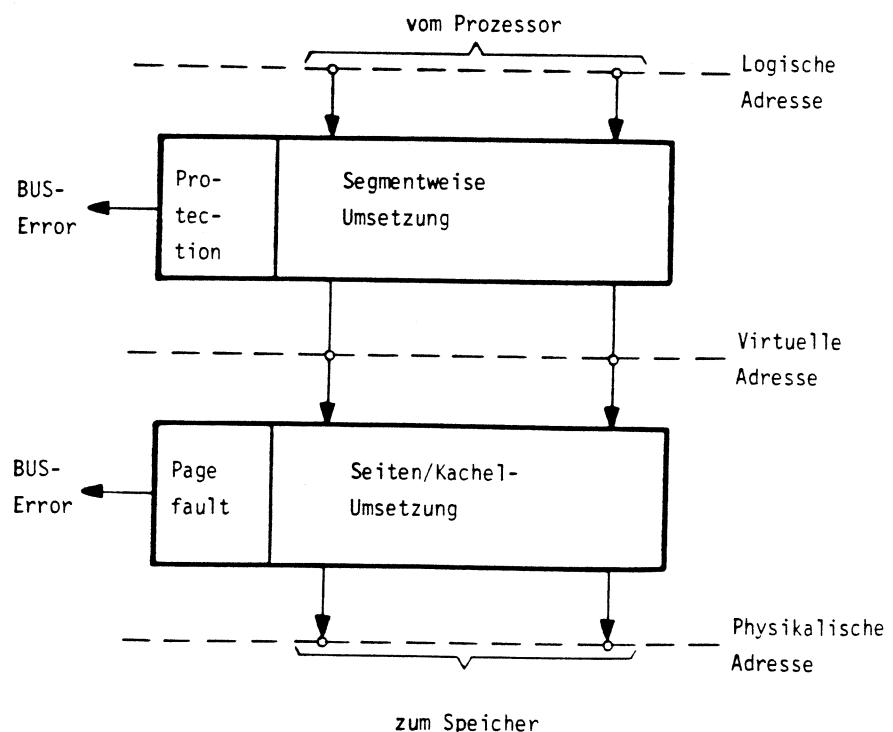


Bild 5.1 2-STUFIGE ADRESS-UMSETZUNG

5.1 UMSETZUNG VON LOGISCHE IN VIRTUELLE ADRESSEN

Das hier gewählte Prinzip orientiert sich stark an der bewährten Speicherverwaltungseinheit der PDP-11-Architektur, welche auch von verschiedenen 16-bit-Mikroprozessoren kopiert wurde (z.B. Z8000) und auch beim PERKEO-Prozessor/6/ Einsatz fand. Daraus resultiert eine Segmentierung des logischen Adressraums.

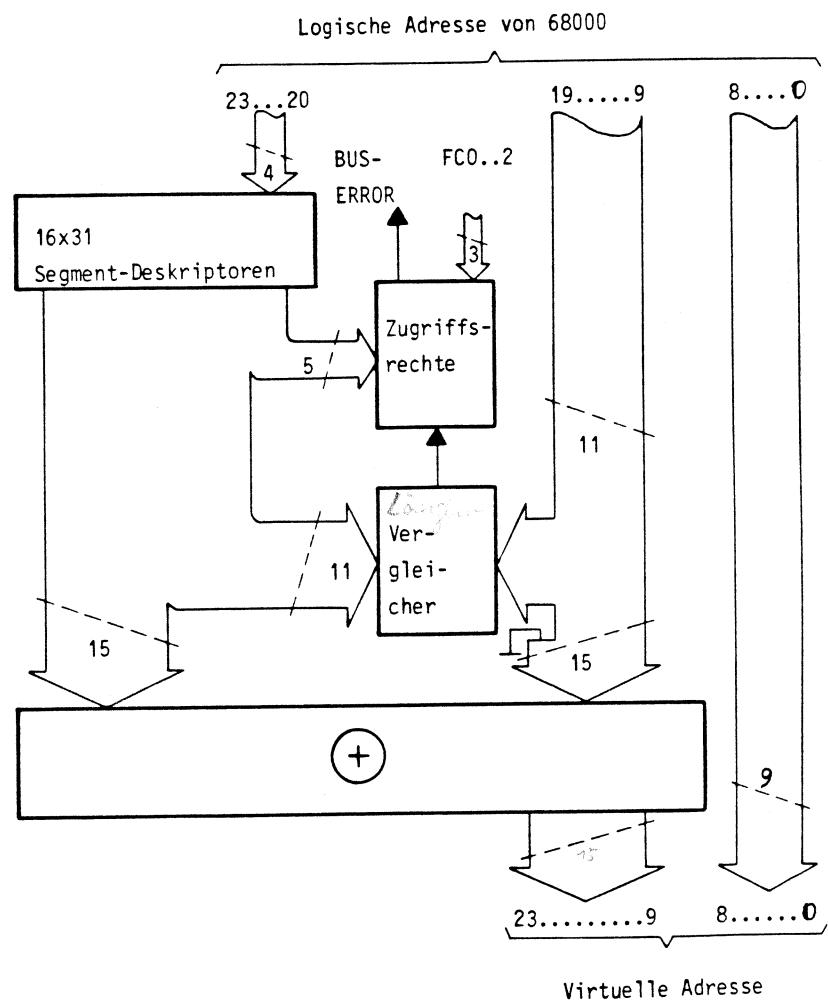


Bild 5.2 UMSETZUNG LOGISCHE/VIRTUELLE ADRESSE

Bild 5.2 zeigt die prinzipielle Struktur der ersten Adress-Umsetztstufe.

- Die vom Prozessor angelieferten höchstwertigen Adressbits 20 bis 23 wählen eines von 16 Segmenten aus, welche eine maximale Länge von 1 MByte haben kann. Die Auswahl der einzelnen Speicherstelle innerhalb des Segments erfolgt über die Adressbit 0 bis 19.

- Zu jedem Segment gehört ein Basis-Adreßregister, dessen Inhalt zu der Segment-relativen Adresse (Prozessor-bit 0 bis 19) addiert wird, um die virtuelle Adresse zu erhalten. Jedes Basis-Adreßregister definiert dabei die Positionen 9 bis 23 (insgesamt 15 bit), während die niedrigerwertigen 9 bit (0 bis 8) der Basis-Adresse als "0" angenommen werden. Dies bedeutet, daß die einzelnen Segmente in Einheiten von 512 byte an einer beliebigen Stelle in den virtuellen Adreßraum eingeschoben werden können.
- Die maximale Segmentlänge kann in Einheiten von 512 byte (256 16-bit-Worten) eingestellt werden. Da die max. Segmentlänge 1 MByte beträgt, wird zur Kennzeichnung der Segmentlänge ein 11 bit-Register benötigt.
- Ein zusätzliches Bit zeigt an, ob das Segment in konventioneller Weise von unten nach oben weist, oder - wie dies beispielsweise beim Stack gewünscht ist - von oben nach unten. Bei einer Überschreitung (bzw. Unterschreitung) der max. zulässigen Segmentlänge wird ein Bus-Error ausgelöst, dessen Ursache in dem bereits in Kap. 4 vorgestellten Statuswort vermerkt ist.
- Zusätzlich sind 4 Steuerbit vorgesehen, welche die Zugriffsrechte zum Segment regeln. Diese sind:
 - o RDWR: Wenn gesetzt, Segment beschreibbar
 - o NOUA: Wenn gesetzt, Zugriffe im User-Mode verboten
 - o NOCD: Wenn gesetzt, Code-Fetch verboten
 - o CDON: Wenn gesetzt, nur Code-Fetch erlaubt (Kopierschutz)

Verletzung dieser Zugriffsrechte, welche durch Verknüpfung mit dem zum jeweiligen Bus-Zyklus gehörenden Funktionscode des MC68000-Prozessors ermittelt wird, führen wiederum zu einem Bus-Error, wobei die Art der Zugriffsrecht-Verletzung im Fehler Statuswort ESR (Error Status Register) vermerkt ist.

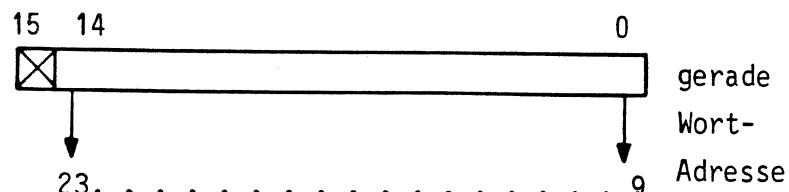
Bild 5.3 gibt eine Beschreibung des Segment-Descriptors wieder, durch welchen jedes von den 16 Segmenten definiert ist. Es handelt sich um zwei 16-Bit-Worte, von denen das erste die Basis-Adresse für die Adreß-Translation enthält (Bit 0 bis 14), während in dem zweiten die Segmentlänge und Segmentrichtungs-Definition sowie die Zugriffsrechte vermerkt sind. Durch das Schreiben der beiden Worte ist jeweils ein Segment festgelegt.

