

K896 - Spezifikation

K 8 9 6 - S P E Z I F I K A T I O N

Stand: 17. Januar 1982

Version: 1.1

Copyright by KONTRON MIKROCOMPUTER GmbH, Eching/München

I N H A L T

1. Übersicht
2. Funktionale Spezifikationen
 - 2.1 Definitionen und Bus Elemente
 - 2.1.1 Der Bus Master (Commander)
 - 2.1.2 Der Bus Slave
 - 2.1.3 Der Supervisor
 - 2.2 Die Busleitungen
 - 2.2.1 Stromversorgungsleitungen
 - 2.2.2 Power On/Fail Leitungen
 - 2.2.3 Adreß-/Datenleitungen
 - 2.2.4 Steuerleitungen
 - 2.2.5 Bustiming Leitungen
 - 2.2.6 Bus Arbitration Leitungen
 - 2.2.7 Interprozessor-Kommunikation
 - 2.2.8 Bus Clock
3. Elektrische Spezifikationen
 - 3.1 Spezifikation der Baugruppe
 - 3.1.1 Stromversorgung
 - 3.1.2 Einschaltsequenz (Power Up)
 - 3.1.3 Power Fail Sequenz
 - 3.2 Spezifikation der Leitungstreiber und -empfänger
 - 3.2.1 Leitungstreiber
 - 3.2.2 Leitungsempfänger
 - 3.2.3 Signallasten
4. Mechanische Spezifikationen
 - 4.1 Baugruppenformat
 - 4.2 Steckerzuordnung
 - 4.3 Ein-/Ausgabestecker
 - 4.4 Platinenausführung
5. Anhang
 - K896 Busbelegung
 - Konstruktionszeichnung (DCB-Baugruppenformat)
 - Implementationsbeispiel des Bus Arbiters
 - Implementationsbeispiel der Interprozessor-Kommunikation

1. Übersicht über das Buskonzept K896

Das Buskonzept K896 spezifiziert die funktionellen, elektrischen und mechanischen Eigenschaften eines 32 bit gemultiplexten Backplanebus, auf Basis einer Standard 64-Pin Steckerleiste (VG-Leiste: DIN 41612).

K896* ist als Prozessor- und Technologie-unabhängiger Kommunikationspfad für heutige 16 und zukünftige 32 Bit Prozessoren konzipiert. Einzel-Prozessorsysteme werden ebenso unterstützt wie Mehrprozessorsysteme mit bis zu 8 potentiellen Bus Masters unterschiedlicher Priorität, wobei zwischen einzelnen potentiellen Bus Master ein einfaches aber leistungsfähiges Verfahren zur Interprozessor-Kommunikation vorhanden ist.

Es wurde insbesondere darauf Rücksicht genommen, eine leichte Realisierbarkeit mit heute verfügbaren Mitteln zu gewährleisten. Das Resultat ist eine Busdefinition, die es erlaubt, mit minimalem Aufwand Bus-kompatible Baugruppen für verschiedenste Anforderungen zu realisieren.

Der Bus selbst ist für hohe Geschwindigkeiten geeignet (10 MHz Taktfrequenz). Ein bei allen wesentlichen Busaktivitäten implementiertes 'Request-Acknowledge'-Prinzip garantiert trotzdem die Unabhängigkeit von der Geschwindigkeit (und damit der Technologie) einzelner Baugruppen.

Alle Baugruppen benötigen eine gewisse Eigenintelligenz, um Aufträge interpretieren, ausführen und dessen Status melden zu können. Reine I/O-Baugruppen (wie beispielsweise im ECB-System) sind in K896 deshalb nicht zugelassen, statt dessen werden sämtliche Ein-/Ausgabe-Aktivitäten von separaten Ein-/Ausgabeprozessoren durchgeführt, was die Zentraleinheit entlastet und damit die Gesamtverarbeitungsgeschwindigkeit drastisch erhöht. Ein typisches K896-System ist in Bild 1 dargestellt.

Der K896 Bus ist auf den CPU- und Speicherbaugruppen der DCB-Familie implementiert:

- DCB/CPU - Z8001 CPU Board (Bus Master) + Memory
- DCB/CMM - Z8001 CPU Board (Bus Master) + Memory Management
- DCB/M25 - K896 Speicherbaugruppe + EDC (256kByte)
- DCB/M38 - K896 Speicherbaugruppe + Parity (384kByte)

Kapitel 2 dieser Druckschrift beschreibt die funktionellen, Kapitel 3 die elektrischen, sowie Kapitel 4 die mechanischen Spezifikationen von K896 bzw. dazu kompatiblen Baugruppen.

* K896 basiert auf dem IEEE-Standardisierungsprojekt P896

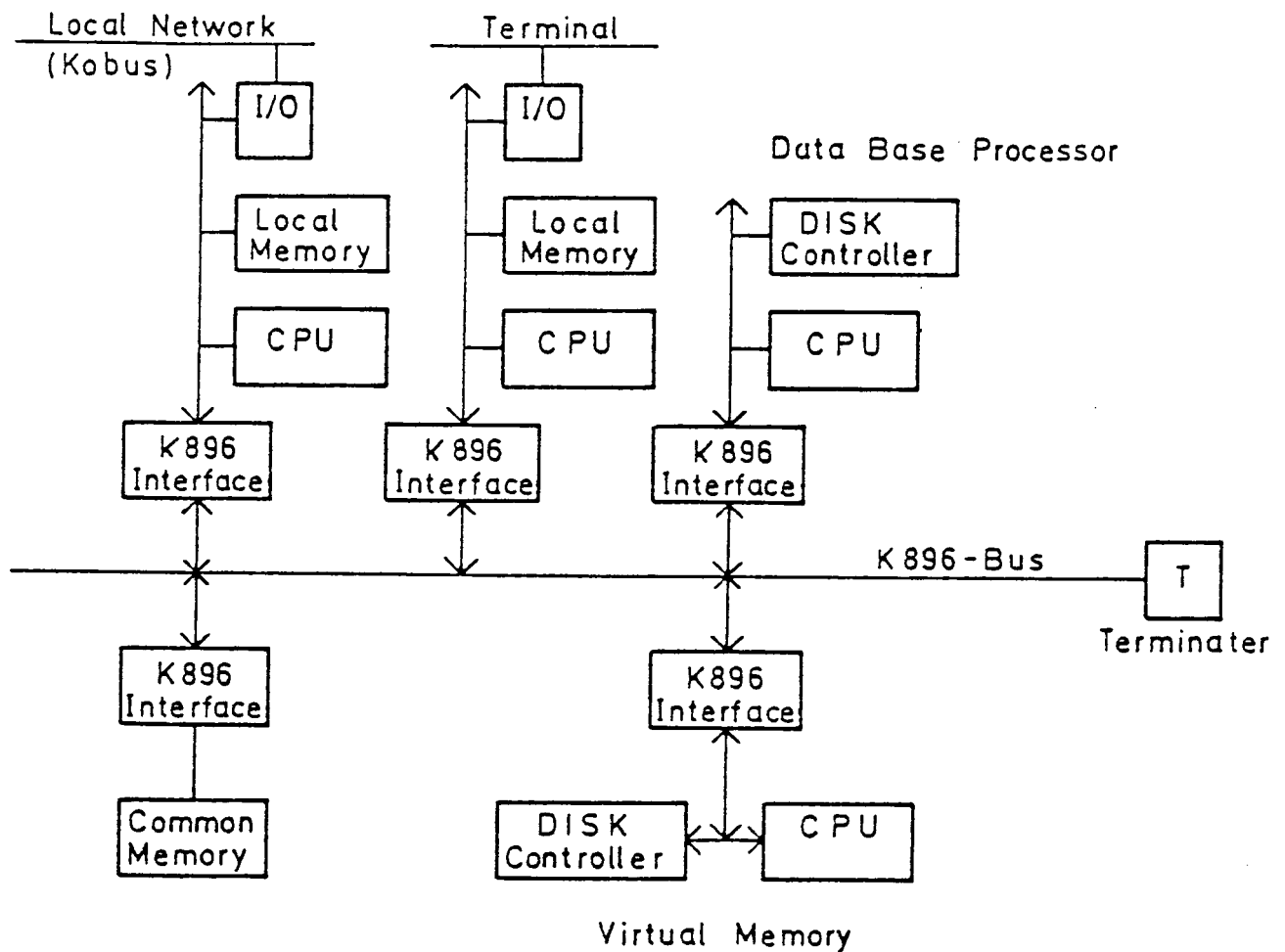


Bild 1: Typisches K896-System

Die charakteristischen Kriterien von K896 sind:

- Prozessor-unabhängig
- Technologie-unabhängig
- 32 Bit Adreß-/Datenbus (gemultiplext)
- Übertragungsraten bis 10 MHz
- Multiprozessor-fähig (bis zu 8 Bus Master)
- Interprozessor-Kommunikation
- Supervisor-fähig
- Open Collector Bus
- Busabschluß Widerstände

2. Funktionale Spezifikationen

2.1 Definitionen und Bus Elemente

Dieser Abschnitt enthält Definitionen, die im folgenden zur Anwendung kommen:

MODUL	Baugruppe (Platine) eines K896-Systems mit K896-Bus Interface
COMMANDER	Das Busmodul, das momentan die Kontrolle über den Bus hat, auch als 'BUS MASTER' bezeichnet. Bevor ein Modul 'COMMANDER' wird, bewirbt es sich über die 'Arbitration Logic' um den Bus.
BUS SLAVE	Das Busmodul, das durch einen Commander adressiert ist. Ein BUS SLAVE dekodiert Adressen- und Steuerinformationen vom Commander, um zu entscheiden, ob es angesprochen ist. Falls ja, führt der 'BUS SLAVE' die vom Commander eingeleitete Aktivität aus.
SUPERVISOR	Das (optionale) Busmodul, das alle Transaktionen überwacht und die Möglichkeit hat, Transaktionen zu ändern, zu verlangsamen oder auszutauschen.
BUS CYCLE	Eine nicht unterbrechbare Folge von Ereignissen in einem K896-System. Ein 'BUS CYCLE' entspricht einem durch BUSRQ nicht unterbrechbaren Maschinenzyklus einer CPU. Er beginnt in K896 immer mit der Übertragung einer Adresse und endet mit der Übertragung von Daten.
ADDRESS CYCLE	Erster Teil des Buszyklus, während dem der Commander Adressen und Steuerinformationen an den Bus Slave überträgt.
DATA CYCLE	Zweiter Teil des Buszyklus, während dem die Datenübertragung zwischen Commander und Bus Slave stattfindet.
BUS ARBITER	Bus-"Schiedsrichter". Eine Logik eines jeden potentiellen COMMANDER's, mit Hilfe derer er sich um den Bus bewirbt. K896 unterstützt bis zu 8 verschiedene Commander unterschiedlicher Priorität.

2.1.1 Der Bus Master (Commander)

Ein Commander kann die Informationsübertragung über den Bus initiieren. **Niemals können mehrere Module gleichzeitig Commander (Bus Master) sein.** Der 'BUS ARBITER' eines jeden potentiellen Commanders garantiert zusammen mit dem **Bus Priority Scheduler** einer K896-Backplane, daß immer nur das momentan höchstpriorisierte Modul, das sich um den Bus bewirbt, diesen zugesprochen bekommt. Die Priorität eines K896-Moduls ist schalterselektierbar. Bis zu 8 potentielle Commander können gleichzeitig in einem System arbeiten.

Ist ein K896-Modul einmal Commander (Bus Master), so sollte es den Bus nur solange behalten, solange es erforderlich ist.

