

IEC-821-VMEbus – Mikroprozessor-Systembus für 1 Byte bis 4 Byte Daten

(IEC 821:1991, modifiziert)
Deutsche Fassung EN 60821:1994

DIN
EN 60821

Diese Norm enthält die deutsche Übersetzung der Internationalen Norm **IEC 821**

ICS 31.080.00

IEC-821-VMEbus – Microprocessor system bus for 1 byte to 4 byte data
(IEC 821:1991, modified);
German version EN 60821:1994

Bus CEI-821-VMEbus – Bus système à microprocesseurs pour données de 1 octet à 4 octets
(CEI 821:1991, modifiée);
Version allemande EN 60821:1994

Die Europäische Norm EN 60821 : 1994 hat den Status einer Deutschen Norm.

Nationales Vorwort

Der Entwurf zu dieser Norm war veröffentlicht als DIN IEC 821 : 1991-09.

Diese Norm wurde vom zuständigen Gemeinschaftsunterkomitee GUK 631.10 "Mikroprozessor-Systeme" der Deutschen Elektrotechnischen Kommission im DIN und VDE (DKE) und des Normenausschusses Informationsverarbeitungssysteme (NI) im DIN bearbeitet.

Diese Norm enthält die deutsche Übersetzung der Internationalen Norm IEC 821 : 1991 "IEC 821 VMEbus – Microprocessor system bus for 1 byte to 4 byte data", die vom Europäischen Komitee für Elektrotechnische Normung (CENELEC) als Europäische Norm EN 60 821 mit gemeinsamen Abänderungen übernommen wurde, siehe Hinweise zu den gemeinsamen Abänderungen.

Der Text der Internationalen Norm enthält neben den gesetzlichen Einheiten auch die Einheiten "Foot" (FT), "Inch" (in), "lbf" und "oz".

Es wird ausdrücklich darauf hingewiesen, daß die Anwendung dieser Einheiten im nationalen amtlichen und geschäftlichen Verkehr aufgrund des Gesetzes über Einheiten im Meßwesen vom 22. 02. 1985 nicht zulässig ist.

Den in dieser Norm genannten IEC-Publikationen entsprechen die folgenden Europäischen bzw. Deutschen Normen:

Europäische Norm	Internationale Norm	Deutsche Norm
HD 493.1 S1	IEC 297-1	–
HD 493.3 S2	IEC 297-3	DIN IEC 297-3
–	IEC 603-2	Normen der Reihe DIN 41612
HD 576 S1	IEC 822	DIN IEC 822
–	IEC 823	–

Bei der Übersetzung wurden deutsche Begriffe verwendet, soweit dies möglich war. Es wurde dabei jedoch berücksichtigt, daß sich im Bereich der Mikroprozessor-Systemtechnik auch im deutschen Sprachgebrauch vielfach englische Fachbegriffe durchgesetzt haben, die zum Teil deutsche Benennungen verdrängen (Beispiel: *Interrupt request* anstelle von *Unterbrechungsanforderung*). Zur leichteren Anwendung dieser Norm wurden insbesondere die verwendeten Namen von Signalen und Datentransfers nicht übersetzt.

Die Internationale Norm enthält eine Vielzahl von Festlegungen zum Systementwurf, die in Form von VORSCHRIFTEN, EMPFEHLUNGEN, VORSCHLÄGEN, ERLAUBNISSEN und ERLÄUTERUNGEN hervorgehoben und numeriert sind.

Fortsetzung Seite 2
und 177 Seiten EN

Deutsche Elektrotechnische Kommission im DIN und VDE (DKE)
Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.

Im Zuge der Überarbeitung von IEC 821, 1. Ausgabe 1987, wurden verschiedene dieser Festlegungen neu formuliert bzw. gestrichen. Dabei wurde von einer Umnummerierung im Hinblick auf die große Anzahl von eingesetzten VMEbus-Systemen abgesehen, so daß in dieser 2. Ausgabe von IEC 821 die numerische Reihenfolge der Festlegungen nicht mehr gegeben ist.

Zu den Nationalen Fußnoten A und B:

- A) Die gemeinsamen Abänderungen beziehen sich auf verschiedene inhaltliche Fehler in der Internationalen Norm und berücksichtigen diverse Druckfehler. Der Text hierfür wurde vom Deutschen Komitee ausgearbeitet. Er ist in die Norm eingearbeitet und am äußeren Seitenrand mit senkrechten Linien gekennzeichnet.
- B) Das in VORSCHRIFT 5.1 beschriebene Verhalten ist das erwünschte Systemzeitverhalten. Es ist jedoch nicht möglich, dieses mit dem POWER-MONITOR-Modul stets sicherzustellen.

Zu den gemeinsamen Abänderungen

Der Text der geänderten VORSCHRIFT 3.14 in Unterabschnitt 3.3.2 lautet im IEC-Original:

“Ein FAIR-REQUESTER MUSS seine Busanforderungsleitung abfragen, wenn festgestellt wird, daß BBSY* high ist, bevor er eine Busanforderung erzeugt.”

Nationaler Anhang NA (informativ)

Literaturhinweise in nationalen Zusätzen

DIN 41612-1	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Gemeinsame Einbaumerkmale, Bauformenübersicht
Beiblatt 1 zu DIN 41612-1	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Allgemeine Hinweise
DIN 41612-2	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauformen B, C und D
DIN 41612-3	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauformen F und G
DIN 41612-4	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauform H
DIN 41612-5	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Kennwerte, Anforderungen, Prüfungen
DIN 41612-6	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauformen Q, R und S
DIN 41612-7	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauform U und V
DIN 41612-8	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauform M
Beiblatt 1 zu DIN 41612-8	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauform M; Übersicht anwendbarer Sonderkontakte für Bauform M
DIN 41612-9	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Maße der Bauform E
DIN 41612-10	Steckverbinder für gedruckte Schaltungen, indirektes Stecken, Rastermaß 2,54 mm – Lehren
DIN IEC 297-3	Maße der 482,6-mm-Bauweise (19-in-Bauweise) – Teil 3: Baugruppenträger und Baugruppen (IEC 297-3 : 1984 + A1 : 1992) Deutsche Fassung HD 493.3 S2 : 1993
DIN IEC 822	IEC 822 VSB – Paralleler Untersystembus für den IEC 821 VMEbus; (Identisch mit IEC 822 : 1988); Englische Fassung HD 576 S1 : 1990

Deutsche Fassung

**IEC-821-VMEbus – Mikroprozessor-Systembus
für 1 Byte bis 4 Byte Daten**

(IEC 821:1991, modifiziert)

IEC 821 VMEbus – Microprocessor system
bus for 1 byte to 4 byte data
(IEC 821 : 1991, modified)

Bus CEI 821 VMEbus – Bus système à
microprocesseurs pour données de 1 octet à
4 octets
(CEI 821 : 1991, modifiée)

Diese Europäische Norm wurde von CENELEC am 1993-12-08 angenommen.

Die CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien und dem Vereinigten Königreich.

CENELEC

EUROPÄISCHES KOMITEE FÜR ELEKTROTECHNISCHE NORMUNG

European Committee for Electrotechnical Standardization

Comité Européen de Normalisation Electrotechnique

Zentralsekretariat: rue de Stassart 35, B-1050 Brüssel

Vorwort

Das CENELEC-Fragebogenverfahren zur unveränderten Annahme der Internationalen Norm IEC 821 : 1991 ergab, daß für die Annahme als Europäische Norm einige gemeinsame Abänderungen notwendig waren.

Das Referenzdokument, mit vom deutschen nationalen Komitee von CENELEC ausgearbeiteten gemeinsamen Abänderungen, wurde danach den CENELEC-Mitgliedern zur formellen Abstimmung vorgelegt.

Der Text des Entwurfs wurde von CENELEC am 1993-12-08 als EN 60821 genehmigt.

Diese Europäische Norm ersetzt HD 524 S1 : 1989.

Nachstehende Daten wurden festgelegt:

- spätestes Datum der Veröffentlichung einer identischen nationalen Norm (dop) : 1994-12-01
- spätestes Datum für die Zurückziehung entgegenstehender nationaler Normen (dow) : 1994-12-01

Anhänge, die als "normativ" bezeichnet sind, gehören zum Norminhalt. Anhänge, die als "informativ" bezeichnet sind, enthalten nur Informationen. In dieser Norm sind die Anhänge A, C und D informativ und die Anhänge B, E und ZA sind normativ.

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 821 : 1991 wurde von CENELEC mit nachstehenden^{A)} gemeinsamen Abänderungen als Europäische Norm angenommen

Inhalt

	Seite		Seite
Vorwort	2	2.3.5 Adressierung	23
Kapitel 0: Einführung		2.3.6 Datentransfer	24
0.1 Anwendungsbereich	7	2.3.7 Blocktransfer	25
0.2 Normative Verweisungen	7	2.3.8 Read-Modify-Write	26
0.3 Hinweis für den Leser	7	2.3.9 Nicht an Wortgrenzen gebundener Transfer	26
Kapitel 1: Einführung in die IEC-821-BUS-Norm		2.3.10 ADDRESS-ONLY-Fähigkeit	29
1.1 Zielsetzung der Bus-Norm	7	2.3.11 Wechselwirkung zwischen den DTB-Funktionsmodulen	30
1.2 Systemelemente	7	2.4 Typischer Betrieb	30
1.2.1 Grundbegriffe	7	2.4.1 Typische Datentransferzyklen	30
1.2.1.1 Begriffe zur Beschreibung des mechanischen Aufbaus	7	2.4.2 Adreß-Pipelining	31
1.2.1.2 Begriffe zur Beschreibung des funktionellen Aufbaus	8	2.5 Datentransferbus-Übernahme	31
1.2.1.3 Zyklusarten des IEC-821-BUSSES	9	2.6 Vorschriften und Erläuterungen zum DTB-Zeitverhalten	36
1.2.2 Allgemeiner Aufbau des IEC-821-BUSSES	9	Kapitel 3: IEC-821-BUS, Zuteilung des Datentransferbus	
1.3 Diagramme in der IEC-821-BUS-Norm	12	3.1 Philosophie der Buszuteilung	66
1.4 In der Norm verwendete Terminologie	12	3.1.1 Arten der Zuteilung	66
1.4.1 Zustände auf den Signalleitungen	12	3.2 Zuteilungsbus-Leitungen	66
1.4.2 Verwendung des Sternchens (*)	13	3.2.1 Busanforderungs- und Buszuteilungsleitungen	66
1.5 Protokollarten	13	3.2.2 Bus-Busy-Leitung (BBSY*)	68
1.5.1 Verschränkte Bussignale	13	3.2.3 Bus-Clear-Leitung (BCLR*)	68
1.5.2 Broadcast-Bussignal	14	3.3 Funktionsmodule	68
1.6 Systembeispiele und Erklärungen	14	3.3.1 ARBITER	69
Kapitel 2: IEC-821-BUS – Datentransfer		3.3.2 REQUESTER	69
2.1 Einführung	14	3.3.3 Datentransferbus MASTER	71
2.2 Leitungen des Datentransferbus (DTB)	14	3.3.3.1 DTB-Freigabe	71
2.2.1 Adreßleitungen	14	3.3.3.2 DTB-Übernahme	73
2.2.2 Adreßmodifizierer-Leitungen	16	3.3.3.3 Weitere Informationen	73
2.2.3 Datenleitungen	18	3.4 Typischer Betrieb	73
2.2.4 Steuerleitungen	18	3.4.1 Verwaltung von Busanforderungen auf zwei verschiedenen Ebenen	73
2.2.4.1 AS*	18	3.4.2 Verwaltung von zwei Busanforderungen auf der gleichen Busanforderungsleitung	74
2.2.4.2 DS0* und DS1*	18	3.5 Laufzeitbedingte Konflikte zwischen MASTER-Busanforderungen und ARBITER-Zuteilungen	74
2.2.4.3 DTACK*	19	Kapitel 4: IEC-821-BUS, INTERRUPT-BUS	
2.2.4.4 BERR*	19	4.1 Einführung	81
2.2.4.5 WRITE*	19	4.1.1 Zentrale Systeme	81
2.3 DTB-Module – Grundlagen	19	4.1.2 Verteilte Systeme	81
2.3.1 MASTER	20	4.2 Interrupt-Bus-Leitungen	81
2.3.2 SLAVE	20		
2.3.3 BUS TIMER	20		
2.3.4 LOCATION MONITOR	20		

^{A)} Nationale Fußnote, siehe Nationales Vorwort

	Seite
4.2.1 Interrupt-Request-Leitungen	81
4.2.2 Interrupt-Acknowledge-Leitung – IACK*	81
4.2.3 Interrupt-Acknowledge-Daisy-Chain – IACKIN*/IACKOUT*	81
4.3 Interrupt-Bus-Module, Grundlagen	81
4.3.1 INTERRUPT HANDLER	83
4.3.2 INTERRUPTER	84
4.3.3 IACK-DAISY-CHAIN-TREIBER	86
4.3.4 Interrupt-Handling-Fähigkeiten	86
4.3.5 Interrupt-Request-Fähigkeiten	88
4.3.6 Fähigkeiten des STATUS/ID-Transfers	88
4.3.7 Möglichkeiten der Interrupt-Beendigung	88
4.3.8 Wechselwirkung zwischen den Interrupt-Bus-Modulen	89
4.4 Typischer Betrieb	92
4.4.1 Ablauf im zentralen Interrupt-System	92
4.4.2 Ablauf im verteilten Interrupt-System	92
4.4.2.1 Verteilte Interrupt-Systeme mit sieben INTERRUPT HANDLERN	92
4.4.2.2 Verteilte Interrupt-Systeme mit zwei bis sechs INTERRUPT HANDLERN	93
4.4.3 Beispiel: Typischer Ablauf im zentralen Interrupt-System	93
4.4.4 Beispiel: Priorisierung von zwei Interrupts in einem verteilten Interrupt-System	93
4.5 Laufzeitbedingte Konflikte	97
4.6 VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten des Interrupt-Bus	97

Kapitel 5: IEC-821-BUS, UTILITY-BUS

5.1 Einführung	115
5.2 Signalleitungen des Utility-Bus	115
5.3 Utility-Bus-Module	115
5.3.1 SYSTEMAKTTREIBER	115
5.3.2 SERIELLER TAKTTREIBER	115
5.3.3 POWER MONITOR	115
5.4 System-Initialisierung und -Selbsttest	115
5.5 Kontakte für die Stromversorgung	119
5.6 RESERVED-Leitung	119

Kapitel 6: IEC-821-BUS, elektrische Eigenschaften

6.1 Einführung	121
6.2 Stromversorgung	121
6.2.1 Spannungswerte	121
6.2.2 Elektrische Grenzwerte der Steckkontakte	121
6.3 Signalspannungen	121
6.4 Bustreiber und -empfänger	122
6.4.1 Bustreiberarten	122
6.4.2 VORSCHRIFTEN für alle Bustreiber und -empfänger	122
6.4.2.1 Vorschriften für Treiber und Empfänger an Three-State-Leitungen für hohe Leistung (AS*, DS0*, DS1*)	123
6.4.2.2 Vorschriften für Treiber und Empfänger an gewöhnliche Three-State-Leitungen (A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, WRITE*)	123
6.4.2.3 Vorschriften für Treiber und Empfänger an Totem-Pole-Leitungen für hohe Leistung (SERCLK, SYSCLK, BCLR*)	124
6.4.2.4 Vorschriften für Treiber und Empfänger an gewöhnlichen Totem-Pole-Leitungen (BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*)	124
6.4.2.5 Vorschriften für Treiber und Empfänger an Open-Collector-Leitungen (BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, IACK*)	124

6.5 Signalleitungen auf der Busplatine	125
6.5.1 Abschlußnetzwerke	125
6.5.2 Wellenwiderstand	125
6.5.3 Zusätzliche Informationen	128
6.6 Benutzerdefinierte Signalleitungen	128
6.7 Signalleitungstreiber und deren Abschlüsse	128

Kapitel 7: IEC-821-BUS, mechanische Eigenschaften

7.1 Einführung	130
7.2 IEC-821-BUS Karten	130
7.2.1 Einfachhohe Karten	130
7.2.2 Doppelhohe Karten	131
7.2.3 Steckverbinder	131
7.2.4 Baugruppen	131
7.2.5 Baugruppenbreite	131
7.2.6 Leiterplatten-Durchbiegung, lötseitige und bestückungsseitige Bauhöhe	132
7.3 Frontplatten	132
7.3.1 Griffe	132
7.3.2 Frontplattenbefestigung	133
7.3.3 Frontplattenmaße	133
7.3.4 Blenden	133
7.3.5 Kartenziehvorrückungen	133
7.4 Busplatinen	133
7.4.1 Busplattenmaße	134
7.4.2 Abschlußnetzwerke der Signalleitungen	134
7.5 Aufbau von Bus-Baugruppenträgern	134
7.5.1 Steckplatzbreiten in Baugruppenträgern	134
7.5.2 Maße der Baugruppenträger	135
7.6 Busplatinen-Steckverbinder und Leiterplatten-Steckverbinder	135
7.6.1 Anschlußbelegungen für J_1/P_1 -Steckverbinder	135
7.6.2 Anschlußbelegungen für J_2/P_2 -Steckverbinder	135

Anhang A: Begriffe der IEC-821-BUS-Norm	157
--	-----

Anhang B: Steckverbinder-/Stiftanschlußbelegung des IEC-821-BUS	159
--	-----

Anhang C: Verwendung der SERCLK- und SERDAT*-Leitungen	160
---	-----

Anhang D: (informativ): Metastabilität und Resynchronisation	162
---	-----

Anhang E: Zulässige Teilmengen von Fähigkeiten	173
---	-----

Anhang ZA (normativ): Andere in dieser Norm zitierte internationale Publikationen mit den Verweisen auf die entsprechenden europäischen Publikationen	177
--	-----

Bilder

1-1: Systembausteine nach dieser Norm	10
1-2: Funktionsmodule und Busse nach dieser Norm	11
1-3: Darstellung des zeitlichen Signalverlaufs	13
2-1: Funktionsschaltbild für den Datentransferbus	15
2-2: Funktionsschaltbild: MASTER	20
2-3: Funktionsschaltbild: SLAVE	21
2-4: Funktionsschaltbild: BUS TIMER	22
2-5: Funktionsschaltbild: LOCATION MONITOR	23
2-6: Die vier Speicherungsarten von 32 Datenbits	28
2-7: Die vier Speicherungsarten von 16 Datenbits	29
2-8: Beispiel für einen Ein-Byte-Lesezyklus	32
2-9: Beispiel für einen Zwei-Byte-Schreibzyklus	33
2-10: Beispiel für einen Vier-Byte-Schreibzyklus	34
2-11: MASTER-Wechsel auf dem Datentransferbus	35
2-12: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei allen Zyklen	47

Bilder	Seite
2-13: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei geradzahligen Ein-Byte-Transfers, ungeradzahligen Ein-Byte-Transfers, Zwei-Byte-Transfers, Vier-Byte-Transfers, nicht an Wortgrenzen gebundenen Transfers	48
2-14: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei Ein-Byte-Blocktransfers, Zwei-Byte-Blocktransfers, Vier-Byte-Blocktransfers.	49
2-15: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei Ein-Byte-RMW-Zyklen, Zwei-Byte-RMW-Zyklen, Vier-Byte-RMW-Zyklen	50
2-16: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR bei BYTE(0)-READ-Zyklen, BYTE(1)-READ-Zyklen, BYTE(2)-READ-Zyklen, BYTE(3)-READ-Zyklen, BYTE(0-2)-READ-Zyklen, BYTE(1-3)-READ-Zyklen, Ein-Byte-BLOCK-READ-Zyklen	51
2-17: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR bei BYTE(0-1)-READ-Zyklen, BYTE(2-3)-READ-Zyklen, BYTE(0-3)-READ-Zyklen, BYTE(1-2)-READ-Zyklen, Zwei-Byte-BLOCK-READ-Zyklen, Vier-Byte-BLOCK-READ-Zyklen	53
2-18: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR bei BYTE(0)-WRITE-Zyklen, BYTE(1)-WRITE-Zyklen, BYTE(2)-WRITE-Zyklen, BYTE(3)-WRITE-Zyklen, BYTE(0-2)-WRITE-Zyklen, BYTE(1-3)-WRITE-Zyklen, Ein-Byte-BLOCK-WRITE-Zyklen.	55
2-19: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR bei BYTE(0-1)-WRITE-Zyklen, BYTE(2-3)-WRITE-Zyklen, BYTE(0-3)-WRITE-Zyklen, BYTE(1-2)-WRITE-Zyklen, Zwei-Byte-BLOCK-WRITE-Zyklen, Vier-Byte-BLOCK-WRITE-Zyklen	57
2-20: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR beim Ein-Byte-RMW-Zyklus	59
2-21: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und LOCATION MONITOR bei Zwei-Byte-RMW-Zyklen, Vier-Byte-RMW-Zyklen	60
2-22: Zeitablauf-Diagramm für den Adreß-Strobe zwischen den Zyklen	61

Bilder	Seite
2-23: Zeitablauf-Diagramm für den Daten-Strobe zwischen den Zyklen Ein Zyklus mit beiden Daten-Strobes auf low, gefolgt von einem Zyklus mit einem oder beiden Daten-Strobes auf low.	62
2-24: Zeitablauf-Diagramm für den Daten-Strobe zwischen den Zyklen Ein Zyklus mit einem Daten-Strobe auf low, gefolgt von einem Zyklus mit einem oder beiden Daten-Strobes auf low.	63
2-25: Zeitablauf-Diagramm für Datentransfer-Signale von MASTER, SLAVE und BUS TIMER beim Time-Out-Zyklus	64
2-26: Zeitablauf-Diagramm für MASTER-Wechsel auf dem Datentransferbus	65
3-1: Funktionsschaltbild: Zuteilungsbuss	67
3-2: Darstellung der verketteten Buszuteilungsleitungen	68
3-3: Funktionsschaltbild: ARBITER.	70
3-4: Funktionsschaltbild: REQUESTER	72
3-5: Flußdiagramm für die Zuteilung: zwei REQUESTER, zwei Anforderungspegel	75
3-6: Ablaufdiagramm für die Zuteilung: zwei REQUESTER, zwei Anforderungspegel	77
3-7: Flußdiagramm für die Zuteilung: zwei REQUESTER auf demselben Anforderungs-pegel	78
3-8: Zeitdiagramm für die Zuteilung: zwei REQUESTER auf demselben Anforderungs-pegel	80
4-1: Funktionsschaltbild: Interrupt-Bus	82
4-2: Struktur eines zentralen Interrupt-Systems	83
4-3: Struktur eines verteilten Interrupt-Systems	84
4-4: IACKIN*/IACKOUT* DAISY-CHAIN	85
4-5: Funktionsschaltbild: INTERRUPT HANDLER	86
4-6: Funktionsschaltbild: INTERRUPTER.	87
4-7: Funktionsschaltbild: IACK-DAISY-CHAIN-TREIBER	88
4-8: Freigabe der Interrupt-Request-Leitungen durch ROAK- und RORA-INTERRUPTER	90
4-9: Ein IACK-DAISY-CHAIN-TREIBER und ein INTERRUPTER auf der gleichen Leiterplatte	91
4-10: Zwei INTERRUPTER auf der gleichen Leiterplatte	92
4-11: Die drei Phasen beim Ablauf eines Interrupt	93
4-12: Zwei INTERRUPT HANDLER, die jeweils eine Interrupt-Request-Leitung überwachen	94
4-13: Zwei INTERRUPT HANDLER, die jeweils mehrere Interrupt-Request-Leitungen überwachen	94
4-14: Flußdiagramm für ein typisches zentrales Interrupt-System	95
4-15: Flußdiagramm für ein typisches verteiltes Interrupt-System, welches zwei INTERRUPT HANDLER enthält	98
4-16: INTERRUPT HANDLER und INTERRUPTER: Zeitablauf-Diagramm für die INTERRUPTER-Auswahl beim Ein-, Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus	106
4-17: IACK-DAISY-CHAIN-TREIBER: Zeitablauf-Diagramm für die INTERRUPTER-Auswahl beim Ein-, Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus.	107
4-18: Teilnehmender INTERRUPTER: Zeitablauf-Diagramm für die INTERRUPTER-Auswahl beim Ein-, Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus.	108
4-19: Reagierender INTERRUPTER: Zeitablauf-Diagramm für die INTERRUPTER-Auswahl beim Ein-, Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus.	109

Bilder	Seite	Bilder	Seite
4-20: INTERRUPT HANDLER: Zeitablauf-Diagramm für den Transfer des STATUS/ID beim Ein-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus	110	D-6: Asynchrone Behandlung der Buszuteilungs-Daisy-Chain	170
4-21: INTERRUPT HANDLER: Zeitablauf-Diagramm für den Transfer des STATUS/ID beim Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus	111	D-7: Synchrone Realisierung der Buszuteilungs-Daisy-Chain	171
4-22: Reagierender INTERRUPTER: Zeitablauf-Diagramm für den Transfer des STATUS/ID beim Ein-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus	112	D-8: Asynchrone Realisierung der Interrupt-Acknowledge-Daisy-Chain	172
4-23: Antwortender INTERRUPTER: Zeitablauf-Diagramm für den Transfer des STATUS/ID beim Zwei- und Vier-Byte-INTERRUPT-ACKNOWLEDGE-Zyklus	113	D-9: Asynchroner VMEbus-ARBITER	172
4-24: IACK-DAISY-CHAIN-TREIBER, antwortender INTERRUPTER und beteiligter INTERRUPTER: Zeitablauf-Diagramm der IACK-Daisy-Chain zwischen den Zyklen	114	Tabellen	
5-1: Funktionsschaltbild: Utility-Bus	116	2-1: Die vier Arten von Byte-Speicherplätzen	14
5-2: Zeitablauf-Diagramm für SYSTEMAKTTREIBER	117	2-2: Verwendung von DS0*, DS1*, A01 und LWORD* zur Auswahl von Byte-Speicherstellen	16
5-3: Funktionsschaltbild des POWER-MONITOR-Moduls	117	2-3: Adreßmodifizierer-Codes	17
5-4: Zeitablauf-Diagramm für den POWER MONITOR bei Spannungsausfall	118	2-4: Verwendung der Datenleitungen, um auf Byte-Speicherstellen zuzugreifen	18
5-5: Zeitablauf-Diagramm für den POWER MONITOR bei Spannungswiederkehr	118	2-5: MASTER: VORSCHRIFTEN und ERLAUBNISSE für das Treiben der gestrichelt gezeichneten Leitungen	19
5-6: Zeitablauf-Diagramm für SYSRESET* und SYSFAIL*	120	2-6: SLAVES: VORSCHRIFTEN und ERLAUBNISSE für das Treiben und Überwachen der gestrichelt gezeichneten Leitungen	21
5-7: Belastbarkeit der Spannungskontakte	120	2-7: BT0()-Mnemonic zur Spezifikation der Time-Out-Periode der BUS TIMER	22
6-1: Signalpegel für den IEC-821-BUS	122	2-8: LOCATION MONITORE: VORSCHRIFTEN und ERLAUBNISSE für das Überwachen der gestrichelt gezeichneten Leitungen	22
6-2: Genormter Busabschluß	126	2-9: Mnemonics, welche die Adressierungsfähigkeiten festlegen	24
6-3: Querschnitt durch eine Signalleitung auf der Busplatine	126	2-10: Mnemonics für Datentransfers	25
6-4: Abhängigkeit von Z_0 von der Leiterbahnbreite	127	2-11: Mnemonics, welche die Blocktransferfähigkeiten festlegen	27
6-5: Abhängigkeit von C_0 von der Leiterbahnbreite	127	2-12: Mnemonic, das die Read-Modify-Write-Fähigkeiten festlegt	27
7-1: Baugruppenträger mit verschiedenen großen Leiterplatten	135	2-13: Transfer von 32-Bit-Daten unter Verwendung von Mehr-Byte-Transfer-Zyklen	28
7-2: Einfachhohe Leiterplatte: Maße	136	2-14: Transfer von 16-Bit-Daten unter Verwendung von Mehr-Byte-Transfer-Zyklen	29
7-3: Doppelthohe Leiterplatte: Maße	137	2-15: Mnemonic, das die Fähigkeit für nicht an Wortgrenzen gebundene Transfers festlegt	29
7-4: Anordnung der Steckverbinder bei einfachhohen und doppelthohen Leiterplatten	138	2-16: Mnemonic, das die ADDRESS-ONLY-Fähigkeit festlegt	29
7-5: Querschnitt durch Leiterplatte, Steckverbinder, Busplatine und Frontplatte	139	2-17: Zeitablauf-Diagramme, die den MASTER-, SLAVE- und LOCATION-MONITOR-Betrieb definieren	37
7-6: Bauelementhöhe, Anschlußdrahtlänge und Leiterplatten-Durchbiegung	140	2-18: Definitionen der in den Tabellen 2-19, 2-20 und 2-21 benutzten Mnemonics	38
7-7: Einfachhohe, einfachbreite Frontplatte	141	2-19: Verwendung der Adreßleitungen zur Auswahl einer 4-Byte-Gruppe	38
7-8: Doppelthohe, einfachbreite Frontplatte	142	2-20: Verwendung von DS1*, DS0*, A01 und LWORD* während der verschiedenen Zyklen	39
7-9: Frontplatten-Montagewinkel und Maße für einfachhohe Leiterplatten	143	2-21: Verwendung der Datenbusleitungen für den Datentransfer	40
7-10: Frontplatten-Montagewinkel und Maße für doppelthohe Karten	144	2-22: Zeitablauf-Parameter von MASTER, SLAVE und LOCATION MONITOR	41
7-11: Einfachhohe Blende	145	2-23: Zeitablauf-Parameter des BUS TIMER	41
7-12: Doppelthohe Blende	146	2-24: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten des MASTER	42
7-13: J_1 - und J_2 -Busplatine	147	2-25: SLAVE: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten	44
7-14: J_1 - und J_2 -Busplatine, detaillierte Maße	148	2-26: LOCATION MONITOR: ERLÄUTERUNGEN zum Zeitverhalten	46
7-15: J_1/J_2 -Busplatine	149	2-27: BUS TIMER: VORSCHRIFTEN zum Zeitverhalten	47
7-16: J_1/J_2 -Busplatine, detaillierte Maße	150	3-1: ARBITER: VORSCHRIFTEN und ERLAUBNISSE zum Treiben und Überwachen der gestrichelt gezeichneten Leitungen	70
7-17: Steckbarer Abschluß der Busplatine (Draufsicht)	151	3-2: REQUESTER: VORSCHRIFTEN und ERLAUBNISSE zum Treiben und Überwachen der gestrichelt gezeichneten Leitungen	72
7-18: Auf die Busplatine montierter Abschluß	152		
7-19: Baugruppenträger mit 21 Steckplätzen	153		
7-20: Ausführliche Beschreibung der Kartenführung	154		
C-1: SERCLK-Zeitablauf-Diagramm	161		
D-1: RS-Flipflop	162		
D-2: Kritische Eingangsbedingungen	163		
D-3: Die zwei Arten von Metastabilität	164		
D-4: Verschiedene Interpretationen eines metastabilen Ausgangssignals	165		
D-5: Struktur der Buszuteilung im VMEbus; gestrichelte Kreise zeigen an, wo kritische Eingangsbedingungen entstehen können	169		

