

## Bussysteme

Grundgedanke: ein einziges Bündel von Signalwegen, an das die Einrichtungen angeschlossen werden.

Die Probleme der elektrischen Ebene sind entscheidend:

- Belastbarkeit / Treibfähigkeit,
- Signallaufzeiten,
- Buskonflikte,
- Bumsumschaltung (Turnaround).

Auslegung:

- Datenbus,
- Adreßbus,
- Steuersignale.

Nutzung im Bereich der Mikrocontroller:

- Erweiterung der E-A-Ports,
- Nutzung von Schaltkreisen und Baugruppen/Funktionseinheiten, die entsprechende Schnittstellen haben,
- Prozessorkopplung; Aufbau von Multiprozessorsystemen.

Ausführungsformen:

- freie Erfindung,
- Anlehnung an Industriestandards,
- Implementierung richtiger Busstandards.

Implementierung der Bushardware (Ankopplung, Steuerung) im Bereich der Mikrocontroller:

- manche Mikrocontroller haben eingebaute Busschnittstellen,
- programmseitige Emulation der Signalspiele über universelle E-A-Ports,
- externe Koppel- und Steuerschaltungen.

Achtung – Verhältnis Aufwand zu Nutzen betrachten. Wenn es zuviel wird, dann ggf. auf andere (= auch ohne Zusatzaufwand hinreichend leistungsfähige) Controllerfamilie ausweichen.

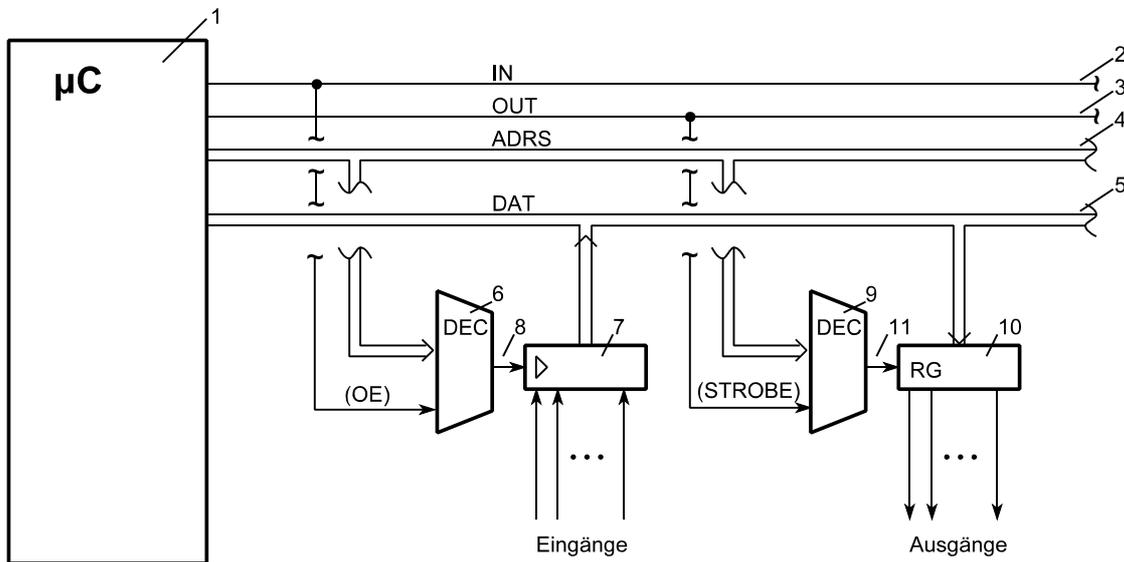
Der typische Bus in Zusammenhang mit Mikrocontrollern

- entspricht typischen Industriestandards,
- wird programmseitig emuliert oder von einem eingebauten Controller unterstützt,
- liegt geschwindigkeitsmäßig in herkömmlicherweise beherrschbarer Größenordnung (Zykluszeiten einige hundert ns),
- ist ein Single-Master-Bus mit dem Mikrocontroller als einzigem Master.

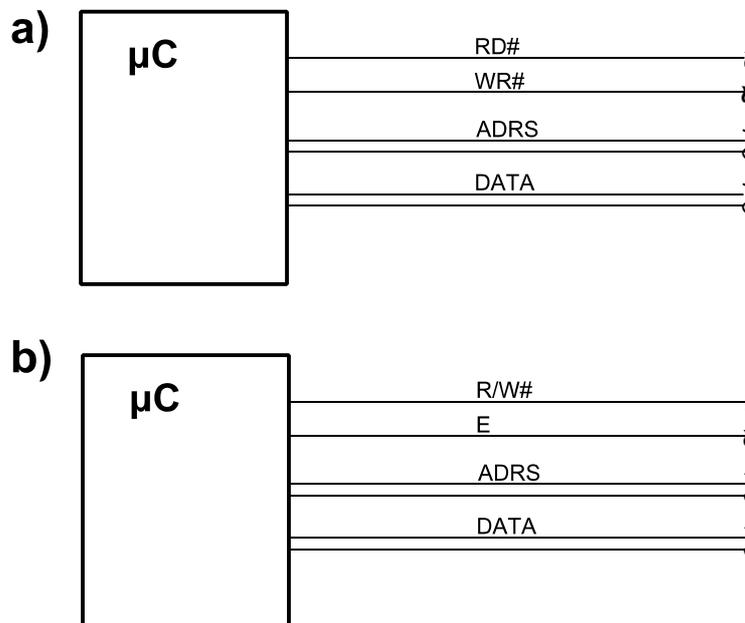
Typische Industriestandards

Sind gegeben durch die Schnittstellen der verbreiteten Buskoppel-, Speicher- und Peripherieschaltkreise sowie durch die herkömmlichen Busstandards (z. B. Multibus, VME, ISA usw.).

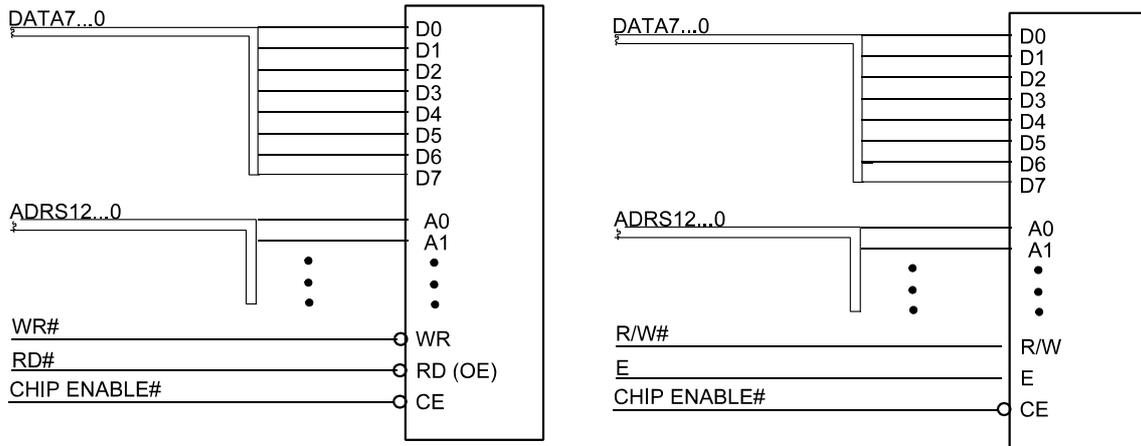
Wichtig: keine allzu exotischen Konzepte. Die Schaltkreise des Massenmarktes müssen ohne viel Umstände passen – und zwar sowohl in funktioneller als auch in elektrischer Hinsicht (Signalpegel, Versorgungsspannungen).