Ein potentieller K896-Commander kann drei Zustände einnehmen:

- LOCAL:** er hat kein Interesse am Bus
- WAITING:** er bewirbt sich mit der ihm eigenen Priorität um den Bus
- ACTIVE:** er ist Bus Master und führt Transferaktivitäten aus, solange dies erforderlich ist, oder kein höher priorisiertes Modul den Bus beansprucht.

2.1.2 Der Bus Slave

Module, die potentielle 'Bus Slaves' sind, überwachen alle Adreßzyklen um zu entscheiden, ob sie angesprochen sind oder nicht. Ein Bus Slave kann zwei Zustände einnehmen:

- SELECTED:** er wurde durch einen Commander adressiert und nimmt am folgenden Datentransferzyklus teil.
- UNSELECTED:** er wurde nicht adressiert und nimmt deshalb auch nicht am folgenden Transferzyklus teil.

2.1.3 Der Supervisor

Ein Supervisor überwacht alle Busaktivitäten. Er hat die Möglichkeit

- a) das Bustiming zu ändern
- b) Adressen und/oder Daten zu ändern
- c) die Art der Operation zu ändern

Ein Supervisor kann zwei Zustände einnehmen:

ACTIVE: er greift in den momentanen Zyklus ein

INACTIVE: er überwacht, aber greift nicht zu

Ein K896-System muß nicht notwendigerweise einen Supervisor haben. Eine typische Anwendung wäre beispielsweise ein Speicherverwaltungsmodul direkt auf dem Bus, das logische in physikalische Adressen ändert.

2.2 Die Busleitungen

Dieser Abschnitt beschreibt die 64 Pin's des K896-Steckers.

Signal(gruppe)	Pinanzahl
+ 5 Volt (Power)	4
Ground	10
Power On/Power Fail	2
Busclock	1
Arbitrationleitungen	5
Interprozessor-Kommunikation	1
Steuerleitungen	4
Adreß-/Datenleitungen	32
Timing	5

Alle K896-Signalleitungen sind mit Open Collector Puffern getrieben und führen invertierte Signale. Wired-or Verknüpfungen können mit allen Signalen verwirklicht werden. Timingdiagramme in dieser Spezifikation zeigen Signale so, wie sie auf dem Bus erscheinen, also invertiert.

2.2.1 Stromversorgung

Ein K896-Modul wird über den Busstecker mit +5 Volt und GROUND versorgt. Eventuell notwendige weitere Spannungen sind auf dem jeweiligen Board selbst zu generieren. Für +5 Volt stehen 4 Anschlüsse, für GROUND 4+6 (signal return) zur Verfügung.

2.2.2 Power On/ Power Fail Leitungen

Signalbezeichnung: POI (Power On Indication)
PFI (Power Fail Indication)

Siehe hierzu die Abschnitte 3.1.2 bzw. 3.1.3!

2.2.3 Adreß-/Datenleitungen

Signalbezeichnung: ADO...AD31 (ADx-Leitungen)

Der 32 Bit breite Adreß-/Datenbus ist gemultiplext und führt Adressen während des Adreßzyklus (AS aktiv) und Daten während des Datenzyklus (DS aktiv).

Theoretisch können sowohl Adressen, als auch Daten die maximale Breite von 32 Bit annehmen, für die DCB-Familie gelten allerdings folgende **Einschränkungen:**

Adressen: 24 Bit von ADO...AD23
Daten: 16 Bit von ADO...AD15

Hinweis:

Alle Speicherboards der DCB-Familie sind bezüglich der Daten 16 Bit orientiert und verwenden 24 Adreßbits zur Adreßdekodierung. Dies entspricht einer maximalen Speicherkapazität von 16 MByte.

Die Kompatibilität zu zukünftigen 32 Bit Systemen ist trotzdem gewahrt, nämlich durch die Definition von zwei Kompatibilitätsstufen für K896:

K896 Level I und
K896 Level II

Die Unterscheidung erfolgt durch die Steuerleitung C2 (siehe nächsten Abschnitt).

Level II kompatible Baugruppen verwenden:

Adressen: 32 Bit von ADO ... AD31
Daten: 32 Bit von ADO ... AD31

2.2.4 Steuerleitungen

Signalbezeichnung: C0...C2 (Cx-Leitungen)
RO (Rückkopplungsstatus)

a) die Cx-Leitungen

Ein 'COMMANDER' (Bus Master) zeigt über diese 3 Leitungen während des Adreßzyklus an, welche Aktivität er mit einem 'BUS SLAVE' durchzuführen gedenkt. Folgende Zuordnung ist definiert:

C2	C1	C0	! Bedeutung
1	0	0	! Byte read)
1	0	1	! Word read) Level I
1	1	0	! Byte write)
1	1	1	! Word write)
			!
0	0	0	! Quadlet read)
0	0	1	! Blocktransfer read)
0	1	0	! Quadlet write) Level II
0	1	1	! Blocktransfer write)

Hinweis: Diese Tabelle zeigt den Logikzustand vor den invertierenden Bustreibern.

Alle Baugruppen der DCB-Familie entsprechen Level I. Level II Steuer codes führen zu einer Fehleranzeige über die Leitung RO.

Die Codierung der Cx-Leitungen ermöglicht eine leichte Dekodierung auf der Empfängerseite (BUS SLAVE). C0 ist beispielsweise die 'Byte/Word Select' Leitung und C1 die 'Read/Write Select' Leitung. Beide Signale können in der Regel direkt von entsprechenden Anschlüssen einer modernen 16 Bit-CPU abgeleitet werden.

Beispiele: Z800x CPU ---> C0=B/W C1=R/W
8086 CPU ---> C0=BHE C1=WR

Für Level I-kompatible Baugruppen genügt es, die Leitungen C0 und C1 zu bedienen. Es ist zulässig aber nicht notwendig, die Cx-Leitungen während eines gesamten Buszyklus' zu treiben.

Bytelage

Die Cx-Leitungen erlauben die Unterscheidung zwischen Datentransfers von 8, 16 und 32 (Level II) Bit. Jeder potentielle Bus Master hat bezüglich der Lage von Bytes bzw. Words innerhalb von ADO ... AD15 folgende Grundsätze zu beachten:

1. Bei Byte Lesezyklen werden immer 16 Bit übertragen (Adresse A0 bleibt unberücksichtigt), wobei die Byteauswahl innerhalb des Commanders erfolgen muß (dies geschieht beispielsweise automatisch bei der Z8001-CPU).
2. Bei Byte Schreibzyklen bestimmt die Adresse A0, ob das nieder- oder höherwertige Byte geschrieben wird. Das niederwertige Byte wird auf ADO ... AD7, das höherwertige auf AD8 ... AD15 erwartet.

Entsprechendes gilt für 32 Bit Operationen (Level II).

Für die Adressierung von Bytes, Words bzw. Quadlets gilt grundsätzlich:

A1	A0	!	für
x	x	!	Bytes (8 bit)
x	0	!	Words (16 bit)
0	0	!	Quadlets (32 bit)
		!	(Level II)

b) Rückkopplungsstatus (RO)

Die Leitung RO dient als Rückkopplungsstatus von einem selektierten 'BUS SLAVE' zum 'COMMANDER'. Sie kann sowohl während des Adreßzyklus als auch während des Datenzyklus aktiviert werden.

RO (Adreßzyklus)		!	RO (Datenzyklus)	
0	OK	!	0	OK
1	Slave busy	!	1	Transfer Fehler

K896 definiert nicht, was ein Commander im Fehlerfall zu tun hat. Dies hängt im Einzelfall von der Hardware des Commanders und dessen Systemsoftware ab.

2.2.5 Bustiming Leitungen

Signalbezeichnung: **AS** - Address Strobe
DS - Data Strobe
AK - Acknowledge
SC - Supervisor Control

Die Bustiming Leitungen kennzeichnen die Gültigkeit bestimmter Informationen auf dem Bus und realisieren außerdem einen vollen Handshake zwischen verschiedenen schnellen Moduln eines Systems.

a) Address Strobe (AS)

Ein Bus Master initiiert eine Transferaktivität durch die Aktivierung des 'Address Strobes'. Die positive Flanke des AS-Signals kennzeichnet die Gültigkeit einer Adresse auf den ADx-Leitungen. Ein prinzipielles Timingdiagramm ist in Bild 2 dargestellt, exakte Angaben sind dem Anhang zu entnehmen.

Hinweis: Die Dauer des AS-Signals unterliegt keinem Handshake. Ein Commander kann erwarten, daß jeder potentielle Bus Slave jederzeit in der Lage ist, Adressen zu 'latchen'.

b) Data Strobe (DS)

Ein Bus Master initiiert den Datenzyklus durch die Aktivierung des 'Data Strobes'. Er darf dies erst tun, nachdem der selektierte Bus Slave den Empfang der Adresse durch ein Acknowledge bestätigt hat. Das Ende eines Datenzyklus wird durch die positive Flanke des AK-Signals eingeleitet.

c) Acknowledge (AK)

Ein Bus Slave bestätigt den Empfang einer Adresse durch die Aktivierung des 'Acknowledge' und gibt dem Commander damit das Startzeichen für den Datenzyklus. Die Dauer des Datenzyklus wird unmittelbar vom Bus Slave bestimmt, da ein Commander erst nach der steigenden Flanke des AK-Signals den Datenzyklus abschließt. Dieses Verfahren ermöglicht die Synchronisation zwischen verschiedenen schnellen Moduln eines K896-Systems.

Jeder Commander ist selbst dafür verantwortlich, einen einmal gestarteten Buszyklus durch entsprechende Timeout-Mechanismen abubrechen, wenn das AK-Signal nicht innerhalb 5 μ s erscheint (Beispiel: ein nicht existenter Speicher oder Bus Slave wird angesprochen).

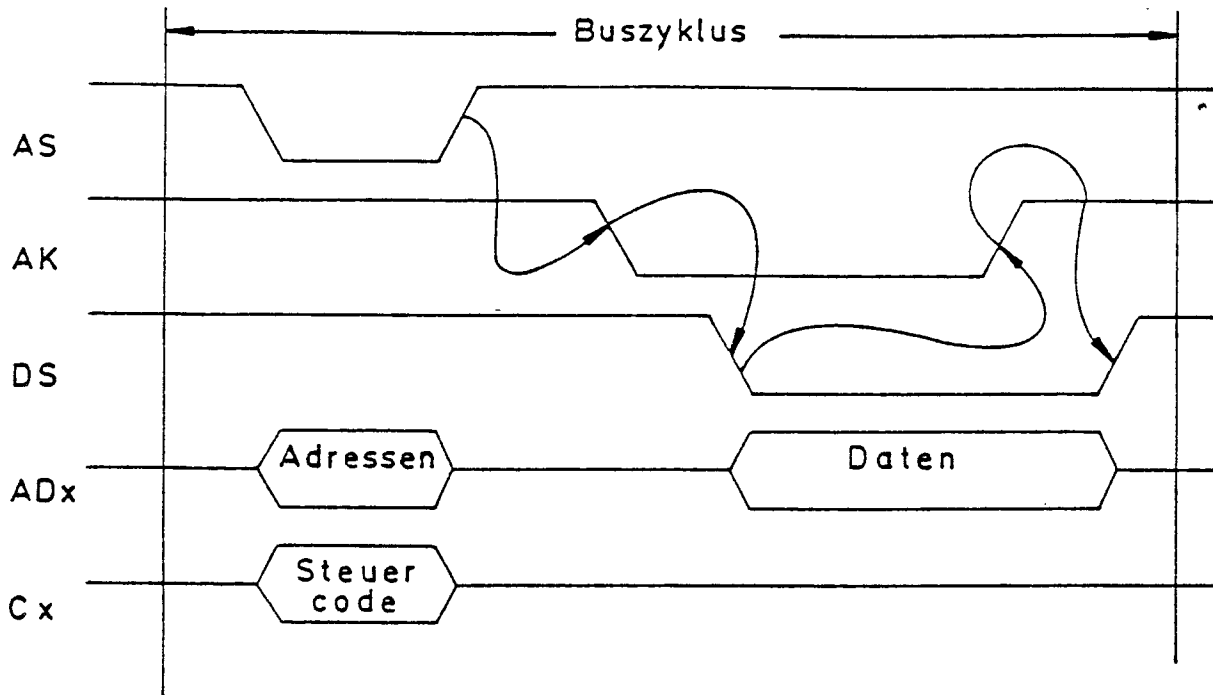


Bild 2: Prinzipdarstellung eines Buszyklus

d) Supervisor Control (SC)

Ein Supervisor überwacht als übergeordnetes Modul alle Busaktivitäten. Er ist in der Lage:

- das Bus Timing zu ändern
- den Steuercode (Cx-Leitungen) zu ändern
- Adressen und/oder Daten zu ändern.