Tabellen	Seite
4-1: INTERRUPT HANDLER: VORSCHRIFTEN und ERLAUBNISSE zum Treiben und Überwachen der gestrichelt gezeichneten Leitungen.	84
4-2: INTERRUPTER: VORSCHRIFTEN und ERLAUBNISSE zum Treiben der gestrichelt gezeichneten Leitungen.	87
4-3: IH()-Mnemonic zur Festlegung der Interrupt-Handling-Fähigkeiten.	88
4-4: I()-Mnemonic zur Festlegung der Fähigkeiten zur Interrupt-Request-Erzeugung.	89
4-5: Mnemonic zur Festlegung der STATUS/ID-Transfer-Fähigkeiten.	89
4-6: Mnemonics zur Festlegung der Interrupt-Request-Freigabe.	90
4-7: 3-Bit-Interrupt-Acknowledge-Code.	97
4-8: Zeitablauf-Diagramme, die den INTERRUPT-HANDLER- und INTERRUPTER-Betrieb definieren.	99
4-9: Zeitablauf-Diagramme, die den Betrieb des IACK-DAISY-CHAIN-TREIBER definieren.	99
4-10: Zeitablauf-Diagramme, die den Betrieb teilnehmender INTERRUPTER definieren.	99
4-11: Zeitablauf-Diagramme, die den Betrieb reagierender INTERRUPTER definieren.	100
4-12: Definitionen der in den Tabellen 4-13, 4-14 und 4-15 verwendeten Mnemonics.	100
4-13: Verwendung von A01-A03 und IACK* während Interrupt-Acknowledge-Zyklen.	100
4-14: Verwendung von DS1*, DS0*, LWORD* und WRITE* während Interrupt-Acknowledge-Zyklen.	100
4-15: Verwendung von D00-D31 für den Transfer des STATUS/ID.	101
4-16: Zeitablauf-Parameter von INTERRUPT HANDLER, INTERRUPTER und IACK-DAISY-CHAIN-TREIBER.	101

Tabellen	Seite
4-17: INTERRUPT HANDLER: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten.	102
4-18: INTERRUPTER: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten.	103
4-19: IACK-DAISY-CHAIN-TREIBER: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten.	105
5-1: Modulausgangssignale während der Ein- bzw. Ausschaltsequenzen der Versorgungsspannung.	119
6-1: Festgelegte Spannungswerte.	122
6-2: Anforderungen an Bustreiber und -empfänger.	123
6-3: Bustreiber-Zusammenfassung.	129
7-1: Anschlußbelegungen von J_1/P_1	155
7-2: Anschlußbelegungen von J_2/P_2	156
C-1: SERCLK-Zeitablauf-Werte.	161
D-1: Daten zur Metastabilität.	167
E-1: Zulässige Teilmengen von Adressierungsfähigkeiten.	173
E-2: Interoperabilität von zulässigen Adressierungsteilmengen.	173
E-3: MASTER: Zulässige Teilmengen von Datentransferfähigkeiten.	174
E-4: SLAVE: Zulässige Teilmengen von Datentransferfähigkeiten.	175
E-5: LOCATION MONITOR: Zulässige Teilmengen von Fähigkeiten zum Erkennen von Datentransfers.	175
E-6: Interoperabilität von zulässigen Datentransfersteilmengen.	175
E-7: Interoperabilität von ARBITERN und REQUESTERN.	176
E-8: Zusammenwirken von INTERRUPTERN und INTERRUPT HANDLERN.	176

Kapitel 0: Einführung

0.1 Anwendungsbereich

Diese Norm beschreibt einen Hochleistungs-Rückwandbus für die Verwendung in Mikroprozessor-Systemen, die einen oder mehrere Mikroprozessoren enthalten. Sie beruht auf der VMEbus-Spezifikation, die im August 1982 von der VMEbus-Herstellervereinigung veröffentlicht wurde. Der Bus schließt vier Unterbusse ein: den Datentransferbus, den Interrupt-Bus, den Zuteilungsbus und den Utility-Bus. Der Datentransferbus unterstützt 8-, 16- und 32-Bit-Transfers über nicht-gemultiplexte Adreß- und Datenleitungen mit 32 Bit Breite. Die Transferprotokolle arbeiten asynchron und mit Quittung. Der Interrupt-Bus stellt dem System Echtzeit-Interrupts zur Verfügung. Die Zuweisung der Vorherrschaft auf dem Bus wird durch den Zuteilungsbus vorgenommen, welcher die Implementierung von Algorithmen sowohl mit Rotations- als auch mit Prioritätszuteilung erlaubt. Der Utility-Bus stellt dem System die Synchronisierung beim Ein- und Ausschalten zur Verfügung. Der mechanische Aufbau der Baugruppen, Busplatinen, der Gestelle und Chassis beruht auf der Norm IEC 297.

0.2 Normative Verweisungen

Die folgenden IEC-Publikationen werden in dieser Norm zitiert:

IEC 297-1 (1982)

Dimensions of mechanical structures of the 482,6 mm (19 in) series – Part 1: Panels and racks

IEC 297-3 (1984)

Dimensions of mechanical structures of the 482,6 mm (19 in) series – Part 3: Subracks and associated plug-in units

IEC 603-2 (1980)

Connectors for frequencies below 3 MHz for use with printed boards, Part 2: Two-part connectors for printed boards for basic grid of 2,54 mm (0.1 in), with common mounting features

IEC 822 (1988)

IEC 822 VSB – Parallel sub-system bus of the IEC 821 VMEbus

IEC 823 (1990)

Microprozessor system bus (VMSbus) – Serial sub-system bus of the IEC 821 Bus (VMEbus)

0.3 Hinweis für den Leser

Der IEC-822-BUS wurde durch das Unterkomitee 47B als Untersystembus für den IEC-821-BUS, der diese Norm darstellt, genormt.

Der IEC-823-BUS stellt den Bus dar, der von JTC 1/SC 26 als serieller Bus des IEC-821-BUS genormt wurde.

Kapitel 1: Einführung in die IEC-821-BUS-Norm

1.1 Zielsetzung der Bus-Norm

Diese IEC-821-BUS-Norm beschreibt ein System zur Verbindung von Baugruppen der Datenverarbeitung, der Datenspeicherung und der Peripheriekontrolle in einer eng gekoppelten Hardware-Konfiguration. Das System ist gemäß den folgenden Zielen konzipiert worden:

- a) Eine Kommunikation zwischen den Baugruppen über den IEC-821-BUS ohne Beeinträchtigung des internen Betriebes anderer am IEC-821-BUS angeschlossenen Baugruppen zu ermöglichen.
- b) Die elektrischen und mechanischen Systemparameter für die Entwicklung von Baugruppen festzulegen, die eine zuverlässige und eindeutige Kommunikation mit anderen am IEC-821-BUS angeschlossenen Baugruppen sicherstellen.
- c) Protokolle festzulegen, die die Wechselwirkung zwischen dem IEC-821-BUS und den angeschlossenen Baugruppen genau definieren.
- d) Terminologie und Begriffe, die das Systemprotokoll beschreiben, anzugeben.
- e) Dem Entwickler genügend Freiheitsgrade bieten, damit er Kosten und/oder Leistungsumfang optimieren kann, ohne die Systemkompatibilität zu beeinträchtigen.
- f) Ein System anzubieten, dessen Leistungsfähigkeit hauptsächlich durch die Baugruppe und nicht durch die Systemschnittstelle begrenzt ist.

1.2 Systemelemente

1.2.1 Grundbegriffe

Der IEC-821-BUS-Aufbau kann unter zwei Gesichtspunkten beschrieben werden: dem mechanischen Aufbau und dem funktionellen Aufbau. Die mechanische Spezifikation be-

schreibt die Maße der Baugruppenträger, der Busplatinen, der Frontplatten, der Baugruppen usw. Die IEC-821-BUS-Funktionsspezifikation beschreibt die Arbeitsweise des Bus, welche Funktionsmodule an einem Vorgang beteiligt sind und die Regeln, die ihr Verhalten bestimmen. Dieser Abschnitt liefert informelle Definitionen für einige Grundbegriffe, die zur Beschreibung des physikalischen wie auch des mechanischen Aufbaus des IEC-821-BUS verwendet werden.

1.2.1.1 Begriffe zur Beschreibung des mechanischen Aufbaus

BUSPLATINE DES IEC-821-BUS

Eine Leiterplatte mit 96poligen Steckverbindern und Signalpfaden, die die Kontaktstifte der Steckverbinder miteinander verbinden. Einige IEC-821-BUS-Systeme haben eine einzige Leiterplatte, J_1 -Busplatine genannt. Diese stellt die Signalfade für die Grundfunktionen zur Verfügung. Andere IEC-821-BUS-Systeme haben wahlweise eine zweite Leiterplatte, genannt J_2 -Busplatine. Diese stellt die für breitere Daten- und Adreßtransfers erforderlichen zusätzlichen 96poligen Steckverbinder und Signalpfade zur Verfügung. Wieder andere haben eine einzige Leiterplatte, J_1/J_2 -Busplatine genannt, welche die Signalleitungen und Steckverbinder der J_1 - und J_2 -Busplatine zur Verfügung stellt.

BAUGRUPPE

Eine gedruckte Leiterplatte, ihre elektronischen Bauteile und entweder ein oder zwei 96polige Steckverbinder, die mit den Steckverbindern der Busplatine des IEC-821-BUS verbunden werden können.

STECKPLATZ

Eine Position, an der eine Baugruppe auf die Busplatine des IEC-821-BUS gesteckt werden kann. Bei einem IEC-821-BUS-System mit je einer J_1 - und J_2 -Busplatine (oder einer

Kombination aus beiden) hat jeder Steckplatz zwei 96polige Steckverbinder. Bei Systemen mit nur einer J_1 -Busplatine hat jeder Steckplatz nur einen 96poligen Steckverbinder.

BAUGRUPPENTRÄGER

Ein fester Rahmen, der mechanischen Halt für die auf die Busplatine gesteckten Baugruppen bietet und sicherstellt, daß die Steckverbinder fest sitzen und die benachbarten Baugruppen keinen Kontakt miteinander haben. Er leitet auch den Kühlluftstrom durch das System, und stellt sicher, daß die eingefügten Baugruppen sich nicht aufgrund von Stoßeinwirkung und Erschütterungen aus der Busplatine lösen.

1.2.1.2 Begriffe zur Beschreibung des funktionellen Aufbaus

Bild 1-1 zeigt ein vereinfachtes Blockschaltbild des funktionellen Aufbaus einschließlich der IEC-821-BUS-Signalleitungen, der Busplatinen-Schnittstellenlogik und der Funktionsmodule.

BUSPLATINEN-SCHNITTSTELLENLOGIK

Eine spezielle Logik, die die Merkmale der Busplatine berücksichtigt: die Signalleitungsimpedanz, die Laufzeit, Abschlußwerte usw. Die IEC-821-BUS-Norm schreibt bestimmte Regeln für den Entwurf dieser Logik vor, die auf der maximalen Länge der Busplatine und der maximalen Anzahl ihrer Steckplätze basieren.

FUNKTIONSMODUL

Eine Zusammenstellung von elektronischen Schaltkreisen, die sich auf einer IEC-821-BUS-Baugruppe befinden und gemeinsam eine bestimmte Aufgabe erfüllen.

DATENTRANSFERBUS

Einer von vier Bussen, die auf der Busplatine des IEC-821-BUS vorhanden sind. Der Datentransferbus erlaubt MASTERN den Transfer binärer Daten zwischen MASTERN und SLAVES zu steuern. (Datentransferbus wird häufig mit DTB abgekürzt.)

DATENTRANSFERBUS-ZYKLUS

Eine Folge von Pegelübergängen auf den Signalleitungen des DTB, die zum Transfer einer Adresse oder einer Adresse und Daten zwischen MASTER und SLAVE führen. Der Datentransferbus-Zyklus ist in zwei Abschnitte aufgeteilt, den Adreßbroadcast und anschließend einen oder mehrere Datentransfers. Es gibt 34 Arten von Datentransferbus-Zyklen. Sie werden später in diesem Kapitel definiert.

MASTER

Ein Funktionsmodul, das DTB-Zyklen auslöst, um Daten zwischen sich und einem SLAVE-Modul zu übertragen.

SLAVE

Ein Funktionsmodul, das von einem MASTER ausgelöste DTB-Zyklen erkennt und, sofern diese Zyklen seine Teilnahme spezifizieren, Daten zwischen sich und dem MASTER überträgt.

LOCATION MONITOR

Ein Funktionsmodul, das den Datentransfer über den DTB überwacht, um Zugriffe auf von diesem zu beobachtende Speicherplätze festzustellen. Erfolgt ein Zugriff auf einen dieser zugewiesenen Speicherplätze, erzeugt der LOCATION MONITOR ein kartenindenes Signal.

BUS TIMER

Ein Funktionsmodul, das die Dauer jedes Datentransfers auf dem DTB mißt und den DTB-Zyklus beendet, wenn der Transfer zu lange dauert. Ohne dieses Modul würde der MASTER beim Versuch, Daten an oder von einem nicht existierenden SLAVE-Speicherplatz zu übertragen, ewig warten.

Der BUS TIMER verhindert dieses durch Beendigung des Zyklus.

INTERRUPT-BUS

Einer von vier Bussen, der auf der Busplatine des IEC-821-BUS zur Verfügung steht. Der Interrupt-Bus erlaubt es den INTERRUPTER-Modulen, Interrupt-Requests an die INTERRUPT-HANDLER-Module zu senden.

INTERRUPTER

Ein Funktionsmodul, das ein Interrupt-Request-Signal auf dem Interrupt-Bus erzeugt und die STATUS/ID-Information bereitstellt, wenn der INTERRUPT HANDLER sie anfordert.

INTERRUPT HANDLER

Ein Funktionsmodul, das von INTERRUPTERN erzeugte Interrupt-Requests erkennt und auf diese Requests mit der Anforderung von STATUS/ID-Informationen reagiert.

DAISY-CHAIN

Eine spezielle IEC-821-BUS-Signalleitung, über die ein Signalpegel von Karte zu Karte verbreitet wird, beginnend am ersten Steckplatz bis zum letzten. Es gibt vier Bus-Grant-Daisy-Chains und eine Interrupt-Acknowledge-Daisy-Chain auf dem IEC-821-BUS.

IACK-DAISY-CHAIN-TREIBER

Ein Funktionsmodul, das die Interrupt-Acknowledge-Daisy-Chain aktiviert, wenn ein INTERRUPT HANDLER einen Interrupt-Request bestätigt. Die Daisy-Chain stellt sicher, daß nur ein INTERRUPTER mit seinem STATUS/ID reagiert, wenn mehrere INTERRUPTER einen Interrupt-Request erzeugt haben.

ZUTEILUNGSBUS

Einer von vier auf der Busplatine des IEC-821-BUS zur Verfügung stehenden Bussen. Dieser Bus ermöglicht es einem ARBITER-Modul und mehreren REQUESTER-Modulen, die Nutzung des DTB zu koordinieren.

REQUESTER

Ein Funktionsmodul, das sich auf der gleichen Baugruppe wie ein MASTER oder INTERRUPT HANDLER befindet und den DTB immer dann anfordert, wenn sein MASTER oder INTERRUPT HANDLER ihn benötigt.

ARBITER

Ein Funktionsmodul, das Busanforderungen von REQUESTER-Modulen annimmt und die Kontrolle über den DTB zu jedem Zeitpunkt einem einzigen REQUESTER erteilt.

UTILITY-BUS

Einer von vier auf der Busplatine des IEC-821-BUS zur Verfügung stehenden Bussen. Dieser Bus umfaßt periodische Taktsignale und solche, die das Ein- und Ausschalten des IEC-821-BUS-Systems koordinieren.

SYSTEMAKTTREIBER

Ein Funktionsmodul, das den Utility-Bus mit einem 16 MHz-Taktsignal versorgt.

SERIELLER TAKTTREIBER

Ein Funktionsmodul, das ein periodisches Taktsignal liefert, welches den Betrieb des IEC-823-BUS synchronisiert. (Obwohl die IEC-821-BUS-Norm einen SERIELLEN TAKTTREIBER für den IEC-823-BUS definiert und obwohl sie zwei Busplatinen-Signalleitungen für diesen Bus reserviert, ist das IEC-823-BUS-Protokoll vollkommen unabhängig vom IEC-821-BUS.) Die Zeitangaben für den SERIELLEN TAKTTREIBER sind im Anhang C festgelegt.

POWER-MONITOR-MODUL

Ein Funktionsmodul, das den Zustand der Primär-Spannungsversorgung des IEC-821-BUS-Systems überwacht und

anzeigt, wenn diese Spannung sich außerhalb der Grenzwerte bewegt, die für eine zuverlässige Systemfunktion erforderlich sind. Da die meisten Systeme an einer Wechselspannungs-Versorgung angeschlossen sind, ist der POWER MONITOR typischerweise für die Feststellung von kurzzeitigem oder totalem Ausfall der Wechselspannung konstruiert.

SYSTEM-CONTROLLER-BOARD

Eine Karte im Steckplatz 1 der Busplatine des IEC-821-BUS, die einen SYSTEMAKTTREIBER, einen ARBITER, einen IACK-DAISY-CHAIN-TREIBER und einen BUS TIMER hat. Einige enthalten zusätzlich einen SERIELLEN AKTTREIBER oder einen POWER MONITOR oder beides.

1.2.1.3 Zyklusarten des IEC-821-BUSSES

LESEZYKLUS

Ein DTB-Zyklus für den Transfer von 1, 2, 3 oder 4 Bytes von einem SLAVE zu einem MASTER. Der Zyklus beginnt, wenn der MASTER eine Adresse und einen Adreßmodifizierer auf den Bus legt. Jeder SLAVE übernimmt den Adreßmodifizierer und die Adresse und überprüft, ob er auf den Zyklus reagieren soll. Falls ja, ruft er die Daten aus seinem internen Speicher ab, legt sie auf den Datenbus und bestätigt den Transfer. Der MASTER beendet daraufhin den Zyklus.

SCHREIBZYKLUS

Ein DTB-Zyklus für den Transfer von 1, 2, 3 oder 4 Bytes von einem MASTER an einen SLAVE. Der Zyklus beginnt, wenn der MASTER eine Adresse und einen Adreßmodifizierer sowie Daten auf den DTB legt. Jeder SLAVE übernimmt Adresse und Adreßmodifizierer und überprüft, ob er auf den Zyklus reagieren soll. Falls ja, speichert er die Daten und bestätigt den Transfer. Der MASTER beendet daraufhin den Zyklus.

BLOCKLESEZYKLUS

Ein DTB-Zyklus für den Transfer von 1 bis 256 Byte großen Datenblöcken von einem SLAVE an einen MASTER. Dieser Transfer erfolgt in Form einer Folge von 1-, 2- oder 4-Byte-Datentransfers. Nachdem der Blocktransfer gestartet worden ist, gibt der MASTER den DTB erst wieder frei, wenn alle Bytes übertragen worden sind. Dieser Zyklus unterscheidet sich von einer Serie von Lesezyklen dadurch, daß der MASTER nur eine Adresse und Adreßmodifizierer auf den Bus legt (am Beginn des Zyklus). Der SLAVE inkrementiert sodann die Adresse bei jedem Transfer, so daß die Daten für den nächsten Transfer vom nächsthöheren Speicherplatz abgerufen werden.

BLOCKSCHREIBZYKLUS

Ein DTB-Zyklus für den Transfer von 1 bis 256 Byte großen Datenblöcken von einem MASTER an einen SLAVE. Er erfolgt in Form einer Folge von 1-, 2- oder 4-Byte-Datentransfers und der MASTER gibt den DTB erst frei, wenn alle Bytes übertragen worden sind. Er unterscheidet sich von einer Serie von Lesezyklen dadurch, daß der MASTER nur eine Adresse und einen Adreßmodifizierer auf den Bus legt (am Beginn des Zyklus). Der SLAVE inkrementiert daraufhin die Adresse bei jedem Transfer, so daß die Daten des nächsten Transfers im nächsthöheren Speicherplatz abgespeichert werden.

READ-MODIFY-WRITE-ZYKLUS

Ein DTB-Zyklus, mit dem man einen SLAVE-Speicherplatz sowohl lesen als auch beschreiben kann, ohne zwischenzeitlich einem anderen MASTER den Zugriff auf diesen Speicherplatz zu erlauben. Dieser Zyklus ist sehr nützlich in einem Multiprozessor-System, wo bestimmte Speicherplätze für Semaphoren-Funktionen benutzt werden.

ADDRESS-ONLY-ZYKLUS

Ein DTB-Zyklus, der aus einem Adreß-Broadcast, jedoch ohne Datentransfer, besteht. SLAVES bestätigen ADDRESS-ONLY-Zyklen nicht und MASTER beenden den Zyklus, ohne eine Bestätigung abzuwarten.

INTERRUPT-ACKNOWLEDGE-ZYKLUS

Ein durch einen INTERRUPT HANDLER ausgelöster DTB-Zyklus, der vom INTERRUPTER den STATUS/ID liest. Ein INTERRUPT HANDLER erzeugt diesen Zyklus jedesmal, wenn er einen Interrupt-Request durch einen INTERRUPTER erkennt und die Kontrolle über den DTB hat.

1.2.2 Allgemeiner Aufbau des IEC-821-BUSSES

Das IEC-821-BUS-Schnittstellensystem besteht aus der Schnittstellenlogik der Busplatine, vier Gruppen von Signalleitungen, genannt Busse, und einer Reihe von Funktionsmodulen, die nach Bedarf konfiguriert werden können. Die Funktionsmodule kommunizieren miteinander über die Signalleitungen der Busplatine.

Die in dieser Norm definierten Funktionsmodule werden zur Erläuterung des Busprotokolls verwendet und sind nicht als Einschränkung beim Logikentwurf anzusehen. Der Entwickler könnte z. B. eine Logik entwerfen, die mit dem IEC-821-BUS in der beschriebenen Weise in Wechselwirkung steht, jedoch andere karteneigene Signale verwendet oder andere IEC-821-BUS-Signale überwacht. IEC-821-BUS-Baugruppen können so entwickelt werden, daß sie jede Kombination der in dieser Norm definierten Funktionsmodule beinhalten.

Der funktionelle Aufbau des IEC-821-BUS ist in vier Kategorien aufgeteilt. Jede besteht aus einem Bus und den zugehörigen Funktionsmodulen, die zusammenarbeiten, um bestimmte Aufgaben zu erfüllen. Bild 1-2 zeigt die IEC-821-BUS-Funktionsmodule und Busse. Zu jeder Kategorie folgt hier eine Kurzbeschreibung:

a) Datentransfer

Die Baugruppen übertragen Daten über den Datentransferbus (DTB), der Transportwege für die Daten, die Adressen und die zugehörigen Steuersignale enthält. Funktionsmodule, genannt MASTER, SLAVE, INTERRUPTER und INTERRUPT HANDLER, verwenden den DTB zum Transfer von Daten untereinander. Zwei weitere Module, genannt BUS TIMER und IACK-DAISY-CHAIN-TREIBER, unterstützen diese dabei.

b) DTB-Zuteilungsbus

Da ein IEC-821-BUS-System mit mehr als einem MASTER oder INTERRUPT HANDLER konfiguriert werden kann, wird eine Methode zur Verfügung gestellt, um die Zuteilung des DTB in geordneter Weise durchzuführen und sicherzustellen, daß zu jedem Zeitpunkt nur einer den DTB kontrolliert. Die Zuteilungsbus-Module, REQUESTER und ARBITER, koordinieren die Weitergabe der Kontrolle.

c) Interrupt-Bus

Die Fähigkeiten des IEC-821-BUS zum Prioritätsinterrupt gibt den Baugruppen die Möglichkeit, die Bedienung durch den INTERRUPT HANDLER anzufordern. Diese Interrupt-Requests können in maximal sieben Prioritätsstufen angeordnet sein. INTERRUPTER und INTERRUPT HANDLER benutzen die Interrupt-Bus-Signalleitungen.

d) Utility-Bus

Taktsignale, Initialisierung und Fehlererkennung werden durch den Utility-Bus zur Verfügung gestellt. Er enthält zwei Taktleitungen, eine System-Reset-Leitung, eine Systemfehlerleitung, eine Spannungsausfall-Leitung und eine serielle Datenleitung.