Alle Fehler und Zugriffsrechtverletzungen, welche in der ersten Adreß-Umsetzphase entstehen können, führen zu einem Bus-Error, der - wie in der MC68000-Prozessor-Beschreibung im Detail zu lesen ist/² - zur Speicherung folgender Zustandsinformation auf dem Supervisor-Stack führen:

- Information über die Art des Bus-Zyklus, in welchem der Fehler auftrat (Funktionscode, Information über Art des Zyklus usw.)
- Die logische Adresse, mit welcher der Bus-Zyklus ausgeführt werden sollte.
- Der Inhalt des Instruktionsregisters bei dem ausführenden Befehl.
- Das Status-Register des Prozessors.
- Den Programmzähler.

Zusätzlich ist im Fehler-Statuswort - ESR - bei Adreß-Umsetzfehlern die genaue Ursache vermerkt.

Die 16 Segment-Descriptor sind über insgesamt 64 Byte-Adressen im lokalen I/O-Adresse Raum (vgl. Bild 3.3) erreichbar.



Basisadresse (* 512)

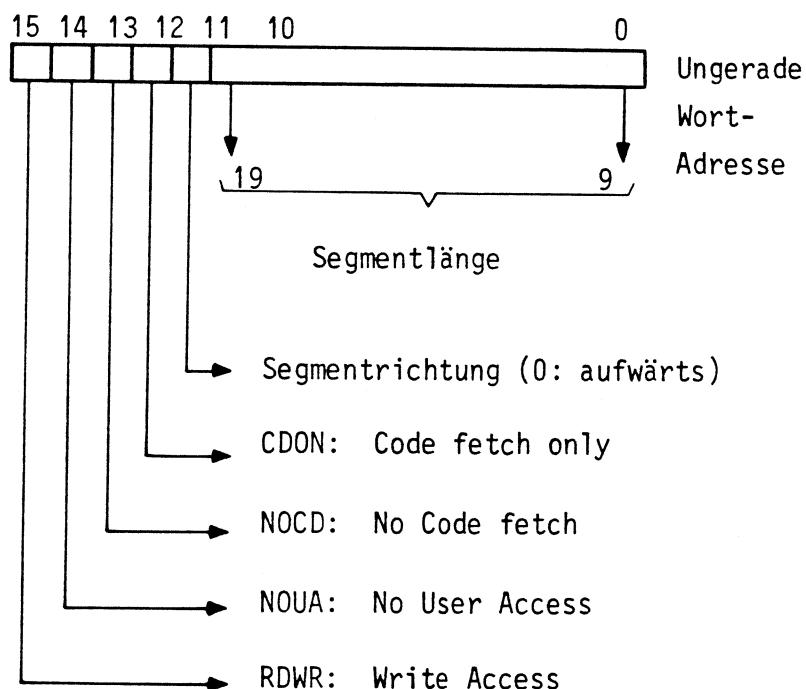


Bild 5.3 SEGMENT-DESKRIPTOR

Bild 5.4 zeigt die Adreß-Anordnung. Hier muß allerdings betont werden daß die Segment-Deskriptoren nur beschrieben werden können: ein Einlesen der aktuellen Segment-Deskriptor-Inhalte ist nicht möglich (wahlweise muß mit einem Speicherabbild gearbeitet werden).

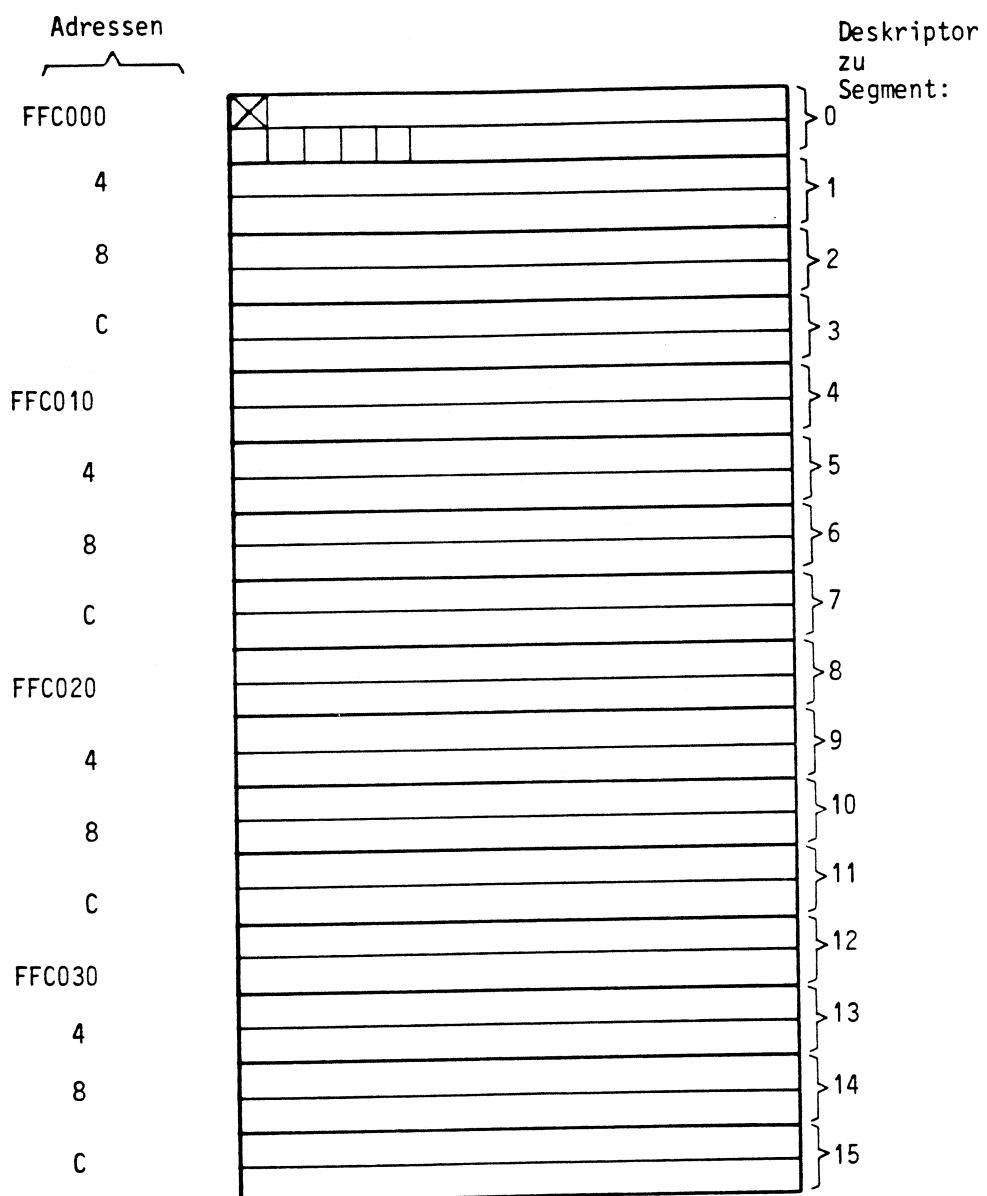


Bild 5.4 : ADRESSIERUNG DER SEGMENT-

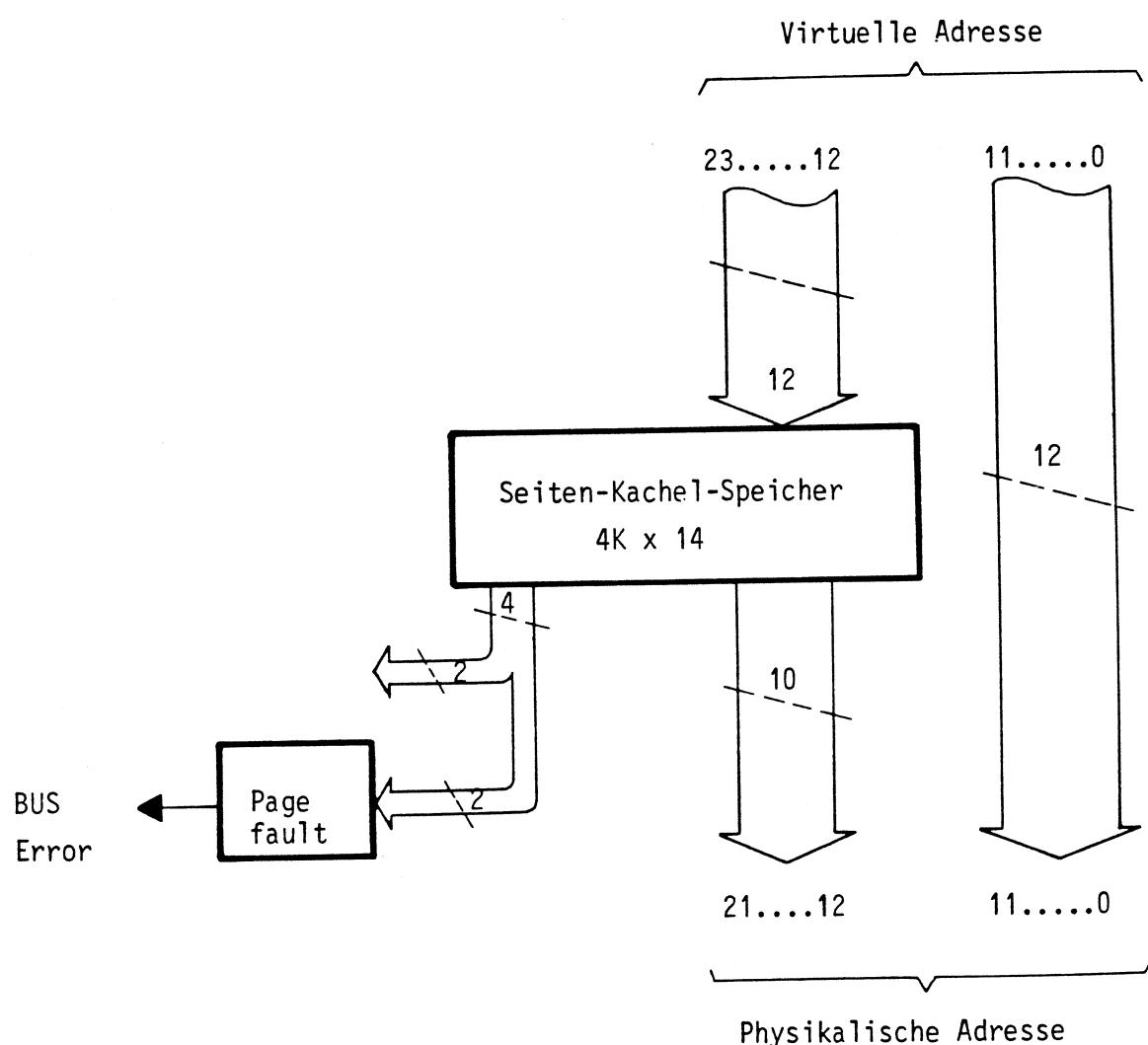
DESKRIPTOREN (WRITE ONLY)