Ein K896-System muß nicht notwendigerweise einen Supervisor enthalten. Es wird allerdings von jedem Modul erwartet, daß es beim Eingreifen eines Supervisors entsprechend der Definition der SC-Leitung reagiert.

Ein Supervisor informiert durch das SC-Signal die übrigen Module, daß er in den laufenden Buszyklus eingegriffen hat. In diesem Fall muß sich der momentane Commander von den ADx- bzw. Cx-Leitungen während des Adreß- bzw. Datenschreibzyklus zurückziehen. Bei Datenlesezyklen muß der Bus Slave entsprechend reagieren.

2.2.6 Bus Arbitration Leitungen

Signalbezeichnung: BPS0...BPS2 (Bus Priority Scheduler: BPSx)
 BB (Bus Busy)
 BRQ (Bus Request)

Ein K896-System darf bis zu 8 potentielle Bus Master enthalten. Zur Auflösung von eventuellen Belegungskonflikten dient die Arbitration Logik. Diese setzt sich zusammen aus dem zentralen Bus Priority Scheduler einer K896-Backplane (BPSx-Leitungen) und der dezentralen Arbitration Logik eines jeden Bus Masters. Die Arbitration Logik wird mit dem Bus Clock (BCLK) synchronisiert.

a) Bus Priority Scheduler (BPSx)

Der Bus Priority Scheduler tastet als zentrale Steuereinheit einer K896-Backplane laufend die potentiellen Bus Master ab. Der Bus Priority Scheduler liefert hierzu über die Leitungen BPS0...BPS2 jeweils eine von acht Adressen (binär codiert). Das Weiterschalten erfolgt mit der Geschwindigkeit des zentralen BUS CLOCK (BCLK = 10 MHz). Ein Zyklus zum Abtasten aller potentiellen Bus Master dauert demnach 800 ns. Dies entspricht einer mittleren Zeit von 400 ns, die ein Modul warten muß, bis es im Falle eines unbenutzten Bus diesen zugesprochen bekommt.

b) Bus Request (BRQ)

Ein K896-Modul, das Bus Master werden will (Requester), aktiviert die Leitung BRQ, während die eigene Adresse auf den BPSx-Leitungen erscheint. Ist der Bus momentan unbenutzt (BB ist inaktiv), so erhält der Requester den Bus zu dem Augenblick, da seine Adresse auf den BPSx-Leitungen erscheint. Zu diesem Zeitpunkt muß der Requester die Bus Busy Leitung aktivieren, um damit die Belegung des Bus durch einen Master anzuzeigen. Ist ein Modul einmal Master geworden, so ist es nicht mehr notwendig, die BRQ-Leitung weiterhin zu aktivieren, denn es hat ja bereits den Bus. Vielmehr muß der momentane Master fortan die BRQ-Leitung überwachen um festzustellen, ob ein höherpriorisiertes Modul den Bus beansprucht. In einem solchen Fall sollte der Bus durch die Deaktivierung von Bus Busy (BB) so bald wie möglich freigegeben werden.

c) Bus Busy (BB)

Diese Leitung zeigt an, daß der Bus momentan von einem Master belegt ist. Sie ist von einem Requester in dem Augenblick zu aktivieren, wo seine eigene Adresse auf den BPSx-Leitungen erscheint.

Hinweis:

Die Implementation der Arbitration Logik auf den CPU-Baugruppen der DCB-Familie ist im Anhang gezeigt.

2.2.7 Interprozessor-Kommunikation

Signalbezeichnung: IPRQ (Interprocessor Request)

In Systemen mit mehreren potentiellen Bus Mastern ist die Kommunikation zwischen einzelnen Bus Mastern von entscheidender Bedeutung. Die IPRQ-Leitung erlaubt, zusammen mit dem zentralen Bus Priority Scheduler den selektiven Request an einen bestimmten Prozessor von jedem anderen potentiellen Bus Master aus, zu senden. Dabei ist es nicht erforderlich, daß der Sender momentan Bus Master ist.

Der Sender aktiviert die IPRQ-Leitung zu dem Augenblick, da die Adresse des gewünschten Zielprozessors auf den BPSx-Leitungen erscheint.

K896 definiert nicht, was die IPRQ-Leitung im Zielprozessor bewirkt. In der Regel wird dadurch ein Interrupt erzeugt.

Hinweis:

Die Logik zur Interprozessor-Kommunikation der DCB-Baugruppen ist ebenfalls im Anhang dargestellt. Sie ist in engem Zusammenhang mit dem Bus Arbiter zu verstehen und arbeitet wie dieser synchron zu BCLK.

2.2.8 Bus Clock

Signalbezeichnung: BCLK (Bus Clock)

Die Frequenz des Bus Clock beträgt 10 MHz. Der Bus Clock wird auf der Backplane eines K896-Systems erzeugt. Es ist nicht definiert, für welche Zwecke ein Modul den Bus Clock verwendet. Die Arbitration Logik wird jedoch immer mit dem Bus Clock synchronisiert.

3. Elektrische Spezifikationen

Dieser Abschnitt beschreibt die elektrischen Spezifikationen für K896-kompatible Baugruppen. Jede K896-kompatible Baugruppe muß diese Spezifikationen zumindest im kommerziellen Temperaturbereich erfüllen. Welche Technologie auf einer bestimmten Baugruppe zur Anwendung kommt hat auf die K896-Kompatibilität keinerlei Einfluß.

3.1 Allgemeine Baugruppen-Spezifikation

3.1.1 Stromversorgung

Eine K896-Baugruppe wird über die K896-Backplane über vier Anschlüsse der 64-poligen Steckerleiste mit +5 Volt versorgt. Alle eventuell notwendigen Zusatzspannungen müssen auf der Baugruppe selbst erzeugt werden. Der Strombedarf einer Baugruppe darf im Mittel 4 Ampere nicht überschreiten. Dies entspricht einem Ampere pro Anschlußpin für 5 Volt bzw. GND. Der Nominalwert der Versorgungsspannung beträgt 5.1 Volt mit einer maximalen Abweichung von +/- 2 % (5.0 V ...5.2 V).

3.1.2 Einschaltsequenz (Power On)

Jede Baugruppe ist für seine eigene 'RESET Operation' verantwortlich. Ein RESET wird über den K896-Bus durch das Signal POI (Power On Indication) angezeigt. Die Logik hierfür befindet sich auf der K896-Backplane. Sie sorgt beim Einschalten der Versorgungsspannung für einen automatischen 'Power On Reset'.

Selbstverständlich kann eine Baugruppe zusätzlich eine eigene 'Power On Reset' Schaltung besitzen. Dies ist insbesondere für solche Baugruppen empfehlenswert, die als 'Single Board Computer' eingesetzt werden sollen.

3.1.3 Power Fail Sequenz

Jede Baugruppe ist für ihre eigene 'Power Fail Funktion' verantwortlich. Ein drohender Power Fail Zustand wird über den K896-Bus durch das Signal PFI (Power Fail Indication) angezeigt. Die Logik hierfür befindet sich auf der K896-Backplane, **wobei diese nur dann arbeiten kann, wenn ein entsprechendes Netzteil vorhanden ist.** Ist dies der Fall, so darf ein K896-Modul annehmen, daß nach dem 'Power Fail Signal' mindestens 20 ms zur Verfügung stehen, bis die Versorgungsspannung auf 4.75 Volt absinkt.

Es ist nicht definiert, was das 'Power fail' Signal innerhalb eines Moduls bewirkt. Üblicherweise wird damit ein NMI (Non maskable Interrupt) generiert.

3.2 Spezifikation der Leitungstreiber und -empfänger

Dieser Abschnitt enthält die elektrischen Spezifikationen für die in K896-Systemen erforderlichen Leitungstreiber bzw. -empfänger. Auch hier spielt es keine Rolle, in welcher Technologie ein Leitungstreiber bzw. -empfänger realisiert ist. Solange die Spezifikationen erfüllt sind, dürfen alle Bausteine, diskrete Transistoren eingeschlossen, als Treiber bzw. Empfänger verwendet werden. Wenn die Wahl zwischen getrennten Treibern und Empfängern oder kombinierten Transceivern besteht, so ist letztere Möglichkeit vorzuziehen.

3.2.1 Leitungstreiber

Jeder K896-kompatible Leitungstreiber (Bustreiber) muß einen Open-Collector Ausgang besitzen und invertierend sein, so daß der aktive Zustand (log. 1) vor den Leitungstreibern 0 Volt auf dem Bus ergibt. Die **erforderliche Treiberkapazität muß 48 mA betragen**, wobei die dann sich einstellende Spannung auf der getriebenen Busleitung **0,8 Volt nicht überschreiten darf**. Dies entspricht der Belastung in einem mit 16 Baugruppen vollbestückten K896-System unter Berücksichtigung der Abschlußwiderstände auf der K896-Backplane.

3.2.2 Leitungsempfänger

Alle Leitungsempfänger mit Schmitt-Trigger-Eigenschaften und einer Hysterese von größer oder gleich 0.40 V sind geeignet, sofern folgende Voraussetzungen gelten:

- a) eine Spannung auf dem Bus von weniger oder gleich 1,20 V wird als logisch 1 erkannt (aktiver Zustand)
- b) eine Spannung auf dem Bus von größer oder gleich 2,40 V wird als logisch 0 erkannt (inaktiver Zustand)

Hinweis: Auf den Baugruppen der DCB-Familie wird der Baustein 74LS642-1 als Bustreiber eingesetzt. Als Empfänger arbeitet je nach Baugruppe ein 'LS240, 'LS373 etc.

3.2.3 Signallasten

Die hier geforderten Werte gelten für eine K896-Baugruppe **insgesamt**. Solange die zulässigen Maximalwerte bezüglich Eingangsstrom und Kapazität nicht überschritten werden, dürfen auf einer Baugruppe mehrere Treiber und/oder Empfänger auf ein und derselben Busleitung arbeiten.

a) Eingangsstrom

Der Eingangsstrom, den eine Baugruppe pro Signalleitung benötigt, darf in Abhängigkeit des Logikpegels auf dem Bus folgende Werte betragsmäßig nicht überschreiten:

- 1,6 mA für Logikpegel $< 0,8$ V (Busleitung aktiv)
- 200 μ A für Logikpegel $> 2,8$ V (Busleitung inaktiv)

In der Praxis bedeutet dies, daß bis zu 4 LS TTL Empfänger auf einer Signalleitung arbeiten können, sofern die Spezifikationen bezüglich der Eingangskapazität eingehalten werden können.

b) Eingangskapazität

Jede Baugruppe darf pro Signalleitung eine Eingangskapazität von höchstens 20 pF aufweisen. Um dies einhalten zu können, ist es im allgemeinen notwendig, die Leitungslängen zwischen Anschlußstecker und Leitungsempfänger bzw. -treiber auf 50 mm zu begrenzen.