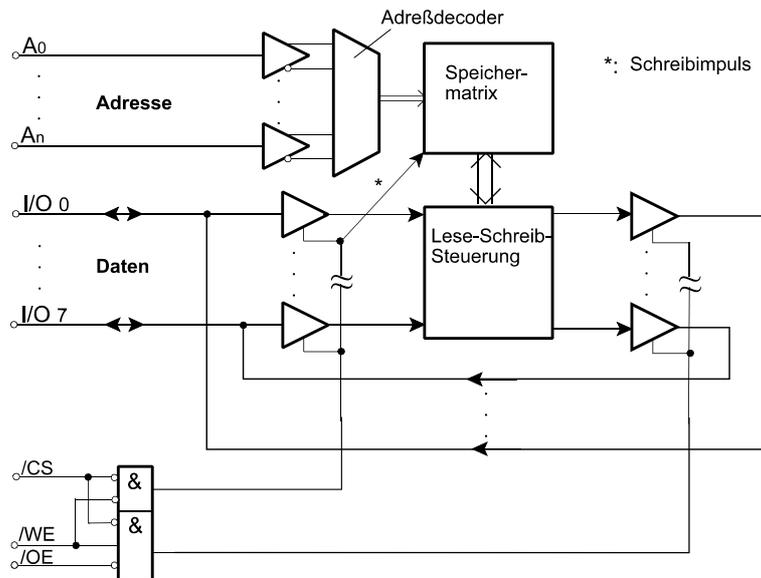
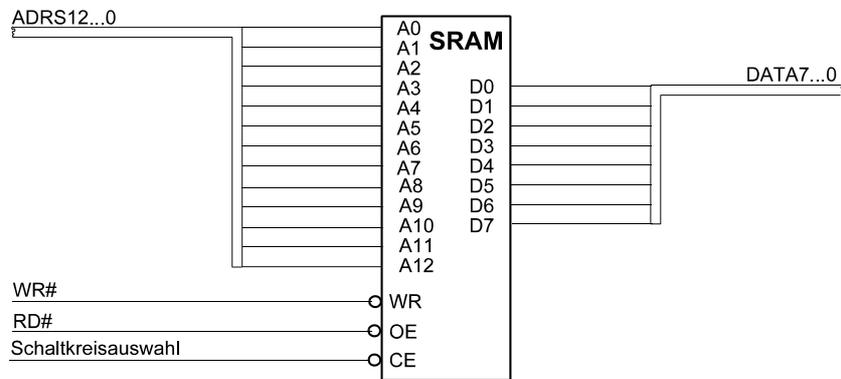
1 - Mikrocontroller; 2 - Eingabesteuersignal (Strobe, Takt); 3 - Ausgabesteuersignal (Output Enable); 4 - Adreßbus; 5 - Datenbus; 6 - Adreßdecoder; 7 - Koppelstufe für Eingabe; 8 - Aufschaltsignal; 9 - Adreßdecoder; 10 - Ausgaberegister; 11 - Übernahmesignal.



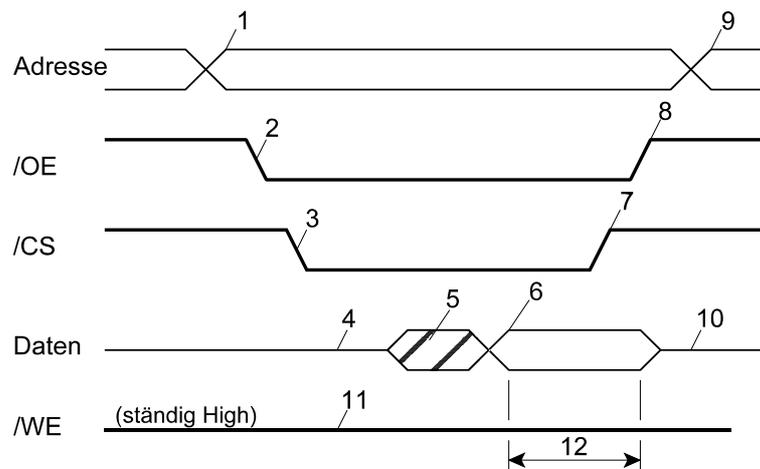
Einfachbussysteme, die Industriestandards sind. Sie unterscheiden sich in der Auslegung der Steuersignale. a) Intel-Prinzip. Je ein Strobe-Signal für Lese- und Schreibzugriffe. Anwendungsbeispiel: Speicherschaltkreise. b) Motorola-Prinzip. Ein Zugriffssteuersignal (R/W#) und ein Strobesignal (E). Anwendungsbeispiel: LCD-Anzeigen mit Busschnittstelle.



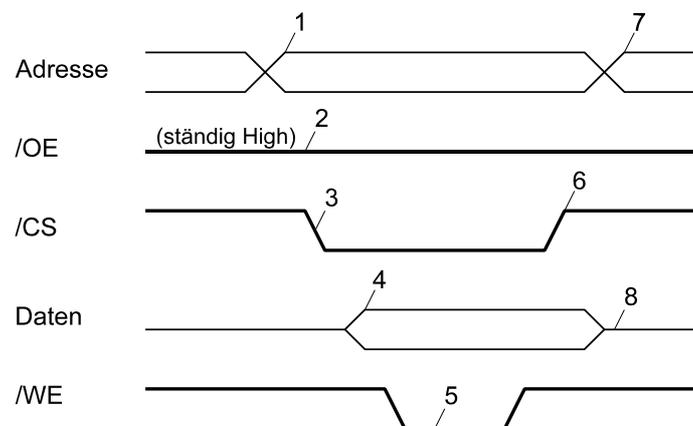
Periphere Einrichtungen mit Busschnittstelle. Links gemäß Intel-, rechts gemäß Motorola-Prinzip



Ein typischer Speicherschaltkreis (SRAM)



Ein Lesezugriff. 1 - die Adresse wird angelegt; 2 - /OE wird aktiviert, um zu signalisieren, daß ein Lesezugriff stattfindet; 3 - der Schaltkreis wird aktiviert; 4 - der Datenausgang ist zunächst hochohmig; 5 - der Datenausgang wird aktiv; 6 - der Inhalt der adressierten Speicherzelle erscheint auf dem Datenausgang (Datenausgang mit gültigen Lesedaten belegt); 7 - der Schaltkreis wird deaktiviert; 8 - /OE wird deaktiviert; 9 - die Adresse verschwindet (Belegung der Adreßeingänge wird geändert); 10 - der Datenausgang wird wieder inaktiv (hochohmig); 11 - der Schreibsteuereingang bleibt während des Lesens inaktiv; 12 - in dieser Zeit müssen die gelesenen Daten vom Datenausgang abgeholt werden.



Ein Schreibzugriff. 1 - die Adresse wird angelegt; 2 - /OE bleibt inaktiv (somit werden die Datenausgänge nicht aktiviert - die Datenanschlüsse bleiben also hochohmig und können von außen belegt werden); 3 - der Schaltkreis wird aktiviert; 4 - die Datenanschlüsse werden mit den Schreibdaten belegt; 5 - ein Schreibimpuls bewirkt das Einschreiben der Daten; 6 - der Schaltkreis wird deaktiviert; 7, 8 - Adresse und Schreibdaten verschwinden (die Schreibdatenbelegung wird abgeschaltet, somit wird der Datenweg wieder hochohmig).

*Zur Wirkungsweise der Steuersignale.* Die folgende Tabelle gibt an, wie die einzelnen Steuersignale wirken. Was als Industriestandard festliegt, ist diese Wirkungsweise und nicht etwa ein Ablaufschema ähnlich den Abbildungen. Wer ein Speichersubsystem zu entwickeln hat, kann sich, gestützt auf die Signalwirkungen und auf die Zeitangaben des Datenblatts, eine Vielfalt von Signalfolgen einfallen lassen.