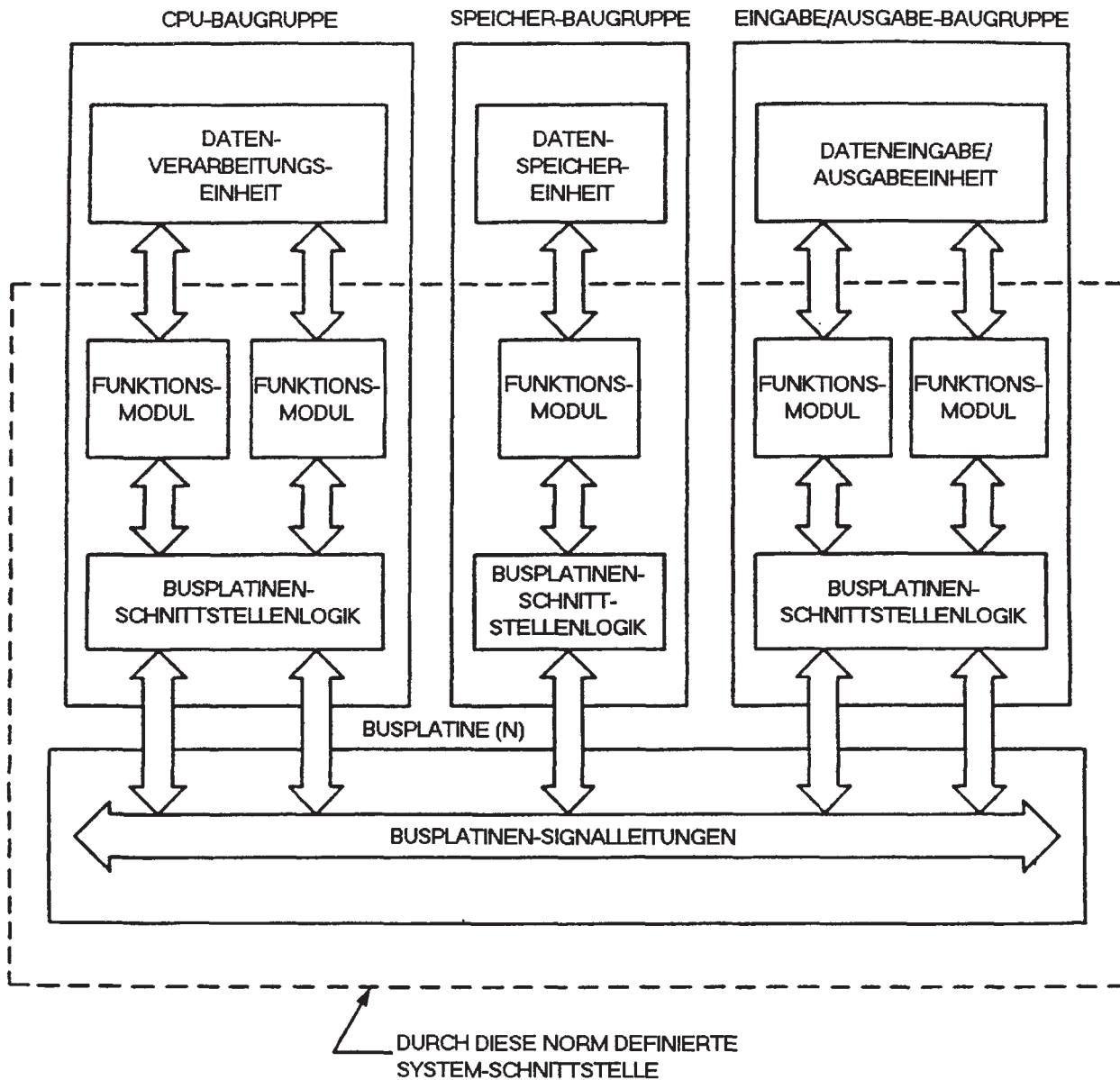


Bild 1-1: Systembausteine nach dieser Norm

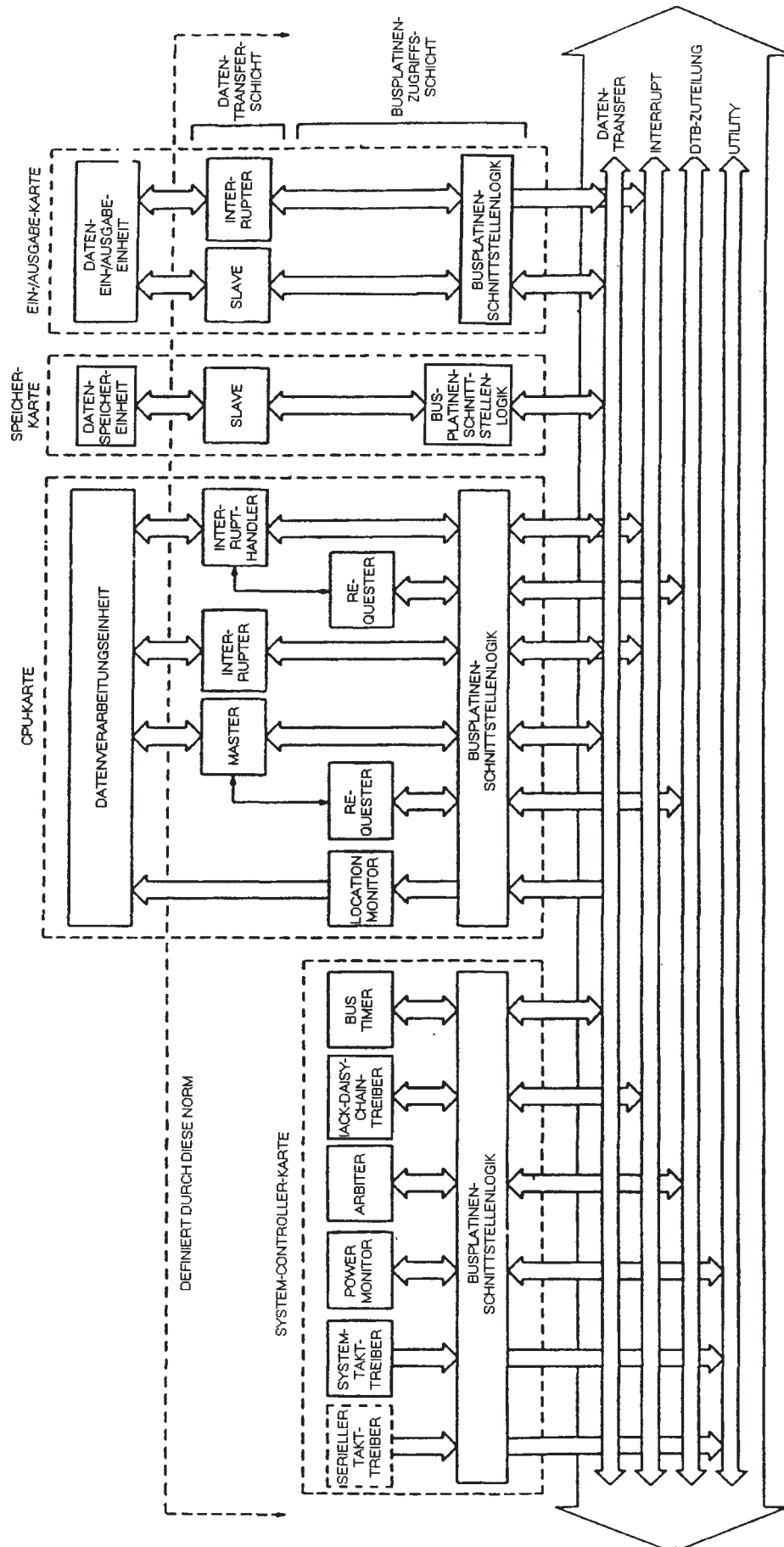


Bild 1-2: Funktionsmodule und Busse nach dieser Norm

1.3 Diagramme in der IEC-821-BUS-Norm

Zur Festlegung und Beschreibung der IEC-821-BUS-Protokolle werden verschiedene Arten von Diagrammen benutzt:

- a) Zeitdiagramme zeigen die zeitlichen Beziehungen der Signalpegelübergänge. Die angegebenen Zeiten haben zugeordnete minimale und/oder maximale Grenzwerte. Einige der Zeiten bestimmen das Verhalten der Schnittstellenlogik der Busplatine, andere das Zusammenwirken der Funktionsmodule.
- b) Ablaufdiagramme sind einem Zeitdiagramm ähnlich, zeigen jedoch nur die zeitliche Aufeinanderfolge von Signalen der Funktionsmodule. Sie zeigen die Abfolge von Ereignissen, anstatt die zugehörigen Zeiten festzulegen. Ein Ablaufdiagramm könnte z. B. aufzeigen, daß das Modul A den Signalübergang B erst dann erzeugen kann, wenn es den durch Modul C erzeugten Signalübergang D festgestellt hat.
- c) Flußdiagramme zeigen den Ablauf von Ereignissen, wie sie während einer IEC-821-BUS-Operation auftreten würden. Die Ereignisse werden verbal beschrieben und ergeben sich aus dem Zusammenspiel von zwei oder mehr Funktionsmodulen. Sie beschreiben die IEC-821-BUS-Operationen in Folgeschritten und zeigen gleichzeitig die Wechselbeziehung der Funktionsmodule.

1.4 In der Norm verwendete Terminologie

Um Verwirrung zu vermeiden und um die Anforderungen bezüglich der Übereinstimmung eindeutig herauszustellen, sind viele Absätze dieser Norm durch Schlüsselbegriffe gekennzeichnet, die die enthaltene Informationsart bezeichnen. Die Schlüsselbegriffe sind:

VORSCHRIFT
EMPFEHLUNG
VORSCHLAG
ERLAUBNIS
ERLÄUTERUNG

Jeder nicht mit einem dieser Schlüsselbegriffe gekennzeichnete Text beschreibt den IEC-821-BUS-Aufbau oder seine Wirkungsweise. Die Beschreibung erfolgt entweder in beschreibender oder berichtender Form. Die Schlüsselbegriffe werden wie folgt angewendet:

VORSCHRIFT Kapitel.Nummer:

Vorschriften bilden das Grundgerüst der IEC-821-BUS-Norm. Sie sind sowohl in Schriftform als auch in Form von Abbildungen, Tabellen oder Zeichnungen ausgedrückt. Alle IEC-821-BUS-Vorschriften MÜSSEN befolgt werden, um die Kompatibilität der IEC-821-BUS-Entwicklungen sicherzustellen. Die Vorschriften sind im Imperativ abgefaßt. Die großgeschriebenen Worte MUSS/MÜSSEN und DARF/DÜRFEN NICHT sind ausschließlich zur Formulierung der Vorschriften in dieser Norm reserviert und werden für keinen anderen Zweck benutzt.

EMPFEHLUNG Kapitel.Nummer:

Immer wenn eine Empfehlung erscheint, sollten Entwickler den gegebenen Ratschlag beachten. Wer anders verfährt, handelt sich möglicherweise tückische Probleme oder schlechte Leistungsdaten ein. Obwohl der IEC-821-BUS zur Unterstützung von Hochleistungssystemen entwickelt wurde, ist es möglich, ein IEC-821-BUS-System zu entwickeln, das allen Vorschriften entspricht und dennoch miserable Leistungsdaten hat. In vielen Fällen benötigt der Entwickler ein gewisses Maß an Erfahrung mit dem IEC-821-BUS, um Baugruppen entwickeln zu können, die ausgezeichnete Leistungsdaten haben. Die in dieser Norm enthaltenen Empfehlungen beruhen auf dieser Erfahrung und sind dazu da, daß beim Entwickler die Lernkurve schnell steigt.

VORSCHLAG Kapitel.Nummer:

Ein Vorschlag enthält einen hilfreichen, jedoch nicht lebens-

wichtigen Rat. Der Leser wird ermuntert, den Ratschlag zu überdenken, bevor er ihn verwirft. Einige Entscheidungen sind schwierig, solange noch keine Erfahrungen mit dem IEC-821-BUS vorliegen. Vorschläge sind zur Unterstützung des Entwicklers, der diese Erfahrung bisher nicht gesammelt hat, beigefügt. Einige Vorschläge betreffen die Entwicklung von Baugruppen, die leicht rekonfiguriert werden können, um die Kompatibilität mit anderen Baugruppen zu erreichen, andere die Entwicklung von Baugruppen, die die Systemfehlerbeseitigung erleichtern.

ERLAUBNIS Kapitel.Nummer:

In einigen Fällen verbietet eine IEC-821-BUS-Vorschrift einen bestimmten Entwurfsansatz nicht ausdrücklich, der Leser könnte jedoch darüber im Zweifel sein, ob dieser Ansatz die Vorschrift verletzt oder zu einem kniffligen Problem führen könnte. Erlaubnisse zeigen dem Leser, daß ein bestimmter Ansatz erlaubt ist und keine Probleme hervorgerufen werden. Das großgeschriebene Wort DARF/DÜRFEN ist allein der Formulierung von Erlaubnissen in dieser Norm vorbehalten und wird nicht für andere Zwecke benutzt.

ERLÄUTERUNG Kapitel.Nummer:

Erläuterungen beinhalten keine besonderen Ratschläge. Normalerweise ergeben sie sich aus dem, was gerade diskutiert wurde. Sie erläutern die Auswirkungen bestimmter IEC-821-BUS-Vorschriften und machen auf Dinge aufmerksam, die gern übersehen werden. Sie begründen auch bestimmte Vorschriften, so daß der Leser versteht, warum die Vorschrift befolgt werden muß.

1.4.1 Zustände auf den Signalleitungen

Die IEC-821-BUS-Protokolle werden in Form von Pegeln und Übergängen auf den Busleitungen beschrieben.

Man geht davon aus, daß eine Signalleitung immer einen von zwei Pegeln hat oder sich im Übergang zwischen diesen Pegeln befindet. Immer wenn der Begriff "high" benutzt wird, bezieht er sich auf einen hohen TTL-Spannungspegel. Der Begriff "low" bezieht sich auf einen niedrigen TTL-Spannungspegel. Eine Signalleitung befindet sich "im Übergang", wenn ihre Spannung sich zwischen diesen Pegeln bewegt. (Siehe Kapitel 6 für die auf dem IEC-821-BUS benutzten Spannungsschwellen.)

Es gibt zwei mögliche Übergänge, die auf einer Signalleitung auftreten können, diese werden "Flanken" genannt. Eine ansteigende Flanke ist die Zeit, in der sich eine Signalleitung im Übergang von einem niedrigen zu einem hohen Pegel befindet. Die abfallende Flanke ist die Zeit, in der die Signalleitung sich im Übergang vom hohen zum niedrigen Pegel befindet.

Einige Bus-Spezifikationen schreiben die maximalen und minimalen Anstiegs- und Abfallzeiten dieser Flanken vor. Das damit verbundene Problem ist, daß die Entwickler der Baugruppen kaum eine Kontrolle über diese Zeiten haben. Wenn die Busplatine stark belastet ist, sind die Anstiegs- und Abfallzeiten lang. Ist sie gering belastet, können diese Zeiten kurz sein. Selbst wenn die Entwickler die maximale und minimale Belastung kennen, müssen sie dennoch Zeit mit Versuchen im Labor verbringen, um herauszufinden, welche Treiber die erforderlichen Anstiegs- und Abfallzeiten liefern.

Tatsächlich sind die Anstiegs- und Abfallzeiten das Ergebnis komplexer Wechselwirkungen, beeinflusst durch die Impedanzen der Signalleitungen der Busplatine, deren Abschlüsse, die Quellimpedanz der Treiber und die kapazitive Belastung der Signalleitung. Um alle diese Faktoren zu berücksichtigen, müßte der Entwickler die Theorie der Übertragungsleitungen sowie gewisse spezifische Parameter der Treiber und Empfänger, die man üblicherweise nicht in den Datenblättern der Hersteller findet, studieren.

Unter Berücksichtigung dieser Tatsachen legt die IEC-821-BUS-Norm keine Anstiegs- und Abfallzeiten fest. Stattdessen

legt sie die elektrischen Eigenschaften der Treiber und Empfänger fest und macht Vorschläge für die Konstruktion der Busplatine. Den Entwicklern wird ebenfalls erläutert, wie die Worst-Case-Busbelastung die Signallaufzeit dieser Treiber beeinflusst, so daß sie, bevor mit der Entwicklung der Baugruppe begonnen wird, sicherstellen können, daß die erforderlichen Zeiten eingehalten werden. Wenn die Entwickler von IEC-821-BUS-Produkten die Erlaubnisse bezüglich der Signallaufzeit befolgen, werden ihre Baugruppen, auch unter Worst-Case-Bedingungen, zusammen mit anderen zum IEC-821-BUS-kompatiblen Baugruppen zuverlässig funktionieren.

1.4.2 Verwendung des Sternchens (*)

Um die Benutzung von Signalen zu definieren, ist ihrem Kurzzeichen bei Bedarf ein Sternchen angefügt:

- Folgt bei Signalen, deren Pegel bedeutsam ist, dem Signalnamen ein Sternchen (*), so zeigt dies, daß das Signal true oder wahr ist, wenn es low ist.
- Folgt bei Signalen, deren Flanke bedeutsam ist, dem Signalnamen ein Sternchen (*), zeigt dies, daß die vom Signal ausgelösten Aktionen beim Übergang von high nach low auftreten.

ERLÄUTERUNG 1.1:

Für die asynchronen Taktleitungen SYSCLK und SERCLK ist ein Sternchen nicht angebracht. Es gibt keine feste Phasenbeziehung zwischen diesen Taktleitungen und anderen IEC-821-BUS-Signalen.

1.5 Protokollarten

Es gibt zwei Schichten im IEC-821-BUS-Protokoll. Die unterste IEC-821-BUS-Schicht, genannt Busplatinenzugriffsschicht, besteht aus der Schnittstellenlogik der Busplatine,

den Utility-Bus-Modulen und den Zuteilungsbuss-Modulen. Die IEC-821-BUS-Datentransferschicht besteht aus dem Datentransferbus und den Interrupt-Bus-Modulen. Bild 1-2 zeigt diese Schichten.

ERLÄUTERUNG 1.2:

Die von den Modulen der Datentransferschicht verwendeten Signalleitungen bilden eine spezielle Klasse, weil sie durch unterschiedliche Module zu verschiedenen Zeiten getrieben werden. Sie werden von Leitungstreibern getrieben, die auf jeder Baugruppe aufgrund der in der Busplatinenzugriffsschicht erzeugten Signale ein- und ausgeschaltet werden können. Es ist sehr wichtig, daß ihre Ein- und Ausschaltzeiten sorgfältig kontrolliert werden, um zu verhindern, daß zwei Treiber versuchen, die gleiche Signalleitung auf verschiedene Pegel zu treiben. Eine besondere Notation der Zeitdiagramme wird in dieser Norm zur Festlegung ihrer Ein- und Ausschaltzeiten verwendet. Dies ist in Bild 1-3 veranschaulicht.

Es werden zwei Grundformen des Protokolls für den IEC-821-BUS verwendet: Protokolle für Quittungsbetrieb und Protokolle ohne Quittung. Protokolle für Quittungsbetrieb verwenden verschränkte Bussignale, während Protokolle ohne Quittung Broadcast-Bussignale verwenden.

1.5.1 Verschränkte Bussignale

Ein verschränktes Bussignal wird von einem bestimmten Modul zu einem anderen bestimmten Modul gesendet. Das Signal wird durch das Empfangsmodul bestätigt. Eine solche Verschränkung besteht zwischen den beiden Modulen so lange, bis das Signal bestätigt worden ist. So kann z. B. ein INTERRUPTER einen Interrupt-Request senden, welcher später mit einem Interrupt-Acknowledge-Signal beantwortet wird (durch die IEC-821-BUS-Norm wird hier keine Zeitbe-

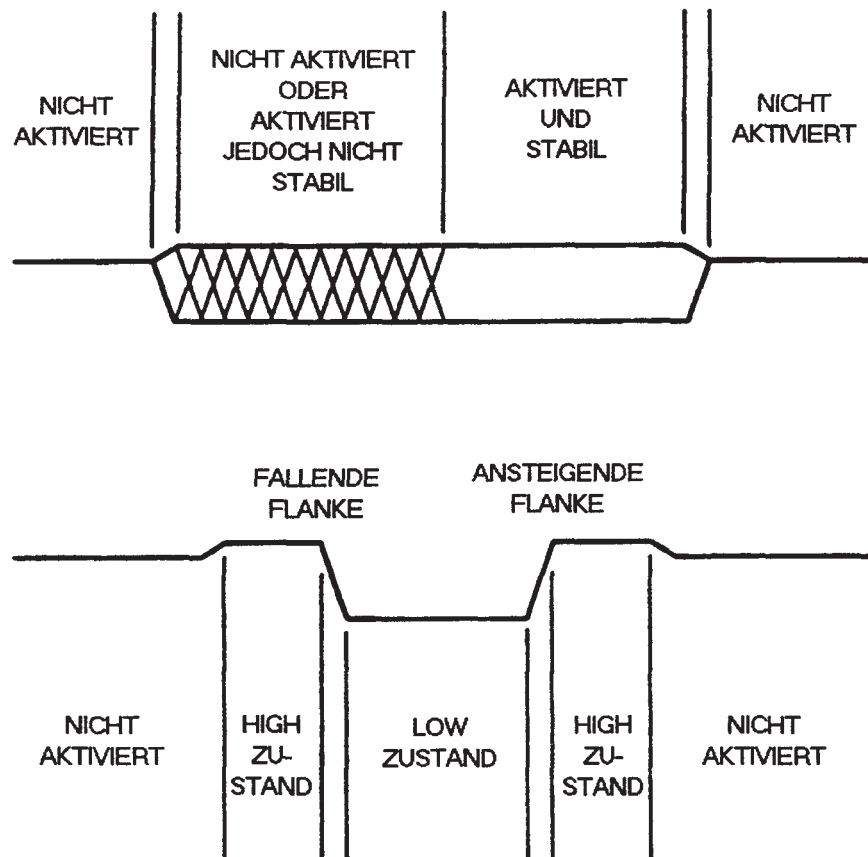


Bild 1-3: Darstellung des zeitlichen Signalverlaufs

grenzung vorgeschrieben). Der INTERRUPTER nimmt den Interrupt-Request erst zurück, wenn der INTERRUPT HANDLER bestätigt hat.

Verschränkte Bussignale koordinieren interne Funktionen des IEC-821-BUS-Systems und stehen nicht in einer Wechselbeziehung mit externen Impulsen. Jedes verschränkte Signal hat ein Quell- und Zielmodul im IEC-821-BUS-System.

Der Adreß-Strobe und die Daten-Strobes sind besonders wichtige verschränkte Signale. Sie sind verschränkt mit dem Datentransfer-Acknowledge- und dem Bus-Error-Signal und koordinieren den Transfer der Adreßinformation und Daten, die die Grundlage für den gesamten Informationsfluß zwischen den Modulen in der Datentransferschicht bilden.

1.5.2 Broadcast-Bussignal

Ein Modul erzeugt ein Broadcast-Signal als Reaktion auf ein Ereignis. Es gibt kein Bestätigungsprotokoll für ein Broadcast-Signal. Stattdessen wird der Broadcast eine festgelegte Mindestzeit aufrechterhalten, lange genug, um sicherzustellen, daß alle in Frage kommenden Module das Signal erkennen. Broadcast-Signale können jederzeit aktiviert werden,

ungeachtet anderer auf dem Bus stattfindender Aktivitäten. Sie werden jeweils über eine fest zugeordnete Signalleitung gesendet. Beispiele sind die Reset- und Spannungsausfall-Leitungen. Diese Signale werden nicht an spezielle Module gesendet, sondern teilen allen Modulen besondere Betriebsbedingungen mit.

1.6 Systembeispiele und Erklärungen

Eine Protokollspezifikation beschreibt ausführlich das Verhalten der verschiedenen Funktionsmodule. Sie erörtert, wie ein Modul auf ein Signal reagiert, ohne zu sagen, woher das Signal kam. Aus diesem Grunde liefert eine Protokollspezifikation dem Leser kein vollständiges Bild. Als Hilfe für den Leser bietet die IEC-821-BUS-Norm Beispiele für typische IEC-821-BUS-Operationen. Jedes Beispiel zeigt einen möglichen Ereignisablauf; andere Abläufe sind ebenfalls möglich. Durch diese gegebenen Beispiele besteht die Gefahr, daß die Leser annehmen, daß der im Beispiel aufgeführte Ablauf der einzig legale ist. Um hier keine Mißverständnisse entstehen zu lassen, werden alle Beispiele im Präsens und in Erzählform erläutert. Dies steht im Gegensatz zur Imperativform, die bei der Erstellung von Vorschriften im Hinblick auf die Übereinstimmung mit der IEC-821-BUS-Norm angewendet wird.

Kapitel 2: IEC-821-BUS – Datentransfer

2.1 Einführung

Der IEC-821-BUS enthält einen schnellen asynchronen, parallelen Datentransferbus (DTB). Bild 2-1 zeigt ein typisches IEC-821-BUS-System einschließlich aller DTB-Funktionsmodule. Der MASTER verwendet den DTB zur Auswahl der vom SLAVE bereitgestellten Speicherplätze und zum Datentransfer von und zu diesen Plätzen. MASTER und SLAVE verwenden entweder alle DTB-Leitungen oder nur eine Teilmenge davon.

LOCATION MONITORE überwachen den Datentransfer zwischen MASTER und SLAVE. Wenn ein Zugriff auf einen der überwachten Byteplätze erfolgt, erzeugt der LOCATION MONITOR ein karteninternes Signal. Zum Beispiel könnte er dem auf der Karte befindlichen Prozessor mittels eines Interrupt-Request etwas melden. Wenn in einer solchen Konfiguration die Prozessorkarte A in einen Speicherplatz des globalen IEC-821-BUS-Speichers schreibt, welche durch den LOCATION MONITOR von Prozessor B überwacht wird, wird Prozessor B unterbrochen.

Nachdem ein MASTER einen Datentransfer ausgelöst hat, wartet er auf die Antwort vom reagierenden SLAVE, bevor der Zyklus abgeschlossen wird. Die asynchrone Arbeitsweise des IEC-821-BUS gibt einem SLAVE ausreichend Zeit, um zu antworten. Wenn ein SLAVE aufgrund eines Fehlers nicht reagiert oder wenn der MASTER versehentlich einen Speicherplatz adressiert, an dem kein SLAVE vorhanden ist, greift der BUS TIMER ein und ermöglicht den Abschluß des Zyklus.

2.2 Leitungen des Datentransferbus (DTB)

Die Datentransferbusleitungen können in drei Kategorien eingeteilt werden:

Adreßleitungen	Datenleitungen	Steuerleitungen
A01-A31	D00-D31	AS*
AM0-AM5		DS0*
DS0*		DS1*
DS1*		BERR*
LWORD*		DTACK*
		WRITE*

ERLÄUTERUNG 2.1:

Die beiden Daten-Strobes (DS0* und DS1*) haben zwei Funktionen:

- 1) Ihre Pegel werden zur Auswahl des oder der Bytes, auf die zugegriffen wird, benutzt.
- 2) Ihre Flanken werden auch als Taktsignale verwendet, welche den Transfer von Daten zwischen MASTER und SLAVE koordinieren.

2.2.1 Adreßleitungen

Die kleinste adressierbare Speichereinheit ist ein Byte. Jedem Byte-Speicherplatz ist eine eindeutige binäre Adresse zugeordnet. Jeder Byte-Speicherplatz kann einer von vier Kategorien gemäß den zwei niedrigstwertigen Bits seiner Adresse (siehe Tabelle 2-1) zugeordnet sein.

Eine Anzahl von Byte-Speicherplätzen, deren Adressen sich nur in den zwei niedrigstwertigen Bits unterscheiden, wird eine 4-Byte-Gruppe oder eine BYTE(0-3)-Gruppe genannt. Einige oder alle Bytes in einer 4-Byte-Gruppe können gleichzeitig in einem einzigen DTB-Zyklus angesprochen werden.

Die MASTER benutzen die Adreßleitungen A02-A31, um auszuwählen, auf welche 4-Byte-Gruppe zugegriffen werden soll. Vier zusätzliche Leitungen, DS1*, DS0*, A01 und LWORD*, werden anschließend benutzt, um die Byte-Speicherplätze in der 4-Byte-Gruppe auszuwählen, die während des Datentransfers angesprochen werden. Mit Hilfe dieser vier Leitungen kann ein MASTER 1-, 2-, 3- oder 4-Byte-Speicherplätze gleichzeitig ansprechen, wie in Tabelle 2-2 gezeigt.

Tabelle 2-1: Die vier Arten von Byte-Speicherplätzen

Kategorie	Byte-Adresse
BYTE(0)	XXXXXX XXXXXX00
BYTE(1)	XXXXXX XXXXXX01
BYTE(2)	XXXXXX XXXXXX10
BYTE(3)	XXXXXX XXXXXX11

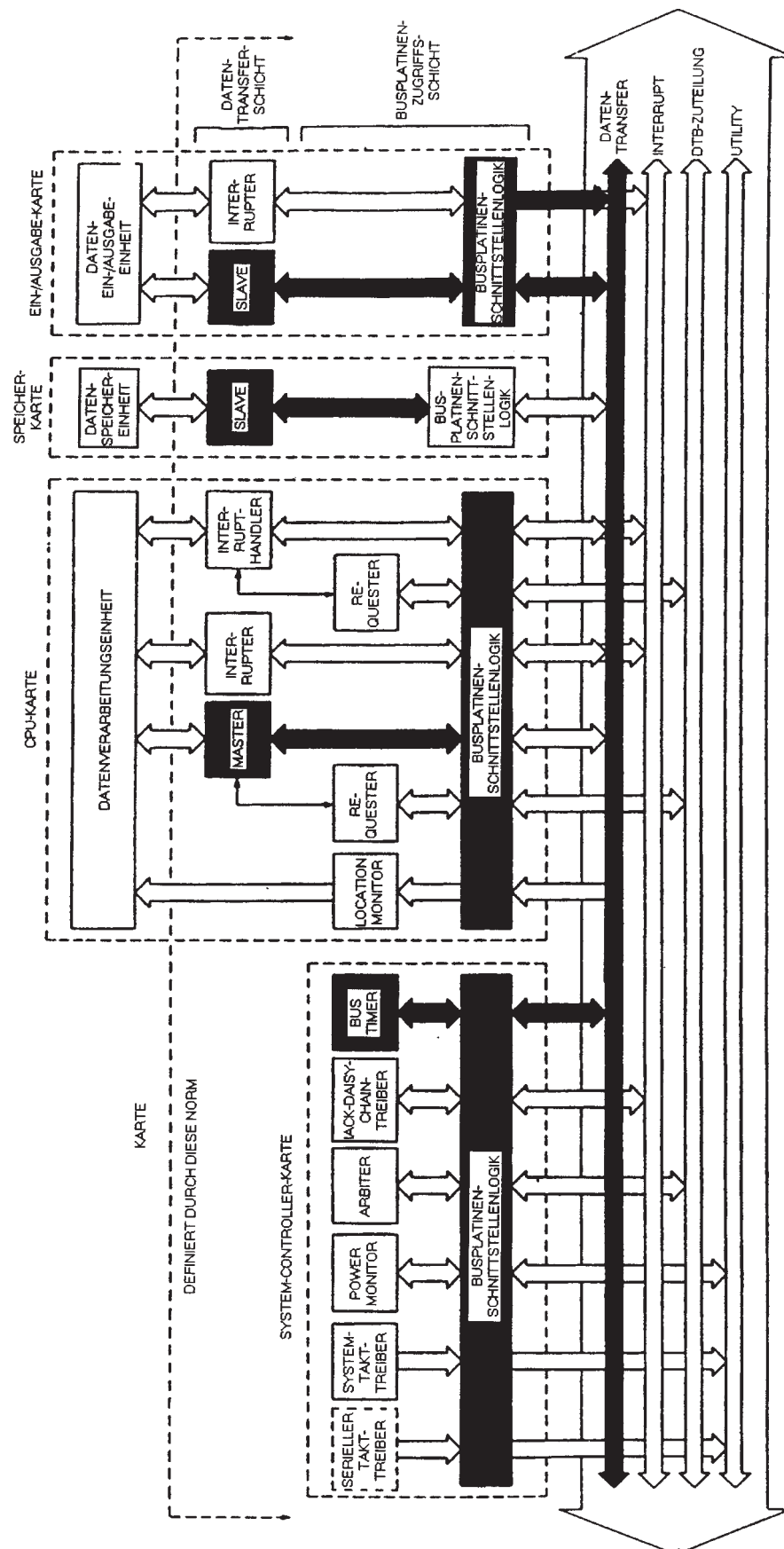


Bild 2-1: Funktionsschaltbild für den Datentransferbus

ERLÄUTERUNG 2.2:

Wenn beide Daten-Strobes auf low getrieben werden, könnte ein Daten-Strobe etwas später als der andere nach low gehen. In diesem Fall stellen die Signalpegel in Tabelle 2-2 die Endpegel dar.

ERLÄUTERUNG 2.3:

Mit den in der Tabelle 2-2 angegebenen vier Signalleitungspegeln sind 16 Pegelkombinationen möglich. Von diesen 16 Kombinationen sind zwei verboten und werden nicht verwendet (siehe Tabelle in VORSCHRIFT 2.1).

VORSCHRIFT 2.1:

MASTER DÜRFEN NICHT DTB-Zyklen erzeugen, in denen die Pegel von DS0*, DS1*, A01 und LWORD* einer der folgenden verbotenen Kombinationen entsprechen:

DS1*	DS0*	A01	LWORD*
high	low	high	low
low	high	high	low

ERLAUBNIS 2.1:

Wenn ein MASTER auf die Byte-Speicherplätze BYTE(1-2) zugreift, DARF er jede der zwei in VORSCHRIFT 2.1 beschriebenen Kombinationen kurzzeitig als Übergangszustände erzeugen (d. h. während ein Daten-Strobe abgefallen ist, der andere jedoch noch nicht).

ERLÄUTERUNG 2.4:

Immer wenn ein MASTER LWORD* low und A01 high treibt, wird er beide Daten-Strobes low treiben. (Jede andere Kom-

bination ist verboten.) Entwickler von IEC-821-BUS-Baugruppen können dies ausnutzen, um die Logik im SLAVE zu vereinfachen.