3.2.4 Signalformen

Jede Baugruppe, die eine K896-Signalleitung treibt, muß die im folgenden festgelegten Signalformen einhalten. Desgleichen darf jeder Empfänger voraussetzen, daß alle Signalformen innerhalb den Spezifikationen liegen.

a) Pegelanforderung

Bild 3 zeigt die Standard-Spannungspegel, wie sie für alle Signale auf einem K896-Bus festgelegt sind. Man beachte, daß dies keine Standard TTL-Pegel sind. Die Minimal-Rauschgrenzen sind zwar identisch, die logischen Pegel jedoch so definiert, daß heute verfügbare Treiberbausteine in der Lage sind, einen mit 16 Einheiten vollbestückten Bus entsprechend den Spezifikationen zu treiben.

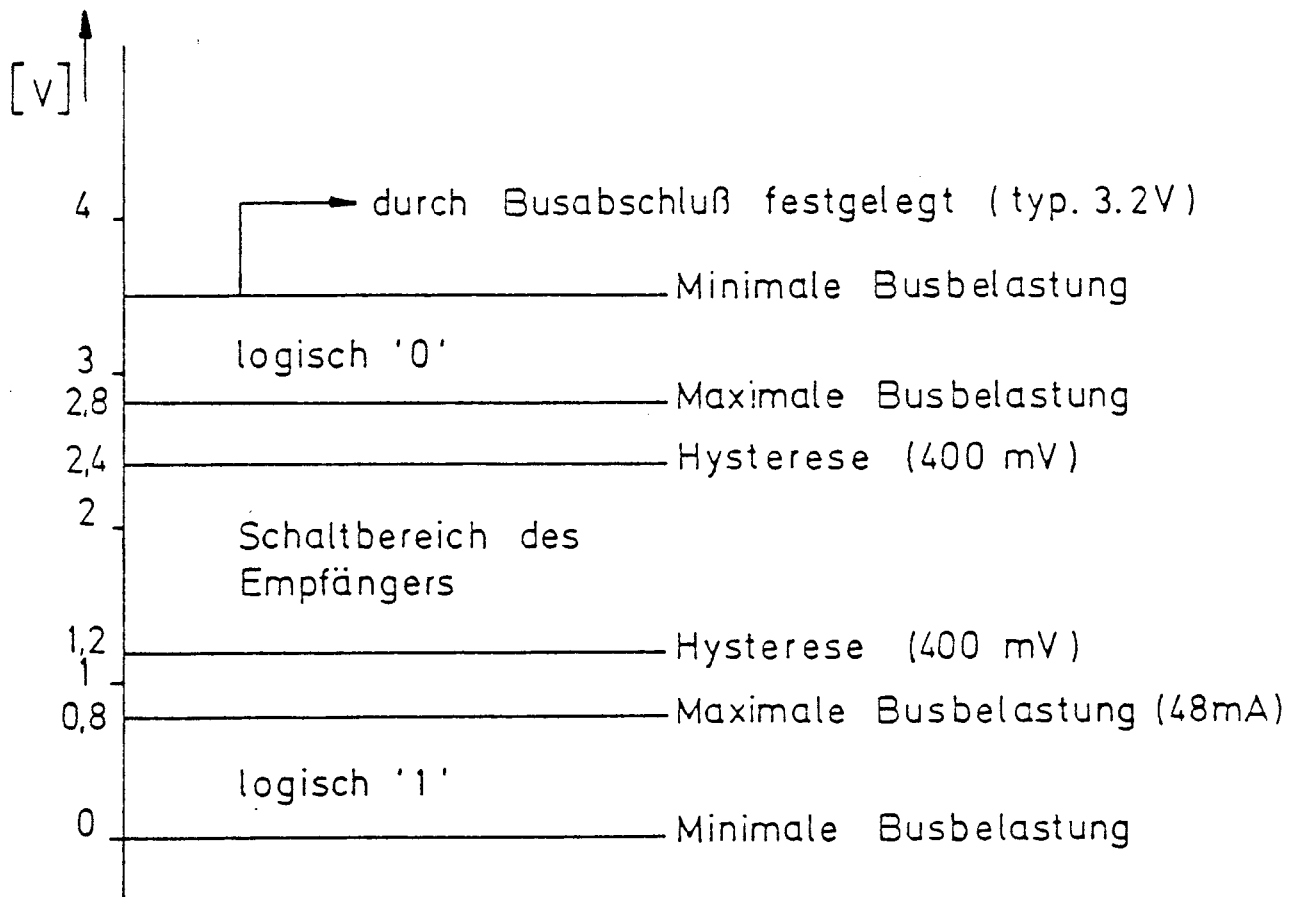


Bild 3: Signalpegel

b) Charakteristische Wellenform

Die charakteristische Wellenform einer K896-Busleitung ist in Bild 4 dargestellt. Unter keinen Umständen darf eine positive Flanke die 1,20 V Schwelle unterschreiten, wenn die 2,40 V Schwelle einmal überschritten wurde, oder eine negative Flanke die 2,40 V Schwelle überschreiten, wenn die 1,20 V Schwelle einmal unterschritten wurde. Im Flankenbereich sind Störungen mit Amplituden kleiner als 0,40 V zulässig (V_{Pr} bzw. V_{Pf}). Die Dauer von Störungen (T_{Pr} bzw. T_{Pf}) darf 5 ns nicht überschreiten.

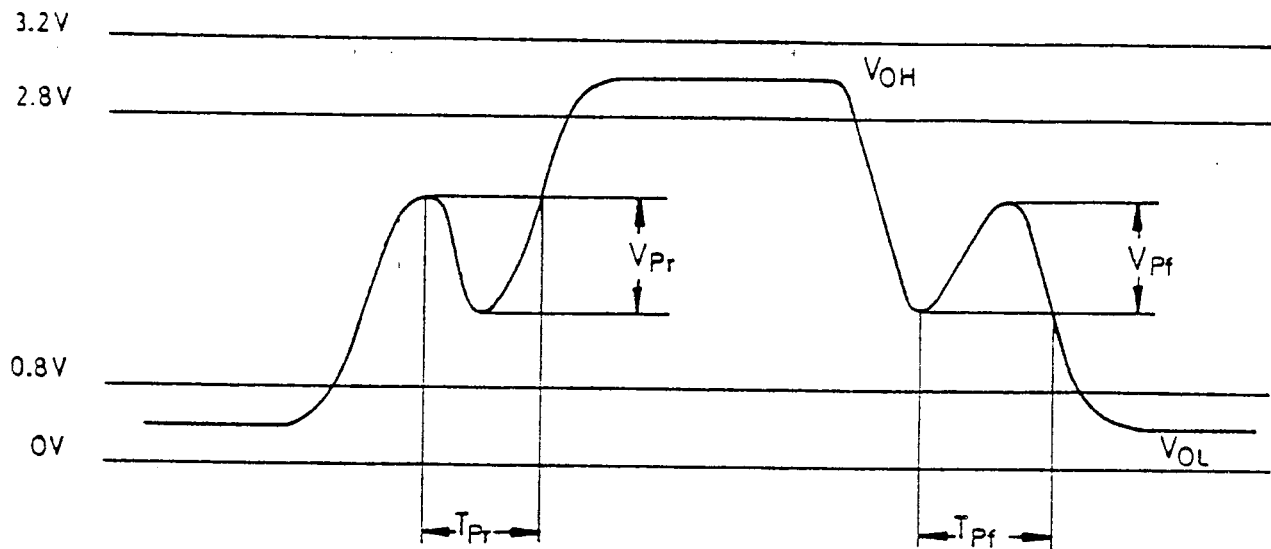


Bild 4: Charakteristische Wellenform

c) Flankensteilheit

Der Wechsel zwischen zwei Logikzuständen auf dem Bus darf maximal 50 ns beanspruchen. Damit ist die Zeit definiert, die während dem Übergang von 0,80 V auf 2,40 V oder umgekehrt vergeht.

4. Mechanische Spezifikationen

Die mechanischen Spezifikationen definieren Format, Stecker und Boardausführung von K896-kompatiblen Baugruppen.

4.1 Baugruppenformat

Das Basisformat einer K896-kompatiblen Baugruppe entspricht dem Dopeleuropaformat (160 mm x 234,4 mm). Alle Baugruppen der DCB-Familie sind in diesem Format ausgeführt. Desweiteren sind Baugruppen im Mehrfachformat zugelassen, sofern die Anschlußsteckerbelegung den Forderungen in Abschnitt 4.2 entspricht.

4.2 Busstecker

Eine K896-kompatible Baugruppe besitzt (mindestens) zwei 64-polige Anschlußstecker für indirekte Steckverbindung. Diese müssen den Spezifikationen in DIN 41612 entsprechen. Ein Stecker (ST-A) führt die Signale des K896-Bus und der danebenliegende Stecker (ST-B) die Signale des weitverbreiteten ECB-Bus, der im DCB-Konzept zur Realisierung von Ein-/Ausgabe für die 16 bit Prozessoren fungiert. Dieser ist in getrennten Spezifikationen beschrieben. Baugruppenformat und Steckeranordnung sind in Bild 5 dargestellt.

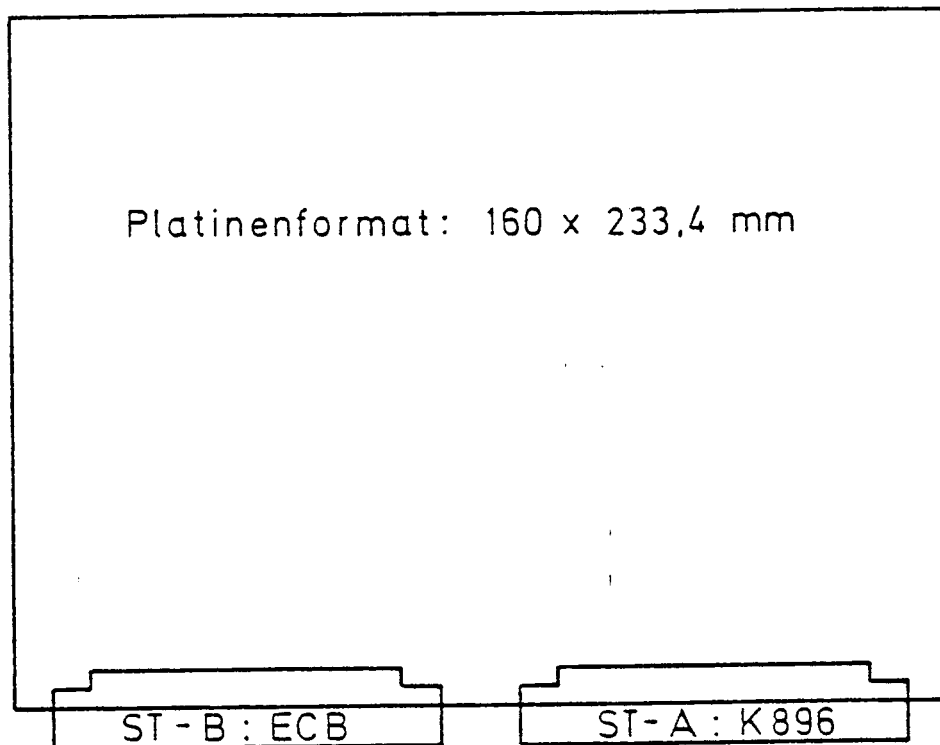


Bild 5: Doppeleuropaformat und Steckerzuordnung

Achtung:

Für jedes Board auf einer K896-Backplane gelten folgende Grundsätze:

- a) Stecker ST-A darf **nur K896-Signale** führen, oder aber überhaupt keine (Dummy Stecker)
- b) Stecker ST-B darf **nur ECB-Signale** führen, oder aber überhaupt keine (Dummy Stecker)

Bei breiteren Baugruppen ist die Zuordnung der übrigen Stecker nicht definiert. Obige Grundsätze müssen jedoch auch bei breiteren Baugruppen gewahrt bleiben.