Funktion	/CS	/OE	/WE	DO	Stromaufnahme ( $I_{CC}$ )
Ruhezustand (Daten halten)	High	X	X	hochohmig	Standby-Strom
Lesen	Low	Low	High	gelesene Bits (Low oder High)	Betriebsstrom
Datenbus freigeschaltet	Low	High	High	hochohmig (kann von außen mit Schreibdaten belegt werden)	Betriebsstrom
Schreiben	Low	X	Low	von außen mit Schreibdaten belegt	Betriebsstrom

X = beliebige Belegung

Zur Wirkungsweise der Steuersignale

*Grundsätze:*

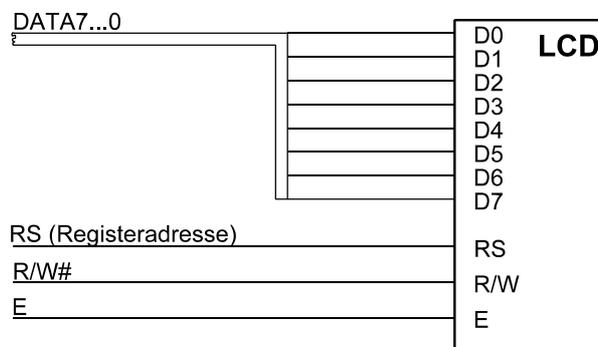
- /CS (andere Bezeichnung: /CE) aktiviert bzw. deaktiviert den gesamten Schaltkreis. Bei inaktivem /CS ist der gesamte Schaltkreis gleichsam abgeschaltet. Alle Tri-State-Ausgänge sind hochohmig, alle anderen Steuersignale sind wirkungslos. Der Schaltkreis tut nichts anderes, als die gespeicherten Daten zu halten. Moderne Typen sparen dabei Strom. Bei aktivem /CS nimmt der Schaltkreis den vollen Betriebsstrom auf und tut gleichsam das, was ihm die anderen Steuersignale sagen.
- /WE bewirkt den Schreibvorgang und deaktiviert gleichzeitig die Datenausgänge (bei aktivem /WE werden die Ausgangstreiber hochohmig<sup>\*)</sup>),
- /OE steuert die Datenausgänge. Ist /OE inaktiv, so sind die Datenausgänge hochohmig<sup>\*)</sup>. Ist /OE (bei aktivem /CS und inaktivem /WE) aktiv, werden die Datenausgänge ebenfalls aktiv; die Lesedaten erscheinen an den Datenanschlüssen.

<sup>\*)</sup>: so daß bidirektionale Anschlüsse von außen belegt werden können.

*Weshalb sind die Steuersignale typischerweise aktiv-Low?*

Das hat zwei Gründe:

- um es dem Entwickler zu erleichtern, bestimmte Signale ständig aktiv zu halten (Verbindung nach Masse genügt),
- Störsicherheit. Vor allem in TTL-Umgebungen sind Low-Signale weniger empfindlich gegen Störungen als High-Signale.



LCD-Anzeige als Beispiel einer nach dem Motorola-Prinzip ausgelegten Einrichtung. Hinweis: Es gibt Typen, die sich wahlweise auf das Intel- oder Motorola-Prinzip einstellen lassen (z. B. durch Jumper).

*RS: Register Select.* Eingang zur Registerauswahl:

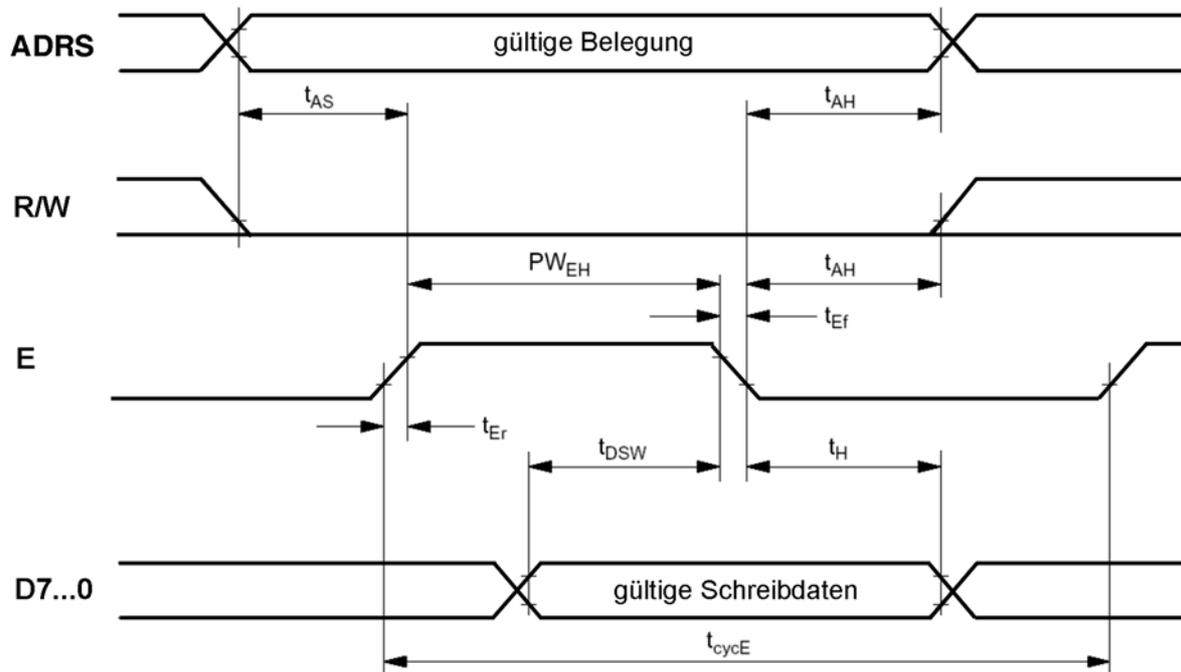
- $RS = 0$ : Kommando- oder Zustandsübertragung,
- $RS = 1$ : Datenübertragung.

*R/W: Read / Write.* Eingang zur Lese-Schreib-Steuerung:

- $R/W = 0$ : Schreibzugriff. Bus wird vom Controller nicht aktiviert (kann also mit zu schreibenden Daten belegt werden).
- $R/W = 1$ : Lesezugriff. Wenn  $E = 1$ , wird Bus vom Controller aktiviert und mit Lesedaten belegt.