ERLAUBNIS 2.2:

Um die erforderliche Logik zu vereinfachen, DÜRFEN SLAVES, die auf einen Zyklus antworten, welche auf BYTE-(1-2)-Speicherstellen zugreifen, ohne Logik entwickelt werden, die zwischen diesen Zyklen und den in VORSCHRIFT 2.1 beschriebenen verbotenen Zyklen unterscheidet.

2.2.2 Adreßmodifizierer-Leitungen

Es gibt sechs Adreßmodifizierer-Leitungen. Sie erlauben dem MASTER, während DTB-Zyklen zusätzliche binäre Informationen an den SLAVE zu übergeben. Tabelle 2-3 zeigt alle 64 möglichen Adreßmodifizierer-(AM-)Codes und teilt jeden einer von drei Kategorien zu:

Festgelegt
Reserviert
Frei verfügbar

Die festgelegten Adreßmodifizierer-Codes können in weitere drei Kategorien unterteilt werden:

- AM-Codes mit Kurzadressierung zeigen an, daß die Adreßleitungen A02-A15 zur Auswahl einer BYTE(0-3)-Gruppe benutzt werden.
- AM-Codes mit Standardadressierung zeigen an, daß die Adreßleitungen A02-A23 zur Auswahl einer BYTE(0-3)-Gruppe benutzt werden.
- AM-Codes mit erweiterter Adressierung zeigen an, daß die Adreßleitungen A02-A31 zur Auswahl einer BYTE(0-3)-Gruppe benutzt werden.

Tabelle 2-2: Verwendung von DS0*, DS1*, A01 und LWORD* zur Auswahl von Byte-Speicherstellen

Ausgewählte Byte-Speicherstellen	DS1*	DS0*	A01	LWORD*
Ein-Byte-Zugriff				
BYTE(0)	low	high	low	high
BYTE(1)	high	low	low	high
BYTE(2)	low	high	high	high
BYTE(3)	high	low	high	high
Zwei-Byte-Zugriff				
BYTE(0-1)	low	low	low	high
BYTE(1-2)	low	low	high	low
BYTE(2-3)	low	low	high	high
Drei-Byte-Zugriff				
BYTE(0-2)	low	high	low	low
BYTE(1-3)	high	low	low	low
Vier-Byte-Zugriff				
BYTE(0-3)	low	low	low	low

Tabelle 2-3: Adreßmodifizierer-Codes

HEX-CODE	ADRESS-MODIFIZIERER						FUNKTION
	5	4	3	2	1	0	
3F	H	H	H	H	H	H	Standard Supervisory Blocktransfer
3E	H	H	H	H	H	L	Standard Supervisory Programmzugriff
3D	H	H	H	H	L	H	Standard Supervisory Datenzugriff
3C	H	H	H	H	L	L	Reserviert
3B	H	H	H	L	H	H	Standard non-privileged Blocktransfer
3A	H	H	H	L	H	L	Standard non-privileged Programmzugriff
39	H	H	H	L	L	H	Standard non-privileged Datenzugriff
38	H	H	H	L	L	L	Reserviert
37	H	H	L	H	H	H	Reserviert
36	H	H	L	H	H	L	Reserviert
35	H	H	L	H	L	H	Reserviert
34	H	H	L	H	L	L	Reserviert
33	H	H	L	L	H	H	Reserviert
32	H	H	L	L	H	L	Reserviert
31	H	H	L	L	L	H	Reserviert
30	H	H	L	L	L	L	Reserviert
2F	H	L	H	H	H	H	Reserviert
2E	H	L	H	H	H	L	Reserviert
2D	H	L	H	H	L	H	Kurzer Supervisory-Zugriff
2C	H	L	H	H	L	L	Reserviert
2B	H	L	H	L	H	H	Reserviert
2A	H	L	H	L	H	L	Reserviert
29	H	L	H	L	L	H	kurzer non-privileged Datenzugriff
28	H	L	H	L	L	L	Reserviert
27	H	L	L	H	H	H	Reserviert
26	H	L	L	H	H	L	Reserviert
25	H	L	L	H	L	H	Reserviert
24	H	L	L	H	L	L	Reserviert
23	H	L	L	L	H	H	Reserviert
22	H	L	L	L	H	L	Reserviert
21	H	L	L	L	L	H	Reserviert
20	H	L	L	L	L	L	Reserviert
1F	L	H	H	H	H	H	frei verfügbar
1E	L	H	H	H	H	L	frei verfügbar
1D	L	H	H	H	L	H	frei verfügbar
1C	L	H	H	H	L	L	frei verfügbar
1B	L	H	H	L	H	H	frei verfügbar
1A	L	H	H	L	H	L	frei verfügbar
19	L	H	H	L	L	H	frei verfügbar
18	L	H	H	L	L	L	frei verfügbar
17	L	H	L	H	H	H	frei verfügbar
16	L	H	L	H	H	L	frei verfügbar
15	L	H	L	H	L	H	frei verfügbar
14	L	H	L	H	L	L	frei verfügbar
13	L	H	L	L	H	H	frei verfügbar
12	L	H	L	L	H	L	frei verfügbar
11	L	H	L	L	L	H	frei verfügbar
10	L	H	L	L	L	L	frei verfügbar
0F	L	L	H	H	H	H	Erweiterter Supervisory Blocktransfer
0E	L	L	H	H	H	L	Erweiterter Supervisory Programmzugriff
0D	L	L	H	H	L	H	Erweiterter Supervisory Datenzugriff
0C	L	L	H	H	L	L	Reserviert
0B	L	L	H	L	H	H	Erweiterter non-privileged Blocktransfer
0A	L	L	H	L	H	L	Erweiterter non-privileged Programmzugriff
09	L	L	H	L	L	H	Erweiterter non-privileged Datenzugriff
08	L	L	H	L	L	L	Reserviert
07	L	L	L	H	H	H	Reserviert
06	L	L	L	H	H	L	Reserviert
05	L	L	L	H	L	H	Reserviert
04	L	L	L	H	L	L	Reserviert
03	L	L	L	L	H	H	Reserviert
02	L	L	L	L	H	L	Reserviert
01	L	L	L	L	L	H	Reserviert
00	L	L	L	L	L	L	Reserviert

L Signalpegel Low H Signalpegel High

VORSCHRIFT 2.2:

Mit Ausnahme der verfügbaren Codes DÜRFEN die in der Tabelle 2-3 definierten Codes NICHT für andere als die vorgeschriebenen Zwecke verwendet werden.

VORSCHRIFT 2.3:

SLAVES DÜRFEN NICHT auf die reservierten Adreßmodifizierer-Codes ansprechen.

ERLÄUTERUNG 2.5:

Die reservierten Adreßmodifizierer-Codes sind für zukünftige Erweiterungen vorgesehen. Wenn SLAVE-Karten auf diese Codes reagieren, könnten später einmal Inkompatibilitäten auftreten, wenn die Benutzung dieser Codes definiert wird.

ERLAUBNIS 2.3:

Frei verfügbare Codes DÜRFEN für jeden Zweck verwendet werden, der den Kartenherstellern oder Anwendern als geeignet erscheint (Seitenwechsel, Speichersicherung, MASTER- oder Task-Identifikation, privilegierter Zugriff auf Programme, usw.).

EMPFEHLUNG 2.1:

Um es den Benutzern zu ermöglichen, den Gebrauch der frei verfügbaren Adreßmodifizierer-Codes ihren eigenen Belangen anzupassen, empfiehlt es sich, diese auf den SLAVE-Karten flexibel zu decodieren. Die Benutzer können die Karte dann so konfigurieren, daß jede von ihrem System verlangte Dekodierung möglich ist.

ERLÄUTERUNG 2.6:

Steckbare programmierbare Bausteine bieten eine flexible Art der Dekodierung von Adreßmodifizierer-Codes.

VORSCHLAG 2.1:

Werden SLAVES mit einem programmierten, gesockelten Baustein (z. B. PROM oder FPLA) hergestellt, so programmiere man den Baustein so, daß der SLAVE auf die folgenden AM-Codes anspricht:

A16 SLAVES mit D08(O), D08(E0), D16 oder D32 Fähigkeit:
29, 2D

A24 SLAVES mit D08(O), D08(E0), D16 oder D32 Fähigkeit:
39, 3A, 3D und 3E

A32 SLAVES mit D08(O), D08(E0), D16 oder D32 Fähigkeit:
09, 0A, 0D und 0E

A24 SLAVES mit BLT Fähigkeit:
3B, 3F

A32 SLAVES mit BLT Fähigkeit:
0B, 0F

Die Mnemonics A16, A24 und A32 sind in Tabelle 2-9 definiert.

Die Mnemonics D08(O), D08(E0), D16, D32 und BLT sind in den Tabellen 2-10 und 2-11 definiert.

2.2.3 Datenleitungen

Systeme können mit einer Busplatinen-Anordnung gebaut werden, die entweder 16 Datenleitungen (D00-D15) oder 32 Datenleitungen (D00-D31) enthält. Busplatinen mit 16 Datenleitungen ermöglichen einem MASTER nur den gleichzeitigen Zugriff auf zwei Byte-Speicherplätze, während ihm jene mit 32 Datenleitungen den Zugriff auf bis zu vier Byte-Speicherplätze erlauben. Wenn der MASTER bei Nutzung der in Abschnitt 2.2.1 beschriebenen Methode 1-, 2-, 3- oder 4-Byte-Speicherplätze ausgewählt hat, kann er binäre Daten zwischen sich und diesen Plätzen über den Datenbus übertragen. Tabelle 2-4 zeigt, wie die Datenleitungen benutzt werden, um auf diese Byte-Speicherplätze zuzugreifen.

2.2.4 Steuerleitungen

Folgende Signalleitungen werden zur Steuerung des Daten-Transfers über die Daten-Transferleitungen verwendet:

AS*	Adreß-Strobe
DS0*	Daten-Strobe
DS1*	Daten-Strobe
BERR*	Bus-Fehler
DTACK*	Daten-Transferbestätigung
WRITE*	Lesen/Schreiben

2.2.4.1 AS*

Eine fallende Flanke auf AS* informiert alle SLAVE-Module, daß die Adresse stabil ist und übernommen werden kann.

2.2.4.2 DS0* und DS1*

Zusätzlich zu ihrer in Abschnitt 2.2.1 beschriebenen Funktion, Bytespeicherplätze für den Datentransfer auszuwählen, haben die Daten-Strobes auch weitere Funktionen. Bei Schreibzyklen gibt die erste fallende Flanke des Daten-Strobes an, daß der MASTER gültige Daten auf den Datenbus gegeben hat. Bei Lesezyklen teilt die erste ansteigende Flanke dem SLAVE mit, wann er gültige Daten vom Datenbus übernehmen kann.

ERLÄUTERUNG 2.7:

Wie in Hauptabschnitt 2.6 beschrieben, dürfen IEC-821-BUS-MASTER keinen der Daten-Strobes low treiben, bevor AS* low getrieben ist. Dennoch könnten SLAVES und LOCATION MONITORE aufgrund der Tatsache, daß AS* auf der Busplatine mehr belastet ist als die Daten-Strobes, eine fallende Flanke auf einem Daten-Strobe feststellen, bevor sie die fallende Flanke von AS* erkennen.

Tabelle 2-4: Verwendung der Datenleitungen, um auf Byte-Speicherstellen zuzugreifen

Zugriff auf Byte-Speicherstellen	D24-D31	D16-D23	D08-D15	D00-D07
BYTE(0) BYTE(1) BYTE(2) BYTE(3)			BYTE(0) BYTE(2)	BYTE(1) BYTE(3)
BYTE(0-1) BYTE(1-2) BYTE(2-3)		BYTE(1)	BYTE(0) BYTE(2) BYTE(2)	BYTE(1) BYTE(3)
BYTE(0-2) BYTE(1-3)	BYTE(0)	BYTE(1) BYTE(1)	BYTE(2) BYTE(2)	BYTE(3)
BYTE(0-3)	BYTE(0)	BYTE(1)	BYTE(2)	BYTE(3)

ERLAUBNIS 2.5:

IEC-821-BUS-SLAVES, die keine Blocktransfer-Fähigkeiten haben, wie in Absatz 2.3.1 beschrieben, und LOCATION MONITORE DÜRFEN darauf ausgelegt werden, die Adresse zu übernehmen, wenn sie eine fallende Flanke auf einem Daten-Strobe anstelle der fallenden Flanke auf AS* feststellen.

ERLÄUTERUNG 2.8:

IEC-821-BUS-SLAVES und LOCATION MONITORE, die die Adresse während der fallenden Flanke des (der) Daten-Strobe(s) übernehmen, brauchen AS* nicht zu überwachen.

ERLÄUTERUNG 2.9:

Um die Vorteile des Address Pipelining, das in Abschnitt 2.4.2 beschrieben ist, vollständig auszunutzen oder Block-Lese- und Schreib-Zyklen durchzuführen, sollte ein SLAVE die Adresse mit der fallenden Flanke von AS* übernehmen.

2.2.4.3 DTACK*

Der SLAVE treibt DTACK* low, um anzuzeigen, daß er die Daten eines Schreibzyklus richtig empfangen hat. Bei einem Lesezyklus treibt der SLAVE DTACK* low, um anzuzeigen, daß er Daten auf den Datenbus gegeben hat.

2.2.4.4 BERR*

BERR* wird durch den SLAVE oder den BUS TIMER auf low getrieben, um dem MASTER anzuzeigen, daß der Datentransfer nicht erfolgreich war. Wenn zum Beispiel der MASTER versucht, in einen Speicherplatz, der ein READ-ONLY-MEMORY enthält, zu schreiben, kann der angesprochene SLAVE BERR* low treiben. Wenn der MASTER versucht, auf einen Speicherplatz zuzugreifen, der nicht durch einen SLAVE besetzt ist, würde der BUS TIMER nach Ablauf einer festgelegten Zeitspanne BERR* low treiben.

VORSCHLAG 2.2:

Man entwickle SLAVES so, daß sie mit einer fallenden Flanke auf BERR* antworten, wenn sie in den Daten, welche sie während eines Lesezyklus aus ihrem internen Speicher entnehmen, einen nicht-korrigierbaren Fehler feststellen.

2.2.4.5 WRITE*

WRITE* ist eine pegelabhängige Signalleitung, die durch die fallende Flanke der ersten Daten-Strobes (DSA*) getaktet

wird und gültig ist, solange einer der Daten-Strobes (DSA* oder DSB*) auf Low-Pegel ist. Sie wird vom MASTER benutzt, um die Richtung der Datentransferoperationen anzuzeigen. Wenn WRITE* low getrieben wird, ist die Datentransferrichtung vom MASTER zum SLAVE. Wenn WRITE* high getrieben wird, ist die Datentransferrichtung vom SLAVE zum MASTER.

2.3 DTB-Module – Grundlagen

Zusätzlich zum ADDRESS-ONLY-ZYKLUS definiert das DTB-Protokoll 33 verschiedene Zyklusarten, die für den Datentransfer verwendet werden. Jede dieser 34 Zyklusarten kann in einem der drei Adreßmodi verwendet werden: kurz (16 Bits), standard (24 Bits) und erweitert (32 Bits). Die Fähigkeiten der MASTER, der SLAVES und der LOCATION MONITORE werden in einer Liste von Mnemonics beschrieben, die angeben, welche Zyklusarten sie erzeugen, annehmen oder überwachen können. Diese Mnemonics werden in den Abschnitten 2.3.5 bis 2.3.10 beschrieben.

Die Abschnitte 2.3.1 bis 2.3.4 enthalten Blockdiagramme für die vier Arten der DTB-Funktionsmodule: MASTER, SLAVE, LOCATION MONITOR und BUS TIMER.

VORSCHRIFT 2.7:

Die in den Bildern 2-2 bis 2-5 angegebenen, durch durchgezogene Linien gekennzeichneten Ausgangssignalleitungen MÜSSEN durch das Modul getrieben werden, wenn es sie nicht ständig high treibt.

ERLÄUTERUNG 2.11:

WENN eine Ausgangssignalleitung nicht getrieben wird,
DANN stellen die Abschlußbeschaltungen auf der Busplatine sicher, daß sie high ist.

VORSCHRIFT 2.8:

Die als durchgehende Linien in den Bildern 2-2 bis 2-5 gezeigten Eingangssignalleitungen MÜSSEN in der entsprechenden Art überwacht und beantwortet werden.

ERLÄUTERUNG 2.12:

VORSCHRIFTEN und ERLAUBNISSE für die Ansteuerung und Überwachung von Signalleitungen, die durch gestrichelte Linien in den Bildern 2-2, 2-3 und 2-5 dargestellt sind, werden in den Tabellen 2-5, 2-6 und 2-8 angegeben.

Tabelle 2-5: MASTER: VORSCHRIFTEN und ERLAUBNISSE für das Treiben der gestrichelt gezeichneten Leitungen

MASTER-Typ	VORSCHRIFTEN und ERLAUBNISSE
D08(E0) und D16	MUSS D00-D15 überwachen und treiben. DARF oder DARF NICHT LWORD* treiben. DARF oder DARF NICHT D16-D31 treiben oder überwachen.
D32	MUSS LWORD* treiben. MUSS D00-D31 überwachen und treiben.
A16	MUSS A01-A15 treiben. DARF oder DARF NICHT A16-A31 treiben.
A24	MUSS A01-A23 treiben. DARF oder DARF NICHT A24-A31 treiben.
A32	MUSS A01-A31 treiben.
ALLE	DÜRFEN oder DÜRFEN NICHT BCLR* oder ACFAIL* überwachen (siehe Kapitel 3 und 5).

ANMERKUNG 1: Die Mnemonics D08(E0), D16 und D32 werden in Tabelle 2-10 definiert.

ANMERKUNG 2: Die Mnemonics A16, A24 und A32 werden in Tabelle 2-9 definiert.

2.3.1 MASTER

Das Funktionsschaltbild des MASTERS wird in Bild 2-2 gezeigt. Die gestrichelten Linien im Diagramm stellen Signale dar, deren Anwendung bei den verschiedenen MASTER-Typen unterschiedlich ist. Tabelle 2-5 legt die Anforderungen zum Treiben und Überwachen dieser Leitungen an die verschiedenen MASTER-Typen fest. Weitere VORSCHRIFTEN, wie die verschiedenen MASTER-Typen, die Adreßleitungen, Datenleitungen, LWORD*, DS0* und DS1* treiben, werden in den Tabellen 2-19, 2-20 und 2-21 angegeben.

2.3.2 SLAVE

Das Funktionsschaltbild des SLAVE wird in Bild 2-3 dargestellt. Die gestrichelten Linien im Diagramm stellen Signale dar, deren Anwendung in den verschiedenen SLAVE-Typen unterschiedlich ist. Tabelle 2-6 legt die Anforderungen zum Treiben und Überwachen dieser Leitungen für die verschiedenen SLAVE-Typen fest. Weitere Informationen, wie die verschiedenen SLAVE-Typen die Datenleitungen treiben, werden in Tabelle 2-21 angegeben.

2.3.3 BUS TIMER

Das Funktionsschaltbild des BUS TIMERS wird in Bild 2-4 dargestellt. Die BUS TIMER können verwendet werden, um

BERR* nach verschiedenen Zeitperioden low zu treiben. Tabelle 2-7 zeigt, wie das BT0()-Mnemonic angewendet wird, um die verschiedenen BUS-TIMER-Typen zu beschreiben.

ERLÄUTERUNG 2.13:

Die in Bild 2-4 aufgezeigten unterbrochenen DTACK*- und BERR*-Leitungen ermöglichen es, einen BUS TIMER in einer der zwei Möglichkeiten zu implementieren:

- BERR* wird low getrieben, wenn der erste Daten-Strobe länger als die Bus-Time-Out-Periode low bleibt, ohne Rücksicht auf die Pegel der DTACK*- und BERR*-Leitungen.
- BERR* wird low getrieben, wenn der erste Daten-Strobe länger als die Bus-Time-Out-Periode low bleibt, jedoch nur, wenn DTACK* und BERR* zum Zeitpunkt des Time-Out high sind.

2.3.4 LOCATION MONITOR

Das Funktionsschaltbild des LOCATION MONITORS wird in Bild 2-5 dargestellt. Die gestrichelt gezeichneten Linien im Diagramm zeigen Signale, deren Anwendung bei den verschiedenen Typen der LOCATION MONITORE verschieden ist. Tabelle 2-8 legt die Anforderungen an das Überwachen dieser Leitungen für die verschiedenen Typen der LOCATION MONITORE fest.

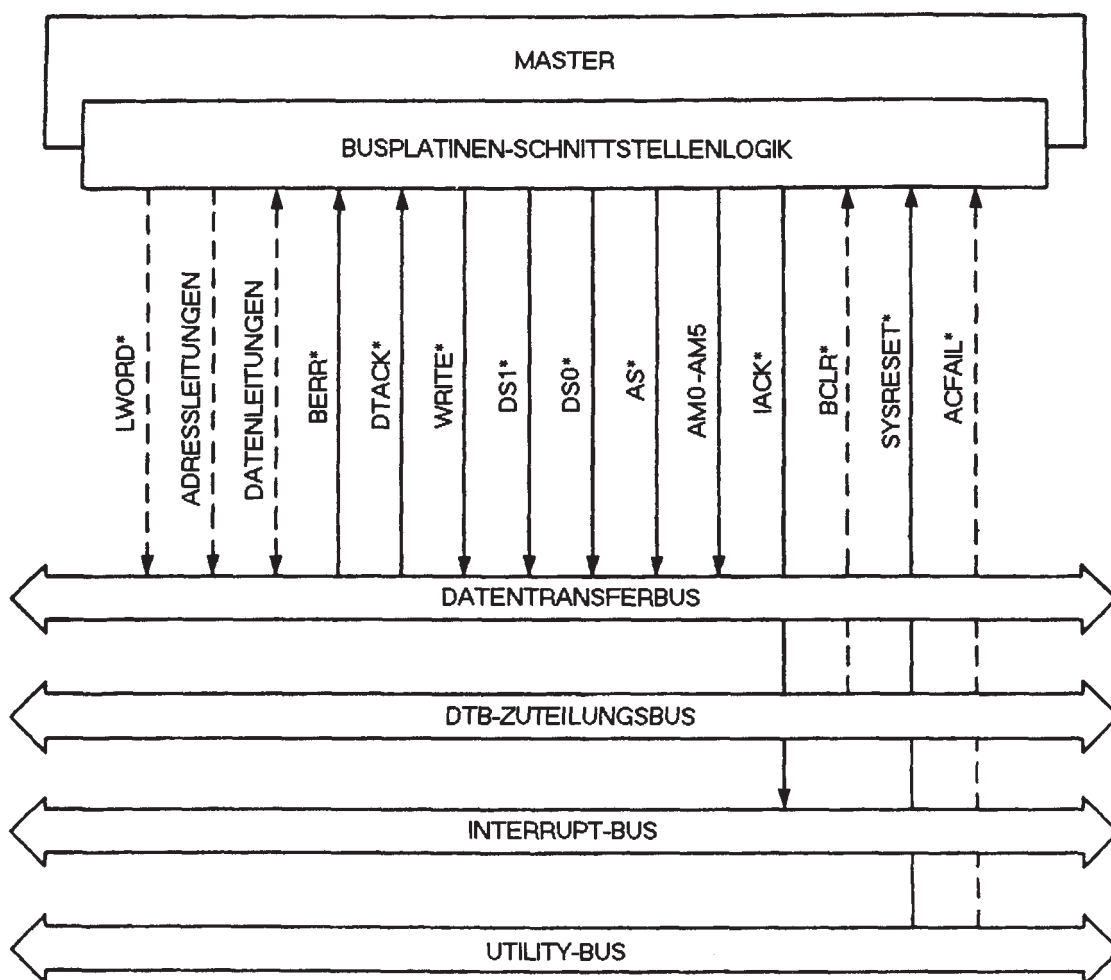


Bild 2-2: Funktionsschaltbild: MASTER

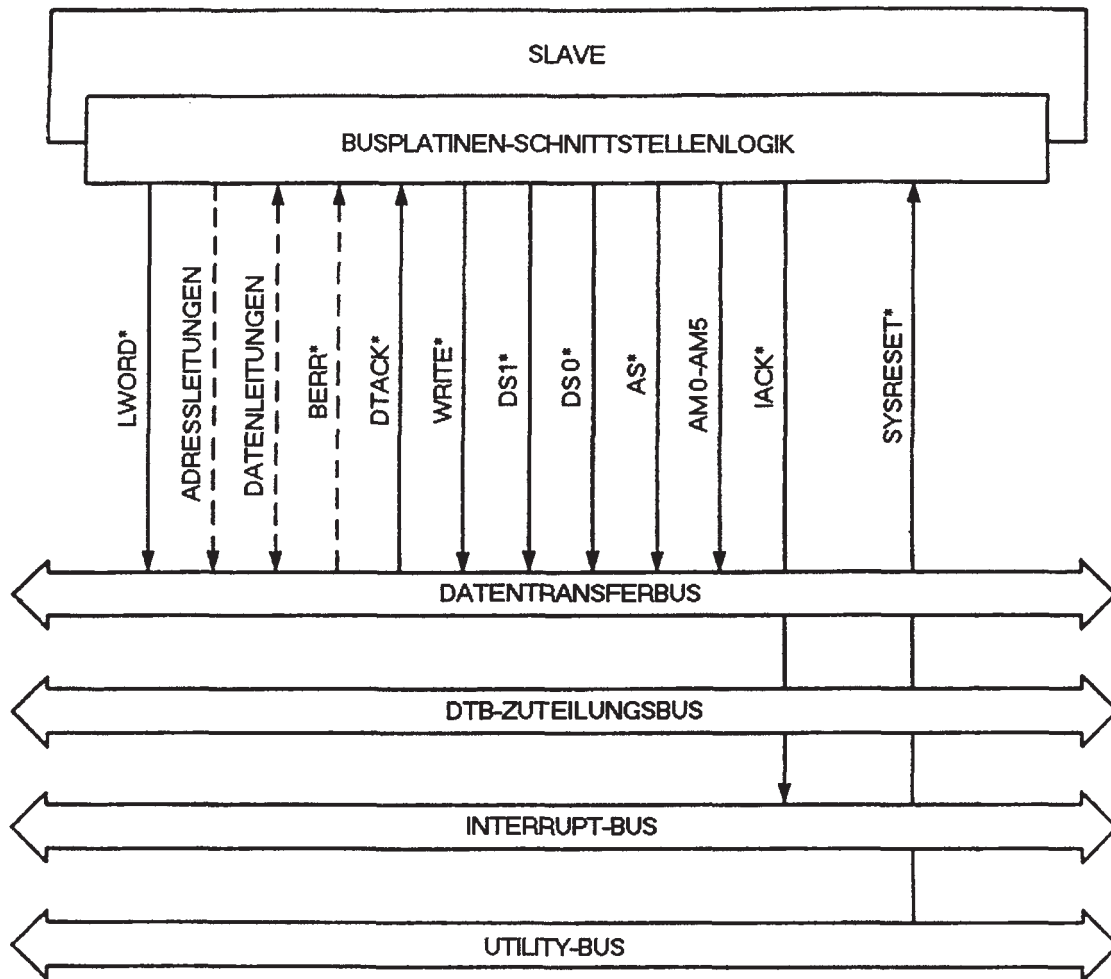


Bild 2-3: Funktionsschaltbild: SLAVE

Tabelle 2-6: SLAVES: VORSCHRIFTEN und ERLAUBNISSE für das Treiben und Überwachen der gestrichelt gezeichneten Leitungen

SLAVE-Typ	VORSCHRIFTEN und ERLAUBNISSE
D08(O)	MUSS D00-D07 überwachen und treiben. DARF oder DARF NICHT AS* überwachen. DARF oder DARF NICHT D08-D31 überwachen oder treiben.
D08(EO) und D16	MUSS D00-D15 überwachen und treiben. DARF oder DARF NICHT AS* überwachen. DARF oder DARF NICHT D16-D31 überwachen oder treiben.
D32	MUSS D00-D31 überwachen und treiben. DARF oder DARF NICHT AS* überwachen.
BLT	MUSS AS* überwachen
A16	MUSS A01-A15 überwachen DARF oder DARF NICHT A16-A31 überwachen.
A24	MUSS A01-A23 überwachen. DARF oder DARF NICHT A24-A31 überwachen.
A32	MUSS A01-A31 überwachen.
ALLE	DARF oder DARF NICHT BERR* treiben.

ANMERKUNG 1: Die Mnemonics D08(O), D08(EO), D16 und D32 werden in Tabelle 2-10 definiert.

ANMERKUNG 2: Mnemonic BLT ist in Tabelle 2-11 definiert.

ANMERKUNG 3: Die Mnemonics A16, A24 und A32 werden in Tabelle 2-9 definiert.