5.2 UMSETZUNG VON VIRTUELLEN IN PHYSIKALISCHE ADRESSEN

Hier findet die klassische Umsetzung von Seiten in Kacheln über eine Seitenkachel-Tabelle in einem besonders schnellen Halbleiter-Speicher statt. Das Ziel dieser Umsetzung besteht darin, einen großen, virtuellen Adreßraum auf einen kleineren physikalischen Speicher abzubilden. Wird der Versuch unternommen, auf eine gerade nicht im Speicher befindliche Seite zuzugreifen, so führt dies zu einem Page-fault. Wie bei virtuellen Speichersystemen üblich, muß dann eine Kachel bereitgestellt werden, in welche aus einem Externspeicher die gewünschte Seite eingeladen wird, worauf das Programm fortgesetzt werden kann. Leider gibt es beim MC68000-Prozessor noch keine echte Page-fault-Behandlung, sie ist allerdings für den Typ MC68010 (1983) angekündigt. Die Konstruktion der MMU ist so, daß diese korrekte Pagefault-Behandlung später problemlos eingefügt werden kann. Vorläufig führt ein Pagefault allerdings zur Auslösung eines Bus-Errors, der voraussichtlich nicht in jedem Fall eine Rekonstruktion des Prozessor-Zustands ermöglicht, wie sie für eine korrekte Pagefault-Behandlung erforderlich wäre.

Es sei darauf hingewiesen, daß die Seitenkachel-Tabelle in Form von 5 integrierten Schaltkreisen auf dem QU68000-Board als Option eingefügt werden kann. Für die Unix-Version (nicht virtuell) werden diese Speicher durch eine 1 : 1-Verdrahtung (Adressen zu Daten) ersetzt.

Bild 5.4 zeigt die Struktur der zweiten Adress-Umsetzung mit Hilfe der Seitenkache-Tabelle.



5.5 STRUKTUR DER SEITEN/KACHEL-UMSETZUNG

Als Seiten- und Kachelgröße wurden 4 K byte festgelegt. Dies bedeutet:

- Bei einem virtuellen Adressraum von 16 MByte, daß max. 4096 Seiten a 4 K byte vorzusehen sind und
- bei einem physikalischen Adressraum von 4 MByte, daß 1024 Kacheln a 4 K byte definiert werden müssen.

Das führt zu einem Umsetspeicher von 4 K x 10 bit, in welchen für jede Seitennummer eine Kachelnummer eingetragen werden kann.

Zusätzlich gibt es 2 bit, welche angeben, ob

- eine Seite überhaupt definiert, also im virtuellen Adressraum festgelegt ist bzw.
- real im Speicher vorhanden ist, oder ob im Moment keine feste Zuordnung zu einer Kachel vorliegt

Für jede Seite gibt es zwei Zustandsbit, welche die Ein- bzw. Auslagerungsstrategien für den Kachelaustausch unterstützen:

- accessed, also ein Bit, welches andeutet, ob auf diese Seite bereits ein Zugriff erfolgt ist und
- modified, ein Bit also, welches einen Schreibzugriff auf eine Seite markiert, woraus im allgemeinen die Notwendigkeit des Wiederauslagerns der Kachel folgt.

Diese beiden Statusbit werden beim Beschreiben eines Seitendeskriptors auf 0 gesetzt, bei einem entsprechenden Bus-Zyklus (Zugriff, Schreibzugriff) auf 1 gesetzt.

Daraus resultiert für jede Seite ein Seitendeskriptor, (Page-Descriptor PD) wie er in Abbildung 5.6 wiedergegeben ist. Neben der 10-bit großen Kachelnummer enthält das 14-bit-Wort die beiden Steuerbit, welche den Definitionszustand der Seite charakterisieren sowie die beiden Statusbit, welche über evtl. Zugriffe Auskunft geben. Diese Seitendescriptor können gelesen und geschrieben werden, wobei die bit 10 und 11 auch "spontan", also durch entsprechende Zugriffe in "1" umbesetzt werden können.

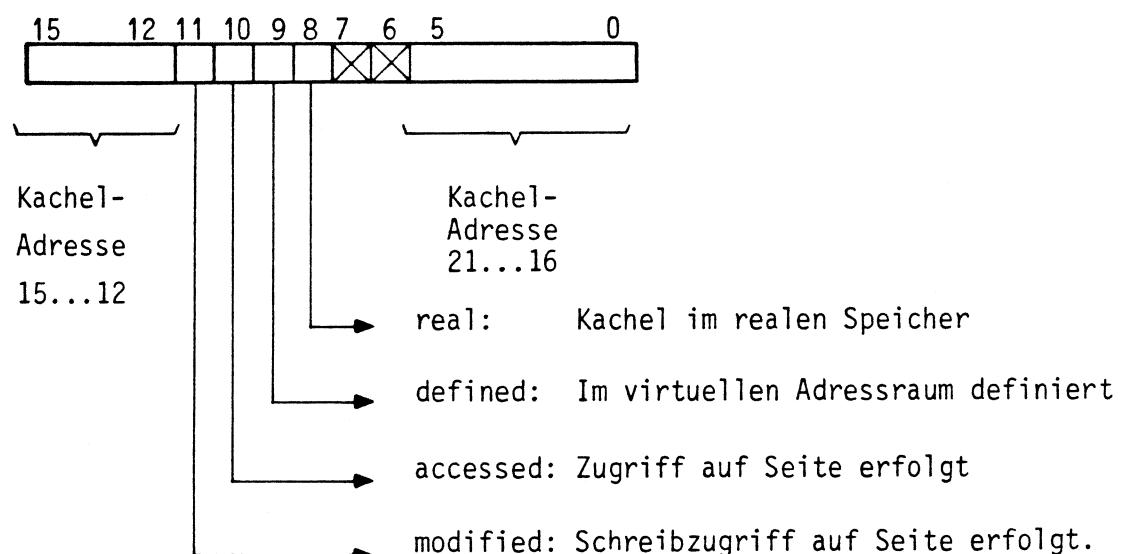


Bild 5.6 DEFINITION DES PAGE-DESKRIPTORS

Das Lesen und Beschreiben der Seitendescriptor erfolgt unter Verwendung des Segment-Descriptors 15, wodurch der Einsatz zusätzlicher, zeitraubender Multiplexer-Hardware vermieden werden kann. Bild 5.7 zeigt diesen "Trick", der unter Verwendung einer Speicheradresse in dem im Supervisor-Mode speziell definierten oberen 32 K byte-Block möglich ist. MC68000-Zugriffe in diesem Bereich führen nicht zu einem Zugriff über die MMU, vielmehr wird dort ein lokaler Speicherzugriff angezeigt. Anderseits erfolgt natürlich über den hardwaremäßige vorhandenen Addierer eine Adreßumsetzung, so daß durch Addition der im Segmentdescriptor 15 enthaltenen Basisadresse mit der festgelegten Adresse (252) jede beliebige Seitenadresse erzeugt werden kann. Damit ist die Adresse für den Seitenkachelspeicher fixiert. Ein Lese-/Schreibzugriff über die festgelegte Adresse PD ermöglicht das Auslesen und Beschreiben eines Seitendescriptor. Diese Zugriffe erfordern also zwei Operationen:

- Zunächst das Laden des Basis-Adreßregisters von Segment 15 mit der Adresse

$$(SN - 252) \times 8,$$

wenn SN die gewünschten Seitennummer ist. Durch die Subtraktion mit 252 wird der Effekt der konstanten Adresse (Bild 5.6) kompensiert, der Faktor 8 gibt den Unterschied zwischen der Seitengröße (4K) und den kleinsten Einheiten für Segment-Definitionen (512 byte) wieder.

- Lese- und Schreiboperationen auf die Adresse PD ermöglichen nun den Zugriff zum Seitendescriptor.