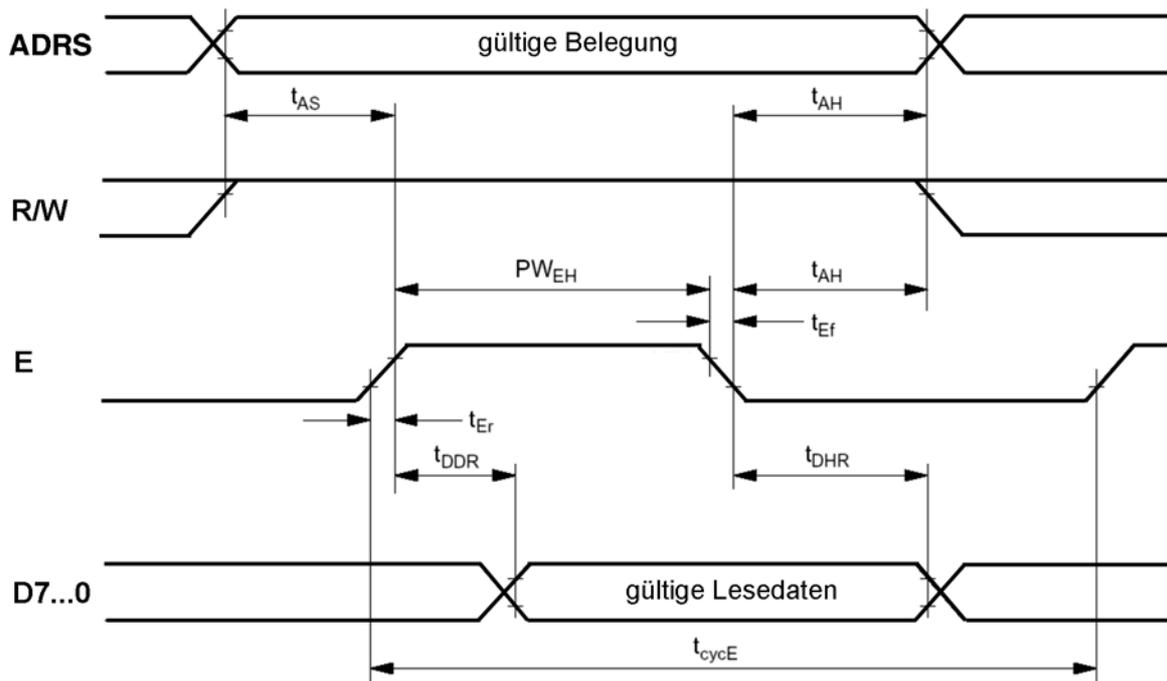
*E: Enable.* Eingang. Allgemeines Erlaubnissignal:

- $E = 0$ : keine Wirkung. Controller inaktiv (nicht ausgewählt). Datenbus frei.
- $E = 1$ : Wirkung gemäß Belegung von RS und R/W.



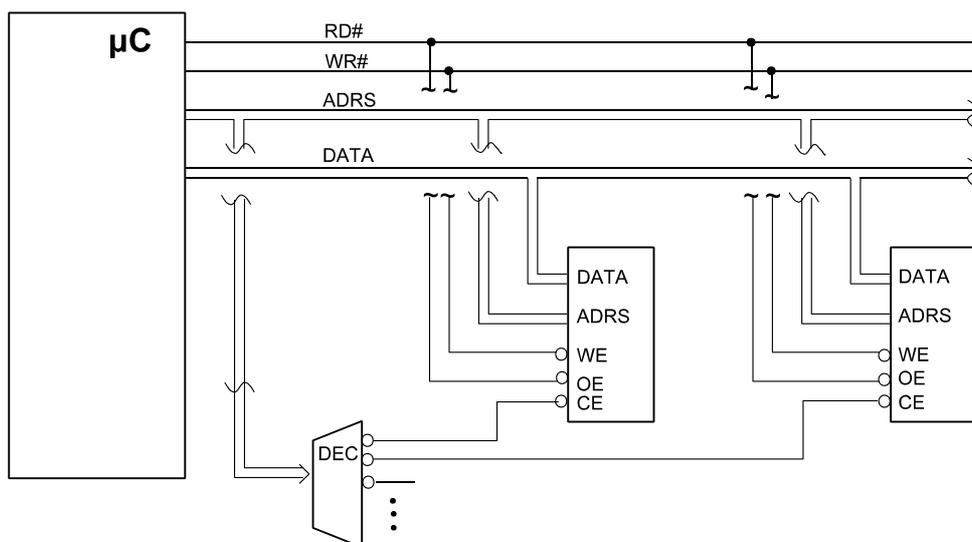
*Der typische Ablauf eines Schreibzugriffs:*

1. RS je nach Zugriff einstellen. R/W auf 0.
2. Schreibdaten auf den Bus legen.
3. von Schritt 1 an müssen wenigstens 140 ns vergangen sein. E-Impuls erzeugen (mindestens 450 ns).
4. Ggf. Schreibdaten vom Bus nehmen und Bus freigeben.
5. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1  $\mu$ s nach Schritt 3 ausgelöst werden

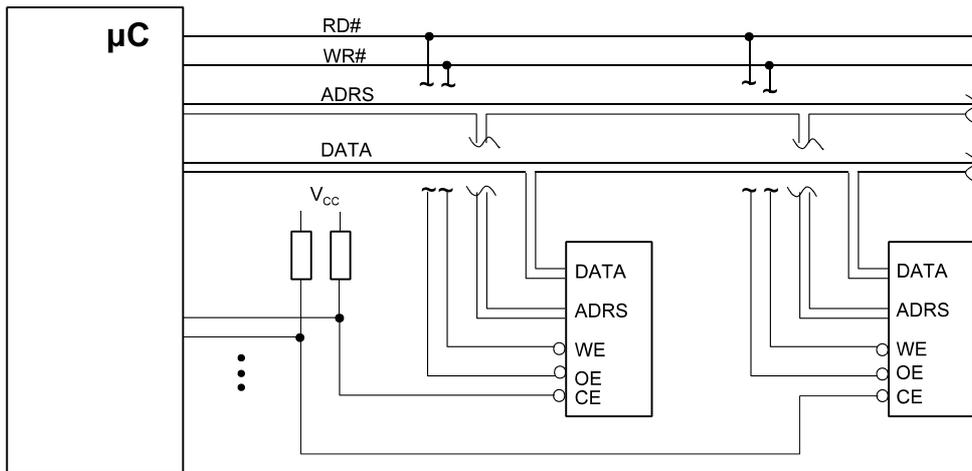


Der typische Ablauf eines Lesezugriffs:

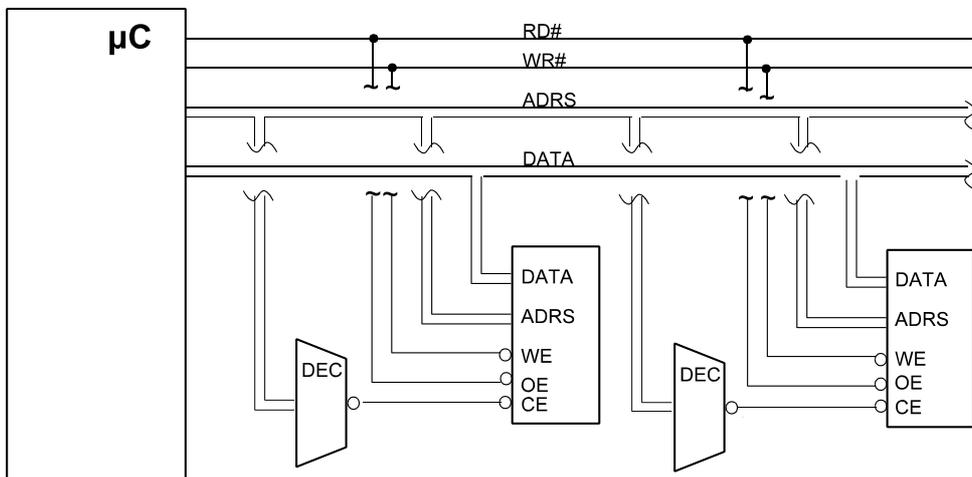
1. ggf. Bus freigeben. Betreffenden Port auf Eingang schalten.
2. RS je nach Zugriff einstellen. R/W auf 1.
3. von Schritt 2 an müssen wenigstens 140 ns vergangen sein. E-Signal auf 1.
4. wenigstens 320 ns abwarten.
5. Lesedaten vom Bus abholen.
6. E-Signal auf 0.
7. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.