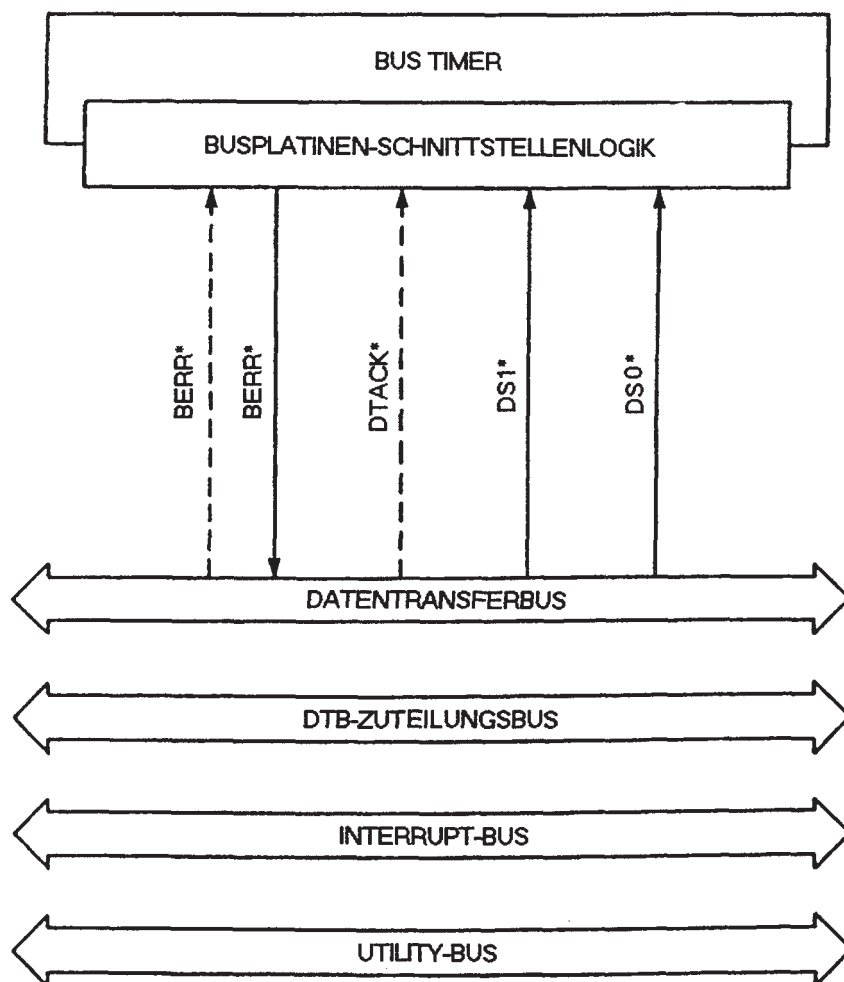


Bild 2-4: Funktionsschaltbild: BUS TIMER

Tabelle 2-7: BT0()-Mnemonic zur Spezifikation der Time-Out-Periode der BUS TIMER

Das folgende Mnemonic	verwendet für einen	bedeutet, daß
BT0(x)	BUS TIMER	BERR* low getrieben wird, wenn der erste Daten-Strobe länger als x Mikrosekunden low bleibt.

Tabelle 2-8: LOCATION MONITORE: VORSCHRIFTEN und ERLAUBNISSE für das Überwachen der gestrichelt gezeichneten Leitungen

Typ des LOCATION MONITORS	VORSCHRIFTEN und ERLAUBNISSE
A16	MUSS A01-A15 überwachen. DARF oder DARF NICHT A16-A31 überwachen.
A24	MUSS A01-A23 überwachen. DARF oder DARF NICHT A24-A31 überwachen.
A32	MUSS A01-A31 überwachen.
ALLE	DÜRFEN oder DÜRFEN NICHT AS* überwachen.

ANMERKUNG: Die Mnemonics A16, A24 und A32 werden in Tabelle 2-9 definiert.

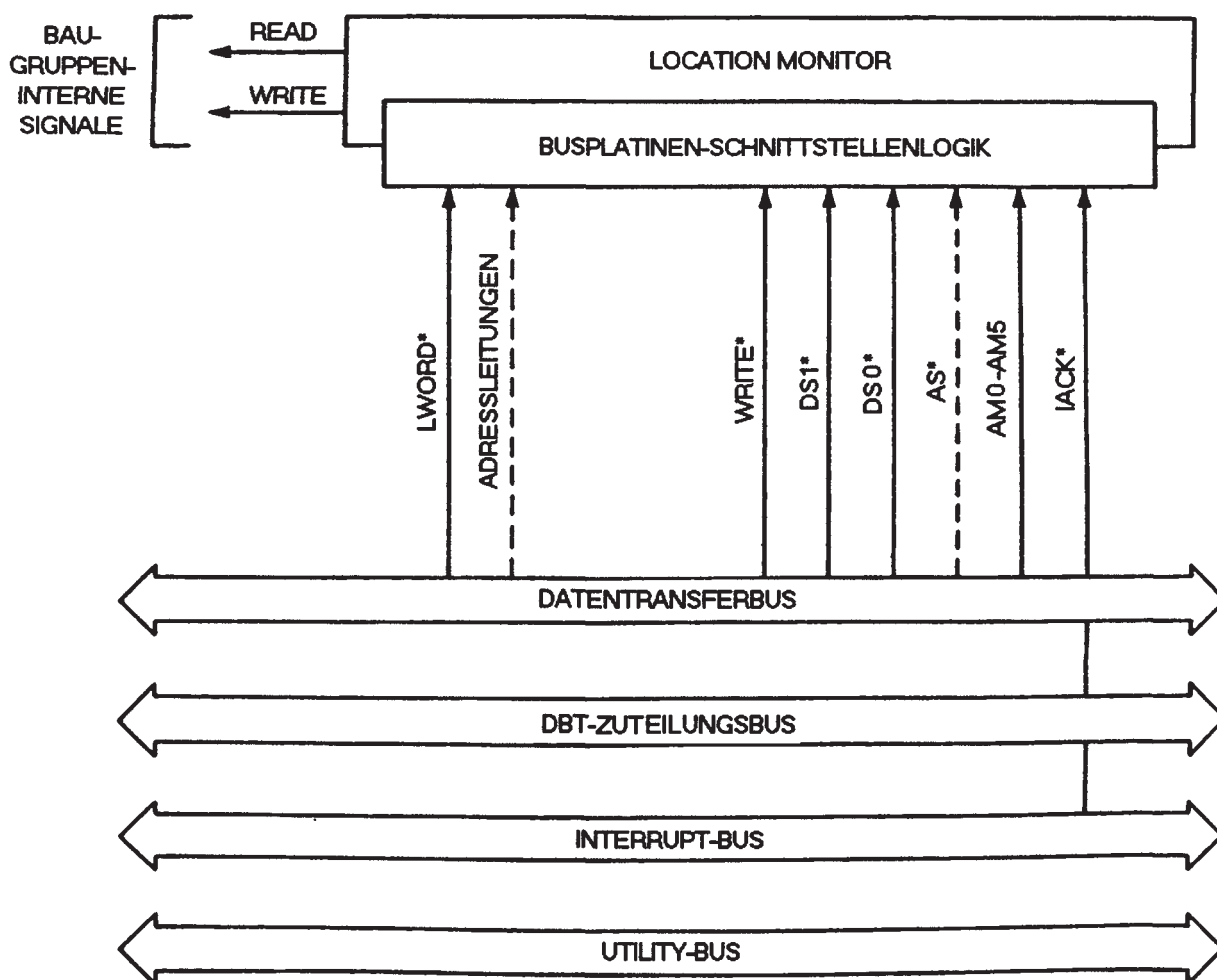


Bild 2-5: Funktionsschaltbild: LOCATION MONITOR

2.3.5 Adressierung

Am Beginn jedes Zyklus legen die MASTER eine Adresse auf den DTB. Diese Adresse könnte, abhängig von den Fähigkeiten des sendenden MASTERS, eine 16-, 24- oder 32-Bit-Adresse sein. 16-Bit-Adressen werden Kurzadressen, 24-Bit-Adressen werden Standardadressen und die 32-Bit-Adressen erweiterte Adressen genannt.

Tabelle 2-9 zeigt die verschiedenen Mnemonics, die zur Beschreibung der Adressierungsmöglichkeiten benutzt werden, und wie jedes benutzt wird, um MASTER, SLAVES und LOCATION MONITORE zu beschreiben.

Der MASTER sendet einen Adreßmodifizierer-Code (AM) mit jeder Adresse, um den SLAVES mitzuteilen, ob es sich um eine Kurz-, Standard- oder erweiterte Adresse handelt.

Kurzadressen werden durch A16-MASTER erzeugt und von A16-SLAVES empfangen. Standardadressen werden durch A24-MASTER erzeugt und durch A24-SLAVES empfangen. Erweiterte Adressen werden durch A32-MASTER erzeugt und durch A32-SLAVES empfangen.

Die Kurzadressierung ist in erster Linie zur Adressierung von Eingabe/Ausgabe-Baugruppen vorgesehen. Sie ermöglicht, daß A16-SLAVES mit weniger Logik ausgelegt werden, da sie nicht so viele Adreßleitungen decodieren müssen. Zwar können Eingabe/Ausgabe-Karten für die Decodierung von Standardadressen und erweiterten Adressen ausgelegt wer-

den, doch macht die Kurzadressierung dieses gewöhnlich überflüssig.

Standard- und erweiterte Adressierungsmodi sind in erster Linie für die Adressierung von Speichern vorgesehen, obwohl keine Vorschrift den Entwurf von Eingabe/Ausgabekarten verbietet, die auch auf diese Adressierungsmodi ansprechen. Standard- und erweiterte Adressierungsmodi ermöglichen einen wesentlich größeren Adressierungsbereich.

VORSCHRIFT 2.9:

SLAVE-Karten MÜSSEN alle Adreßmodifizierer-Leitungen decodieren.

ERLÄUTERUNG 2.14:

Die VORSCHRIFT 2.9 läßt zu, daß ein SLAVE zwischen Kurzadressen, Standardadressen und erweiterten Adressen unterscheidet.

ERLÄUTERUNG 2.15:

Zusätzlich zu den hier beschriebenen drei Adressierungsmodi besteht ein vierter Modus, der in den INTERRUPT-ACKNOWLEDGE-Zyklen verwendet wird (siehe Kapitel 4). Diese INTERRUPT-ACKNOWLEDGE-Zyklen können von den Datentransferzyklen durch die Tatsache unterschieden werden, daß die IACK*-Signalleitung low anstelle von high ist.

VORSCHRIFT 2.10:

Immer wenn ein MASTER eine Adresse über den Adreßbus broadcastet, MUSS er sicherstellen, daß IACK* high ist.

Tabelle 2-9: Mnemonics, welche die Adressierungsfähigkeiten festlegen

Das folgende Mnemonic	verwendet für einen	bedeutet, daß dieser
A16	MASTER	Zyklen mit Kurzadressen (16 Bit) erzeugen kann.
	SLAVE	Zyklen mit Kurzadressen (16 Bit) empfangen kann.
	LOCATION MONITOR	Zyklen mit Kurzadressen (16 Bit) überwachen kann.
A24	MASTER	Zyklen mit Standardadressen (24 Bit) erzeugen kann.
	SLAVE	Zyklen mit Standardadressen (24 Bit) empfangen kann.
	LOCATION MONITOR	Zyklen mit Standardadressen (24 Bit) überwachen kann.
A32	MASTER	Zyklen mit erweiterten Adressen (32 Bit) erzeugen kann.
	SLAVE	Zyklen mit erweiterten Adressen (32 Bit) empfangen kann.
	LOCATION MONITOR	Zyklen mit erweiterten Adressen (32 Bit) überwachen kann.

ERLAUBNIS 2.7:

Ein MASTER KANN entweder IACK* während der Adreß-aussendung high setzen oder IACK* ungetrieben lassen. (Die Busabschlüsse werden dann das Signal auf high halten.)

VORSCHRIFT 2.11:

SLAVES DÜRFEN NICHT auf DTB-Zyklen ansprechen, wenn IACK* low ist.

VORSCHRIFT 2.61:

A32-MASTER MÜSSEN A24- und A16-Fähigkeiten einschließen.

VORSCHRIFT 2.62:

A24-MASTER MÜSSEN die A16-Fähigkeit einschließen.

VORSCHLAG 2.6:

Man gehe nicht davon aus, daß obige VORSCHRIFTEN den Lesern von Produktbeschreibungen bekannt sind. Vielmehr spezifiziere man A32-MASTER-Produkte als "A32 und A24 und A16". Genauso spezifiziere man ein A24-MASTER-Produkt als "A24 und A16".

2.3.6 Datentransfer

In Verbindung mit dem DTB gibt es vier Daten-Transfer-Möglichkeiten: D08(EO) (geradzahliges oder ungeradzahliges Byte), D08(O) (nur ungeradzahliges Byte), D16 und D32. Diese Fähigkeiten ermöglichen Flexibilität bei der Verbindung verschiedener Prozessoren und Peripheriebaugruppen mit dem Bus.

Acht-Bit-Prozessoren können am Bus als D08(EO)-MASTER angeschlossen werden. Sechzehn-Bit-Prozessoren können als D16-MASTER angeschlossen werden. Das D16-SLAVE-Modul ist geeignet, um 16-Bit-Speicherbausteine oder 16-Bit-Eingabe/Ausgabe-SLAVES an den DTB anzuschließen.

Viele erhältliche Peripherie-ICs besitzen Register, die nur 8 Bit breit sind. Da diese ICs häufig mehrere dieser Register besitzen, können sie nicht gleichzeitig die Inhalte von zwei Registern liefern, wenn ein D16-MASTER den Zugriff auf zwei benachbarte Speicherplätze mit einem Zwei-Byte-Lesezyklus versucht. Diese 8-Bit-Peripherie-ICs können an den DTB als D08(O)-SLAVE angeschlossen werden. D08(O)-SLAVES liefern nur BYTE(1)- oder BYTE(3)-Speicherplätze und sprechen nur auf Ein-Byte-Zugriffe an. Da ungeradzahlige Ein-Byte-Zugriffe immer über D00-D07 erfolgen, vereinfacht dies die Schnittstellenlogik von D08(O)-SLAVES.

VORSCHRIFT 2.63:

D32-MASTER, D32-SLAVES und D32-LOCATION-MONITORE MÜSSEN die D08(EO)- und D16-Fähigkeiten einschließen.

VORSCHRIFT 2.64:

D16-MASTER, D16-SLAVES und D16-LOCATION-MONITORE MÜSSEN die D08(EO)-Fähigkeit einschließen.

VORSCHLAG 2.7:

Man gehe nicht davon aus, daß obige VORSCHRIFTEN den Lesern von Produktbeschreibungen bekannt sind. Vielmehr spezifiziere man Produkte mit D32-Fähigkeit als "D32 und D16 und D08(EO)". Genauso spezifiziere man D16-Produkte als "D16 und D08(EO)".

VORSCHRIFT 2.4:

Während Zyklen, die Zugriff auf die Byte-Speicherplätze BYTE(1-2), BYTE(0-2), BYTE(1-3) oder BYTE(0-3) fordern, DÜRFEN D16-SLAVES NICHT antworten, indem sie DTACK* low treiben.

VORSCHRIFT 2.5:

Während Zyklen, die Zugriff auf die Byte-Speicherplätze BYTE(0-1), BYTE(1-2), BYTE(2-3), BYTE(0-2), BYTE(1-3) oder BYTE(0-3) fordern, DÜRFEN D08(EO)-SLAVES NICHT antworten, indem sie DTACK* low treiben.

VORSCHRIFT 2.65:

Während Zyklen, die Zugriff auf die Byte-Speicherplätze BYTE(0), BYTE(2), BYTE(0-1), BYTE(1-2), BYTE(2-3), BYTE(0-2), BYTE(1-3) oder BYTE(0-3) fordern, DÜRFEN D08(EO)-SLAVES NICHT antworten, indem sie DTACK* low treiben.

VORSCHLAG 2.8:

Man entwickle SLAVES so, daß sie in folgenden Situationen mit einer fallenden Flanke auf BERR* reagieren:

- 1) Wenn ein D08(O)-, D08(EO)-, oder D16-SLAVE aufgefordert wird, einen Vier-Byte-Zyklus auszuführen.
- 2) Wenn ein D08(O)- oder D08(EO)-SLAVE aufgefordert wird, einen Zwei-Byte-Zyklus auszuführen.
- 3) Wenn ein D08(O)-, D08(EO) oder D16-SLAVE aufgefordert wird, einen nicht an Wortgrenzen gebundenen Transfer auszuführen (d. h. einen Drei-Byte-Transfer oder einen Zwei-Byte-BYTE(1-2)-Transfer).
- 4) Wenn ein D08(O)-SLAVE aufgefordert wird, einen BYTE(0)- oder BYTE(2)-Transfer auszuführen.



Documents

Document Sharing Hub

Tabelle 2-10 zeigt die verschiedenen Mnemonics, die zur Beschreibung der grundsätzlichen Datentransfer-Möglichkeiten genutzt werden, und wie jedes einzelne zur Beschreibung der MASTER, SLAVES und LOCATION MONITORE verwendet wird.

ERLÄUTERUNG 2.16:

Es scheint logisch zu sein, "Even-Byte-only"-SLAVES zu definieren, die auf geradzahlige Bytespeicherplätze von zugehörigen D08(O)-SLAVES ansprechen. Dies kann jedoch nicht getan werden, da es nur eine DTACK*-Leitung gibt. Wenn ein Master gleichzeitig einen geradzahligen und einen ungeradzahligen Byte-Speicherplatz auswählen sollte, indem er einen Zwei-Byte-Transfer ausführt, würden beide SLAVES DTACK* gleichzeitig treiben und der MASTER wüßte nicht, ob beide Karten den Zugriff bestätigten.

ERLÄUTERUNG 2.17:

Da D08(O)-SLAVES nur auf ungeradzahlige Byteadressen ansprechen, können sie keine zusammenhängenden Speicherbereiche zur Verfügung stellen. D08(O)-SLAVES sind nur geeignet für Ein-/Ausgabe, Zustands- oder Steuerregister, während D08(EO)-, D16- und D32-SLAVES auch für Speicher geeignet sind.

2.3.7 Blocktransfer

Die MASTER greifen häufig in aufsteigender Reihenfolge auf verschiedene Speicherplätze zu. Wenn dies der Fall ist, sind die Blocktransferzyklen sehr nützlich. Sie ermöglichen dem MASTER, eine einzelne Adresse zu liefern und dann auf Daten in diesem Speicherplatz und in solchen mit höheren Adressen zuzugreifen, ohne zusätzliche Adressen zu erzeugen.

Wenn ein MASTER einen Blocktransferzyklus auslöst, speichert der antwortende SLAVE die Adresse in einem internen Adreßzähler. Der MASTER verhindert, daß der Adreß-Strobe während der Vollendung des ersten Datentransfers (d. h., indem er die Daten-Strobes high treibt) auf high geht. Stattdessen treibt er aufgrund von Datentransfer-Bestätigungen durch den SLAVE wiederholt den/die Daten-Strobe(s) low, und überträgt Daten zu und von sequentiellen Speicherplätzen in aufsteigender Reihenfolge.

Um auf den oder die nächsten Speicherplätze zuzugreifen, inkrementiert der SLAVE einen auf der Karte vorhandenen Zähler, der die Adresse für jeden Daten-Strobe-Wechsel erzeugt.

Tabelle 2-10: Mnemonics für Datentransfers

Das folgende Mnemonic	verwendet für einen	bedeutet, daß dieser
D08(EO)	MASTER SLAVE LOCATION MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Ein-Byte-Lesezyklen: BYTE(0) READ BYTE(1) READ BYTE(2) READ BYTE(3) READ Ein-Byte-Schreibzyklen: BYTE(0) WRITE BYTE(1) WRITE BYTE(2) WRITE BYTE(3) WRITE
D08(O)	SLAVE	die folgenden Zyklen übernehmen MUSS: Ein-Byte-Lesezyklen: BYTE(1) READ BYTE(3) READ Ein-Byte-Schreibzyklen: BYTE(1) WRITE BYTE(3) WRITE
D16	MASTER SLAVE LOCATION MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Zwei-Byte-Lesezyklen: BYTE(0-1) READ BYTE(2-3) READ Zwei-Byte-Schreibzyklen: BYTE(0-1) WRITE BYTE(2-3) WRITE
D32	MASTER SLAVE LOCATION MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Vier-Byte-Lesezyklen: BYTE(0-3) READ Vier-Byte-Schreibzyklen: BYTE(0-3) WRITE

ANMERKUNG: (EO) bedeutet "gerade/ungerade", und (O) bedeutet "nur ungerade".

ERLÄUTERUNG 2.18:

Blocktransferzyklen von unbestimmter Länge erschweren den Entwurf von Speicherkarten. Insbesondere müßten alle Blocktransfer-SLAVES (sowohl derjenige, der anspricht, als auch diejenigen, die nicht ansprechen) die Anfangsadresse puffern und dann den Adreßzähler bei jeder Busübertragung inkrementieren. Alle SLAVES müßten dann die inkrementierten Adressen decodieren, um festzustellen, ob der Blocktransfer eine Kartengrenze in ihren Adreßbereich hinein überschritten hat. Da dies gewiß möglich ist, begrenzt eine derartige Adreßdecodierung typischerweise die Zugriffszeiten der SLAVES. Um den Entwurf der SLAVES zu vereinfachen und schnellere Zugriffszeiten zu erlauben, wurde die VORSCHRIFT 2.12 formuliert.

VORSCHRIFT 2.12:

Blocktransferzyklen DÜRFEN NICHT über eine 256-Byte-Begrenzung des Adreßraumes hinweggehen.

ERLÄUTERUNG 2.19:

Die VORSCHRIFT 2.12 begrenzt die maximale Länge der Blocktransfers auf 256 Bytes. Jedoch vereinfacht die Kenntnis, daß nur A01 bis A07 im Laufe eines Blocktransfers wechseln, den Entwurf von Blocktransfer-SLAVES. Die oberen Adreßleitungen müssen nur einmal, am Anfang des Blocktransferzyklus, dekodiert werden, was eine viel geringere Zugriffszeit bei allen folgenden Datentransfers ermöglicht.

ERLÄUTERUNG 2.20:

In einigen Fällen könnte es nötig sein, einen größeren Datenblock, der eine oder mehrere 256-Byte-Begrenzungen übersteigt, zu übermitteln. Wenn die Hardware auf den Karten, die den Blocktransfer durchführt, so ausgelegt ist, daß sie das Erreichen der 256-Byte-Begrenzung anzeigt, kann diese Hardware in einem solchen Fall kurzfristig AS* high treiben und einen anderen Blocktransfer auslösen, ohne daß die System-Software eingreift.

Der Blocklesezyklus ist einer Folge von Lesezyklen sehr ähnlich. Genauso ist der Blockschreibzyklus einer Folge von Schreibzyklen sehr ähnlich. Der Unterschied ist, daß nur die Anfangsadresse durch den MASTER gesendet und der Adreß-Strobe während des gesamten Datentransfers low gehalten wird.

ERLÄUTERUNG 2.21:

Die DTB-Steuerung kann während des Blocktransfers nicht gewechselt werden, weil AS* während aller Datentransfers low gehalten wird und die DTB-Steuerung nur gewechselt werden kann, während AS* high ist.

VORSCHRIFT 2.66:

SLAVES mit Blocktransfer-Fähigkeit MÜSSEN AS* überwachen und MÜSSEN die Adreßinformation übernehmen, wenn sie eine fallende Flanke auf AS* feststellen.

ERLÄUTERUNG 2.86:

Bei Ein-Byte-Blocktransfers werden die Daten 8bitweise über D00-D07 oder D08-D15 übertragen. Nachfolgend wird ein Beispiel gegeben:

	D08-D15	D00-D07
Erster Datentransfer	BYTE(2)	BYTE(1)
	BYTE(0)	BYTE(3)
	BYTE(2)	BYTE(1)
Letzter Datentransfer	BYTE(2)	BYTE(3)

ERLÄUTERUNG 2.87:

Bei Zwei-Byte-Blocktransfers werden die Daten 16bitweise über D00-D15 übertragen. Nachfolgend wird ein Beispiel gegeben:

	D08-D15	D00-D07
Erster Datentransfer	BYTE(2)	BYTE(3)
	BYTE(0)	BYTE(1)
	BYTE(2)	BYTE(3)
	BYTE(0)	BYTE(1)
	BYTE(2)	BYTE(3)
Letzter Datentransfer	BYTE(0)	BYTE(1)

Tabelle 2-11 legt die Mnemonics fest, die zur Beschreibung der Blocktransfer-Fähigkeiten und zur Beschreibung der MASTER, SLAVES und LOCATION MONITORE verwendet werden.

2.3.8 Read-Modify-Write

In Mehrprozessorsystemen, die Betriebsmittel wie Speicher und Eingabe/Ausgabe gemeinsam nutzen, wird eine Methode zur Zuweisung dieser Betriebsmittel benötigt. Ein sehr wichtiges Ziel dieses Zuweisungs-Algorithmus ist es, sicherzustellen, daß ein durch eine Aufgabe besetztes Betriebsmittel nicht gleichzeitig durch ein anderes benutzt werden kann. Das Problem wird am besten durch ein Beispiel beschrieben:

Zwei Prozessoren in einem Multiprocessing-System teilen sich ein gemeinsames Betriebsmittel (z. B. einen Drucker). Zu jedem Zeitpunkt kann nur ein Rechner das Betriebsmittel nutzen. Das Betriebsmittel wird durch ein Bit im Speicher zugewiesen, d. h. wenn das Bit gesetzt ist, ist das Betriebsmittel besetzt; wenn es gelöscht ist, ist das Betriebsmittel verfügbar. Um die Nutzung des Betriebsmittels zu erlangen, liest und überprüft Rechner A das Bit um festzustellen, ob es gelöscht ist. Ist das Bit gelöscht, setzt Rechner A das Bit, um Rechner B auszusperrern. Diese Operation erfordert zwei Datentransfers: eine Leseoperation, um das Bit zu überprüfen, und eine Schreiboperation, um das Bit zu setzen. Ein Problem könnte jedoch auftreten, wenn der Bus zwischen diesen zwei Transfers für Rechner B freigegeben wird. Rechner B könnte dann ebenfalls das Bit gelöscht vorfinden und annehmen, daß das Betriebsmittel verfügbar ist. Beide Rechner werden dann im nächstmöglichen Zyklus das Bit setzen und versuchen, das Betriebsmittel zu nutzen.

Dieser Konflikt wird durch die Definition eines Read-Modify-Write-Zyklus vermieden, der eine Übergabe der Steuerung des DTB zwischen dem Lese- und dem Schreibabschnitt des Zyklus verhindert. Dieser Zyklus ist einem Lesezyklus, dem unmittelbar ein Schreibzyklus folgt, sehr ähnlich. Der Unterschied besteht darin, daß der Adreß-Strobe während beider Transfers low gehalten wird. Dies stellt sicher, daß, anders als bei einem Lesezyklus und einem folgenden Schreibzyklus, die DTB-Steuerung während des Read-Modify-Write-Zyklus nicht übergeben werden kann, da dies nur möglich ist, während der Adreß-Strobe high ist.

Die Tabelle 2-12 legt das Mnemonic fest, das zur Beschreibung der Read-Modify-Write-Fähigkeit benutzt wird und wie es zur Beschreibung der MASTER, SLAVES und LOCATION MONITORE benutzt wird.

2.3.9 Nicht an Wortgrenzen gebundener Transfer

Einige 32-Bit-Mikroprozessoren speichern und laden Daten ohne Bindung an Wortgrenzen. Ein 32-Bit-Wert könnte beispielsweise auf vier verschiedene Arten gespeichert sein, wie in Bild 2-6 dargestellt.

Tabelle 2-11: Mnemonics, welche die Blocktransferfähigkeiten festlegen

Das folgende Mnemonic	verwendet für einen	bedeutet, daß dieser
BLT	D08(E0)-MASTER D08(E0)-SLAVE D08(E0)-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Blocklesezyklen: SINGLE BYTE BLOCK READ Blockschreibzyklen: SINGLE BYTE BLOCK WRITE
	D16-MASTER D16-SLAVE D16-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Blocklesezyklen: DOUBLE BYTE BLOCK READ Blockschreibzyklen: DOUBLE BYTE BLOCK WRITE
	D32-MASTER D32-SLAVE D32-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Blocklesezyklen: QUAD BYTE BLOCK READ Blockschreibzyklen: QUAD BYTE BLOCK WRITE

Tabelle 2-12: Mnemonic, das die Read-Modify-Write-Fähigkeiten festlegt

Das Mnemonic	verwendet für einen	bedeutet, daß dieser
RMW	D08(E0)-MASTER D08(E0)-SLAVE D08(E0)-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Ein-Byte Read-Modify-Write-Zyklen: BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE
	D08(O)-SLAVE	die folgenden Zyklen übernehmen MUSS: Ein-Byte Read-Modify-Write-Zyklen: BYTE(1) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE
	D16-MASTER D16-SLAVE D16-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Zwei-Byte Read-Modify-Write-Zyklen: BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE
	D32-MASTER D32-SLAVE D32-LOCATION-MONITOR	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: die folgenden Zyklen überwachen MUSS: Vier-Byte Read-Modify-Write-Zyklen: BYTE(0-3) READ-MODIFY-WRITE

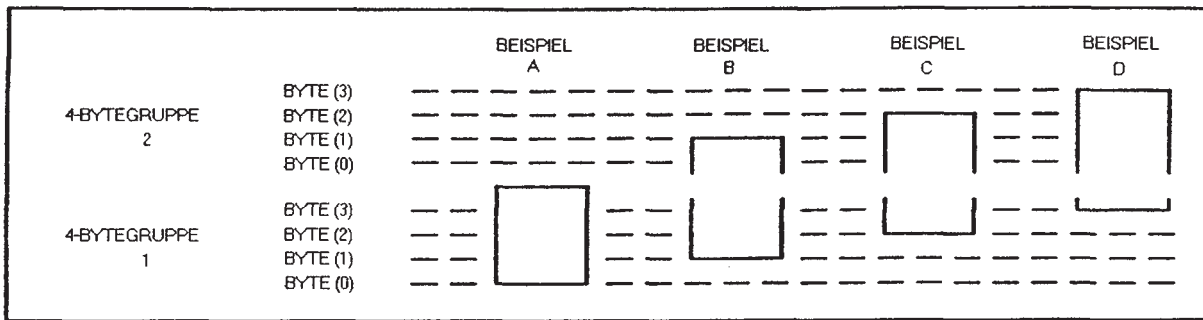


Bild 2-6: Die vier Speicherungsarten von 32 Datenbits

Tabelle 2-13: Transfer von 32-Bit-Daten unter Verwendung von Mehr-Byte-Transfer-Zyklen

Beispiel	Zyklusfolgen, die zur Erledigung des Transfers verwendet werden	Genutzte Datenbusleitungen	Zugriff auf die Bytespeicherplätze (siehe Bild 2-6)
A	Vier-Byte-Transfer	D00-D31	Gruppe 1, BYTE(0-3)
B	Ein-Byte-Transfer Zwei-Byte-Transfer Ein-Byte-Transfer oder Drei-Byte-Transfer Ein-Byte-Transfer	D00-D07 D00-D15 D08-D15 D00-D23 D08-D15	Gruppe 1, BYTE(1) Gruppe 1, BYTE(2-3) Gruppe 2, BYTE(0) Gruppe 1, BYTE(1-3) Gruppe 2, BYTE(0)
C	Zwei-Byte-Transfer Zwei-Byte-Transfer	D00-D15 D00-D15	Gruppe 1, BYTE(2-3) Gruppe 2, BYTE(0-1)
D	Ein-Byte-Transfer Zwei-Byte-Transfer Ein-Byte-Transfer oder Ein-Byte-Transfer Drei-Byte-Transfer	D00-D07 D00-D15 D08-D15 D00-D07 D08-D31	Gruppe 1, BYTE(3) Gruppe 2, BYTE(0-1) Gruppe 2, BYTE(2) Gruppe 1, BYTE(3) Gruppe 2, BYTE(0-2)

Der MASTER kann die 32 Datenbits durch Verwendung mehrerer verschiedener Folgen von DTB-Zyklen übertragen. Zum Beispiel kann er die Daten byteweise durch Verwendung von vier Ein-Byte-Datentransfers übertragen. Ein MASTER kann den Transfer jedoch viel schneller durch die Verwendung einer der in Tabelle 2-13 aufgeführten Zyklusfolgen erledigen.