Zugriffsfehler (Zugriffsversuche auf nicht definierte oder nicht real vorhandene Seiten) führen, wie bereits erwähnt, zu einem Bus-Error, wobei die Ursache wiederum im eingetragenen Fehler-Status-Wort ESR registriert sind.

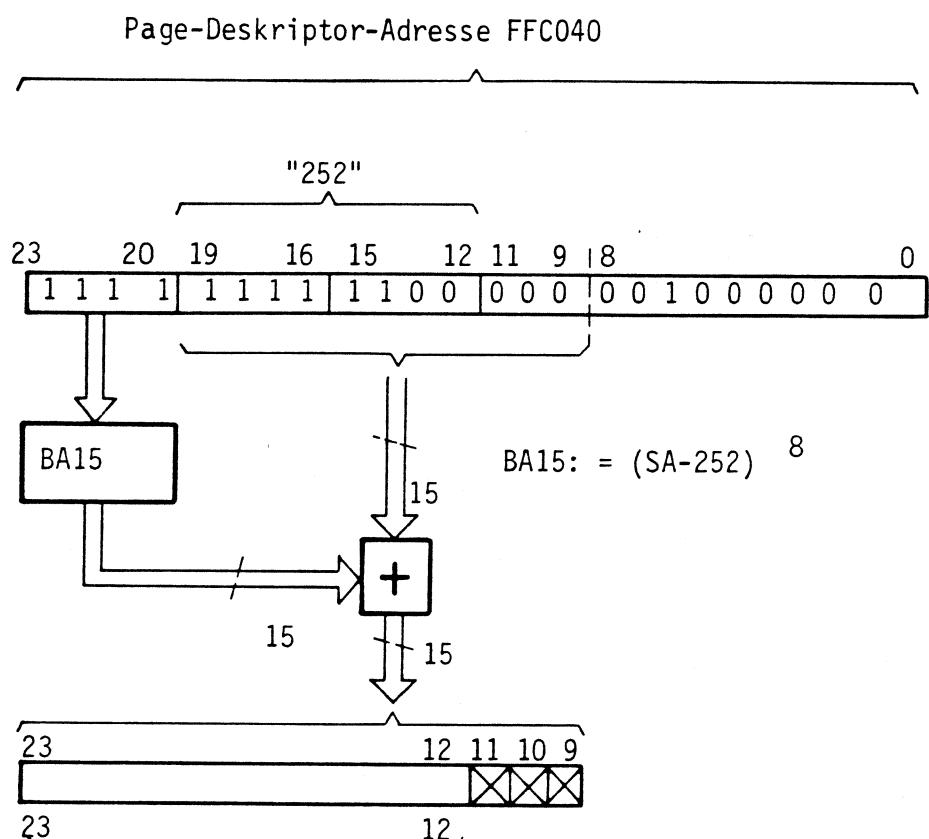


Bild 5.7: ADRESSIERUNG DES SEITEN/KACHEL-SPEICHERS ÜBER

DIE BASIS-ADRESSE DES SEGMENTDESKRIPTORS 15

6. SPEICHERBUS (S-BUS)

Ein besonderes Problem ist die Fehlanpassung der Übertragungsgeschwindigkeiten am Q-Bus und der insbesondere bei den schnelleren Versionen des MC68000-Prozessors (12 MHz) möglichen Geschwindigkeiten. Der Q-Bus stellt hier einen absoluten Engpaß dar, der dazu führen würde, daß die Geschwindigkeit des Prozessors maximal zur Hälfte ausgelastet werden könnte (zahlreiche Wartezeiten). Dies führt zu der Notwendigkeit, zusätzliche lokale Speicher vorzusehen, welche nicht den Q-Bus zur Datenübertragung benötigen.

Die Aufgabe des Speicherbus (S-Bus) besteht darin, spezielle, schnelle Speicherkarten direkt mit dem QU68000-Prozessor zuverbinden. Es handelt sich um einen spezialisierten, nur zur Ansteuerung von Speichern geeigneten Bus; die Speichermodule sind reine Slaves, das bedeutet, daß die Adresse immer vom Prozessor angelegt wird, und der Speicher hat den dabei spezifizierten Auftrag (Lesen, Schreiben) auszuführen. Auch irgendeine spontane Meldungen (Interrupts) sind in diesem Bus nicht vorgesehen.

Die Struktur des Speicherbusses ist an die dynamischen 64 K-Speicherbausteine mit Adreßmultiplexing angepaßt. So wird die 16-bit-Adresse, welche eine Speicherzelle im 64 K-Baustein auswählt, in zwei 8-bit-Einheiten übertragen: Zunächst die 8-bit-LSB-bit, welche direkt vom MC68000-Prozessor geliefert werden, welche also nicht über die MMU geführt werden müssen. Dies führt dazu, daß die Speicherzykluszeit dadurch verkürzt werden kann, daß die Row-Address schon zur Speicherlogik gelangen kann, bevor die MMU ihren Umsetzvorgang beendet hat. Dadurch wird es möglich, auch bei schnellen MC68000-Prozessoren (12MHz) mit 1 bis 2 Wartezeiten auszukommen.

Der Refresh-Vorgang für die dynamischen Speicher erfolgt über Refresh-Schaltkreise, welche dezentral auf den einzelnen Speicherplatten untergebracht werden. (siehe Anhang: S-Bus Definition)

Auf diese Weise entsteht gewissermaßen eine CPU (Prozessor und Speicher), welche den Q-Bus als Ein/Ausgabe-Bus benutzt. Mit einer Dual-port-Struktur der privaten Speicher können auch DMA-Zugriffe direkt vom Q-Bus aus ausgeführt werden, ohne daß der Datenaustausch zwischen Prozessor und Speicher behindert wird. Die gesamte Peripherie ist an den Q-Bus angeschlossen (zahlreiche Controller sind hier auf dem Markt verfügbar), zusätzlich können natürlich beliebige Q-Bus-Speicher eingesetzt werden.

6.1 ADRESS- UND DATENBUSSIGNAL

Die Datenleitung SDO ... SD15 dienen dem Datenverkehr zwischen Prozessor und Speichereinheit. Die Signale sind "aktiv High" und werden über bidirektionale Treiber getrieben bzw. empfangen. (siehe Tabelle im Anhang.)

Die Adreßleitungen lassen sich in S-BUS Adresse (SA0 ... SA8) und Bank-Select (SBO ... SB4) aufteilen. Prozessorseitig werden die unteren 16 Adreßleitungen (AD1 ... AD16) gemultiplext. Dies verringert zum einen die Zahl der S-BUS-Signale, zum anderen kommt es den Konzept des S-BUS-Speichers entgegen (RAS,CAS).

Die oberen 5 Adressen (AD17AD21) werden dem Speicher als Bank-Select zur Verfügung gestellt. Bei einer Speicherkapazität von 256K Worten selektieren SB2 ... SB4 maximal 8 Speicherplatten, SBO ... SB1 wählen eines von vier 64K Modulen aus.

Um die angekündigten 256K * 1 dynamischen Ram-Bausteine einsetzen zu können, liefert der Prozessor das Signal SA8 , ein Multiplexsignal aus AD9 und AD18. Die Umschaltung zwischen beiden Betriebsarten erfolgt mittels einer Brücke.

6.2 STEUERSIGNAL

Der S-BUS besitzt 8 Steuersignale, von denen 6 prozessorseitig, und 2 auf dem Speicher generiert wurden.

Die Signale Read/Write (SR/W), Upper-/Lower Data Strobe (SUDS,SLDS) und Adress-Strobe (SAS) sind direkte MC68000-Signale, die über Lowaktive Treiber zum Speicher gehen. SUDS und SLDS dienen u.a. der Byte-Adressierung. SAS ermöglicht die Erkennung eines Read-Modify-Zyklus. SR/W gibt an, ob ein Schreib - oder Lesezyklus vorliegt.

Das Signal SSYNC signalisiert dem Speicher, daß die RAS-Adresse übernommen werden kann. Da BSSYNC in einer ganz bestimmten Prozessortaktphase aktiv wird (während S4 - ca. 20 ns nach positiver Taktflanke) und die Column-Adresse einen definierten Zeitraum später am Bus liegt, kann eine Delay-Line-Steuerung das Speichertiming übernehmen.

Als Antwort auf SSYNC gibt der Speicher das Acknowledge-Signal SRPLY. Dieses Signal verzögert sich bei Refresh sowie bei aktionen Q-Bus Speicherzugriff.

Eine Besonderheit stellt der MC68000-Befehl "TAS" dar. In diesem Fall wird ein nicht trennbarer Read-Modify-Write Zyklus ausgeführt. Prozessorseitig ist SSYNC sowohl bei Read als auch bei dem nachfolgendem Write aktiviert. Während bei der Phasen bleibt AS aktiv, um auf die Besonderheit dieses Zyklus hinzuweisen.

Mit SPGE wird auf dem Memory-Board ein Parity-Error erzeugt. Die Erkennung dieses bzw. eines tatsächlich auftretenden Paritätsfehlers erfolgt über das Signal SPE. Bedingt durch das MC68000 - Konzept wird der Fehler jedoch erst im nächsten Befehlszyklus erkannt.