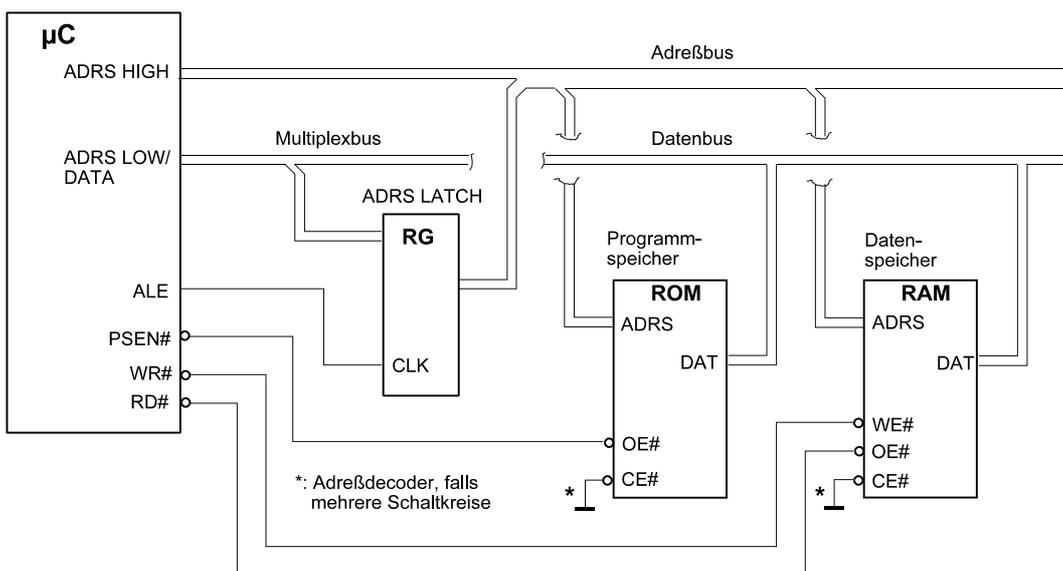
Speicher- oder E-A-Subsystem mit zentraler Adreßdecodierung.



Speicher- oder E-A-Subsystem mit zentraler Einzelauswahl.



Speicher- oder E-A-Subsystem mit lokaler Adreßdecodierung.

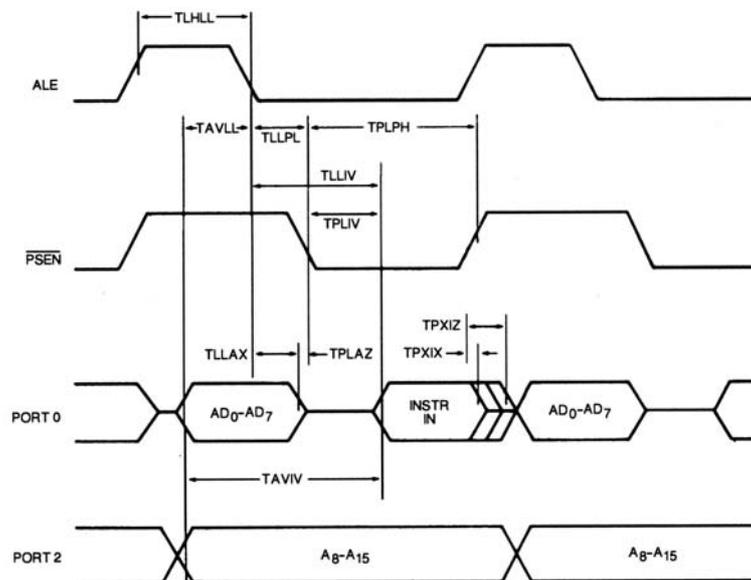


Der Intel-Multiplexbus – ein wirklicher Industriestandard (u. a. 8051, PIC 18x und Atmel AVR).

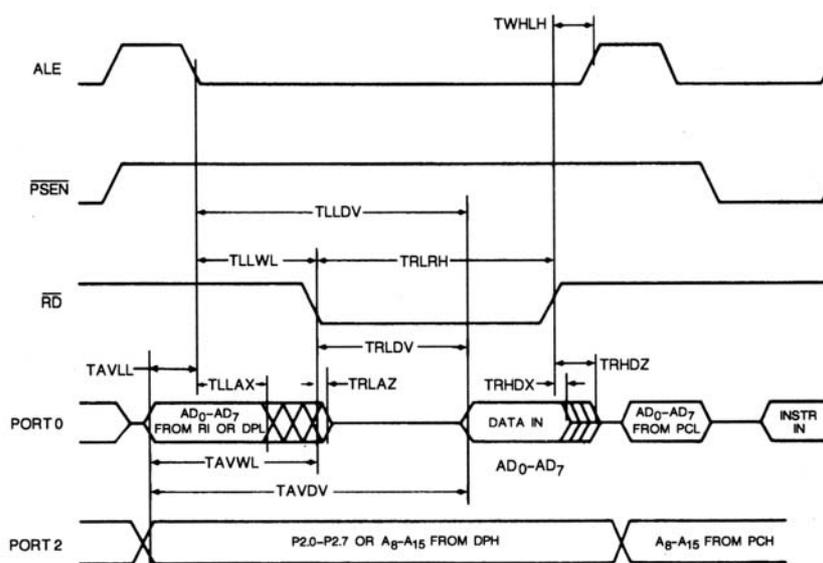
Um Schaltkreisanschlüsse einzusparen, werden Adressen und Daten über gemeinsame Busleitungen übertragen. Die Gültigkeit der Adresse wird mit dem Strobe-Signal ALE gekennzeichnet. ALE veranlaßt die Übernahme der Adresse in ein Haltereister (Adreß-Latch) oder die Adreßauswertung in den angeschlossenen Schaltkreisen. Nach Ausgabe der Adresse wird der Bus zur Datenübertragung benutzt.

Die Steuersignale:

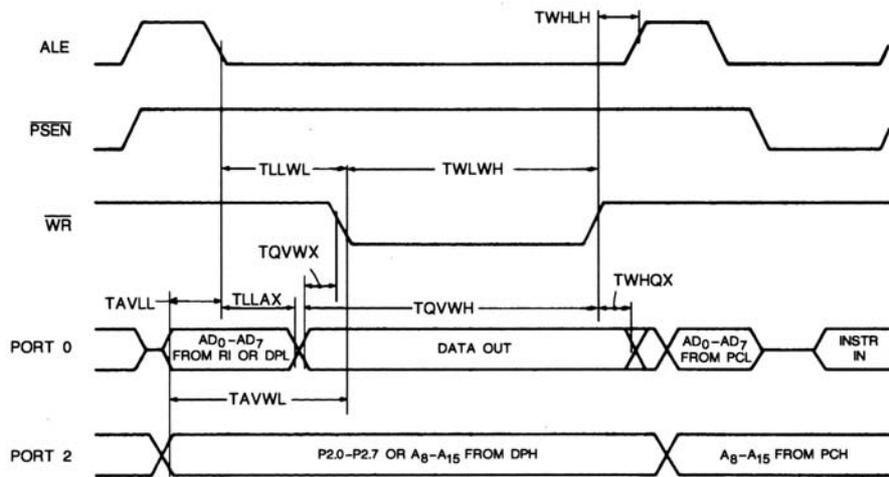
- RD# = Lesezugriff,
- WR# = Schreibzugriff,
- PSEN# = Befehlslesen. PSEN# ist eine Besonderheit z. B. des 8051 (Harvard-Architektur). Andere Typen haben kein PSEN#; sie unterscheiden nicht zwischen Befehls- und Datenzugriffen (v. Neumann-Architektur).