4.3 Ein-/Ausgabestecker

Ein-/Ausgabestecker befinden sich grundsätzlich auf der den Bussteckern gegenüberliegenden Seite. Sie sind entsprechend der Konstruktionszeichnung im Anhang direkt einzulöten. Ebenso ist im Layout Befestigungsmöglichkeit für eine Steckerblende (siehe Konstruktionszeichnung) vorzusehen.

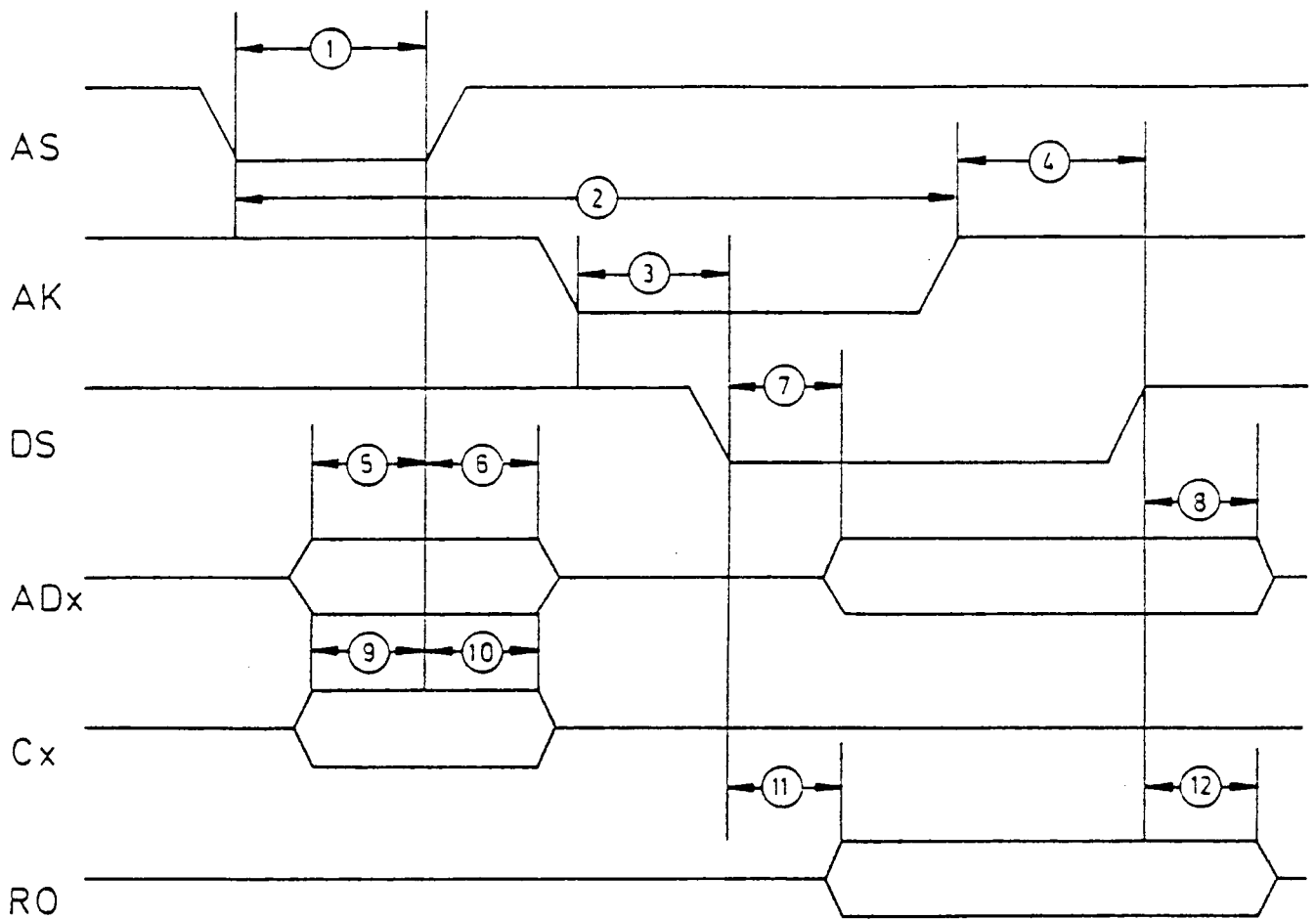
4.4 Platinenausführung

Dies ist keine Spezifikation, sondern lediglich eine Empfehlung. Alle K896-kompatiblen Baugruppen sollten mindestens als vierlagiger Multilayer ausgeführt sein, wobei in allen Fällen zwei Lagen für die Spannungsversorgung dienen sollten. Es ist empfehlenswert, die beiden Außenlagen der Platine für die Spannungsversorgung zu verwenden.

5. Anhang: K896 Busbelegung (Stecker ST-A der DCB-Baugruppen)
res. = reserved

PIN	Reihe: a	c		
1	GND	GND	Power	
2	+5V	+5V		
3	res.	BCLK	Bus Clock (10 MHz)	
4	GND	GND	Signal Return	
5	IPRQ	BB	Bus Arbitration	
6	BPS2	BRQ		
7	BPS0	BPS1		
8	POI	PFL	Power On/Fail	
9	GND	GND	Signal Return	
10	DS	AS	Bus Timing	
11	SC	AK		
12	C2	RO	Bus Control	
13	C0	C1		
14	AD1	AD0	Address/Data	
15	AD3	AD2		
16	AD5	AD4		
17	AD7	AD6		
18	AD9	AD8		
19	AD11	AD10		
20	AD13	AD12		
21	AD15	AD14		
22	GND	GND		Signal Return
23	AD17	AD16		Address/Data
24	AD19	AD18		
25	AD21	AD20		
26	AD23	AD22		
27	AD25	AD24		
28	AD27	AD26		
29	AD29	AD28		
30	AD31	AD30		
31	+5V	+5V	Power	
32	GND	GND		

Anhang: K896 Bus Timing



Signal	Symbol	Bedeutung	min.	max.
1	t AWS	Address strobe width	50	-
2	t BCT	Bus cycle time	-	5000
3	t ADL	Acknowledge low to data strobe low delay	25	-
4	t ADH	Acknowledge high to data strobe high delay	40	BM (b)
5	t AST	Address setup time	25	-
6	t AHT	Address hold time	10	-
7	t DSD	Data Strobe low to data valid delay	50	-
8	t DHT	Data hold time	0	-
9	t CST	Control code setup time	25	-
10	t CHT	Control code hold time	10	-
11	t DST	Data strobe low to feedback status valid delay	50	-
12	t RHT	Feedback status hold time	0	-

Hinweise:

- a) alle Größen in Nanosekunden (ns)
- b) der Wert BM wird vom momentanen Bus Master und dessen Geschwindigkeit bestimmt.

K896 BUS ARBITER

K896 INTERPROZESSOR-KOMMUNIKATION

Stand: 17. Januar 1982

Version: 1.1

Copyright by KONTRON MIKROCOMPUTER GmbH, Eching/München

Inhalt

Teil A: K896 Bus Arbiter

1. Allgemeines Funktionsprinzip
 - 1.1 Bus Priority Scheduler (BPS0...BPS2)
 - 1.2 Bus Request (BRQ)
 - 1.3 Bus Busy (BB)

2. Die Implementation eines K896-Bus Arbiters
 - 2.1 Steuersignale des Bus Arbiters
 - 2.2 Arbeitsweise der PAL's
 - 2.3 Zeitdiagramme

Teil B: K896 Interprozessor-Kommunikation

1. Allgemeines Funktionsprinzip

2. Die Implementation der Interprozessor-Kommunikation
 - 2.1 Steuersignale der Interprozessor-Kommunikation
 - 2.2 Arbeitsweise der PAL's
 - 2.3 Zeitdiagramme

Teil C: Anhang und Schaltpläne

K896 Bus Arbiter

K896 unterstützt bis zu 8 potentielle Bus Master in einem System. Potentielle Bus Master werden im folgenden oftmals als **'Prozessor'** bezeichnet, unabhängig davon, ob sie bereits Bus Master (Commander im Sinne von K896) sind oder sich erst bewerben, Bus Master zu werden. Ein sich bewerbender Prozessor wird auch als **'Requester'** bezeichnet.

Jedem Prozessor ist eine Priorität von 0 bis 7 zugeordnet. Definitionsgemäß gilt:

- 0 entspricht der höchsten Priorität
- 7 entspricht der niedersten Priorität

Es ist Aufgabe des sogenannten **'Bus Arbiter'** (deutsch: Bus Schiedsrichter) in folgenden Fällen zu einem eindeutig definierten Zeitpunkt für eine klare Entscheidung zu sorgen:

- a) Der Bus ist unbenutzt; **einer** der Prozessoren bewirbt sich um den Bus.
- b) Der Bus ist unbenutzt; **mehrere** Prozessoren bewerben sich **gleichzeitig** um den Bus.
- c) Der Bus ist benutzt; ein **weiterer** Prozessor bewirbt sich um den Bus.

Alle im Sinne von K896 potentiellen Bus Master müssen eine entsprechende Logik besitzen, um in den eben geschilderten Fällen ein eindeutig definiertes Verhalten zu zeigen. Die hier vorliegende Dokumentation beschreibt das allgemeine Funktionsprinzip der K896-Bus Arbitration, sowie als Implementationsbeispiel den Bus Arbiter der Prozessorbaugruppen der DCB-Baugruppenfamilie.

1. Allgemeines Funktionsprinzip der Bus Arbitration

Die Bus Arbitration erfordert 5 Leitungen des 64-poligen K896-Steckers:

BPS0...BPS2	(Bus Priority Scheduler)
BB	(Bus Busy)
BRQ	(Bus Request)

Außerdem wird der Bus Clock (BCLK) verwendet.

Die gesamte Arbitration Logik eines Mehrprozessorsystems setzt sich in K896 zusammen aus einem zentralen und einem dezentralen Teil. Der zentrale Teil wird als **'Bus Priority Scheduler'** bezeichnet und befindet sich auf der K896-Backplane. Der dezentrale Teil wird als **'Bus Arbiter'** bezeichnet und befindet sich in **funktional identischer** Form auf jeder Prozessorbaugruppe eines K896-Systems.

1.1 Bus Priority Scheduler (BPS0...BPS2)

Die zentrale Steuereinheit 'Bus Priority Scheduler' liefert periodisch die binär codierten Adressen aller potentiellen Bus Master über die Leitungen BPS0 bis BPS2 (BPSx-Leitungen). Der Bus Clock (BCLK) eines K896-Systems bestimmt unmittelbar die Zähhfrequenz des Bus Priority Schedulers. Sie beträgt normalerweise 10 MHz. Daraus ergibt sich eine Zykluszeit von 800 ns, bis alle potentiellen Bus Master abgetastet sind. Dies wiederum entspricht einer mittleren Zeit von 400 ns (latency time) die ein Requester warten muß, bis er im Falle eines unbenutzten Bus diesen zugesprochen bekommt.