6.3 SPEZIELLE SIGNALE

Bei der Adressierung einer Speichereinheit muß der Prozessor feststellen, ob S-BUS Speicher oder Q-BUS-Speicher angesprochen werden. Zu diesem Zweck werden die Memory-Present-Leitungen SMP0 ... SMP7 verwendet (Bild 6.1)

Korrespondieren die Kartenselekt-Signale SB2 ... SB4 mit der Schalterstellung auf dem S-BUS-Board, wird das Signal S-BUS aktiv.

Da es sich bei den Leitungen SMP0 ... SMP7 um statische Leitungen handelt, kann man diese als "Quasi-Masseleitung" ansehen. Diese Tatsache reduziert die Zahl der verwendeten S-BUS-Leitungen.

6.4 INTERFACEBAUSTEINE

Als Treiber/Empfängerbausteine dienen die Typen F 241, F245 sowie der schnelle AMD-Treiber 29828.

6.5 MECHANISCHES KONZEPT

Die Bilder A.9 - A.11 im Anhang zeigen die mechanische Realisierung des S-BUS: QU68050 und Dual-Port-Speicher erhalten je zwei 40 pol. 3M Stecker, über die mittels Flachbandkabel die Verbindung hergestellt wird.

Zur Erhöhung der Störsicherheit wird jede zweite Leitung auf Masse bzw. Quasi- Masse (vgl. 2.3) gelegt.

Die Steckerbelegung von J1 und J2 kann dem Anhang entnommen werden.

7. MULTIPROZESSOR-SYSTEME

Im Prinzip ist es möglich, einen zweiten oder weitere derartige Prozessoren mit privatem Speicher in ein Q-Bus-System einzufügen; hierzu muß man bestimmte Funktionen deaktivieren. So darf beispielsweise nur ein einziger Prozessor für den Q-Bus Arbiter spielen und auch die Peripherie-Interrupts sollten nur an einen einzigen Prozessor weitergeleitet und von diesem verarbeitet werden. Damit entsteht eine Konfiguration mit mehreren Prozessoren, welche einen gemeinsamen Ein/Ausgabe-Bus, den Q-Bus, benutzen.

Damit diese Anordnung ihre volle Leistung erreichen kann, ist es notwendig, die Prozessoren parallel zueinander mit ihren privaten Speichern arbeiten zu lassen und nur dann eine Unterbrechung (DMA-Request) vorzunehmen, wenn über den Q-Bus eine DMA-Übertragung in einen der privaten Speicher vorgenommen werden muß. Um dies konsistent zu ermöglichen, ist es erforderlich, für alle Prozessoren einen gemeinsamen Adreßraum bereitzustellen: Jede Speicherzelle im Mehrrechnersystem hat damit eine eindeutig identifizierbare Adresse. In der ersten Version wird die Multiprozessor-Fähigkeit des QU68000 nicht für die Anwender verfügbar sein.

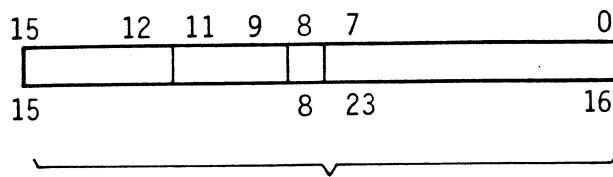
8. UNTERSCHIEDE DER VERSIONEN QU68050 UND QU68030

In der Version QU68030 ist der volle Funktionsumfang des QU68050-Prozessors nicht implementiert.

- a) Es gibt keinen S-Bus-Anschluß (und damit keine S-Bus Speichermodule).
- b) Es ist nur die Stufe 1 der MMU realisiert, (der QU68010-Prozessor, der virtuelle Adressierung voll unterstützt ist noch nicht verfügbar.)

Die 2. Stufe der MMU ist beim QU68050-Board nur dann sinnvoll ausnutzbar, wenn die Karte mit der MC68010-CPU bestückt ist. Für reine UNIX-Systeme kann aber die Seiten-Kachel-Tabelle entfallen.

Für die Definition des Zugriffs zum Page-Deskriptor ergibt sich aus diesem Unterschied folgende Änderung: Bild 8.1. macht deutlich, daß jetzt unter der PD-Adresse (FFC040) nur mehr gelesen werden kann, und zwar der Wert, der aus der 1. MMU-Stufe gelieferten Adresse (bit 9...23). Bei dieser Option entfällt also die Umsetzung von virtuellen in physikalische Adressen. Die "virtuelle Adresse ist gleich der physikalischen Adresse für den Q-Bus (S-Bus wobei die Adressbit 22 und 23 nicht verwendet werden.



(Physikalische) Adresse aus MMU

Bild 8.1 BEILEGUNG DES PD-REGISTERS (READ ONLY)

BEI DER VERSION QU68030 (ADRESSE FFC040)

9. ANHANG: Zusammenfassung

Im folgenden sind die Adressen und funktionellen Kurzbeschreibungen zusammengestellt, welche für den Programmierer des QU68000-Prozessors von Bedeutung sind.

Bild A1 gibt nochmals den Überblick über den 32 Kbyte-Bereich, der im Anlaufzustand 512-mal in den Adreßraum des MC68000 hineigespiegelt ist, und der im initialisierten Zustand die obersten 32 K des MC68000-16Mbyte-Adreßraums im Supervisor-Mode belegt. Die vier dort dargestellten Teilbereiche können wie folgt kurz charakterisiert werden:

- Der EPROM-Bereich in den ersten 8 KByte ist bezüglich der Speicherzellen 0 bis 7 durch den Initialisierungsvektor (Anfangsadresse, Anfangsstatus) des MC68000 belegt. Im übrigen entfällt das EPROM, wie erwähnt, das ODT zur Bedienung des Prozessors, ein Initialisierungsprogramm sowie ein minimales Diagnoseprogramm zur Analyse der onboard-Resourcen.
- Der zweite 8 KByte-Bereich ist mit 4 KByte RAM-Speicher belegt. Er dient zur Zwischenspeicherung von lokalen Größen sowie zur Aufnahme des am Anfang benötigten Supervisor-Stack, die 2. Hälfte des 8 Kbyte-Bereichs ist nicht belegt.
- Der dritte 8 KByte-Bereich ist zu einem kleinen Teil (128 byte) mit den Ein-/Ausgabe-Adressen der Onboard-Resourcen belegt, welche im folgenden näher beschrieben werden;
- Der vierte 8 KByte-Bereich entspricht der External-Page, dient also zur Adressierung der Ein-/Ausgabe-Register, die am Q-Bus angeordnet sind.

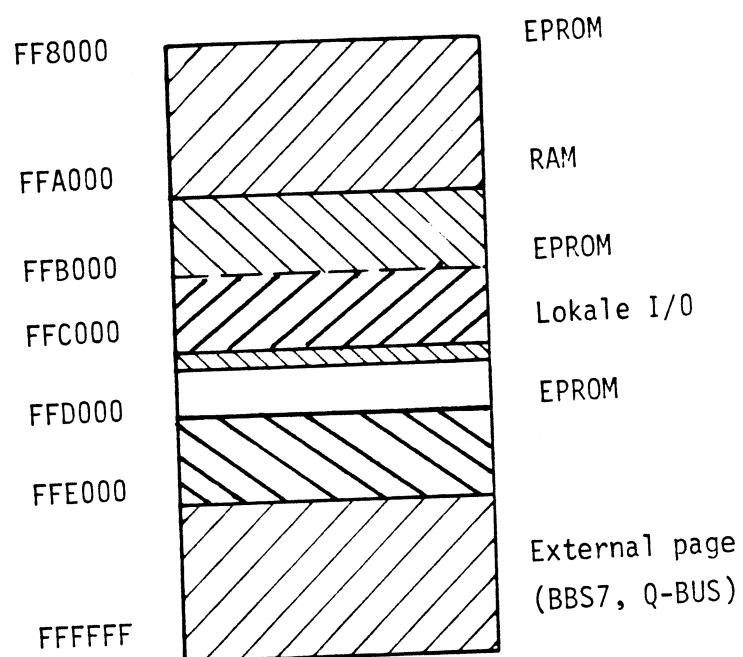


Bild A1: LAYOUT DES LOKALEN ADRESSRAUMS

Bild A2 zeigt im Detail die Adreß-Zuordnung in dem lokalen Ein/Ausgabe-Bereich (128 byte-Adressen). Folgende Funktionen werden dabei unterstützt:

- Die ersten 64 Byte dienen zur Adressierung der 16 Segmentdeskriptoren SD, wie sie in Abbildung A3 nochmals ausführlicher beschrieben sind.
- Eine Wortadresse dient zur Adressierung des Page-Deskriptors, über welchen die Seitenkachel-Tabelle geladen bzw. abgefragt werden kann. Bild A4 gibt die Struktur dieses Deskriptors wieder (vgl. Kap. 7 der Restriktionen in Version 1).
- Die nächste Wortadresse betrifft das Prozessor-Kontroll-Register PCR, dessen Struktur in Bild A5 näher wiedergegeben ist. Es dient zur Steuerung des Interrupt-Systems, der netzgetriebenen Echtzeituhr, des Anlaufzustands sowie der 68000-Programm-auslösbarer INIT-Pulses auf dem Q-Bus. Zusätzlich ist es - zu Diagnosezwecken - möglich, in Speichermodulen Paritätsfehler zu generieren und DMA-Requests auszulösen.
- Die nächste Wortadresse dient der Abfrage des Error-Statusregisters ESR, welches in Bild A6 nochmals wiedergegeben wird. Dieses Register zeigt immer den Fehlerzustand beim letzten Bus-Error.
- Vier Wortadressen dienen schließlich zur Definition der DMA-Extension-Register DER 0 bis DER 3, deren Belegung und Bedeutung aus Bild A8 hervorgeht.