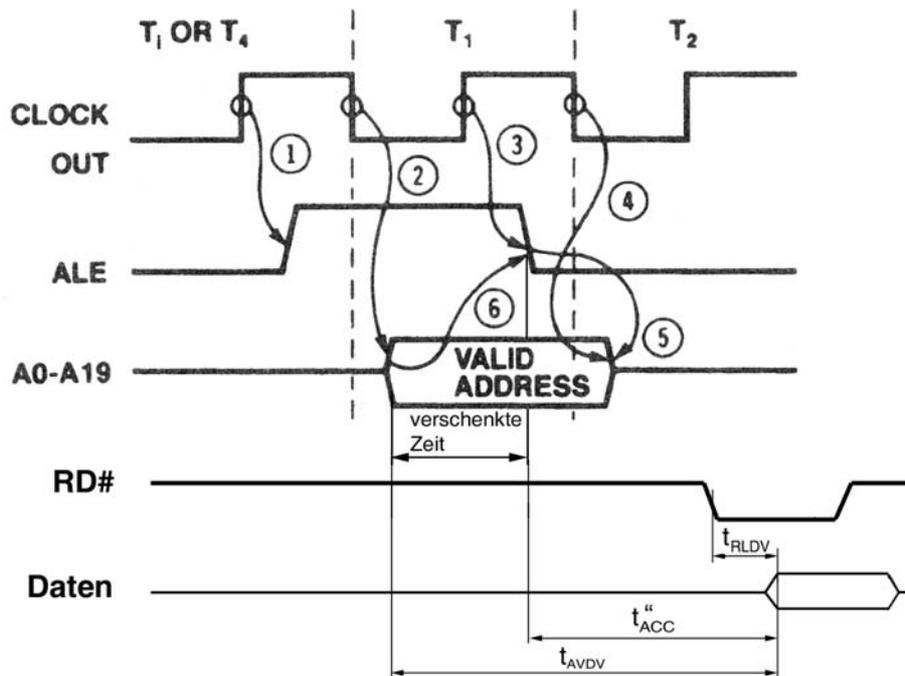
Befehlslesezugriff (8051).



Datenlesezugriff (8051).



Datenschreibzugriff (8051).

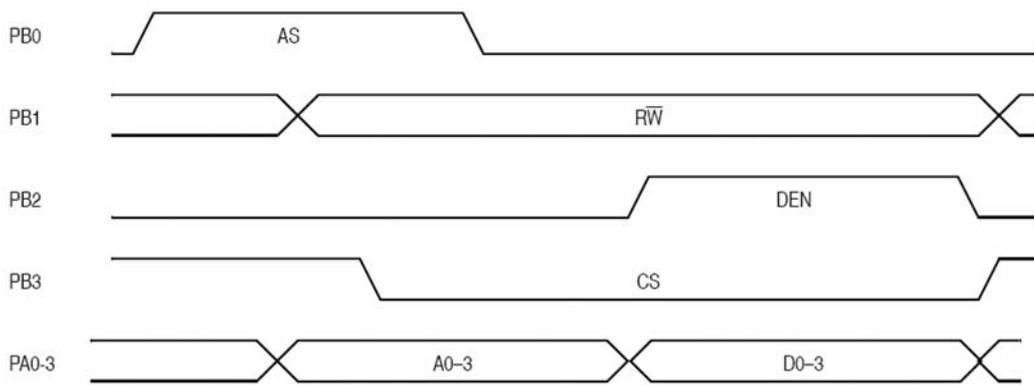


Adreßausgabe und ALE-Signal (nach Intel). Wird ALE aktiv, so liegt noch keine gültige Adresse auf dem Bus. Adreßauswertung: mit der folgenden Low-High-Flanke des Taktes (falls zugänglich) oder mit der High-Low-Flanke von ALE.

Was geschieht, wenn man das Adreßhalterregister mit D-Flipflops aufbaut? – Wird ALE als Takt verwendet, wird keine gültige Adresse übernommen. Negiert man ALE, so verschenkt man Zeit – die verfügbare Zugriffszeit verkürzt sich von  $t_{AVDV}$  auf  $t_{ACC}$ .

Das Adreßhalterregister muß ein Latch-Register sein, damit die Adresse schon dann an dessen Ausgängen erscheint, wenn sie auf dem Bus gültig wird. In dieser Zeit kann somit schon die Speicheradressierung anlaufen.

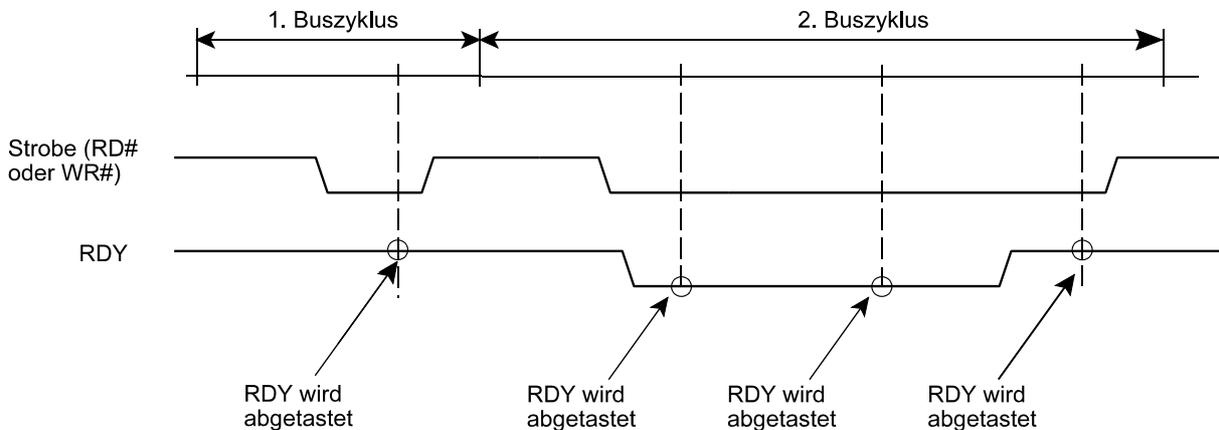
Wenn die angeschlossenen Schaltkreise die Adressierung – durch ALE veranlaßt – intern auswerten, braucht man kein Halterregister. ALE schaltet so, daß man bei aktivem ALE mit der nächsten Low-High-Flanke des Taktsignals die Adreßbelegung auswerten kann.



Adreß-Multiplexing nach dem Motorola-Prinzip. ALE heißt hier AS (Address Strobe). Nach der Adreßausgabe findet der eigentliche Zugriff statt, der von den Signalen R/W und DEN (= E) gesteuert wird.