Der Bus Priority Scheduler bestimmt in eindeutiger Weise den Zeitpunkt, zu dem ein Requester den Bus erhält. Dieser Zeitpunkt ist dann gekommen, wenn die Adresse des Requesters auf den BPSx-Leitungen erscheint und die Bus Busy Leitung (BB) nicht aktiv ist. Daraus ergibt sich der oberste Grundsatz des K896-Arbiter Systems:

Ein Requester kann den Bus nur dann erhalten, wenn die Bus Busy Leitung inaktiv ist. Es ist Aufgabe eines Requesters, zum Zeichen der Bus-Übernahme die Bus Busy Leitung zum oben geschilderten Zeitpunkt zu aktivieren.

Da der Bus Priority Scheduler für jeden potentiellen Requester einen absolut eindeutigen Reaktionszeitpunkt festlegt, gibt es keine Konfliktsituationen für den Fall, daß mehrere Requester gleichzeitig den Bus beanspruchen. Hat ein Prozessor den Bus einmal bekommen, so ist er definitionsgemäß zum **'Commander'** geworden.

Daraus leitet sich der zweite Grundsatz des K896-Arbiter Systems ab:

Nur der momentane Commander bestimmt den Zeitpunkt der Wiederfreigabe des Bus, selbst dann, wenn ein höher priorisierter Prozessor den Bus beansprucht.

Grundsätzlich sollte allerdings gelten:

- a) ist kein höher priorisierter Requester vorhanden, so sollte ein Commander den Bus nur solange belegen, solange es für seine Aktivitäten notwendig ist.
- b) ist ein höher priorisierter Requester vorhanden, so sollte ein Commander den Bus **sobald wie möglich** abgeben.

K896 definiert weder eine dafür zulässige Maximalzeit, noch das Verhalten eines Commanders nach dem Erkennen eines höher priorisierten Requesters.

1.2 Bus Request (BRQ)

Ist der Bus von einem Prozessor beliebiger Priorität belegt, so zeigt ein Requester über die Bus Request Leitung an, daß er den Bus beansprucht. Hierzu aktiviert ein Requester die Bus Request Leitung während des Zeitfensters, wo seine Adresse auf den BPSx-Leitungen erscheint. Auf diese Weise teilt er gleichzeitig seine Priorität dem momentanen Commander mit, der nun entscheidet, ob er den Bus freigeben will. Wie bereits erwähnt, kann kein Commander von außen gezwungen werden, den Bus freizugeben. Wie schnell er dies im Falle eines höher priorisierten Requesters tut, hängt im Einzelfall von Hard- und Software eines Commanders ab.

1.3 Bus Busy (BB)

Die Bus Busy Leitung ist die einzige statische Leitung des K896-Arbiter Systems. Sie sagt aus, ob der Bus momentan von einem Prozessor benutzt wird. Ist der Bus unbenutzt, so aktiviert ein Requester die Bus Busy Leitung ab dem Zeitpunkt, wo seine Adresse auf den BPSx-Leitungen erscheint.

2. Die Implementation eines K896-Bus Arbiters

Die hier vorgestellte Implementation ist auf den Z8001-basierenden Prozessorbaugruppen DCB/CPU und DCB/CMM realisiert. Der Bus Arbiter ist bezüglich seiner Funktion prozessor-unabhängig, so daß die hier vorgestellte Realisierung auch auf anderen Prozessorbaugruppen Anwendung finden kann.

Lediglich Erzeugung und Wirkung der mit (*) gekennzeichneten Signale sind Z8001- oder Baugruppen-spezifisch. Die Schaltung ist mit zwei PAL Bausteinen realisiert, die außerdem die Logik zur Interprozessor-Kommunikation enthalten. Daneben wird zur Einstellung der Prozessorpriorität ein 3-fach Schalter benötigt: PPS0...PPS1 Processor Priority Switch.

2.1 Steuersignale des Bus Arbiters

Zur besseren Verständlichkeit sollen hier zunächst die in den Schaltbildern verwendeten Symbole erläutert werden.

Symbol	Bedeutung
BCLK	Bus Clock
BB	Bus Busy
BRQ	Bus Request
LBRQ (*)	Local Bus Request
LBAK (*)	Local Bus Acknowledge
HPR (*)	High Priority Request
HPRA (*)	High Priority Request Acknowledge
BPSx	Bus Priority Scheduler (3 lines)
PPSx	Processor Priority Switch (3 lines)

Alle Signale sind 'low aktiv'; ihre Funktion ist in folgendem beschrieben:

a) Local Bus Request (LBRQ)

Der Prozessor zeigt dem Bus Arbiter mit diesem Signal an, daß er den Bus beansprucht.

b) Local Bus Acknowledge (LBAK)

Der Bus Arbiter zeigt dem Prozessor mit diesem Signal an, daß er den Bus nun bekommen hat. Der Bus Arbiter hält die Bus Busy Leitung aktiv, bis der Prozessor das LBRQ-Signal deaktiviert.

c) High Priority Request (HPR)

Der Bus Arbiter zeigt mit diesem Signal an, daß ein höher priorisierter 'Requester' den Bus beansprucht. Die Wirkung dieses Signals ist implementations-abhängig. Auf den Prozessorbaugruppen der DCB-Familie verursacht das Signal HPR einen NVI (Non Vectored Interrupt).

Für hoch priorisierte Prozessoren ist es denkbar, das HPR-Signal überhaupt nicht auszuwerten. Höher priorisierte Prozessoren haben dann allerdings keine Möglichkeit mehr, ihre Busbeanspruchung einem solchen Prozessor kund zu tun.

d) High Priority Request Acknowledge (HPRA)

Der Prozessor zeigt dem Bus Arbiter mit diesem Signal an, daß ein 'High Priority Request' akzeptiert wurde, worauf der Bus Arbiter das HPR-Signal wieder wegnimmt. Ein Prozessor sollte unmittelbar darauf das Signal LBRQ (Local Bus Request) deaktivieren, um den Bus für den höher priorisierten Requester freizugeben. Auf den Prozessorbaugruppen der DCB-Familie ist das Signal HPRA identisch mit dem Signal NVIAK (Non Vectored Interrupt Acknowledge).

Bild 1 zeigt den Stromlaufplan eines allgemeinen K896-kompatiblen Bus Arbiters, wobei hier bewußt die eben definierten Symbole verwendet wurden. Im Anhang ist schließlich die entsprechende Schaltung der DCB-Familie zusammen mit der Logik zur Interprozessor-Kommunikation dargestellt. Timing Diagramme sind den Bildern 2 bis 5 zu entnehmen.

2.2 Arbeitsweise der PAL's (Bild 1)

PAL 1 (PAL16R6) enthält das synchrone Steuerwerk des Bus Arbiters einer K896-kompatiblen Prozessorbaugruppe. PAL 2 (PAL16L8) arbeitet im wesentlichen als Vergleicher.

Das Zeitverhalten des synchronen Steuerwerks (PAL1) ist in den Bildern 2 und 3 dargestellt. Infolge der synchronen Arbeitsweise des Bus Arbiters zeigt sich die Wirkung eines Signals bzw. einer Signalverbindung grundsätzlich einen Taktzyklus später.

Beispiel:

Prozessor #4 beansprucht den Bus, dann aktiviert der Bus Arbiter von Prozessor #4 die Bus Request Leitung, während auf den BPSx-Leitungen die Adresse 5 erscheint.

PAL 2 (PAL16L8) liefert die für PAL1 erforderlichen Vergleichssignale.

a) BPSx = PPSx

Der Stand des Bus Priority Schedulers entspricht der Prozessor-eigenen Priorität.

b) BPSx # 7

Der Stand des Bus Priority Schedulers ist ungleich sieben. PAL 1 benötigt dieses Signal für ein internes Hilfssignal (IHS). Es wird außerhalb von PAL 1 nicht benötigt.

2.3 Zeitdiagramme

Die Bilder 2 und 3 enthalten Zeitdiagramme für folgende Fälle:

Fall 1: Der Bus ist unbenutzt; Prozessor #4 beansprucht und bekommt den Bus (Bild 2 oben).

Fall 2: Der Bus ist unbenutzt; die Prozessoren #2 und #4 beanspruchen gleichzeitig den Bus. Prozessor #4 erhält den Bus (Bild 2 unten).

Fall 3: Prozessor #4 hat den Bus; Prozessor #2 beansprucht ihn als höher priorisierter Requester (Bild 3 oben).

Fall 4: Prozessor #4 gibt den Bus ab; Prozessor #2 erhält ihn (Bild 3 unten).