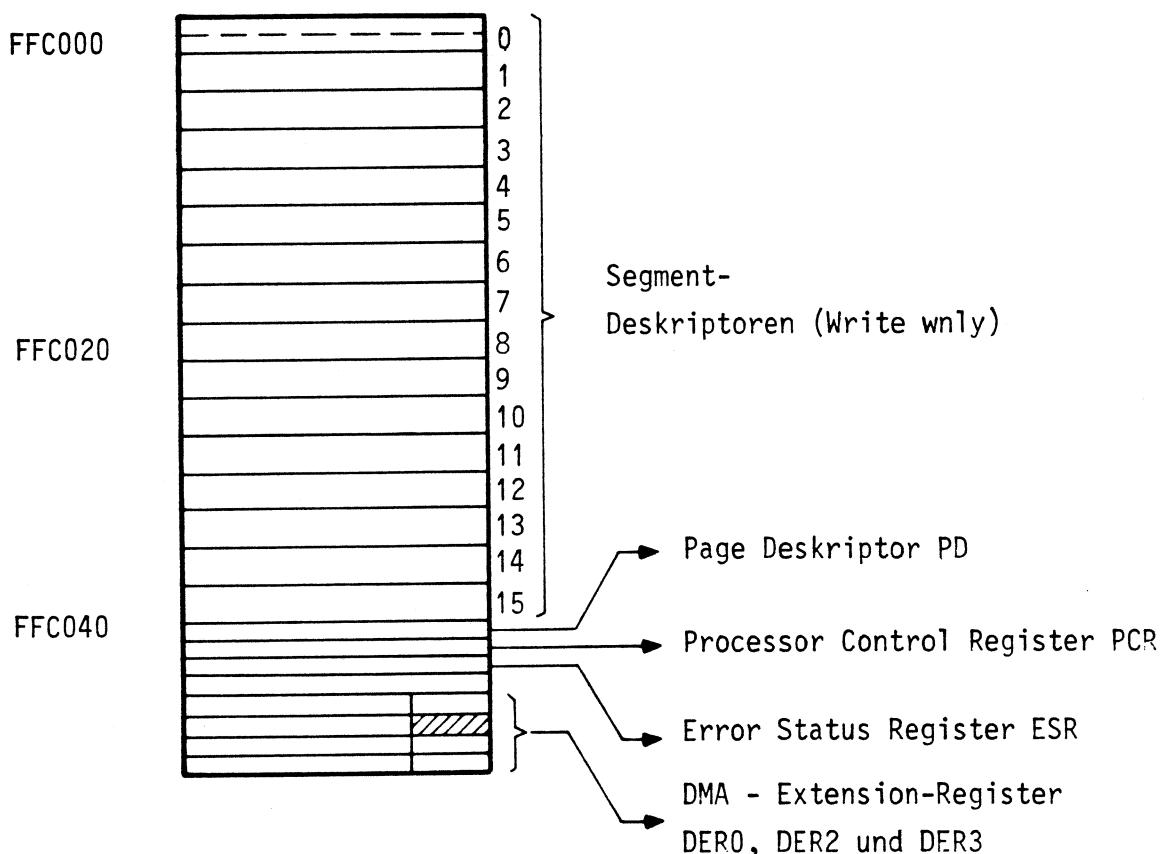


Bild A.2 ADRESSE DER LOKALEN I/O-REGISTER

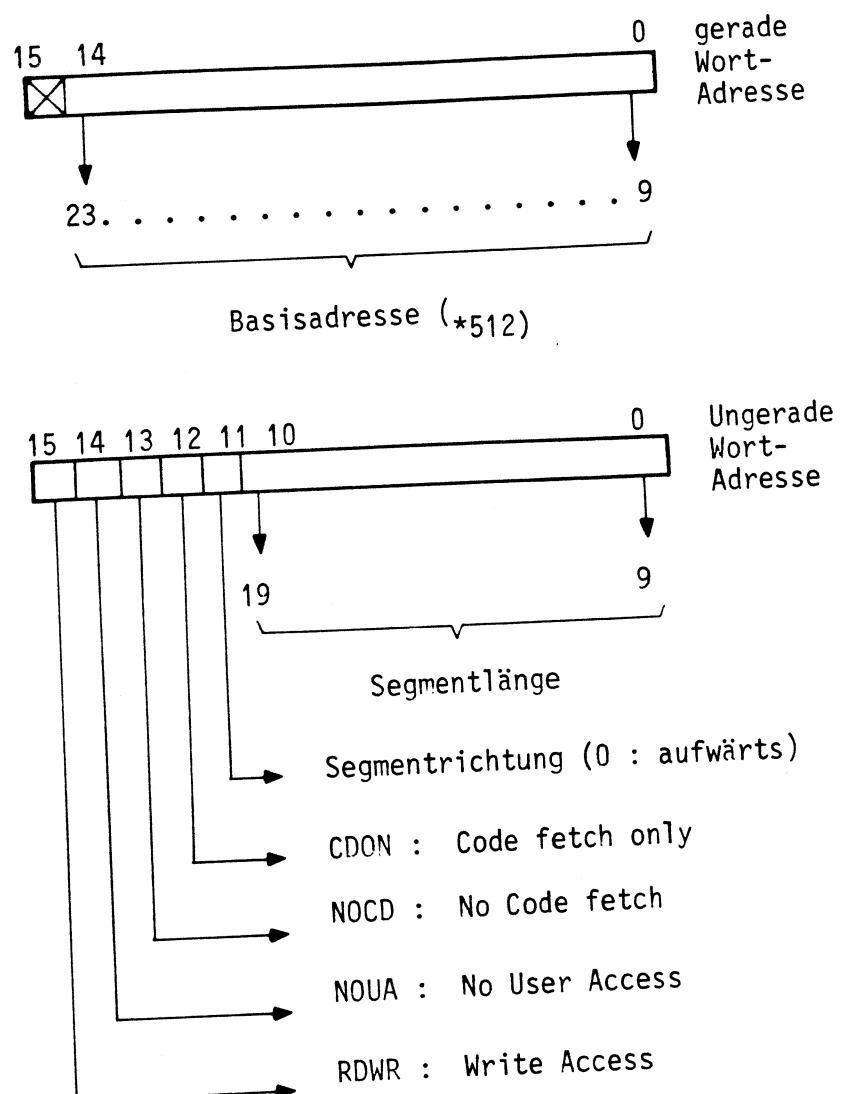


Bild A.3 SEGMENT-DESKRIPTOR

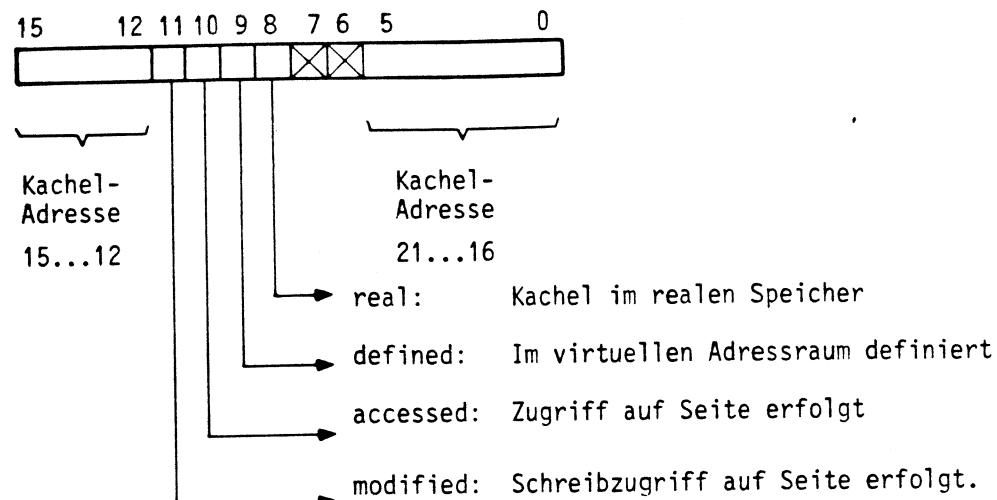


Bild A.4 DEFINITION DES PAGE-DESKRIPTORS

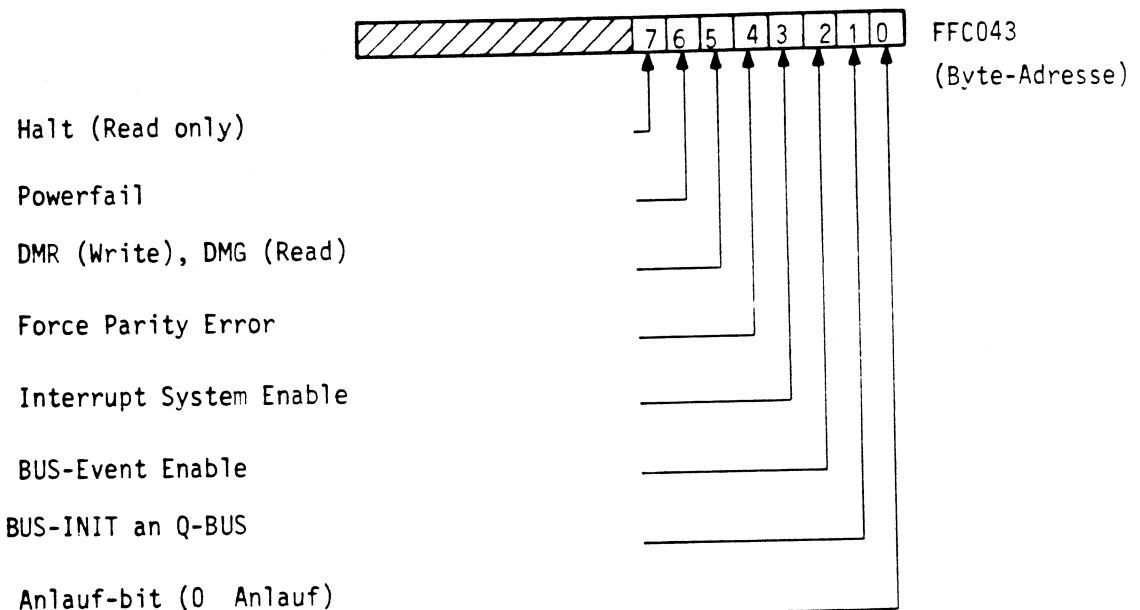


Bild A. 5 PROCESSOR CONTROL REGISTER PCR

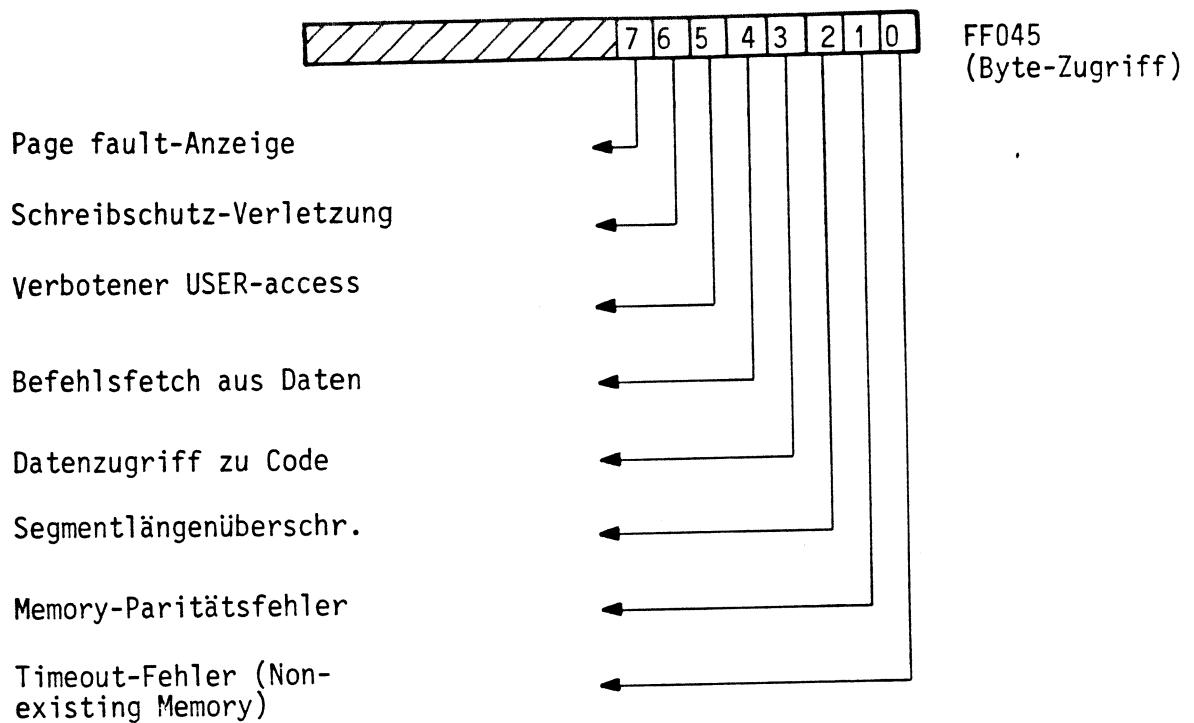


Bild 4.6 BUS-ERROR STATUS REGISTER ESR

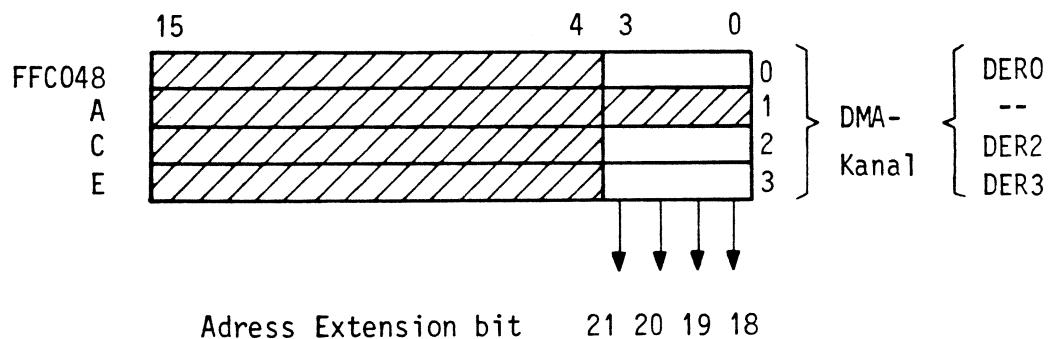


Bild A.7 DMA-EXTENSION-REGISTER DERO, 2 und 3

Schließlich soll nochmals auf die Belegung der sieben Interrupt-Ebenen des MC68000-Prozessors eingegangen werden:

- Level 1: Q-Bus-Level 4
- Level 2: Q-Bus-Level 5
- Level 3: Bus-Event, aus dem Netzclock abgeleitetes Echtzeit-Interrupt-Signal. Hier wird der - durch Autovektor erzeugte- Interrupt-Vektor 108 generiert.
- Level 4: Q-Bus-Level 6
- Level 5: Q-Bus-Level 7
- Level 6: Interrupt, der durch das Q-Bus-Signal B HALT ausgelöst wird. Der Autovektor wird hier die Interrupt-Vektor-Adresse 120 erzeugt.
- Level 7: Interrupt, der durch das Powerfail-Signal des Q-Bus (BPOKH) erzeugt wird. Per Autovektor wird hier die Vektoradresse 124 ausgelöst.