*Anpassung von Zyklus- und Zugriffszeiten:*

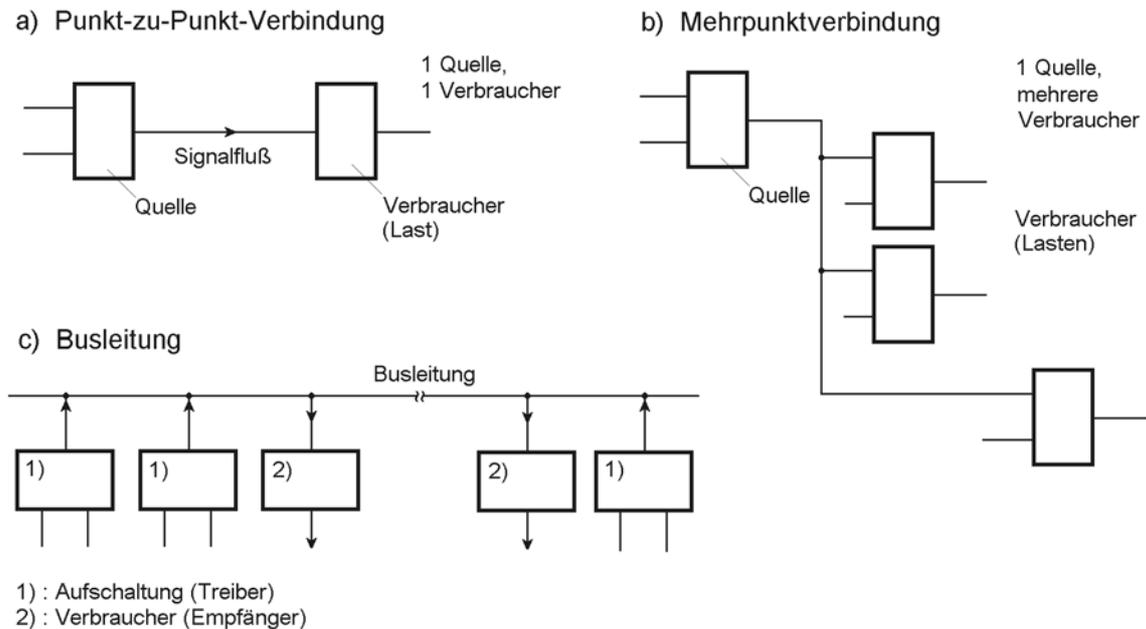
1. by design (= durch entsprechende Bauelementeauswahl und Festlegung der Taktfrequenz) – es muß einfach alles passen. Die einfachste Lösung.
2. programmierbare Wartezustände (pauschal über alles oder selektiv für bestimmte Adreßbereiche).
3. extern steuerbare Wartezustände über zusätzliche Busleitung (RDY, WAIT# o. dergl.). Die Standardlösung der herkömmlichen Mikroprozessoren (Z 80, 8086, ISA-Bus der PCs usw.).
4. Taktfrequenzänderung oder kurzzeitiges Anhalten des Taktes. Eine typische Notlösung. Nachteil: die interne Peripherie ist mitbetroffen (Zeitgeber, Schnittstellen usw.).



Einfügen von Wartezuständen. Zu einem bestimmten Zeitpunkt im Zyklus wird die RDY-Leitung abgetastet. Führt sie High-Signal, wird der Zyklus zu Ende gebracht. Führt sie Low-Signal, wird der Zyklus um ein bestimmtes Zeitintervall (z. B. eine Taktperiode) verlängert; dann wird wieder abgetastet usw.

## Grundlagen der Busverbindungen

Ein wesentliches Kennzeichen von Bussystemen ist die Anschaltung mehrerer Einrichtungen an gemeinsam genutzte Verbindungsleitungen. Dafür haben sich zwei Prinzipien durchgesetzt: (1) das Open-Collector-Prinzip und (2) das Tri-State-Prinzip. Die Namen gehen auf Einzelheiten der Schaltungstechnik zurück.

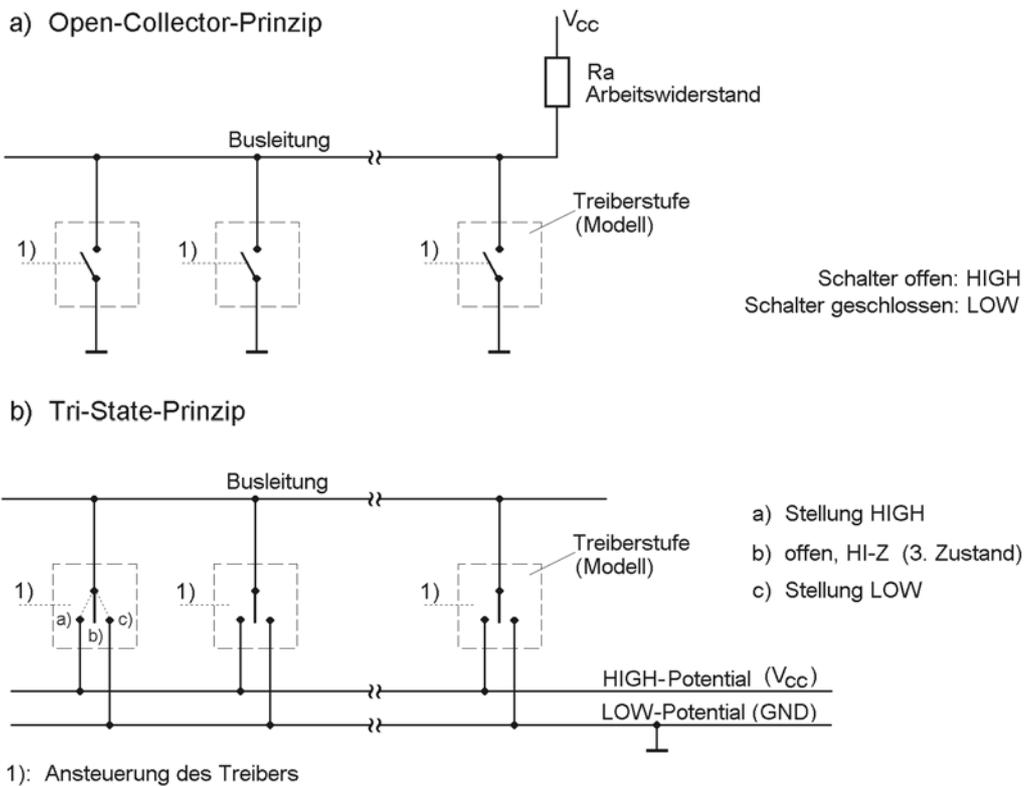


### Einzweckverbindungen und Busleitungen

Die Grundsatzfrage: wie bekommt man es hin, daß jede Einrichtung die Busleitung belegen kann?

Wenn eine Einrichtung ein binäres Signal auf eine Busleitung aufschalten will, so muß sie diese Leitung entweder auf Low- oder auf High-Potential bringen. Hierfür gibt es zwei Möglichkeiten, die wir uns zunächst anhand von Modellvorstellungen klarmachen wollen:

- die Leitung führt im „Freizustand“ (es ist keine der angeschlossenen Einrichtungen aktiv) eines der beiden Potentiale, beispielsweise High. Will eine Einrichtung High aufschalten, so tut sie gar nichts. Will sie hingegen Low aufschalten, so stellt sie über einen Schalter eine Verbindung zum Low-Potential (das bedeutet praktisch: zur Masse) her. Würde man als High unmittelbar die Versorgungsspannung ( $V_{CC}$ ) verwenden, so wäre eine Direktverbindung zwischen High ( $V_{CC}$ ) und Low (Masse, GND) aus elektrischer Sicht ein Kurzschluß. Man legt deshalb die Busleitung über einen Arbeitswiderstand an die Speisespannung, der bei Durchschaltung zu Low den fließenden Strom auf einen vertretbaren Wert begrenzt. Dies ist das Modell des Open-Collector- bzw. Open-Drain-Prinzips.
- die Leitung führt im Freizustand (dann, wenn keine der angeschlossenen Einrichtungen aktiv ist) gar kein Signal, sie hängt also elektrisch gleichsam in der Luft. Jede Einrichtung, die die Leitung erregen kann, hat eine Art Wechselschalter, mit dem die Leitung wahlweise auf High- oder auf Low-Potential geschaltet werden kann. Dies ist das Modell des Tri-State-Prinzips.