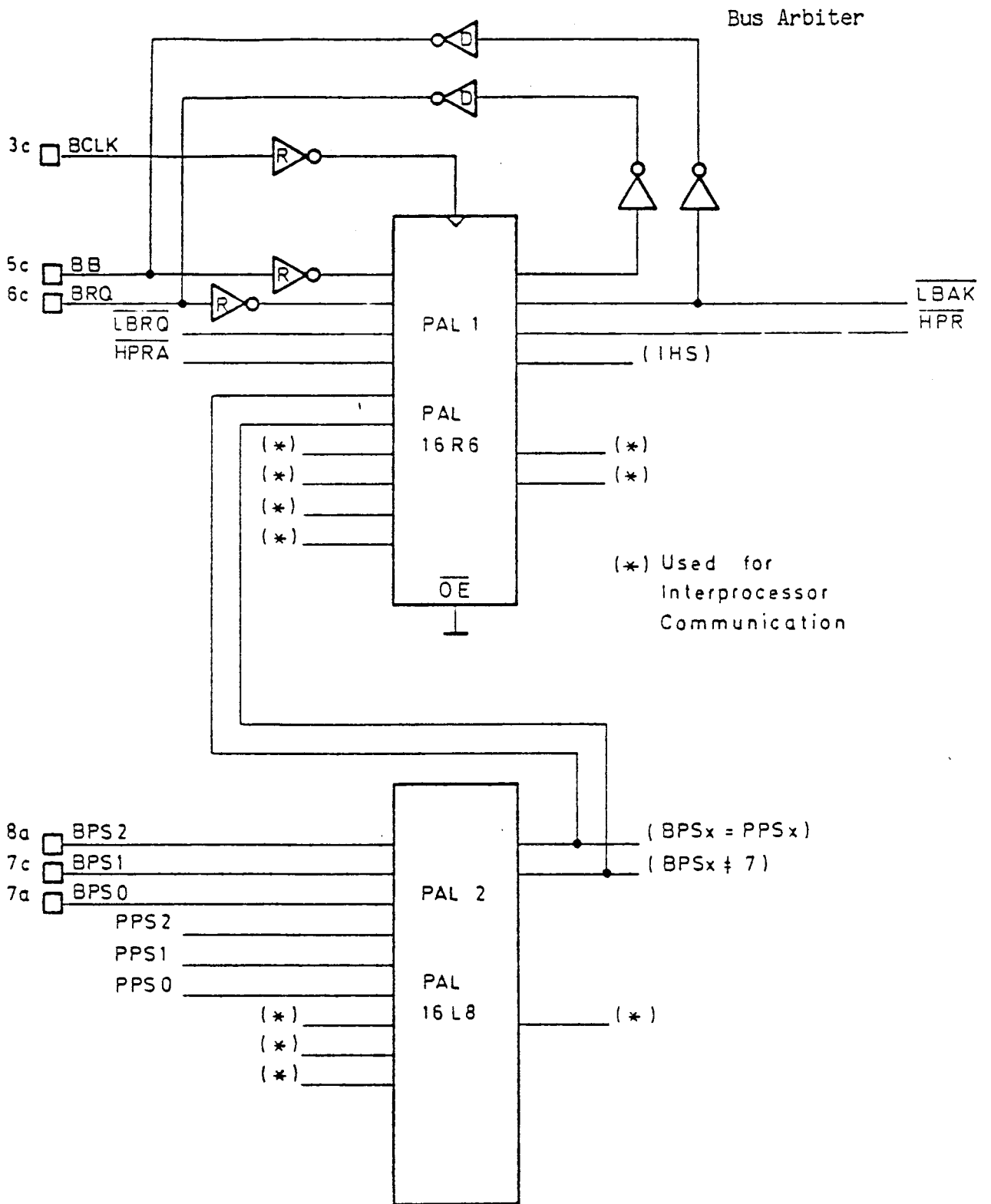


Bild 1: Schaltung des K896 Bus Arbiter

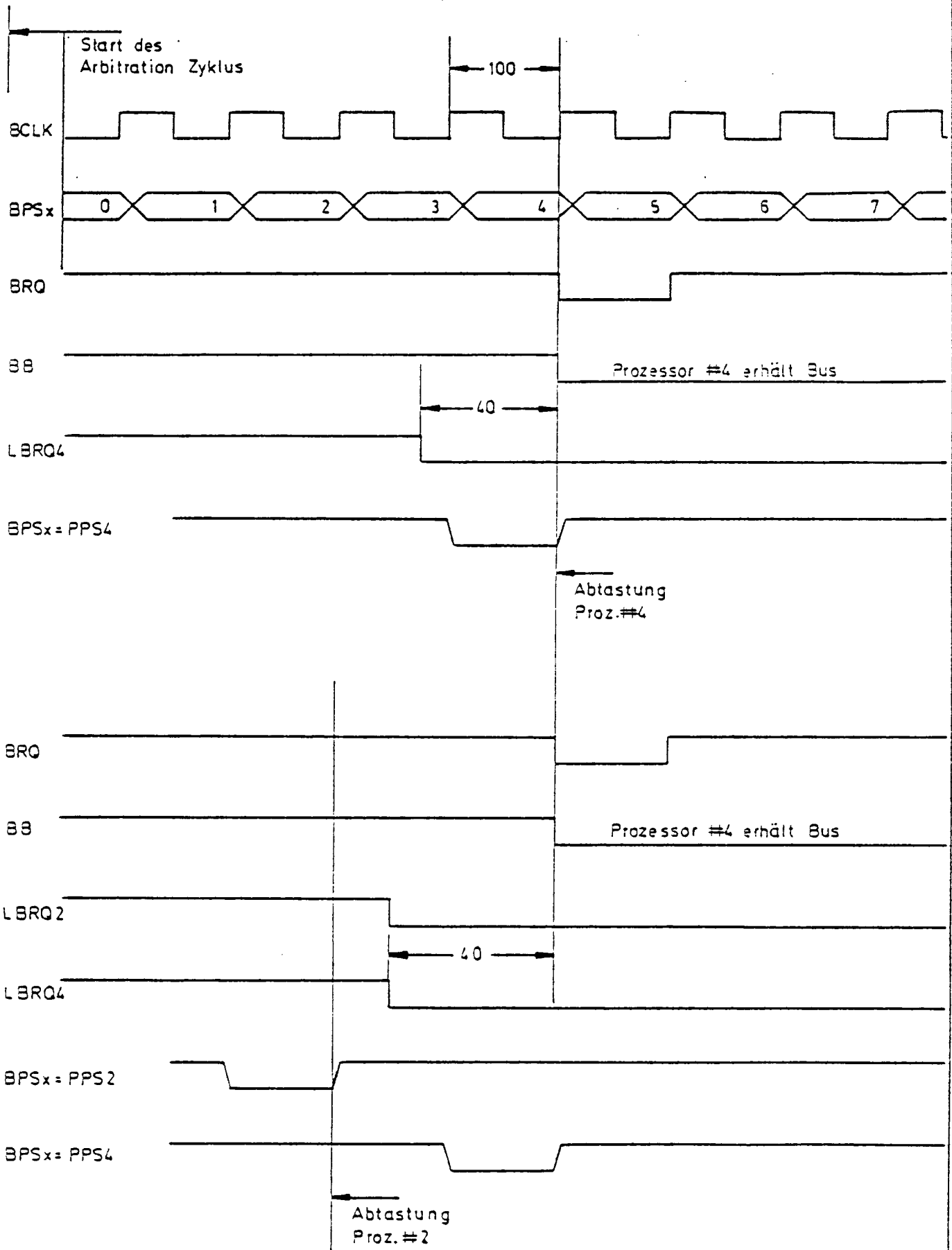


Bild 2: Impulsdiagramm zweier typischer Arbitration Zyklen (Fälle 1 und 2)

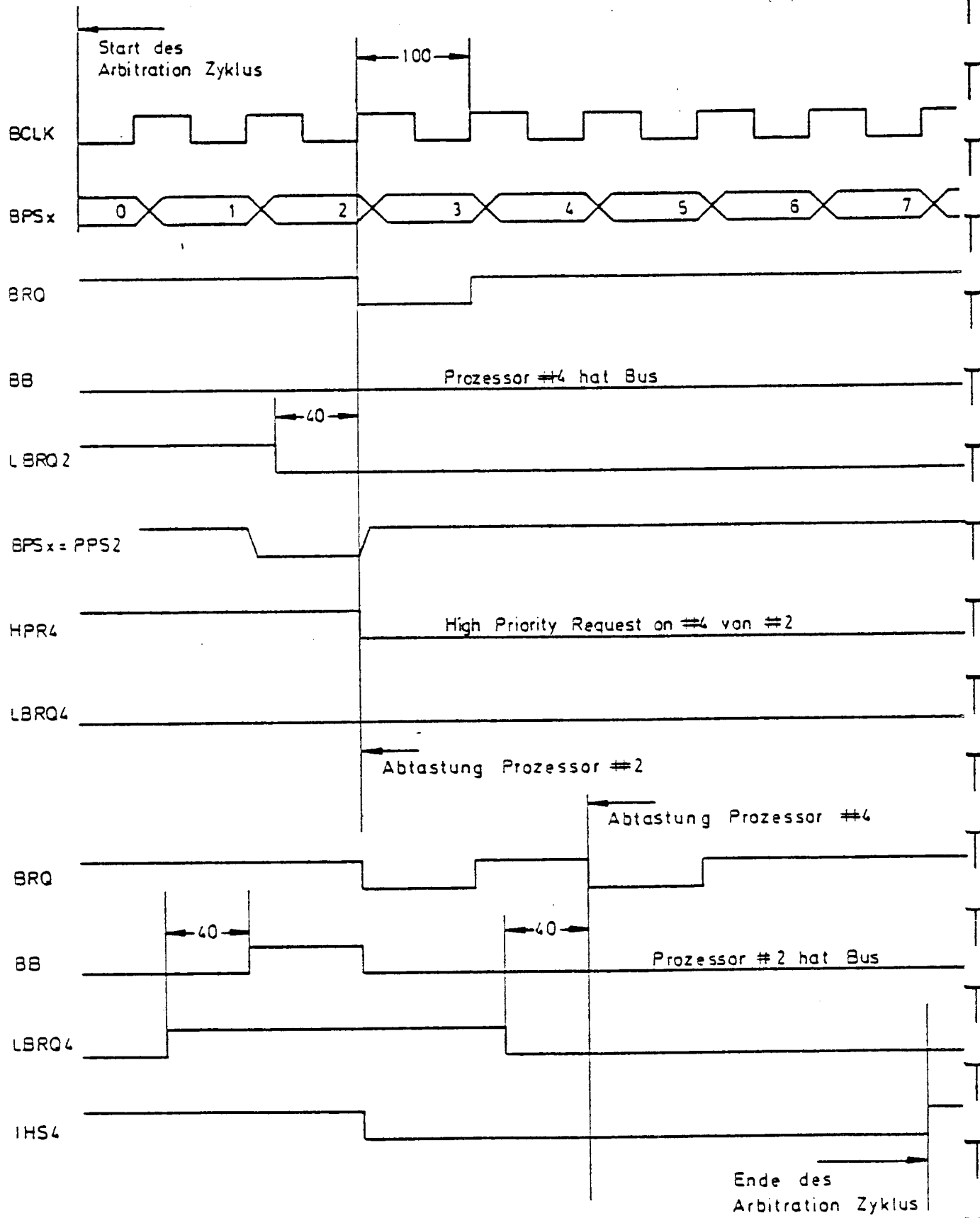


Bild 3: Impulsdiagramm zweier typischer Arbitration Zyklen (Fälle 3 und 4)

K896 Interprozessor-Kommunikation

K896 unterstützt bis zu 8 potentielle Bus Master in einem System. Potentielle Bus Master werden im folgenden oftmals als 'Prozessor' bezeichnet.

Das in K896 definierte Verfahren der Interprozessor-Kommunikation ermöglicht es jedem Prozessor, ein Interruptsignal zu einem beliebigen Zielprozessor zu schicken. Es ist nicht notwendig, daß der Sender momentan Bus Master (Commander) ist. Die Adresse des Zielprozessors ist durch dessen Priorität (siehe Bus Arbiter) festgelegt.

1. Allgemeines Funktionsprinzip

Die Interprozessor-Kommunikation erfordert eine Leitung des 64-poligen K896-Steckers:

IPRQ (Inter Processor Request)

Die Interprozessor-Kommunikation basiert, wie der Bus Arbiter, auf dem zentralen Bus Priority Scheduler, sowie einer dezentralen Steuerlogik, die in funktional identischer Form auf jeder K896-kompatiblen Prozessorbaugruppe vorhanden sein muß.

Will ein Prozessor die Aufmerksamkeit eines anderen erregen, so aktiviert er die IPRQ-Leitung während des Zeitfensters, wo die Adresse des 'anderen' auf den BPSx-Leitungen erscheint. Die Kommunikations-Steuerlogik des Zielprozessors erkennt, daß der Request über die IPRQ-Leitung für den eigenen Prozessor bestimmt war und gibt eine entsprechende Meldung an diesen weiter, beispielsweise als Interrupt.

Da keine Informationen von Prozessor zu Prozessor übergeben werden können, ist im allgemeinen ein sogenanntes 'Mail Box Memory' notwendig, über das die einzelnen Prozessoren ihre Informationen austauschen.

2. Die Implementation der Interprozessor-Kommunikation

Die hier vorgestellte Implementation ist auf den Z8001-basierenden Prozessorbaugruppen DCB/CPU und DCB/CMM realisiert. Die Steuerlogik zur Interprozessor-Kommunikation ist bezüglich seiner Funktion processor-unabhängig, so daß die im folgenden beschriebene Realisierung auch auf anderen Prozessorbaugruppen Verwendung finden kann. Lediglich Erzeugung und Wirkung der mit (*) gekennzeichneten Signale sind Z8001- oder Baugruppen-spezifisch.

Die Schaltung ist, zusammen mit dem Bus Arbiter, mit zwei PAL-Bausteinen realisiert.