ANMERKUNG: Die Verwendung der UAT-Option des DTB-Protokolls ist nur in einer D32-Umgebung möglich, unabhängig von nicht an Wortgrenzen gebundenen 16-Bit- oder 24-Bit-Transfers.

ERLÄUTERUNG 2.22:

Die in Tabelle 2-13 aufgeführten Folgen wären typisch für einen MASTER, der in aufsteigender Reihenfolge auf die Bytespeicherplätze zugreift. Das IEC-821-BUS-Protokoll erfordert dieses nicht.

Wie in Tabelle 2-13 gezeigt, kann jeder dieser 32-Bit-Transfers mit einer Kombination von Ein-Byte- und Zwei-Byte-Transfers erledigt werden. Die Beispiele B und D benötigen jedoch drei Buszyklen, wenn sie derart ausgeführt werden. Aus diesem Grund enthält das DTB-Protokoll auch zwei Drei-Byte-Transfer-Zyklen. Wenn diese in Verbindung mit einem Ein-Byte-Zyklus angewendet werden, ermöglichen diese Drei-Byte-Zyklen, die Daten mit nur zwei Buszyklen zu speichern, wie in den Beispielen B und D gezeigt.

Einige 32-Bit-Mikroprozessoren speichern und laden auch 16 Datenbits auf einmal ohne Bindung an Wortgrenzen, wie in Bild 2-7 aufgezeigt wird.

Der MASTER kann die 16 Datenbits durch Verwendung mehrerer, verschiedener Folgen von DTB-Zyklen übertragen, wie in Tabelle 2-14 dargestellt.

ERLÄUTERUNG 2.23:

Die in Tabelle 2-14 aufgeführten Folgen wären typisch für einen MASTER, der in aufsteigender Reihenfolge auf die Bytespeicherplätze zugreift. Das IEC-821-BUS-Protokoll erfordert dieses nicht.

Wie in Bild 2-14 gezeigt wird, kann der 16-Bit-Transfer aus Beispiel F mit zwei Ein-Byte-Transfers erreicht werden. Dieses erfordert jedoch zwei Buszyklen. Aus diesem Grund enthält das DTB-Protokoll auch einen Zwei-Byte-Transfer-Zyklus, der das Speichern von Daten in nur einem Buszyklus ermöglicht, wie im Beispiel F dargestellt.

ERLÄUTERUNG 2.24:

Da die nicht an Wortgrenzen gebundenen Transfers alle 32 Datenleitungen nutzen, können nur D32-MASTER und D32-SLAVES nicht an Wortgrenzen gebundene Transfers durchführen.

VORSCHRIFT 2.67:

D32-SLAVES und LOCATION MONITORE MÜSSEN die UAT-Fähigkeit einschließen.

VORSCHRIFT 2.6:

Während eines Zyklus, der auf die Byte-Speicherplätze BYTE(1-2), BYTE(0-2) oder BYTE(1-3) zugreift, DÜRFEN D08(O)-, D08(EO)- und D16-SLAVES NICHT antworten, indem sie DTACK* low treiben.

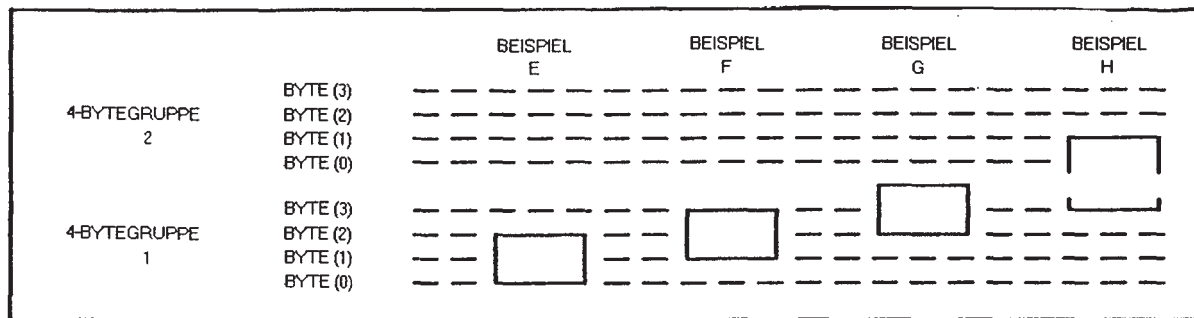


Bild 2-7: Die vier Speicherungsarten von 16 Datenbits

Tabelle 2-14: Transfer von 16-Bit-Daten unter Verwendung von Mehr-Byte-Transfer-Zyklen

Beispiel	Zyklusfolgen, die zur Abwicklung des Transfers verwendet werden	Genutzte Datenbusleitungen	Zugriff auf die Bytespeicherplätze (siehe Bild 2-7)
E	Zwei-Byte-Transfer	D00-D15	Gruppe 1, BYTE(0-1)
F	Ein-Byte-Transfer Ein-Byte-Transfer oder Zwei-Byte-Transfer	D00-D07 D08-D15 D08-D23	Gruppe 1, BYTE(1) Gruppe 1, BYTE(2) Gruppe 1, BYTE(1-2)
G	Zwei-Byte-Transfer	D00-D15	Gruppe 1, BYTE(2-3)
H	Ein-Byte-Transfer Ein-Byte-Transfer	D00-D07 D08-D15	Gruppe 1, BYTE(3) Gruppe 2, BYTE(0)

Tabelle 2-15: Mnemonic, das die Fähigkeit für nicht an Wortgrenzen gebundene Transfers festlegt

Das Mnemonic	verwendet für einen	bedeutet, daß dieser
UAT	D32-MASTER D32-SLAVE	die folgenden Zyklen erzeugen kann: die folgenden Zyklen übernehmen MUSS: Vier-Byte-Lese-Zyklen: BYTE(0-3) READ Vier-Byte-Schreib-Zyklen: BYTE(0-3) WRITE Drei-Byte-Lese-Zyklen: BYTE(0-2) READ BYTE(1-3) READ Drei-Byte-Schreib-Zyklen: BYTE(0-2) WRITE BYTE(1-3) WRITE Zwei-Byte-Lese-Zyklen: BYTE(1-2) READ Zwei-Byte-Schreib-Zyklen: BYTE(1-2) WRITE

Tabelle 2-15 führt auf, wie das Mnemonic für nicht an Wortgrenzen gebundenen Transfer (UAT) zur Beschreibung der MASTER verwendet wird.

2.3.10 ADDRESS-ONLY-Fähigkeit

Der ADDRESS-ONLY-Zyklus ist der einzige Zyklus auf dem DTB, der nicht zum Datentransfer benutzt wird. Er beginnt wie ein typischer DTB-Zyklus mit der Belegung der Leitungen für Adresse, Adreßmodifizierer, IACK* und LWORD* und, nach einer Set-up-Zeit, der fallenden Flanke von AS*. Die Daten-Strobes werden jedoch nie low getrieben. Nachdem die verschiedenen durch AS* validierten Leitungen für eine vorgeschriebene Mindest-Zeitspanne stabil gehalten wurden, beendet der MASTER den Zyklus, ohne darauf zu warten, daß DTACK* oder BERR* low gehen. (Der ADDRESS-ONLY-Zyklus ist im übrigen die einzige DTB-Zyklusart, die zum Abschluß keine Quittung erfordert.)

Tabelle 2-16: Mnemonic, das die ADDRESS-ONLY-Fähigkeit festlegt

Das Mnemonic	verwendet für einen	bedeutet, daß dieser
ADO	MASTER	ADDRESS-ONLY-Zyklen erzeugen kann

Die Tabelle 2-16 führt auf, wie das Mnemonic für ADDRESS-ONLY (ADO) zur Beschreibung der MASTER benutzt wird.

ERLÄUTERUNG 2.25:

ADDRESS-ONLY-Zyklen können zur Verbesserung der Leistungsfähigkeit von IEC-821-BUS-Baugruppen benutzt werden, indem sie einer CPU-Karte erlauben, eine Adresse zu senden, bevor diese ermittelt hat, ob die Adresse einen

SLAVE auf dem IEC-821-BUS auswählt oder nicht. Das Senden der Adresse ermöglicht es IEC-821-BUS-SLAVES, die Adresse gleichzeitig mit der CPU-Karte zu decodieren.

VORSCHRIFT 2.68:

Alle SLAVES MÜSSEN so konstruiert werden, daß ADO-Zyklen ohne Datenverlust oder fehlerhafte Operation ermöglicht werden.

2.3.11 Wechselwirkung zwischen den DTB-Funktionsmodulen

Datentransfers finden zwischen MASTER und SLAVES statt. Der MASTER ist das den Transfer steuernde Modul. Der SLAVE, der die Adresse als seine eigene erkennt, ist der reagierende SLAVE. Alle übrigen SLAVES sind nichtreagierende SLAVES.

Nach Einleitung eines Datenübertragungszyklus wartet der MASTER auf eine Antwort vom reagierenden SLAVE. Wenn der MASTER die Antwort des SLAVES feststellt, steuert er seine Daten-Strobes und den Adreß-Strobe high und beendet den Zyklus. Der SLAVE antwortet durch Freigabe seiner Antwortleitung.

ERLÄUTERUNG 2.26:

Obwohl der zeitliche Ablauf für Adresse und Daten weitgehend voneinander unabhängig ist, gibt es zwei Ausnahmen. Erstens wartet der MASTER bis er AS* low getrieben hat, bevor er entweder DS0* oder DS1* low treibt. Zweitens bestätigt der SLAVE sowohl AS1* und DS0* und DS1*, entweder mit DTACK* oder BERR*.

VORSCHRIFT 2.13:

WENN ein SLAVE auf einen Datenübertragungszyklus reagiert,

DANN MUSS er entweder DTACK* oder BERR* low treiben, jedoch DARF er NICHT beide low treiben.

ERLÄUTERUNG 2.27

Aufgrund möglicher Laufzeitunterschiede auf dem Bus, die der unterschiedlichen Belastung von AS* und DS0* und DS1* zuzuschreiben sind, könnten die fallenden Flanken von DS0* und DS1* vom SLAVE kurz vor der fallenden Flanke von AS* festgestellt werden.

ERLÄUTERUNG 2.28:

Die WRITE*-Leitung ist high/low, um einen Lese-/Schreibzyklus zu kennzeichnen, bevor der erste Daten-Strobe low getrieben wird, und bleibt stabil, bis beide Daten-Strobes high sind.

VORSCHRIFT 2.14:

Bevor er den Datenbus treibt, MUSS der MASTER sicherstellen, daß der vorherige reagierende SLAVE die Ansteuerung des Datenbusses beendet hat. Er tut dies, indem er überprüft, daß DTACK* und BERR* beide high sind, bevor er den (die) Daten-Strobe(s) in irgendeinem Zyklus low treibt und bevor er irgendwelche Datenleitungen in einem Schreibzyklus treibt.

VORSCHRIFT 2.15:

Am Ende eines Lesezyklus MUSS der reagierende SLAVE den Datenbus freigeben, bevor er DTACK* high gehen läßt.

VORSCHRIFT 2.16:

Während Lesezyklen MUSS der SLAVE gültige Daten auf dem Datenbus beibehalten, bis der MASTER den ersten Daten-Strobe wieder high treibt.

ERLÄUTERUNG 2.29:

Die Adressierungsinformation auf dem IEC-821-BUS könnte wechseln, sobald ein Modul DTACK* oder BERR* low und bevor der MASTER DS0* und DS1* high treibt.

VORSCHLAG 2.3:

Um optimale Leistung zu erreichen, konstruiere man MASTER so, daß sie DS0* und DS1* möglichst schnell high treiben, nachdem DTACK* oder BERR* low gehen. Man konstruiere auch SLAVES so, daß sie den Datenbus und DTACK* möglichst bald freigeben, nachdem sie festgestellt haben, daß DS0* und DS1* high sind. Dadurch wird eine maximale Datentransferrate auf dem Bus ermöglicht.

Eine dritte Modulart, LOCATION MONITOR genannt, überwacht den Datentransfer und erzeugt ein oder zwei karteneigene Signale, wenn auf einen der von ihm überwachten Bytespeicherplätze zugegriffen wird. Wenn der Zugriff ein Schreibzyklus ist, wird das karteneigene WRITE-Signal erzeugt. Wenn der Zugriff ein Lesezyklus ist, wird das karteneigene READ-Signal erzeugt. Wenn ein Read-Modify-Write-Zyklus durchgeführt wird, werden beide karteneigenen Signale erzeugt.

Wenn der Zyklus zu lange dauert, schaltet sich ein viertes Modul, BUS TIMER genannt, ein, indem es BERR* low treibt und damit das Datentransfer-Protokoll beendet und dem Bus die Wiederaufnahme der Tätigkeit ermöglicht.

VORSCHRIFT 2.17:

Es besteht eine feste Verriegelung zwischen den steigenden und fallenden Flanken der Daten-Strobes und DTACK*/BERR*. Sobald ein MASTER entweder DS0* oder DS1* low getrieben hat, DARF er sie NICHT high treiben und einen Transfer beenden, bevor er entweder DTACK* oder BERR* low festgestellt hat.

ERLÄUTERUNG 2.30:

Enthält eine Platine einen Prozessor, der Datentransfers zwischen sich und anderen IEC-821-BUS-Platinen ausführen muß, würde sie ein MASTER-Modul enthalten. Wenn die gleiche Platine außerdem einen vom IEC-821-BUS zugänglichen Speicher hat, würde sie auch ein SLAVE-Modul enthalten. Ein Gleitkommaprozessor oder eine intelligente Peripheriegerätesteuerung könnte Befehle über eine SLAVE-Schnittstelle von einer Universalprozessor-Platine empfangen. Danach könnte er als MASTER arbeiten, um auf den allgemeinen IEC-821-BUS-Speicher zuzugreifen, um den empfangenen Befehl auszuführen.

2.4 Typischer Betrieb

Die MASTER leiten Datenübertragungen über den DTB ein. Der adressierte SLAVE bestätigt dann die Übertragung. Nach dem Empfang der Datentransfer-Bestätigung beendet der MASTER den Datentransferzyklus. Die asynchrone Natur des DTB ermöglicht es dem SLAVE, die für die Übertragung verwendete Zeit zu steuern.

Vor jeglicher Durchführung einer Datenübertragung muß einem MASTER die alleinige DTB-Steuerung zugeteilt sein. Dies stellt sicher, daß nicht mehrere MASTER gleichzeitig versuchen, den DTB zu nutzen. Der MASTER erlangt das DTB-Steuerungsrecht unter Verwendung der Module und Signalleitungen des Zuteilungsbusses. (Kapitel 2.5 erklärt dieses näher.) Die folgende Erörterung setzt voraus, daß der MASTER die DTB-Steuerung bereits zugeteilt bekommen und übernommen hat.

2.4.1 Typische Datentransferzyklen

Bild 2-8 zeigt einen typischen Ein-Byte-Lese-Zyklus. Um den Transfer zu starten, treibt der MASTER die Adreßleitungen mit der gewünschten Adresse und dem Adreßmodifiziercode. Da dieses Beispiel ein BYTE(1)-READ-CYCLE ist, treibt der MASTER LWORD* high und A01 low. Da er keinen Interrupt-Acknowledge-Zyklus erzeugt, treibt er IACK* nicht low. Der MASTER wartet dann eine festgelegte Set-up-Zeit ab, bevor er AS* low treibt, um den Adreßleitungen und den

Adreßmodifizierer-Leitungen zu ermöglichen, sich zu stabilisieren, bevor die SLAVES sie abfragen.

Jeder SLAVE stellt durch Überprüfung der Leitungspegel auf der/den Adreß-, den Adreßmodifizierer- und der IACK*-Leitung(en) fest, ob er antworten sollte. Während dieses geschieht, treibt der MASTER WRITE* high, um eine Lese-Operation anzuzeigen. Dann überprüft der MASTER, ob DTACK* und BERR* high sind, um sicherzustellen, daß der SLAVE des vorausgegangenen Zyklus nicht länger den Datenbus treibt. Wenn dieses der Fall ist, treibt der MASTER DS0* low und hält DS1* high.

Der reagierende SLAVE ermittelt dann, auf welche 4-Byte-Gruppe und auf welches Byte dieser Gruppe der Zugriff erfolgen soll und beginnt den Transfer. Nachdem er die Daten aus dem internen Speicher gelesen und auf die Datenbusleitungen D00-D07 gelegt hat, signalisiert der SLAVE dies dem MASTER, indem er DTACK* low treibt. Der SLAVE hält dann DTACK* low und hält die Daten so lange gültig, wie der MASTER DS0* low hält.

Wenn der MASTER DTACK* low empfängt, übernimmt er die Daten von D00-D07, gibt die Adreßleitungen frei und treibt DS0* und AS* high. Der SLAVE antwortet durch Freigabe von D00-D07 und dadurch, daß er DTACK* high setzt.

ERLÄUTERUNG 2.31:

Der MASTER in Bild 2-8 gibt am Ende des Datentransfers alle DTB-Leitungen frei. Dies ist nicht erforderlich, es sei denn, der REQUESTER des MASTERS hat BBSY* während des Datentransfers freigegeben, wie in Kapitel 2.5 beschrieben.

Die Zyklusabläufe für Zwei-Byte- und Vier-Byte-Datentransferzyklen sind dem Ein-Byte-Zyklus sehr ähnlich. Flußdiagramme für diese Zyklen werden in den Bildern 2-9 und 2-10 gezeigt.

2.4.2 Adreß-Pipelining

Der IEC-821-BUS taktet die Adresse und die Daten mit getrennten Strobe-Signalen. Dies erlaubt einem MASTER, die Adresse für den nächsten Zyklus auszusenden, während die Datenübertragung des vorhergehenden Zyklus noch ausgeführt wird. Dies wird "Adreß-Pipelining" genannt.

ERLAUBNIS 2.8:

Sobald der MASTER feststellt, daß der SLAVE DTACK* oder BERR* low getrieben hat, DARF der MASTER die Adresse ändern und, nachdem er AS* für einen Mindestzeitraum high getrieben hatte, AS* wieder low treiben.

Wenn beispielsweise ein SLAVE DTACK* oder BERR* in einem Lesezyklus low treibt, darf der MASTER eine neue Adresse auf dem Adreß-Bus anlegen, während er die Daten vom Bus liest. Dies führt zu einem Überlappen des einen Zyklus mit dem folgenden und erlaubt höhere Geschwindigkeiten auf dem IEC-821-BUS.

VORSCHRIFT 2.18:

Alle SLAVES MÜSSEN so konstruiert sein, daß Adreß-Pipelining ohne Datenverlust oder fehlerhafte Operation ermöglicht wird.

Es folgen zwei Entwicklungsvorschläge, welche die Anforderung erfüllen, das Adreß-Pipelining sicherzustellen:

ERLÄUTERUNG 2.32:

Der reagierende SLAVE könnte seine Adresse feststellen und sehr schnell auf den DTACK*- oder BERR*-Leitungen antworten. Da dem MASTER erlaubt ist, die Adresse zu entfernen, nachdem der reagierende SLAVE DTACK* oder BERR* low getrieben hat, könnten die nichtreagierenden SLAVES nicht imstande sein, die Adressierungsinformation zu decodieren, bevor der MASTER sie vom Bus entfernt hat.

ERLÄUTERUNG 2.33:

Weil ein MASTER eine neue Adresse aussenden könnte, während ein vorhergehender Zyklus erst noch beendet wird, muß der Entwickler sicherstellen, daß das Anlegen des zweiten Adreß-Strobes nicht die erste Adresse ungültig macht, solange sie von der Platinenlogik zur Aufrechterhaltung der Daten auf dem Bus benötigt wird.

VORSCHLAG 2.4:

Man konstruiere die SLAVES so, daß sie die Adressierungsinformation mit der fallenden Flanke von AS* übernehmen.

ERLÄUTERUNG 2.34:

Ein MASTER könnte AS* für den neuen Zyklus low treiben, bevor er entweder DS0* oder DS1* des vorhergehenden Zyklus high treibt. Deshalb könnte während der Zyklusüberlappung ein Zeitraum entstehen, in dem AS* für den neuen Zyklus mit mindestens einem der Signale DS0* oder DS1* des vorhergehenden Zyklus zusammentrifft.

VORSCHLAG 2.5:

Man konstruiere SLAVES so, daß sie Datenübertragungen zum und vom Bus einleiten, wenn einer oder beide von DS0* und DS1* low und DTACK* und BERR* beide high sind, anstatt bei gleichzeitigem low-Pegel von AS* und einem oder beiden von DS0* und DS1*.

ERLAUBNIS 2.9:

MASTER DÜRFEN ohne Fähigkeit zum Adreß-Pipelining konstruiert werden (z. B. DÜRFEN sie warten, bis der reagierende SLAVE DTACK* und BERR* freigibt, bevor sie AS* für den nächsten Zyklus low treiben).

2.5 Datentransferbus-Übernahme

VORSCHRIFT 2.19:

Bevor er irgendwelche Daten auf dem DTB übermittelt, MUSS ein MASTER die Nutzungsgenehmigung erhalten.

Mehrere MASTER könnten den DTB gleichzeitig nutzen wollen. Die Entscheidungsprozedur, welcher MASTER den DTB nutzen darf, wird Zuteilung genannt und in Kapitel 3 erörtert. Da die Zuteilung eng mit dem Betrieb des DTB verknüpft ist, wird sie hier kurz beschrieben.

Bild 2-11 zeigt zwei Beispiele, die die möglichen Abläufe für den Fall aufzeigen, daß ein MASTER, MASTER A genannt, die DTB-Nutzung beendet und den Ablauf der Zuteilung ermöglicht.

Im Beispiel 1 zeigt MASTER A innerhalb des letzten Transfers an, daß er den DTB nicht länger benötigt. Dieses tut er, indem er seinen REQUESTER veranlaßt, BBSY* freizugeben. Da MASTER A diese frühzeitige Meldung, daß der DTB bald verfügbar ist, abgibt, wird die Zuteilung während des letzten Datentransfers durchgeführt. Die Zuteilung wird vollendet, und MASTER B erhält die Genehmigung zur DTB-Nutzung, bevor MASTER A seinen Zyklus beendet hat. Er wartet jedoch, bis MASTER A AS* freigibt. Dies stellt sicher, daß MASTER B den DTB nicht zu treiben beginnt, bevor MASTER A seinen letzten Datentransfer beendet hat.

Im Beispiel 2 wartet MASTER A bis nach dem letzten Transfer; d. h. bis AS* freigegeben ist, bevor er BBSY* freigibt. In diesem Fall ist der DTB während der Zuteilung unbenutzt. Dem MASTER B wird dann der Bus zugeteilt, und da AS* schon high ist, beginnt er sofort mit der DTB-Nutzung.

VORSCHRIFT 2.20:

Hat der REQUESTER eines MASTERS BBSY* bereits auf high freigegeben, so DARF der MASTER AS* NICHT mehr von high auf low treiben; d. h. er DARF NICHT einen neuen Zyklus beginnen, bevor sein REQUESTER eine neue Buszuteilung erhält.