Die Interrupts, welche den Q-Bus-Ebenen 4 bis 7 entsprechen, erhalten ihre Interrupt-Vektoren über den Q-Bus, wobei aufgrund der MC68000-Eigenschaft die übertragenen Vektoradressen mit 4 multipliziert werden. Dies führt dazu, daß bei standardmäßiger Interruptvektor-Erzeugung von Q-Bus-Controlern nur jeder vierte Interruptvektor im MC68000-Adreßraum belegt ist.

ANHANG ZUR S-BUS-SPEZIFIKATION

		PROZ. SPEICHER
16	Datenleitungen	$\overline{\text{BSD}_0}$ - $\overline{\text{BSD}_{15}}$
9	S-Bus-Adresse	$\overline{\text{BSA}_0}$ - $\overline{\text{BSA}_8}$
5	Bank-Select	$\overline{\text{BSB}_0}$ - $\overline{\text{BSB}_4}$
1	Read/Write	$\overline{\text{BSR} \downarrow \overline{W}}$
1	Upper Data Strobe	$\overline{\text{BSUDS}}$
1	Lower Data Strobe	$\overline{\text{BSLDS}}$
1	Adress Strobe	$\overline{\text{BSAS}}$
1	Syme	$\overline{\text{BSSYNC}}$
1	Reply	$\overline{\text{BSRPLY}}$
1	Parity-Gen.-Error	$\overline{\text{BSPGE}}$
1	Parity-Error	$\overline{\text{BSPE}}$
+8	Memory Present	$\overline{\text{BSMP}_0}$ - $\overline{\text{BSMP}_8}$

Anmerkung: Alle Signale activ-low.