2.1 Steuersignale der Interprozessor-Kommunikation

Symbol	Bedeutung
BCLK	Bus Clock
IPRQ	Interprocessor Request
BPSx	Bus Priority Scheduler
PPSx	Priority Switch
LIPRQ (*)	Local Interprocessor Request
LIPRQA (*)	Local Interprocessor Request Acknowledge
PRI (*)	Processor Request Interrupt
PRIA (*)	Processor Request Interrupt Acknowledge
PA0...PA2 (*)	Zielprozessor Adresse

a) Local Interprocessor Request (LIPRQ)

Dieses Signal zeigt der Steuerlogik an, daß ein Request an den Zielprozessor mit der Adresse PAX abzugeben ist.

b) Local Interprocessor Request Acknowledge (LIPRQA)

Die Steuerlogik zeigt mit diesem Signal an, daß der Request an den Zielprozessor abgesetzt wurde. Das Signal LIPRQ muß daraufhin weggenommen werden, sonst wird beim nächsten Zyklus des Bus Priority Schedulers ein erneuter Request an den Zielprozessor abgesandt.

c) Processor Request Interrupt (PRI)

Die Steuerlogik informiert über dieses Signal den Prozessor, daß ein Request für ihn über den Bus abgegeben wurde.

d) Processor Request Interrupt Acknowledge (PRIA)

Der Prozessor bestätigt mit diesem Signal die Annahme des Requests. Die Steuerlogik nimmt daraufhin das PRI-Signal wieder weg.

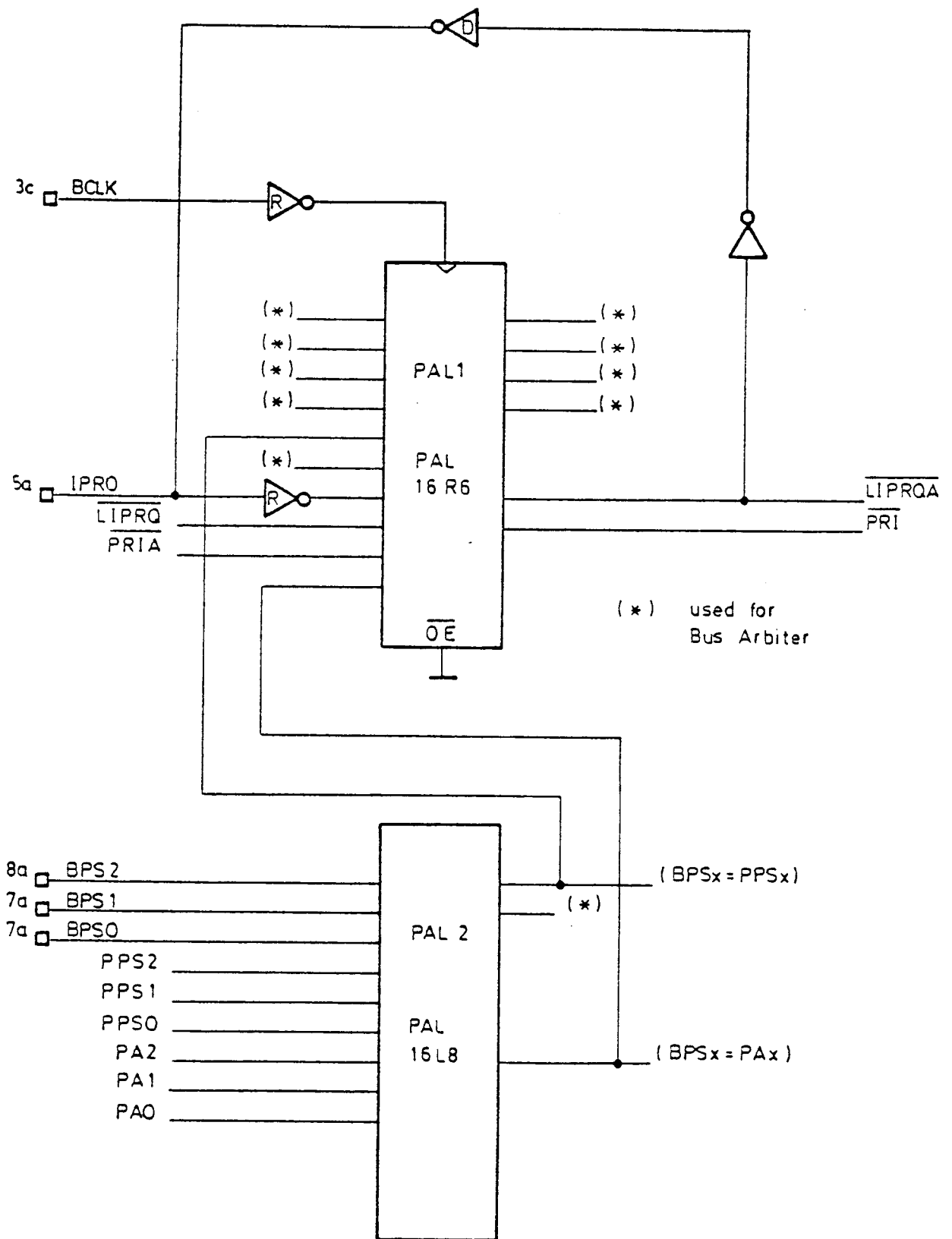
2.2 Arbeitsweise der PAL's (Bild 1)

PAL 1 (PAL16R6) enthält das synchrone Steuerwerk der K896-kompatiblen Interprozessor-Kommunikation. PAL 2 arbeitet als Vergleicher und liefert das Signal BPSx=Pax, welches aussagt, daß die BPSx-Leitungen momentan die Adresse Pax des Zielprozessors führen, sowie das Signal BPSx=PPSx, welches aussagt, daß die BPSx-Leitungen momentan die Prozessor-eigene Adresse führen.

Das Baugruppen-interne Signal LIPRQ (Local Interprocessor Request) initiiert den Vorgang der Interprozessor-Kommunikation. Die Steuerlogik setzt zum nächst möglichen Zeitpunkt (BPSx = Pax) einen Request über die Busleitung IPRQ ab. Dieser Request sollte Baugruppen-intern als LIPRQA (Local Interprocessor Request Acknowledge) zur Wegnahme des Signals LIPRQ verwendet werden. Ist dieses Signal beim nächsten Zyklus des Bus Priority Schedulers immer noch aktiv, so setzt PAL 1 einen weiteren Request über den Bus ab.

Auf den DCB-Baugruppen wird das Signal LIPRQ von einem Flipflop abgeleitet, das durch die Special I/C-Befehle der Z8001-CPU softwaremäßig setzbar ist. Durch das Signal LIPRQA wird dieses Flipflop wieder rückgesetzt. Das Signal PRI (Processor Request Interrupt) generiert auf den DCB-Prozessorbaugruppen indirekt einen Non Vectored Interrupt (NVI).

Bild 1: Schaltung der K896 Interprozessor-Kommunikation



2.3 Zeitdiagramme

Bild 2 zeigt das Zeitdiagramm für den Fall, daß Prozessor #2 einen Request zum Zielprozessor #4 sendet.

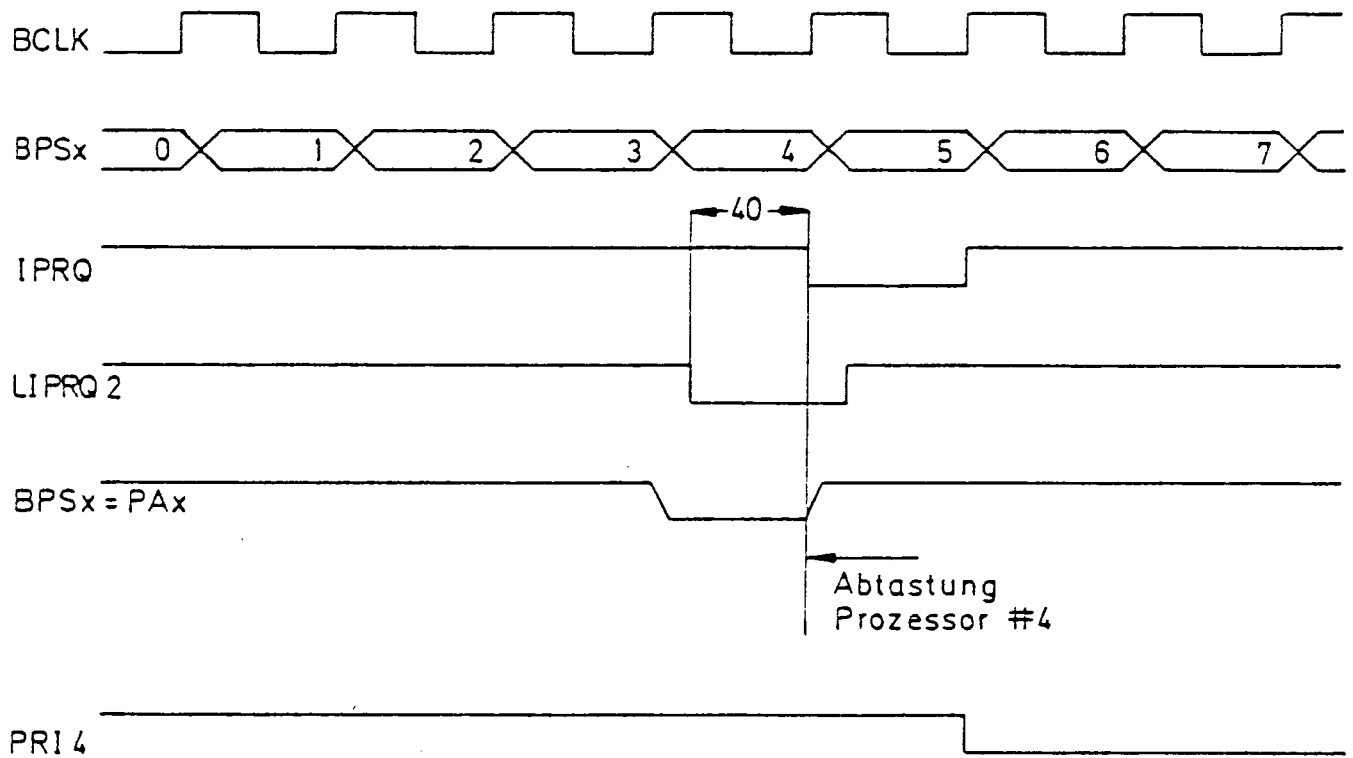


Bild 2: Impulsdiagramm der Interprozessor-Kommunikation

Teil C: Anhang zur Dokumentation über K896 Bus
 Arbiter und K896 Interprozessor-Kommunikation

Der Anhang enthält den Schaltplan der Realisation von Bus Arbiter und Interprozessor-Kommunikation auf den DCB-Baugruppen DCB/CPU und DCB/CMM.

Die Prozessor-spezifischen Signale sind:

allgemeines Signal	spezifisches Signal
LBRQ	MO (Multi micro out)
LBAK	MI (Multi micro in)
HPR	NVI (Non vectored interrupt)
HPRA	NVIAK (NVI acknowledge)
LIPRQ	SIO.REF (special I/O-reference über D-Flipflop)
PRI	NVI (non vectored interrupt)
PAX	LAX (latched address)

Hinweis:

HPR (High priority request) und PRI (Processor request interrupt) verursachen beide einen NVI (Non vectored interrupt). HPR und PRI werden in PAL 2 'Oder-verknüpft' und von dort an den NVI-Eingang der Z8001-CPU geführt. Während des NVI-Acknowledge Cycles liefert PAL 2 außerdem über das Datenbit DO (Anschluß: IADO) einen Hinweis, was den NVI verursacht hat:

DO = 0 NVI verursacht durch HPR
 DO = 1 NVI verursacht durch PRI