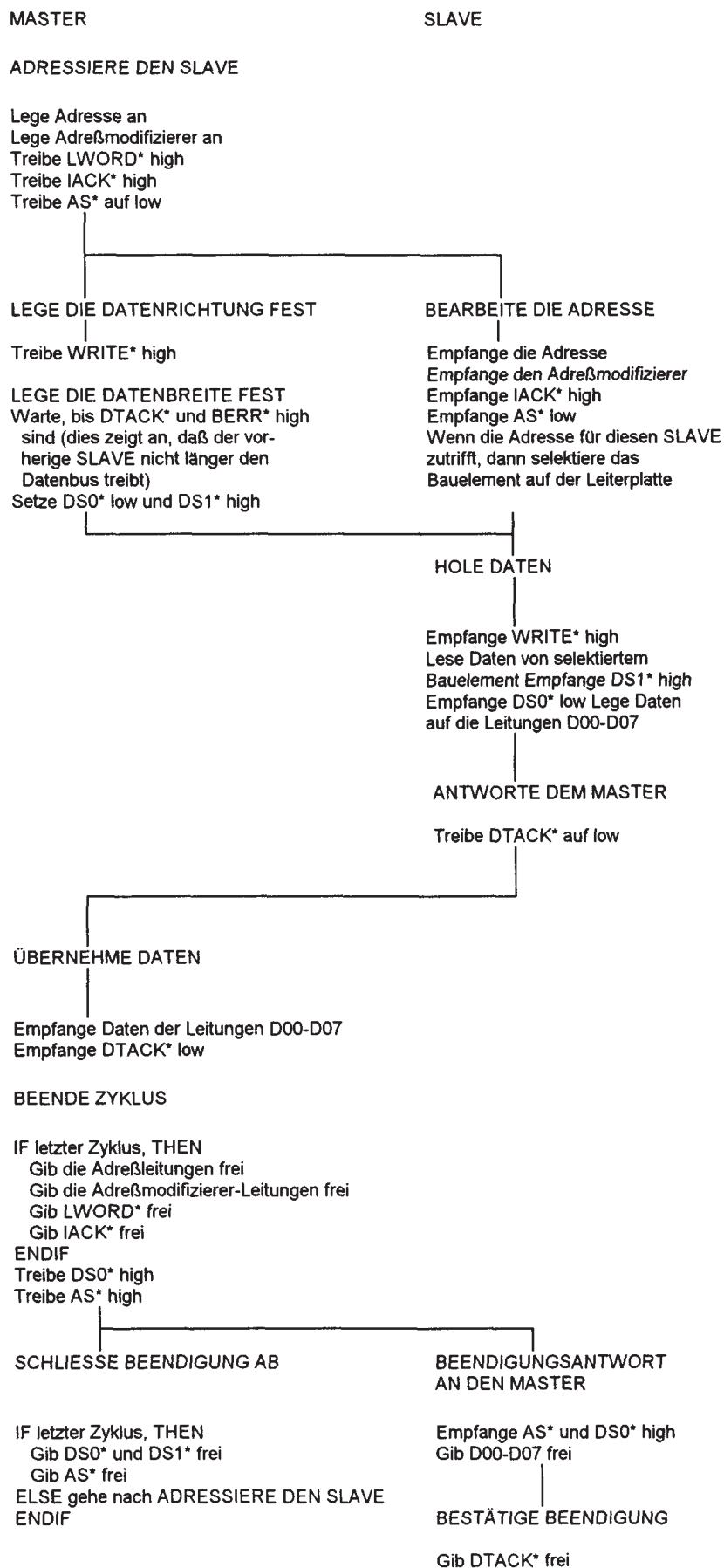


Bild 2-8: Beispiel für einen Ein-Byte-Lesezyklus

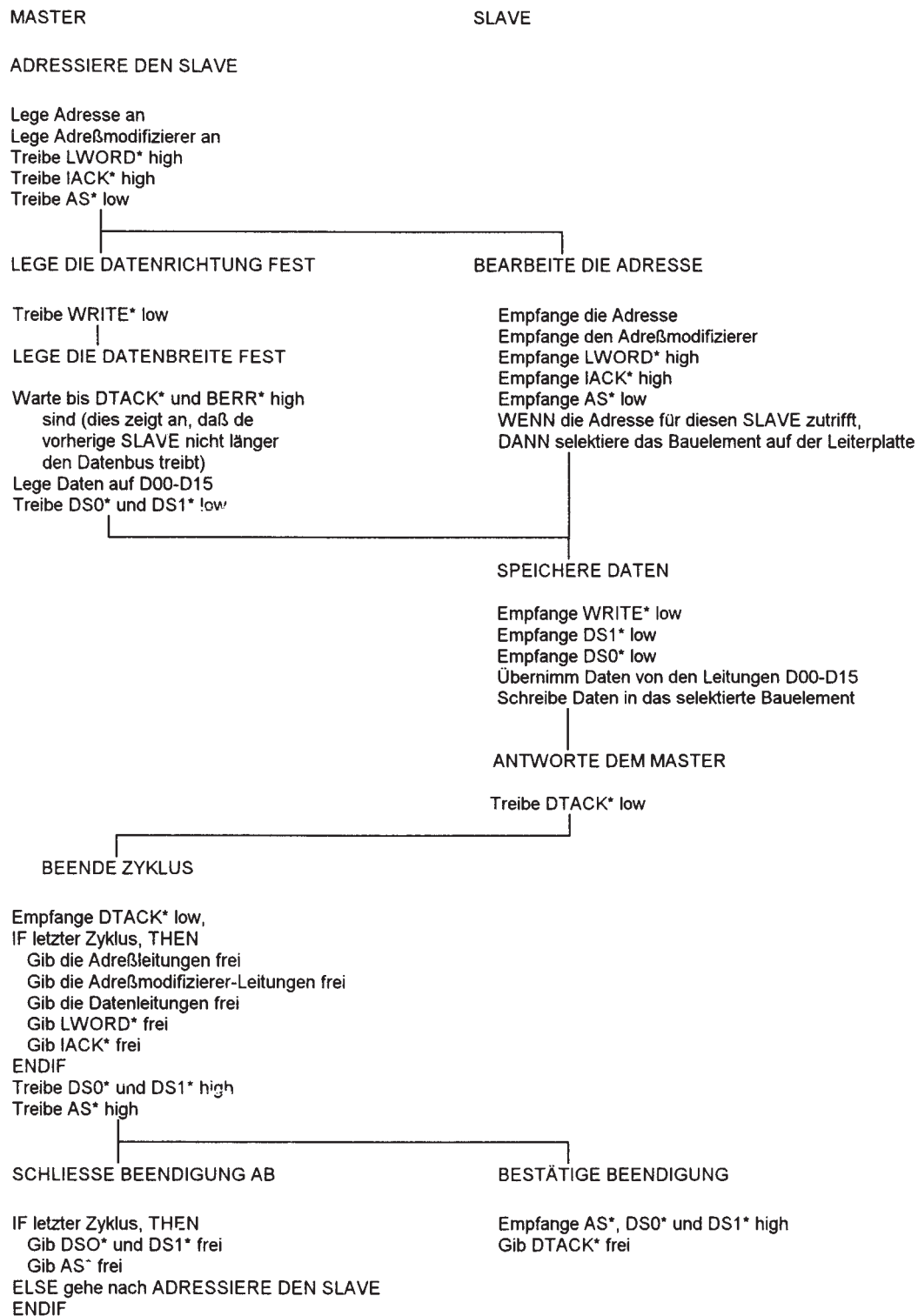


Bild 2-9: Beispiel für einen Zwei-Byte-Schreibzyklus

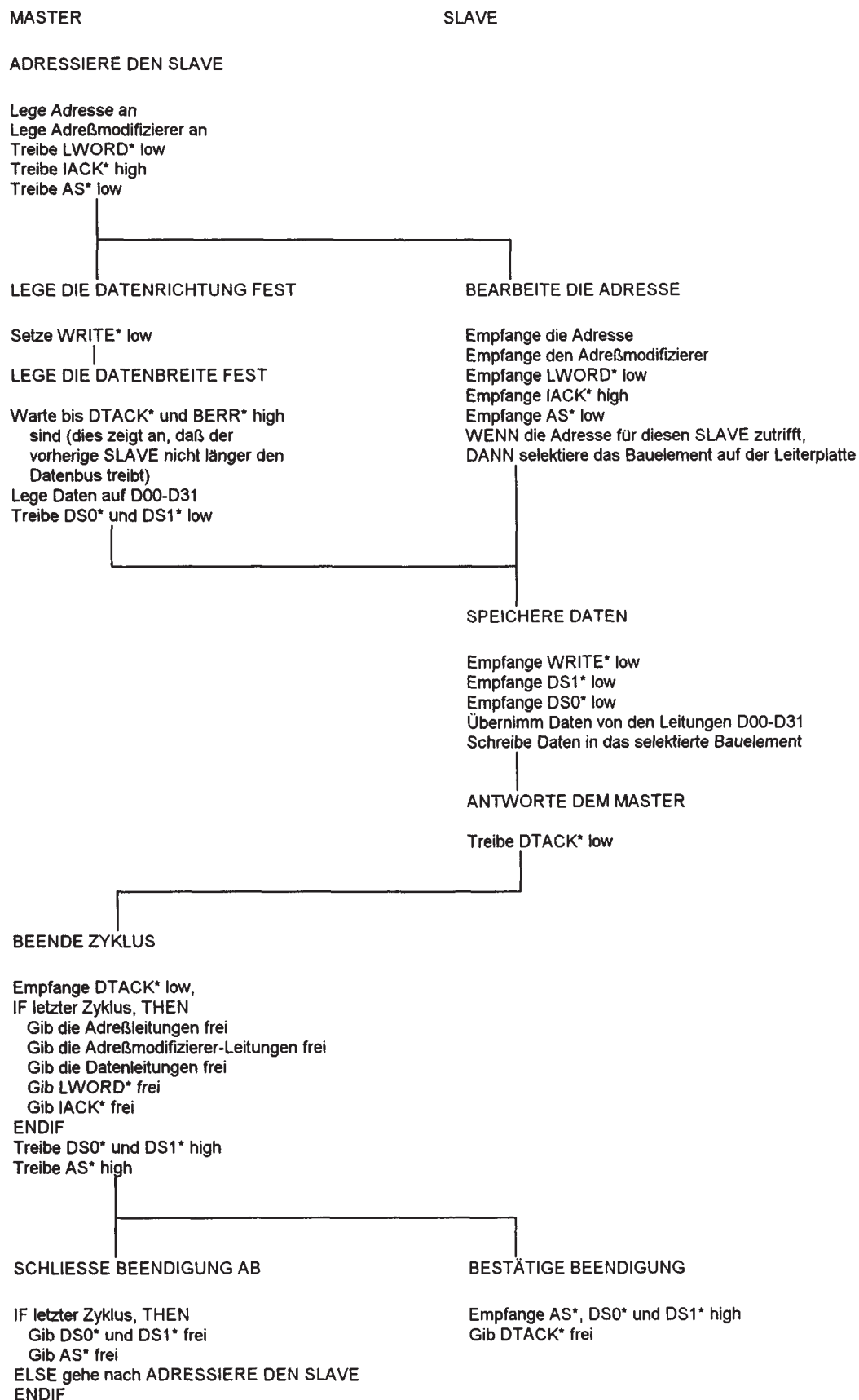
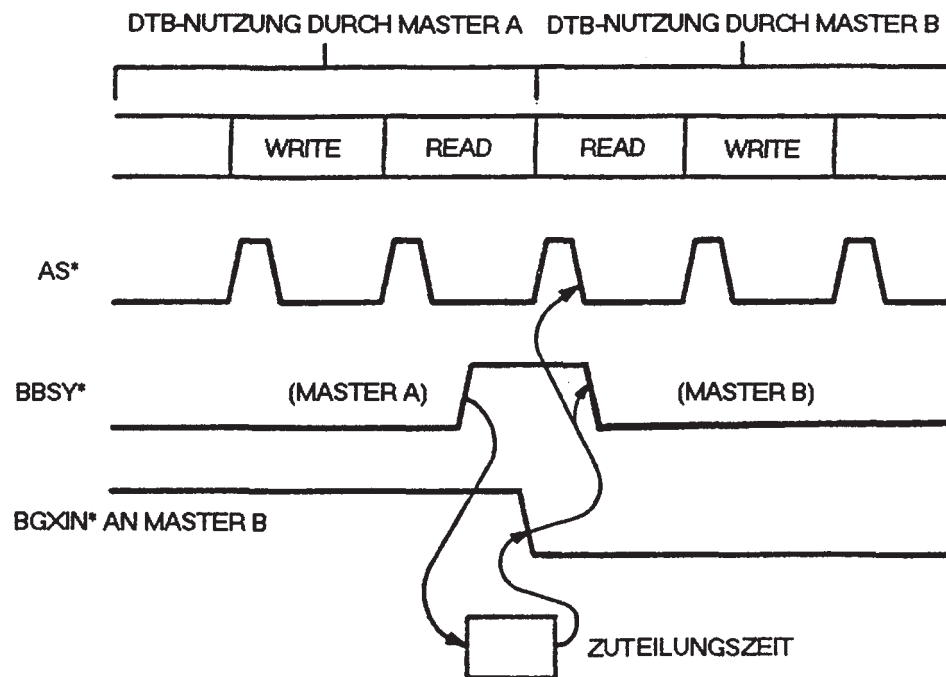


Bild 2-10: Beispiel für einen Vier-Byte-Schreibzyklus

BEISPIEL 1:
Zuteilung WÄHREND des letzten Datentransfers



BEISPIEL 2:
Zuteilung NACH dem letzten Datentransfer

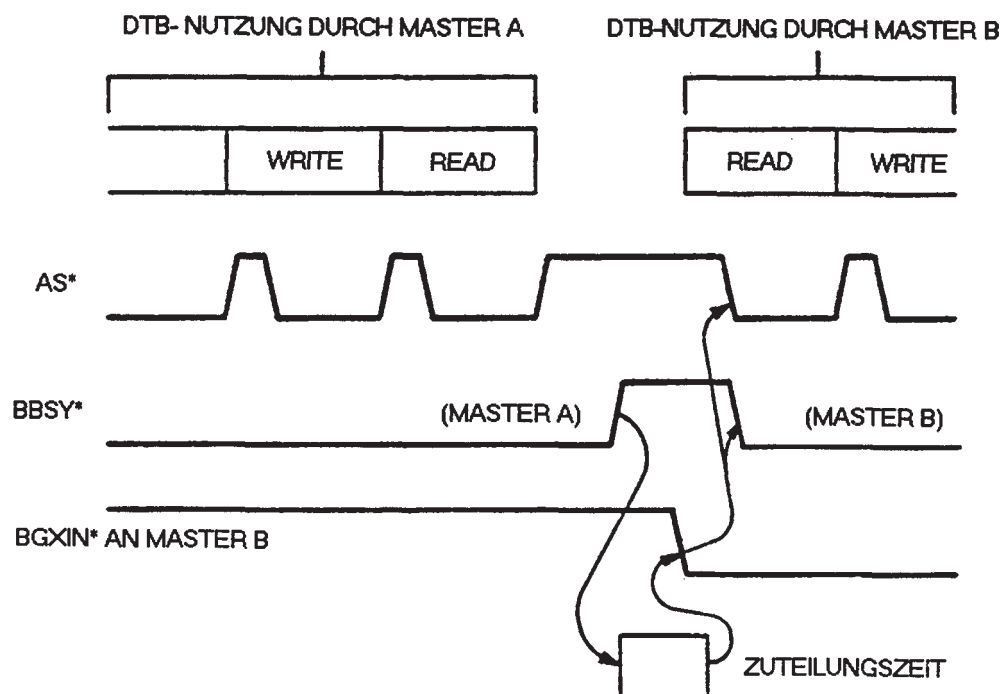


Bild 2-11: MASTER-Wechsel auf dem Datentransferbus

2.6 Vorschriften und Erläuterungen zum DTB-Zeitverhalten

Dieser Abschnitt beschreibt die VORSCHRIFTEN und ERLÄUTERUNGEN für den zeitlichen Ablauf, die das Verhalten der MASTER und SLAVES regeln. Diese Information zum zeitlichen Ablauf wird in Form von Bildern und Tabellen gegeben:

Tabelle 2-17 enthält eine Zeitablauf-Tabelle und Zeitablauf-Diagramme, die den Betrieb von MASTER-, SLAVE- und LOCATION MONITOR festlegen.

Tabelle 2-18 beschreibt die verschiedenen, in diesem Hauptabschnitt benutzten Mnemonics.

Die Tabellen 2-19 bis 2-21 legen die Benutzung der DTB-Signale fest.

Die Tabellen 2-22 bis 2-27 legen die Zeitablauf-Parameter des DTB fest. (Die Referenzziffern, die in den Tabellen 2-24 bis 2-27 benutzt werden, entsprechen den Ziffern der Zeitablauf-Parameter in den Tabellen 2-22 und 2-23.)

Die Bilder 2-12 bis 2-15 legen die Zeitablauf-VORSCHRIFTEN und ERLÄUTERUNGEN während der Zeit des Sendens der Adresse fest.

Die Bilder 2-16 bis 2-21 legen Zeitablauf-VORSCHRIFTEN und ERLÄUTERUNGEN für MASTER, SLAVES und LOCATION MONITORE während der Datentransferzeit fest.

Die Bilder 2-22 bis 2-24 legen die Zeitablauf-VORSCHRIFTEN und ERLÄUTERUNGEN für MASTER und SLAVES zwischen DTB-Zyklen fest.

Bild 2-25 ist das Zeitablauf-Diagramm für MASTER, SLAVE und BUS TIMER für einen Time-Out-Zyklus.

Bild 2-26 zeigt den Zeitablauf während eines Masterwechsels auf dem DTB.

Um den festgelegten Zeitablauf-VORSCHRIFTEN gerecht zu werden, müssen Systementwickler die Worst-Case-Durchlaufverzögerungen der Bus-Treiber und -Empfänger, die auf ihren IEC-821-BUS-Karten benutzt werden, berücksichtigen. Die Durchlaufverzögerung der Treiber ist abhängig von den Ausgangsbelastungen, und die Herstellerspezifikationen geben nicht immer genug Information, diese Durchlaufverzögerung bei verschiedenen Belastungen zu berechnen. Um die IEC-821-BUS-Systementwickler zu unterstützen, werden in Kapitel 6 einige Vorschläge unterbreitet.

Die ERLÄUTERUNGEN legen das Zeitablaufverhalten der Signalfanken für Eingangsleitungen fest. Auf diese Zeiten kann man sich verlassen, solange die VORSCHRIFTEN über die Busplatten-Belastung in Kapitel 6 nicht verletzt werden. Die VORSCHRIFTEN für die Busabschlüsse in Kapitel 6 garantieren, daß die Zeitablauf-Parameter für Signalleitungen, die nach einer Belegung freigegeben werden, eingehalten werden.

Typischerweise gibt es für jede Zeitablauf-VORSCHRIFT eine entsprechende ERLÄUTERUNG. Die in der ERLÄUTERUNG garantierte Zeit kann dabei von der in der VORSCHRIFT festgelegten abweichen. Z. B. ergibt eine sorgfältige Überprüfung der Zeitablauf-Diagramme, daß vom MASTER verlangt wird, daß er 35 ns Adressen- und Daten-Set-Up-Zeit zur Verfügung stellt, dem SLAVE jedoch nur 10 ns zugestanden werden. Der Grund dafür ist, daß die Adreß- und Daten-Bustreiber nicht immer imstande sind, die Busplatten-Signalleitungen vollkommen durch den Schwellwertbereich von low auf high zu treiben, bevor die Flanke sich zum Busplattenende ausgebreitet hat und reflektiert wird. Typischerweise durchlaufen jedoch die fallenden Flanken der Adreß- und Daten-Strobes den 0,8-V-Schwellwertbereich, ohne daß sie durch Reflektion gestört werden. Die resultierende Set-up-Zeit für den SLAVE ist die Set-up-Zeit des MASTERS abzüglich zweier Busausbreitungszeiten.

Eine spezielle Notation wird verwendet, um das Daten-Strobe-Zeitablaufverhalten zu beschreiben. DS0* und DS1* führen ihre Flankenwechsel nicht immer gleichzeitig durch. In diesen Zeitablauf-Diagrammen stellt DSA* den Daten-Strobe dar, der als erster den Flankenwechsel vornimmt, sei es DS0* oder DS1*. DSB* stellt den Daten-Strobe dar, der als zweiter den Flankenwechsel vornimmt, sei es DS1* oder DS0*. Die für die Zeit stabiler Daten-Strobes unterbrochen gezeichnete Linie zeigt an, daß der Daten-Strobe mit zuerst fallender Flanke nicht der erste mit ansteigender Flanke sein muß, d. h. DSA* kann an der fallenden Flanke DS0* und an der steigenden Flanke DS1* repräsentieren.

Die Tabellen 2-19, 2-20 und 2-21 zeigen, wie die verschiedenen Signalleitungen des DTB zum Senden von Adressen und zum Transfer von Daten benutzt werden. Auf diese Tabellen wird durch die folgenden verschiedenen Zeitablauf-Diagramme Bezug genommen. Um diese Tabellen kompakt zu halten, werden Mnemonics zur Beschreibung, wann und wie die verschiedenen Leitungen getrieben werden, benutzt. Diese Mnemonics werden in Tabelle 2-18 definiert.

Tabelle 2-17: Zeitablauf-Diagramme, die den MASTER-, SLAVE- und LOCATION-MONITOR-Betrieb definieren
(Siehe Tabelle 2-23 für Zeitablauf-Werte)

Mnemonic	Art der Zyklen	Zeitablauf-Diagramme beim Senden der Adresse	Datentransfer- Zeitablauf-Diagramme
ADO	ADDRESS-ONLY	Bild 2-12	Nicht vorhanden
D08(E0)	geradzahlige Ein-Byte-Transfers BYTE(0) READ BYTE(2) READ BYTE(0) WRITE BYTE(2) WRITE	Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13	Bild 2-16 Bild 2-16 Bild 2-18 Bild 2-18
D08(E0) oder D08(O)	ungeradzahlige Ein-Byte-Transfers BYTE(1) READ BYTE(3) READ BYTE(1) WRITE BYTE(3) WRITE	Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13	Bild 2-16 Bild 2-16 Bild 2-18 Bild 2-18
D16	Zwei-Byte-Transfers BYTE(0-1) READ BYTE(2-3) READ BYTE(0-1) WRITE BYTE(2-3) WRITE	Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13	Bild 2-17 Bild 2-17 Bild 2-19 Bild 2-19
D32	Vier-Byte-Transfers BYTE(0-3) READ BYTE(0-3) WRITE	Bilder 2-12, 2-13 Bilder 2-12, 2-13	Bild 2-17 Bild 2-19
D08(E0):BLT	Ein-Byte-Blocktransfers SINGLE-BYTE-BLOCK READ SINGLE-BYTE-BLOCK WRITE	Bilder 2-13, 2-14 Bilder 2-13, 2-14	Bild 2-16 Bild 2-18
D16:BLT	Zwei-Byte-Blocktransfers DOUBLE-BYTE-BLOCK READ DOUBLE-BYTE-BLOCK WRITE	Bilder 2-13, 2-14 Bilder 2-13, 2-14	Bild 2-17 Bild 2-19
D32:BLT	Vier-Byte-Blocktransfers QUAD-BYTE-BLOCK READ QUAD-BYTE-BLOCK WRITE	Bilder 2-13, 2-14 Bilder 2-13, 2-14	Bild 2-17 Bild 2-19
D08(E0):RMW	Ein-Byte-RMW-Transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	Bilder 2-13, 2-15 Bilder 2-13, 2-15 Bilder 2-13, 2-15 Bilder 2-13, 2-15	Bild 2-20 Bild 2-20 Bild 2-20 Bild 2-20
D16:RMW	Zwei-Byte-RMW-Transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	Bilder 2-13, 2-15 Bilder 2-13, 2-15	Bild 2-21 Bild 2-21
D32:RMW	Vier-Byte-RMW-Transfers BYTE(0-3) READ-MODIFY-WRITE	Bilder 2-13, 2-15	Bild 2-21
D32:UAT	nicht an Wortgrenzen gebundene Transfers BYTE(0-2) READ BYTE(1-3) READ BYTE(1-2) READ BYTE(0-2) WRITE BYTE(1-3) WRITE BYTE(1-2) WRITE	Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13 Bilder 2-12, 2-13	Bild 2-16 Bild 2-16 Bild 2-17 Bild 2-18 Bild 2-18 Bild 2-19

Tabelle 2-18: Definitionen der in den Tabellen 2-19, 2-20 und 2-21 benutzten Mnemonics

Mnemonic	Beschreibung	Bemerkungen
DVBM	DURCH MASTER GÜLTIG GETRIEBEN	VORSCHRIFT 2.21: Der MASTER MUSS DVBM-Leitungen auf einen gültigen Pegel treiben
DLBM	DURCH MASTER LOW GETRIEBEN	VORSCHRIFT 2.22: Der MASTER MUSS DLBM-Leitungen auf low treiben.
DHBM	DURCH MASTER HIGH GETRIEBEN	VORSCHRIFT 2.23: Der MASTER MUSS DHBM-Leitungen auf high treiben
dhbm?	DURCH MASTER HIGH GETRIEBEN?	ERLAUBNIS 2.10: Der MASTER DARF dhbm?-Leitungen high treiben. VORSCHRIFT 2.24: Der MASTER DARF dhbm?-Leitungen NICHT low treiben.
dxbm?	IST DURCH MASTER GETRIEBEN?	ERLAUBNIS 2.11: Der MASTER DARF dxbm?-Leitungen treiben oder DARF diese Leitungen ungetrieben lassen. (Wenn dxbm?-Leitungen getrieben werden, führen sie keine gültige Information.)
DVBS	DURCH SLAVE GÜLTIG GETRIEBEN	VORSCHRIFT 2.25: Der SLAVE MUSS DVBS-Leitungen auf einen gültigen Pegel treiben
dxbs?	DURCH DEN SLAVE GETRIEBEN?	ERLAUBNIS 2.12: Der SLAVE DARF dxbs?-Leitungen treiben oder DARF diese Leitungen ungetrieben lassen. (Wenn dxbs?-Leitungen getrieben werden, führen sie keine gültige Information.)
DVBB	DURCH BEIDE, SLAVE UND MASTER, GÜLTIG GETRIEBEN	VORSCHRIFT 2.26: Während des "Read"-Teils eines Read-Modify-Write-Zyklus MUSS der SLAVE DVBB-Leitungen mit gültigen Daten treiben. Während des "Write"-Teils eines Read-Modify-Write-Zyklus MUSS der MASTER DVBB-Leitungen mit gültigen Daten treiben.
dxbb?	DURCH BEIDE, SLAVE UND MASTER, GETRIEBEN?	ERLAUBNIS 2.13: Während des "Read"-Teils eines Read-Modify-Write-Zyklus DARF der SLAVE dxbb?-Leitungen treiben oder sie ungetrieben lassen. Während des "Write"-Teils eines Read-Modify-Write-Zyklus DARF der MASTER dxbb?-Leitungen treiben oder sie ungetrieben lassen. (Wenn dxbb?-Leitungen getrieben werden, führen sie keine gültige Information.)

Tabelle 2-19: Verwendung der Adreßleitungen zur Auswahl einer 4-Byte-Gruppe

Mnemonic	Adressierungsmodus	A02-A15 (siehe Anmerkung)	A16-A23	A24-A31	IACK*
A16	SHORT	DVBM	dxbm?	dxbm?	dhbm?
A24	STANDARD	DVBM	DVBM	dxbm?	dhbm?
A32	EXTENDED	DVBM	DVBM	DVBM	dhbm?

ANMERKUNG: A01 wird in Verbindung mit LWORD* und den zwei Daten-Strobes zur Auswahl benutzt, auf welche der vier Bytes in der 4-Byte-Gruppe der Zugriff erfolgt (siehe Tabelle 2-20).

Tabelle 2-20: Verwendung von DS1*, DS0*, A01 und LWORD* während der verschiedenen Zyklen

Mnemonic	Art der Zyklen	DS1*	DS0*	A01	LWORD*
ADO	ADDRESS-ONLY	dhbm?	dhbm?	dxbm?	dxbm?
D08(EO)	geradzahlige Ein-Byte-Transfers BYTE(0) READ oder WRITE BYTE(2) READ oder WRITE	DLBM DLBM	dhbm? dhbm?	DLBM DHBM	dhbm? dhbm?
D08(EO) oder D08(O)	ungeradzahlige Ein-Byte-Transfers BYTE(1) READ oder WRITE BYTE(3) READ oder WRITE	dhbm? dhbm?	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D16	Zwei-Byte-Transfers BYTE(0-1) READ oder WRITE BYTE(2-3) READ oder WRITE	DLBM DLBM	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D32	Vier-Byte-Transfers BYTE(0-3) READ oder WRITE	DLBM	DLBM	DLBM	DLBM
D08(EO):BLT	Ein-Byte-Blocktransfers SINGLE-BYTE-BLOCK READ oder WRITE	Anmerkung 1			dhbm?
D16:BLT	Zwei-Byte-Blocktransfers DOUBLE-BYTE-BLOCK READ oder WRITE	DLBM	DLBM	Anmerkung 2	dhbm?
D32:BLT	Vier-Byte-Blocktransfers QUAD-BYTE-BLOCK READ oder WRITE	DLBM	DLBM	DLBM	DLBM
D08(EO):RMW	Ein-Byte-RMW-Transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	DLBM dhbm? DLBM dhbm?	dhbm? DLBM dhbm? DLBM	DLBM DLBM DHBM DHBM	dhbm? dhbm? dhbm? dhbm?
D16:RMW	Zwei-Byte-RMW-Transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	DLBM DLBM	DLBM DLBM	DLBM DLBM	dhbm? dhbm?
D32:RMW	Vier-Byte-RMW-Transfers BYTE(0-3) READ-MODIFY-WRITE	DLBM	DLBM	DLBM	DLBM
D32:UAT	nicht an Wortgrenzen gebundene Transfers BYTE(0-2) READ oder WRITE BYTE(1-3) READ oder WRITE BYTE(1-2) READ oder WRITE	DLBM dhbm? DLBM	dhbm? DLBM DLBM	DLBM DLBM DHBM	DLBM DLBM DLBM

ANMERKUNG 1: Während der Ein-Byte-Blocktransfers werden die beiden Daten-Strobes abwechselnd low getrieben. Während der ersten Übertragung kann jeder der beiden Daten-Strobes low getrieben werden. Wenn der erste Zugriff auf den BYTE-Speicherplatz BYTE(0) oder BYTE(2) erfolgt, treibt der MASTER zuerst DS1* low. Wenn der erste Zugriff auf den Byte-Speicherplatz BYTE(1) oder BYTE(3) erfolgt, treibt er zuerst DS0* low. A01 ist nur während des ersten Datentransfers gültig (d. h., bis der SLAVE DTACK* oder BERR* das erste Mal low treibt) und kann entweder high oder low sein, abhängig vom Anfangs-Byte des Ein-Byte-Blocktransfers. Wenn die erste Byte-Stelle BYTE(0) oder BYTE(1) ist, treibt der MASTER A01 low. Wenn die erste Byte-Stelle BYTE(2) oder BYTE(3) ist, treibt der MASTER A01 high.

Ein Beispiel für die Nutzung von DS0*, DS1*, A01 und LWORD* während eines Ein-Byte-Blocktransfer-Zyklus, der mit BYTE(2) beginnt, ist unten aufgeführt:

	DS1*	DS0*	A01	LWORD*
Erster Datentransfer	DLBM	DHBM	DHBM	dhbm?
BYTE(2)	DHBM	DLBM	dxbm?	dxbm?
BYTE(3)	DLBM	DHBM	dxbm?	dxbm?
BYTE(0)	DHBM	DLBM	dxbm?	dxbm?
BYTE(1)	DLBM	DHBM	dxbm?	dxbm?
Letzter Datentransfer	DLBM	DHBM	dxbm?	dxbm?
BYTE(2)				

ANMERKUNG 2: Bei einem Zwei-Byte-Blocktransfer ist A01 nur gültig während des ersten Datentransfers (d. h., bis der SLAVE DTACK* oder BERR* zum ersten Mal low treibt) und wird high oder low getrieben, abhängig von der Zwei-Byte-Blockgruppe, mit der der Zwei-Byte-Blocktransfer beginnt. Wenn die erste Zwei-Byte-Blockgruppe BYTE(0-1) ist, treibt der MASTER A01 low. Wenn die erste Zwei-Byte-Blockgruppe BYTE(2-3) ist, treibt der MASTER A01 high.

Tabelle 2-21: Verwendung der Datenbusleitungen für den Datentransfer

Mnemonic	Art der Zyklen	D24-D31	D16-D23	D08-D15	D00-D07
ADO	ADDRESS-ONLY	dxbm?	dxbm?	dxbm?	dxbm?
D08(E0)	geradzahlige Ein-Byte-Transfers BYTE(0) READ BYTE(2) READ BYTE(0) WRITE BYTE(2) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM	dxbs? dxbs? dxbm? dxbm?
D08(E0) oder D08(O)	ungeradzahlige Ein-Byte-Transfers BYTE(1) READ BYTE(3) READ BYTE(1) WRITE BYTE(3) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM
D16	Zwei-Byte-Transfers BYTE(0-1) READ BYTE(2-3) READ BYTE(0-1) WRITE BYTE(2-3) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM	DVBS DVBS DVBM DVBM
D32	Vier-Byte-Transfers BYTE(0-3) READ BYTE(0-3) WRITE	DVBS DVBM	DVBS DVBM	DVBS DVBM	DVBS DVBM
D08(E0):BLT	Ein-Byte-Blocktransfers SINGLE-BYTE-BLOCK READ SINGLE-BYTE-BLOCK WRITE	dxbs? dxbm?	dxbs? dxbm?	siehe Anmerkung siehe Anmerkung	
D16:BLT	Zwei-Byte-Blocktransfers DOUBLE-BYTE-BLOCK READ DOUBLE-BYTE-BLOCK WRITE	dxbs? dxbm?	dxbs? dxbm?	DVBS DVBM	DVBS DVBM
D32:BLT	Vier-Byte-Blocktransfers QUAD-BYTE-BLOCK READ QUAD-BYTE-BLOCK WRITE	DVBS DVBM	DVBS DVBM	DVBS DVBM	DVBS DVBM
D08(E0):RMW	Ein-Byte-RMW-Transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	dxbb? dxbb? dxbb? dxbb?	dxbb? dxbb? dxbb? dxbb?	DVBB dxbb DVBB dxbb	dxbb? DVBB dxbb? DVBB
D16:RMW	Zwei-Byte-RMW-Transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	dxbb? dxbb?	dxbb? dxbb?	DVBB DVBB	DVBB DVBB
D32:RMW	Vier-Byte-RMW-Transfers BYTE(0-3) READ-MODIFY-WRITE	DVBB	DVBB	DVBB	DVBB
D32:UAT	nicht an Wortgrenzen gebundene Transfers BYTE(0-2) READ BYTE(1-3) READ BYTE(1-2) READ BYTE(0-2) WRITE BYTE(1-3) WRITE BYTE(1-2) WRITE	DVBS dxbs? dxbs? DVBM dxbm? dxbm?	DVBS DVBS DVBS DVBM DVBM DVBM	DVBS DVBS DVBS DVBM DVBM DVBM	dxbs? DVBS dxbs? dxbm? DVBM dxbm?