Bild: A. 8

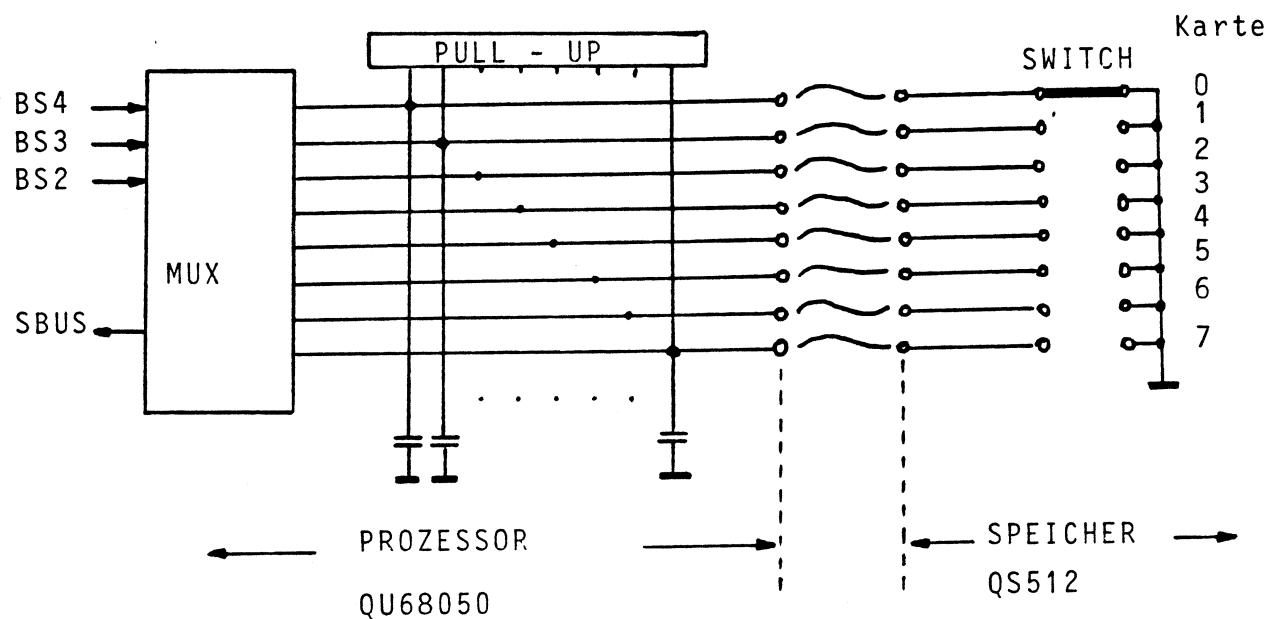


Bild A.9

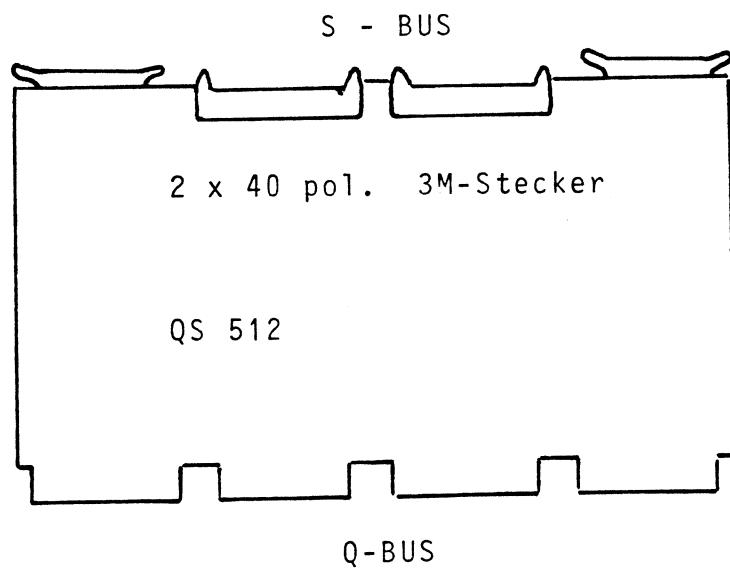


Bild A.10

Seite -58-

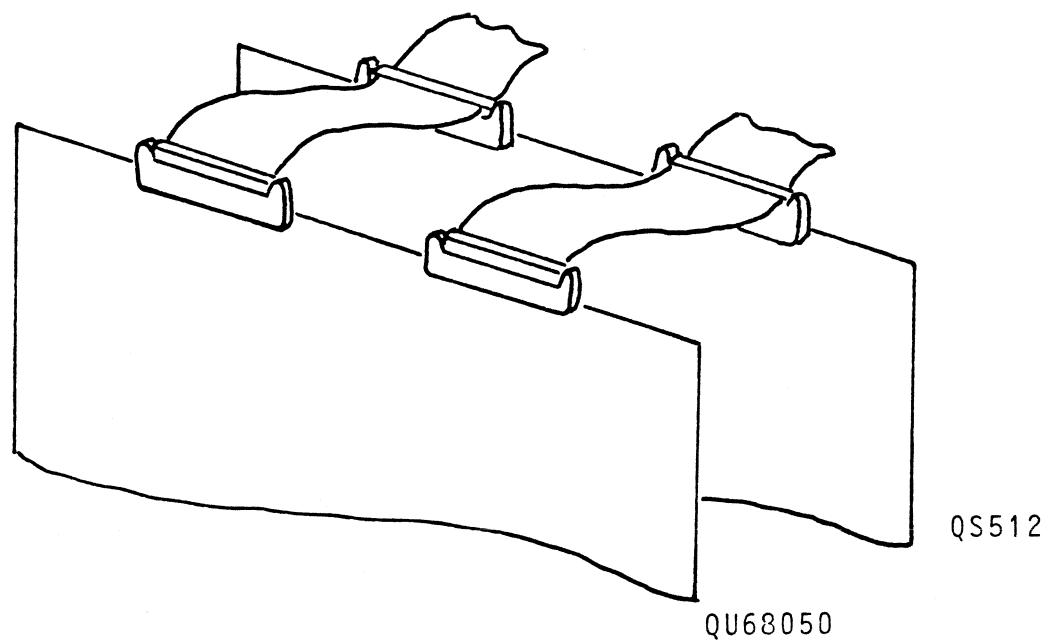
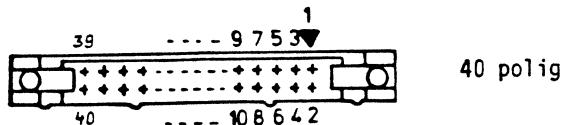


Bild A.11

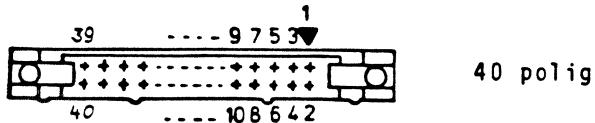
**Belegung:
3M Scotchflex - Steckerleiste**



Pin	Signalname	Bemerkungen	Pin	Signalname	Bemerkungen
1	BSD0		26	GND	
2	GND		27	BSD 13	
3	BSD1		28	GND	
4	GND		29	BSD14	
5	BSD2		30	GND	
6	GND		31	BSD15	
7	BSD3		32	GND	
8	GND		33	BSR/W	
9	BSD4		34	GND	
10	GND		35	BSUDS	
11	BSD5		36	GND	
12	GND		37	BSLDS	
13	BSD 6		38	GND	
14	GND		39	BSAS	
15	BSD7		40	GND	
16	GND		41		
17	BSD8		42		
18	GND		43		
19	BSD9		44		
20	GND		45		
21	BSD10		46		
22	GND		47		
23	BSD11		48		
24	GND		49		
25	BSD12		50		

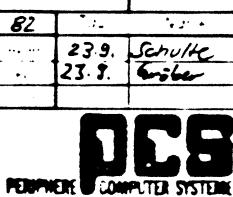
AVO 7/11 3/78	82	70	71	Steckerbelegung für pol.Scotchflex - Steckerleiste	
-----	23.9.	Schulte	23.9.		
 DCS DEUTSCHE COMPUTER SYSTEME				S-Bus Schnittstelle (J1)	
				P	
				1 / 2	

Belegung:
3M Scotchflex - Steckerleiste



Pin	Signalname	Bemerkungen	Pin	Signalname	Bemerkungen
1	<u>BSA0</u>		26	GND	
2	<u>BSMP0L</u>		27	<u>BSB4</u>	
3	<u>BSA1</u>		28	GND	
4	<u>BSMP1L</u>		29	<u>BSPE</u>	
5	<u>BSA2</u>		30	GND	
6	<u>BSMP2L</u>		31	<u>BSRPL</u>	
7	<u>BSA3</u>		32	GND	
8	<u>BSMP3L</u>		33	<u>BSPGE</u>	
9	<u>BSA4</u>		34	GND	
10	<u>BSMP4L</u>		35	<u>BSSYNC</u>	
11	<u>BSA5</u>		36	GND	
12	<u>BSMP5L</u>		37	NC	
13	<u>BSA6</u>		38	GND	
14	<u>BSMP6L</u>		39	NC	
15	<u>BSA7</u>		40	GND	
16	<u>BSMP7L</u>		41		
17	<u>BSA8</u>		42		
18	GND		43		
19	<u>BSB0</u>		44		
20	GND		45		
21	<u>BSB1</u>		46		
22	GND		47		
23	<u>BSB2</u>		48		
24	GND		49		
25	<u>BSB3</u>		50		

AVO 7/11 3/78



Steckerbelegung für
pol. Scotchflex - Steckerleiste

VL-NR

2 / 2

LITERATUR

- /1/ Microcomputer processor handbook
Firmenschrift der Fa. DEC, 1979/1980.
- /2/ MC 68000 L 10, 16-bit-Microprocessor, Advance Information Fa. Motorola Inc., 1981
- /3/ MC 68451 L41618, Memory Management Unit, Advance Information Fa. Motorola Inc., 1981
- /4/ UNIX Time Sharing System. The Bell System Technical Journal, Juli - August 1978, Vol. 57, N.6, Part 2, Bell Systems, 1978.
- /5/ Microsoft Memory Management Document (XENIX for microprocessors). Fa. Microsoft, 1981.
- /6/ PERKEO - a Hardware/Software System for "Personal Scientific Computing". Fa Siemens AG, Januar 1981.