ANMERKUNG: Bei Ein-Byte-Blocktransfers werden die Daten 8bitweise über D00-D07 oder D08-D15 übertragen. Ein Beispiel für SINGLE-BYTE-BLOCK READ ist unten aufgeführt.

	D08-D15	D00-D07
Erster Datentransfer	DVBS dxbs? DVBS dxbs? DVBS dxbs?	dxbs? DVBS dxbs? DVBS dxbs?
Letzter Datentransfer	DVBS	dxbs?

Tabelle 2-22: Zeitablauf-Parameter von MASTER, SLAVE und LOCATION MONITOR

PARAMETER NUMMER	MASTER siehe auch Tabelle 2-24		SLAVE siehe auch Tabelle 2-25		LOCATION MONITOR siehe auch Tabelle 2-26	
	min.	max.	min.	max.	min.	max.
1	0	—	—	—	—	—
2	0	—	—	—	—	—
3	60	—	—	—	—	—
4	35	—	10	—	10	—
5	40	—	30	—	30	—
6	0	—	0	—	—	—
7	0	—	0	—	—	—
8	35	—	10	—	—	—
9	0	—	0	—	—	—
10	0	—	—10	—	—10	—
11	40	—	30	—	30	—
12	35	—	10	—	10	—
13	—	10	—	20	—	20
14	0	—	0	—	—	—
15	0	—	0	—	—	—
16	0	—	0	—	—	—
17	40	—	30	—	30	—
18	0	—	0	—	—	—
19	40	—	30	—	30	—
20	0	—	0	—	—	—
21	0	—	0	—	—	—
22	0	—	0	—	—	—
23	10	—	0	—	0	—
24A	0	—	—	—	—	—
24B	0	—	—	—	—	—
25	—	25	—	—	—	—
26	0	—	0	—	—	—
27	—25	—	0	—	—	—
28	30	2T	30	—	—	—
29	0	—	0	—	—	—
30	0	—	0	—	—	—
31	0	—	0	—	—	—
32	—	—	10	—	10	—
33	—	—	30	—	30	—

ANMERKUNG 1: Alle Zeiten sind in Nanosekunden angegeben.

ANMERKUNG 2: T = Time-Out-Zeit, in Mikrosekunden.

Tabelle 2-23: Zeitablauf-Parameter des BUS TIMER
(siehe auch Tabelle 2-28)

PARAMETER NUMMER	MASTER	
	min.	max.
28	T	2T
30	0	—

ANMERKUNG: T = Time-Out-Zeit in Mikrosekunden.

Tabelle 2-24: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten des MASTER

ANMERKUNG: Die Nummern entsprechen den in Tabelle 2-22 festgelegten Zeitablauf-Parametern

1. VORSCHRIFT 2.27:
Wenn der MASTER die Steuerung des IEC-821-BUS übernimmt, DARF er IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1* oder AS* NICHT treiben, bevor der vorhergehende MASTER AS* erlaubt, den Low-Pegel zu verlassen.
ERLÄUTERUNG 2.35:
Kapitel 3 beschreibt, wie dem REQUESTER eines MASTER der IEC-821-BUS zugeteilt wird.
2. VORSCHRIFT 2.28:
Wenn der MASTER die Steuerung des IEC-821-BUS übernimmt, DARF er IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1* oder AS* NICHT treiben, bevor sein REQUESTER den Bus zugeteilt bekommt.
ERLÄUTERUNG 2.36:
Kapitel 3 beschreibt, wie dem REQUESTER eines MASTER der IEC-821-BUS zugeteilt wird.
3. VORSCHRIFT 2.29:
Wenn der MASTER die Steuerung des IEC-821-BUS übernimmt, DARF er AS* NICHT vor Ablauf dieser Zeit low treiben, nachdem der vorhergehende MASTER AS* erlaubt, den Low-Pegel zu verlassen.
ERLÄUTERUNG 2.37:
VORSCHRIFT 2.29 stellt sicher, daß der Zeitablauf-Parameter 5 für SLAVES garantiert wird, wenn ein Wechsel des DTB-MASTER erfolgt.
4. VORSCHRIFT 2.30:
Der MASTER DARF NICHT AS* low treiben, bevor IACK* high und die benötigten Leitungen von A01-A31, AM0-AM5 und LWORD* für diese Mindestzeit gültig waren.
ERLÄUTERUNG 2.38:
Tabelle 2-19 legt die Leitungen aus A01-A31, die der MASTER treiben muß, fest, Tabelle 2-3 legt fest, wie er AM0-AM5 benutzt und Tabelle 2-20 legt fest, wie er LWORD* benutzt.
5. VORSCHRIFT 2.31:
Bei Benutzung des DTB für zwei aufeinanderfolgende Zyklen DARF der MASTER AS* NICHT low treiben, bevor AS* für diese Mindestzeit high war.
6. VORSCHRIFT 2.32:
Nach einem Lesezyklus DARF der MASTER KEINE der Leitungen D00-D31 treiben, bevor DTACK* und BERR* beide high sind.
7. VORSCHRIFT 2.33:
Während Lesezyklen DARF der MASTER NICHT DSA* low treiben, bevor er alle Leitungen D00-D31 freigegeben hat.
8. VORSCHRIFT 2.34:
Während Schreibzyklen DARF der MASTER NICHT DSA* low treiben, bevor die benötigten Leitungen von D00-D31 für diese Mindestzeit gültig sind.
ERLÄUTERUNG 2.39:
Tabelle 2-21 legt die Leitungen von D00-D31 fest, die ein MASTER treiben muß.
9. VORSCHRIFT 2.35:
Der MASTER DARF NICHT DSA* low treiben, bis sowohl DTACK* als auch BERR* high sind.
10. VORSCHRIFT 2.36:
Der MASTER DARF NICHT DSA* low treiben, bis er AS* low getrieben hat.
11. VORSCHRIFT 2.37:
Der MASTER DARF NICHT DSA* low treiben, bis DS0* und DS1* beide gleichzeitig für diese Mindestzeit high waren.
12. VORSCHRIFT 2.38:
Der MASTER DARF NICHT DSA* low treiben, bis WRITE* für diese Mindestzeit gültig war.
13. VORSCHRIFT 2.39:
Während Zyklen, in denen der MASTER beide Daten-Strobes low treibt, MUSS er innerhalb dieser Maximalzeit DSB* low treiben, nachdem er DSA* low getrieben hat.
ERLÄUTERUNG 2.40:
Der Zeitablauf-Parameter 13 ist nicht anwendbar auf Transfers, bei denen nur DS0* oder DS1* low getrieben wird.

(fortgesetzt)

Tabelle 2-24 (fortgesetzt)

14. **VORSCHRIFT 2.40:**
Während aller Datentransfer-Zyklen, ausgenommen Read-Modify-Write-Zyklen, MUSS der MASTER A01-A03 gültig halten und den entsprechenden Pegel von LWORD* aufrechterhalten, bis er die erste fallende Flanke von DTACK* oder BERR* feststellt.
ERLÄUTERUNG 2.41:
Während aller Datentransfer-Zyklen, ausgenommen Blocktransfer- und Read-Modify-Write-Zyklen, gibt es nur eine fallende Flanke von DTACK* oder BERR*.
15. **VORSCHRIFT 2.41:**
Während Read-Modify-Write-Zyklen MUSS der MASTER A01-A03 gültig halten und den entsprechenden Pegel von LWORD* aufrechterhalten, bis er die zweite fallende Flanke von DTACK* oder BERR* feststellt.
16. **VORSCHRIFT 2.42:**
Während aller Datentransfer-Zyklen MUSS der MASTER einen gültigen Adreßmodifizierer-Code aufrechterhalten und sicherstellen, daß IACK* high bleibt, bis er die letzte fallende Flanke von DTACK* oder BERR* feststellt.
ERLÄUTERUNG 2.42:
Während aller Datentransfer-Zyklen, ausgenommen Blocktransfer- und Read-Modify-Write-Zyklen, gibt es nur eine fallende Flanke von DTACK* oder BERR*.
17. **VORSCHRIFT 2.43:**
Der MASTER DARF die Pegel von IACK*, A01-A31, AM0-AM5 und LWORD* während dieser Mindestzeit NICHT ändern, nachdem er AS* low getrieben hat.
18. **VORSCHRIFT 2.44:**
Während aller Datentransfer-Zyklen MUSS der MASTER AS* low halten, bis er die letzte fallende Flanke von DTACK* oder BERR* feststellt.
19. **VORSCHRIFT 2.45:**
Der MASTER MUSS während dieser Mindestzeit AS* low halten.
20. **VORSCHRIFT 2.46:**
Wenn ein MASTER DSA* low getrieben hat, MUSS er diese Leitung low halten, bis er bei DTACK* oder BERR* low feststellt.
21. **VORSCHRIFT 2.47:**
Wenn ein MASTER DSB* low getrieben hat, MUSS er diese Leitung low halten, bis er bei DTACK* oder BERR* low feststellt.
22. **VORSCHRIFT 2.48:**
Wenn der MASTER während Schreibzyklen DSA* low getrieben hat, DARF er KEINE der Leitungen D00-D31 ändern, bis er bei DTACK* oder BERR* low feststellt.
23. **VORSCHRIFT 2.49:**
Wenn ein MASTER DSA* low getrieben hat, DARF er NICHT den Pegel der WRITE*-Leitung während dieser Mindestzeit ändern, nachdem DS0* und DS1* beide high sind.
- 24A. **VORSCHRIFT 2.50:**
WENN der MASTER AS* auf high treibt oder freigibt, nachdem sein REQUESTER BBSY* freigibt,
DANN MUSS er IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0* und DS1* freigeben, bevor er AS* gestattet, den Low-Pegel zu verlassen.
ERLÄUTERUNG 2.43:
Kapitel 3 beschreibt, wie der REQUESTER eines MASTER die BBSY*-Leitung freigibt.
- 24B. **VORSCHRIFT 2.51:**
WENN der MASTER AS* auf high treibt oder freigibt, bevor sein REQUESTER BBSY* freigibt,
DANN MUSS er AS*, IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0* und DS1* freigeben, bevor er dem REQUESTER erlaubt, BBSY* freizugeben.
ERLÄUTERUNG 2.44:
Kapitel 3 beschreibt, wie der REQUESTER eines MASTER die BBSY*-Leitung freigibt.
25. **VORSCHRIFT 2.52:**
WENN der MASTER AS* auf high treibt oder freigibt, bevor sein REQUESTER BBSY* freigibt,
DANN MUSS er AS* innerhalb dieser Zeit freigeben, nachdem er AS* erlaubt hat, den Low-Pegel zu verlassen.
ERLÄUTERUNG 2.45:
Kapitel 3 beschreibt, wie der REQUESTER eines MASTER die BBSY*-Leitung freigibt.

(fortgesetzt)

Tabelle 2-24 (abgeschlossen)

26.	ERLÄUTERUNG 2.46: Der Zeitablauf-Parameter 26 garantiert, daß während der Lesezyklen der Datenbus nicht getrieben wird, bis der MASTER DSA* low treibt.
27.	ERLÄUTERUNG 2.47: Während Lesezyklen wird dem MASTER garantiert, daß Datenbus innerhalb dieser Zeit, nachdem DTACK* auf low geht, gültig ist. Diese Zeit tritt bei Zyklen, in denen der SLAVE BERR* anstelle von DTACK* low treibt, nicht auf.
28.	ERLÄUTERUNG 2.48: Dem MASTER wird garantiert, daß weder DTACK* noch BERR* während dieser Mindestzeit low gehen, nachdem er DSA* low getrieben hat. Der BUS TIMER garantiert dem MASTER, daß der BUS TIMER BERR* low treibt, wenn DTACK* nicht nach Ablauf der Time-Out-Periode und innerhalb der doppelten Time-Out-Periode low gegangen ist.
29.	ERLÄUTERUNG 2.49: Während Lesezyklen wird dem MASTER garantiert, daß der Datenbus gültig bleibt, bis er DSA* high treibt.
30.	ERLÄUTERUNG 2.50: Der Zeitablauf-Parameter 30 garantiert, daß weder DTACK* noch BERR* high gehen, bis der MASTER sowohl DS0* als auch DS1* high treibt.
31.	ERLÄUTERUNG 2.51: Während Lesezyklen wird dem MASTER garantiert, daß der Datenbus während des Zeitabschnitts, in dem DTACK* und BERR* high sind, freigegeben ist.

Tabelle 2-25: SLAVE: VORSCHRIFTEN und ERLÄUTERUNGEN zum Zeitverhalten

ANMERKUNG: Die Nummern entsprechen den in Tabelle 2-22 festgelegten Zeitablauf-Parametern.

4.	ERLÄUTERUNG 2.52 Allen SLAVES wird garantiert, daß IACK*, A01-A31, AM0-AM5 und LWORD* mindestens seit dieser Zeit gültig sind, wenn sie eine fallende Flanke von AS* feststellen.
5.	ERLÄUTERUNG 2.53 Allen SLAVES wird zwischen DTB-Zyklen diese Mindestzeit mit High-Pegel auf AS* garantiert.
6.	ERLÄUTERUNG 2.54: Während Lesezyklen wird dem reagierenden SLAVE garantiert, daß keine der Leitungen D00-D31 durch irgendein anderes Modul getrieben wird, bis der reagierende SLAVE DTACK* und BERR* auf high freigibt.
7.	ERLÄUTERUNG 2.55: Während Lesezyklen wird dem reagierenden SLAVE garantiert, daß der Datenbus durch alle anderen Module freigegeben wird, wenn DSA* low wird.
8.	ERLÄUTERUNG 2.56: Während Schreibzyklen wird dem reagierenden SLAVE garantiert, daß der Datenbus für diese Mindestzeit gültig war, wenn er eine fallende Flanke von DSA* feststellt.
9.	ERLÄUTERUNG 2.57: Dem reagierenden SLAVE wird garantiert, daß weder DS0* noch DS1* auf low gehen werden, bis DTACK* und BERR* aus dem vorhergegangenen Zyklus high geworden sind.
10.	ERLÄUTERUNG 2.58: Aufgrund der Buslaufzeit-Differenz könnten SLAVES auf dem DTB eine fallende Flanke von DSA* vor der fallenden Flanke von AS* feststellen. Den SLAVES ist jedoch garantiert, daß eine fallende Flanke von DSA* der fallenden Flanke von AS* nicht mehr als diese Zeit voreilt.
11.	ERLÄUTERUNG 2.59: Den SLAVES wird diese Mindestzeit garantiert, während der DS0* und DS1* beide zwischen aufeinander folgenden Datentransfers gleichzeitig high sind.
12.	ERLÄUTERUNG 2.60: Den SLAVES wird garantiert, daß WRITE* für diese Mindestzeit vor einer fallenden Flanke von DSA* gültig bleibt.

(fortgesetzt)

Tabelle 2-25 (fortgesetzt)

13. ERLÄUTERUNG 2.61:
WENN der MASTER DS0* und DS1* beide low treibt,
DANN wird dem reagierenden SLAVE garantiert, daß DSB* innerhalb dieser Maximalzeit low wird, nachdem DSA* low wurde.
14. ERLÄUTERUNG 2.62:
Während aller Datentransfer-Zyklen, ausgenommen Read-Modify-Write-Zyklen, wird dem reagierenden SLAVE garantiert, daß A01-A31 und LWORD* gültig bleiben, bis er DTACK* oder BERR* das erste Mal low treibt, vorausgesetzt, dies geschieht innerhalb der Time-Out-Periode des Busses.
15. ERLÄUTERUNG 2.63:
Während aller Read-Modify-Write-Zyklen wird dem reagierenden SLAVE garantiert, daß A01-A31 und LWORD* gültig bleiben, bis er DTACK* oder BERR* das zweite Mal low treibt, vorausgesetzt, er führt dies innerhalb der Bus-Time-Out-Periode durch.
16. ERLÄUTERUNG 2.64:
Dem reagierenden SLAVE wird garantiert, daß IACK* und AM0-AM5 gültig bleiben, bis er DTACK* oder BERR* das letzte Mal low treibt, vorausgesetzt, dies geschieht innerhalb der Time-Out-Periode des Busses.
17. ERLÄUTERUNG 2.65:
Den SLAVES wird garantiert, daß IACK*, A01-A31, AM0-AM5 und LWORD* für diese Mindestzeit nach der fallenden Flanke von AS* gültig bleiben. Während der ADDRESS-ONLY-Zyklen wird diese Zeit durch den MASTER garantiert. Während aller anderen Zyklusarten wird diese Zeit aus den Zeitablauf-Parametern 10, 14, 16 und 28 abgeleitet.
18. ERLÄUTERUNG 2.66:
Dem reagierenden SLAVE wird garantiert, daß AS* low bleibt, bis er DTACK* oder BERR* low treibt, vorausgesetzt, dies geschieht innerhalb der Time-Out-Periode des Busses.
19. ERLÄUTERUNG 2.67:
Den SLAVES wird garantiert, daß AS* für diese Mindestzeit low bleibt.
20. ERLÄUTERUNG 2.68:
Dem reagierenden SLAVE wird garantiert, daß DSA*, einmal auf low gegangen, low bleibt, bis der SLAVE DTACK* oder BERR* low treibt, vorausgesetzt, der SLAVE führt dies innerhalb der Time-Out-Periode des Busses durch.
21. ERLÄUTERUNG 2.69:
Dem reagierenden SLAVE wird garantiert, daß DSB*, einmal nach low gegangen, low bleibt, bis der SLAVE DTACK* oder BERR* low treibt, vorausgesetzt, der SLAVE führt dies innerhalb der Time-Out-Periode des Busses durch.
22. ERLÄUTERUNG 2.70:
Dem reagierenden SLAVE wird während Schreibzyklen garantiert, daß der Datenbus gültig bleibt, bis er DTACK* oder BERR* low treibt, vorausgesetzt, er führt dies innerhalb der Bus-Time-Out-Periode durch.
23. ERLÄUTERUNG 2.71:
Dem reagierenden SLAVE wird garantiert, daß die WRITE*-Leitung gültig bleibt, bis beide Daten-Strobes high sind.
26. VORSCHRIFT 2.53:
Während Lesezyklen DARF der reagierende SLAVE NICHT den Datenbus treiben, bis DSA* auf low geht.
27. VORSCHRIFT 2.54:
Während Lesezyklen DARF der reagierende SLAVE NICHT DTACK* treiben, bevor er die Datenleitungen mit gültigen Daten versorgt.
ERLÄUTERUNG 2.72:
Diese Zeit gilt nicht bei Zyklen, in denen der reagierende SLAVE BERR* anstelle von DTACK* auf low treibt.
28. VORSCHRIFT 2.55:
Der reagierende SLAVE MUSS diese Mindestzeit, nachdem DSA* low geworden ist, warten, bevor er DTACK* oder BERR* auf low treibt.
29. VORSCHRIFT 2.56:
Wenn während Lesezyklen der reagierende SLAVE DTACK* einmal low getrieben hat, DARF er D00-D31 NICHT verändern, bis DSA* high wird.
30. VORSCHRIFT 2.57:
Hat der reagierende SLAVE DTACK* oder BERR* einmal low getrieben, DARF er diese NICHT freigeben, bis er feststellt, daß DS0* und DS1* beide high sind.

(fortgesetzt)

Tabelle 2-25 (abgeschlossen)

- | |
|---|
| <p>31. VORSCHRIFT 2.58:
Während Lesezyklen MUSS der reagierende SLAVE alle Leitungen D00-D31 freigeben, bevor er DTACK* oder BERR* auf high freigibt.</p> <p>32. ERLÄUTERUNG 2.73:
Den SLAVES wird garantiert, daß IACK*, LWORD*, A00-A31 und AM0-AM5 für diese Mindestzeit gültig sind, wenn sie eine fallende Flanke von DSA* feststellen. Diese Zeit wird aus den Zeitablauf-Parametern 4 und 10 abgeleitet.</p> <p>33. ERLÄUTERUNG 2.74:
Während Datentransfer-Zyklen wird den SLAVES garantiert, daß entweder DS0* oder DS1* für wenigstens diese Mindestzeit low bleiben. Diese Zeit wird vom Zeitablauf-Parameter 28 abgeleitet, wobei der reagierende SLAVE eine Mindestzeit warten muß, bevor er BERR* oder DTACK* low treibt.</p> |
|---|

Tabelle 2-26: LOCATION MONITOR: ERLÄUTERUNGEN zum Zeitverhalten

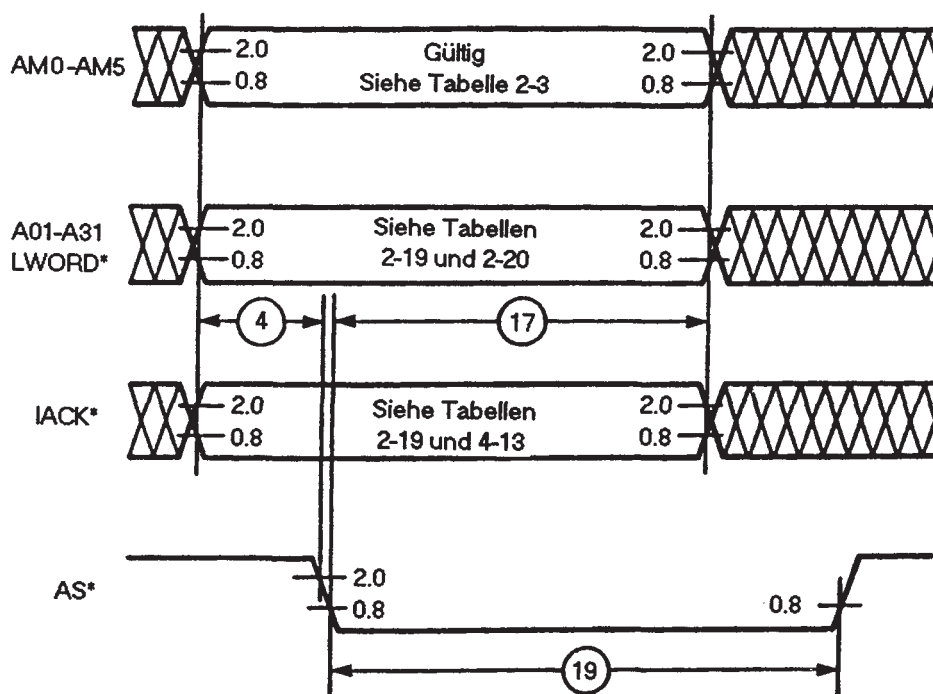
ANMERKUNG: Die Nummern entsprechen den in Tabelle 2-22 festgelegten Zeitablauf-Parametern.

- | |
|--|
| <p>4. ERLÄUTERUNG 2.75:
Dem LOCATION MONITOR wird garantiert, daß IACK*, A01-A31, AM0-AM5 und LWORD* für diese Mindestzeit gültig waren, wenn er eine fallende Flanke von AS* feststellt.</p> <p>5. ERLÄUTERUNG 2.76:
Dem LOCATION MONITOR wird diese Mindestzeit mit High-Pegel an AS* zwischen DTB-Zyklen garantiert.</p> <p>10. ERLÄUTERUNG 2.77:
Aufgrund der Buslaufzeit-Differenz könnten LOCATION MONITORE auf dem DTB eine fallende Flanke von DSA* vor der fallenden Flanke von AS* feststellen. Dem LOCATION MONITOR wird jedoch garantiert, daß die fallende Flanke von DSA* der fallenden Flanke von AS* nicht mehr als diese Zeit voreilt.</p> <p>11. ERLÄUTERUNG 2.78:
Dem LOCATION MONITOR wird diese Mindestzeit garantiert, während der DS0* und DS1* beide zwischen aufeinanderfolgenden Datentransfers gleichzeitig high sind.</p> <p>12. ERLÄUTERUNG 2.79:
Dem LOCATION MONITOR wird garantiert, daß WRITE* für diese Mindestzeit gültig war, wenn er eine fallende Flanke von DSA* feststellt.</p> <p>13. ERLÄUTERUNG 2.80:
WENN der MASTER DS0* und DS1* beide low treibt,
DANN wird dem LOCATION MONITOR garantiert, daß DSB* innerhalb dieser Maximalzeit low wird, nachdem DSA* low wurde.</p> <p>17. ERLÄUTERUNG 2.81:
Dem LOCATION MONITOR wird garantiert, daß IACK*, A01-A31, AM0-AM5 und LWORD* für diese Mindestzeit nach der fallenden Flanke von AS* gültig bleiben. Während ADDRESS-ONLY-Zyklen wird diese Zeit durch den MASTER garantiert. Während aller anderen Zyklusarten wird diese Zeit aus den Zeitablauf-Parametern 10, 14, 16 und 28 abgeleitet.</p> <p>19. ERLÄUTERUNG 2.82:
Dem LOCATION MONITOR wird garantiert, daß AS* für diese Mindestzeit low bleibt.</p> <p>23. ERLÄUTERUNG 2.83:
Dem LOCATION MONITOR wird garantiert, daß WRITE* gültig bleibt, bis DS0* und DS1* beide high sind.</p> <p>32. ERLÄUTERUNG 2.84:
Dem LOCATION MONITOR wird garantiert, daß IACK*, LWORD*, A01-A31 und AM0-AM5 für diese Mindestzeit gültig waren, wenn er eine fallende Flanke von DSA* feststellt.</p> <p>33. ERLÄUTERUNG 2.85:
Während Datentransfer-Zyklen wird dem LOCATION MONITOR garantiert, daß DS0* und/oder DS1* für wenigstens diese Mindestzeit low bleiben. Diese Zeit wird vom Zeitablauf-Parameter 28 abgeleitet, wobei der reagierende SLAVE eine Mindestzeit warten muß, bevor er BERR* oder DTACK* low treibt.</p> |
|--|

Tabelle 2-27: BUS TIMER: VORSCHRIFTEN zum Zeitverhalten

ANMERKUNG: Die Nummern entsprechen den in Tabelle 2-23 festgelegten Zeitablauf-Parametern.

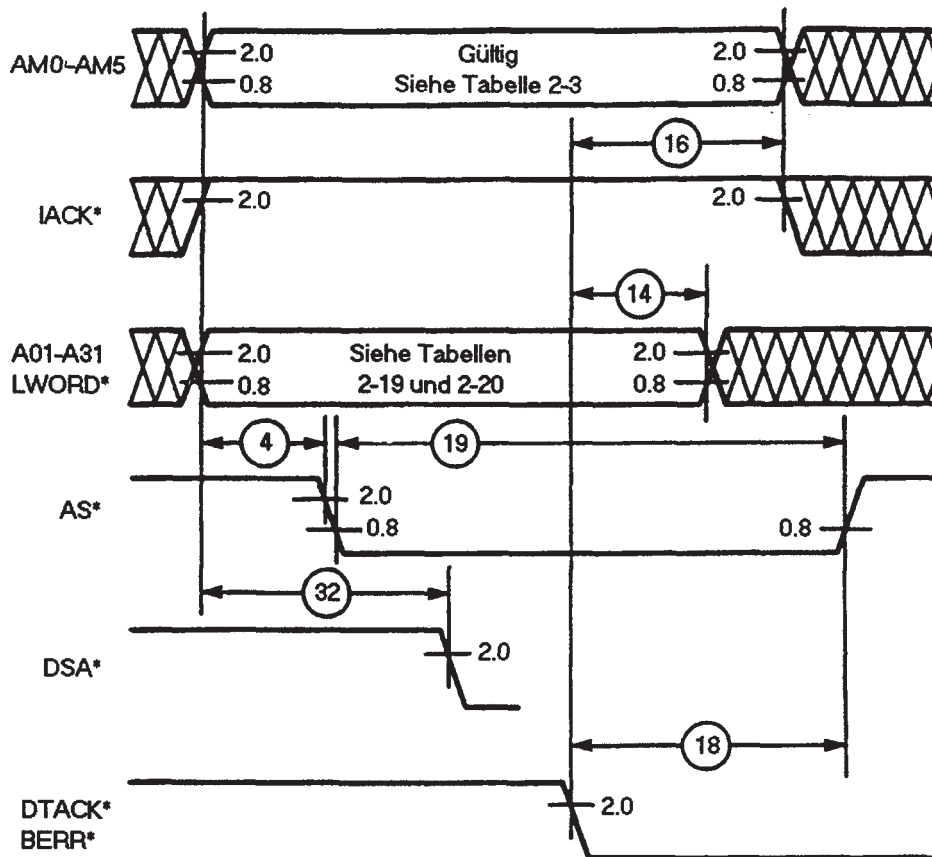
28. VORSCHRIFT 2.59:
Der BUS TIMER MUSS, nachdem DSA* low wird, mindestens seine Time-Out-Periode, jedoch nicht länger als die doppelte Time-Out-Periode abwarten, bevor er BERR* low treibt.
30. VORSCHRIFT 2.60:
Wenn der BUS TIMER BERR* einmal low getrieben hat, DARF er BERR* NICHT freigeben, bis er feststellt, daß DS0* und DS1* beide high sind.



PARAMETER NUMMER	MASTER		SLAVE		LOCATION MONITOR	
	min.	max.	min.	max.	min.	max.
4	35	—	10	—	10	—
17	40	—	30	—	30	—
19	40	—	30	—	30	—

ANMERKUNG: Alle Zeiten sind in Nanosekunden angegeben.

Bild 2-12: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei allen Zyklen



PARAMETER NUMBER	MASTER		SLAVE		LOCATION MONITOR	
	min.	max.	min.	max.	min.	max.
4	35	—	10	—	10	—
14	0	—	0	—	—	—
16	0	—	0	—	—	—
18	0	—	0	—	—	—
19	40	—	30	—	30	—
32	—	—	10	—	10	—

ANMERKUNG: Alle Zeiten sind in Nanosekunden angegeben.

Bild 2-13: Zeitablauf-Diagramm für Adreß-Broadcast von MASTER, SLAVE und LOCATION MONITOR bei

geradzahligen Ein-Byte-Transfers,
ungeradzahligen Ein-Byte-Transfers,
Zwei-Byte-Transfers,
Vier-Byte-Transfers,
nicht an Wortgrenzen gebundenen Transfers