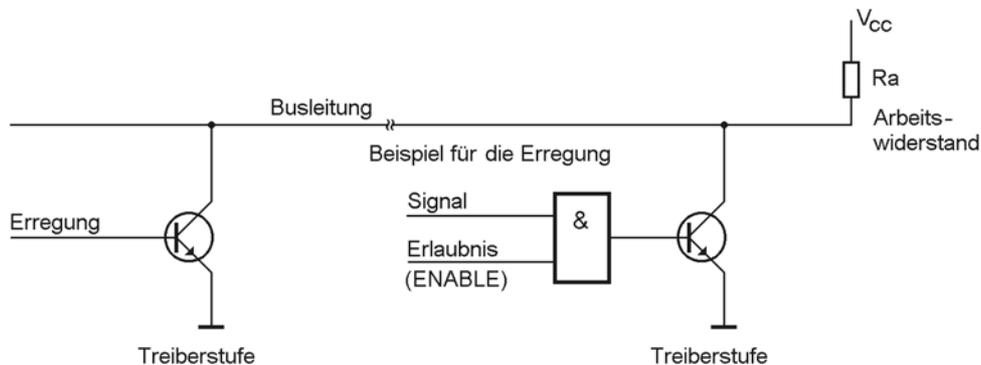
Prinzipien der Busanschaltung (Modellvorstellungen)

**Das Open-Collector- bzw. Open-Drain-Prinzip**

Die Busleitung selbst ist über einen Arbeitswiderstand (Pull-up-Widerstand)  $R_a$  an die Betriebsspannung  $V_{CC}$  angeschlossen. Jeder Bustreiber hat einen Transistor, dessen Kollektor mit der Busleitung verbunden ist (der Name des Prinzips rührt eben daher; der Kollektor des Treiber-Transistors ist ohne weitere Beschaltung auf den Schaltkreisanschluß geführt, liegt also praktisch „offen“). Wird der Transistor aktiviert (durchgesteuert), so zieht er die Busleitung auf Low-Potential.

*Open Collector und Open Drain*

Beide Begriffe bezeichnen das gleiche Prinzip. Open Collector bezieht sich auf bipolare Transistoren, Open Drain auf Feldeffekttransistoren.



Open-Collector-Prinzip

Was geschieht, wenn mehrere Treiber gleichzeitig aktiv sind? - Eigentlich nichts! Auch mehrere aktive

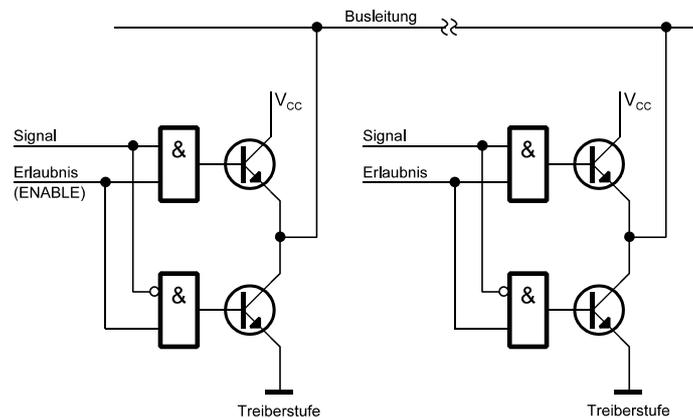
Bustreiber können nicht mehr tun, als die Busleitung auf Low zu ziehen. Bezüglich des Low-Potentials verhält sich also die Busleitung wie eine ODER-Verknüpfung (bei positiver Logik wie ein NOR – es genügt eine Eins an der Basis eines der Transistoren, um die Busleitung auf Null zu ziehen). Deshalb ist auch die Bezeichnung „Wired OR“ (also „verdrahtetes ODER“) üblich.

Beim Open-Collector-Prinzip setzen sich die aktiven Bustreiber gleichsam gegen den Arbeitswiderstand durch („Low gewinnt“). Es schadet deshalb nichts, wenn mehrere Einrichtungen gleichzeitig aufschalten.

**Das Tri-State-Prinzip**

Die Treiberstufe ist eine Anordnung aus zwei Transistoren, die jeweils alternativ angesteuert werden. Es gibt insgesamt drei Zustände (daher der Name):

1. der untere Transistor ist durchgesteuert und zieht somit die Busleitung auf Low-Potential,
2. der obere Transistor ist durchgesteuert und zieht somit die Busleitung auf High-Potential,
3. beide Transistoren sind gesperrt; damit ist die Busleitung gleichsam freigegeben. Aus elektrischer Sicht ist sie „hochohmig“ (dieser Sachverhalt wird in der Literatur oft mit „tristated“, „HI-Z“, „high impedance“ oder „ $Z = \infty$ “ bezeichnet).



Tri-State-Prinzip

Eine Tri-State-Treiberstufe hat zwei Eingänge:

1. das eigentliche (aufzuschaltende) Datensignal,
2. ein Aufschalterlaubnissignal (Enable-Signal). Ist dieses Signal inaktiv, so befindet sich die Stufe ausgangsseitig im hochohmigen Zustand, ist es aktiv, wird die Datensignalbelegung zur Busleitung durchgeschaltet.

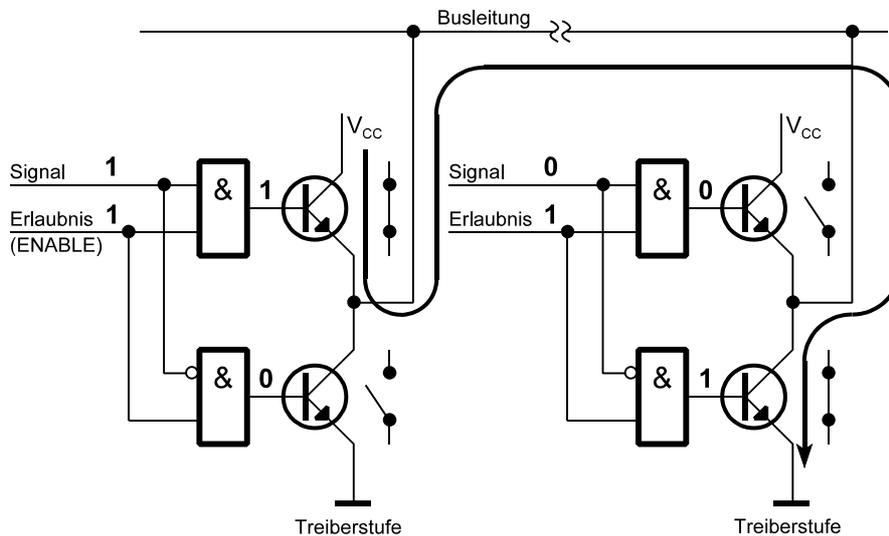
Erlaubnissignal	Datensignal	Busleitung
0 (inaktiv)	0	4 (hochohmig)
0 (inaktiv)	1	4 (hochohmig)
1 (aktiv)	0	0 (folgt dem Datensignal)
1 (aktiv)	1	1 (folgt dem Datensignal)

Zur Wirkungsweise einer Tri-State-Stufe

Was geschieht, wenn mehrere Treiber gleichzeitig aktiv sind? – Das darf einfach nicht vorkommen! Liefern alle aktiven Treiber das gleiche Ausgangspotential (Low oder High), so schadet es nichts, aber

wehe, wenn einer auf Low und einer auf High geschaltet ist: Dieser Betriebsfall bedeutet nämlich einen Kurzschluß von  $V_{CC}$  nach Masse.

*Grundsatz:* Auf eine nach dem Tri-State-Prinzip ausgelegte Busleitung darf zu einer Zeit nur eine einzige Einrichtung aufschalten. Ansonsten gibt es einen sog. Buskonflikt.



So entsteht ein Buskonflikt ...

#### *Bus Contention und Open Collector*

Bei Open-Collector-Bustreibern gibt es keine Buskonflikte. Busleitungen, bei denen es funktionell notwendig ist, daß mehrere Treiber gleichzeitig aktiv sein können (Wired OR), *muß* man deshalb gemäß dem Open-Collector-Prinzip auslegen. Beispiele: Anforderungs- und Fehlersignalleitungen.

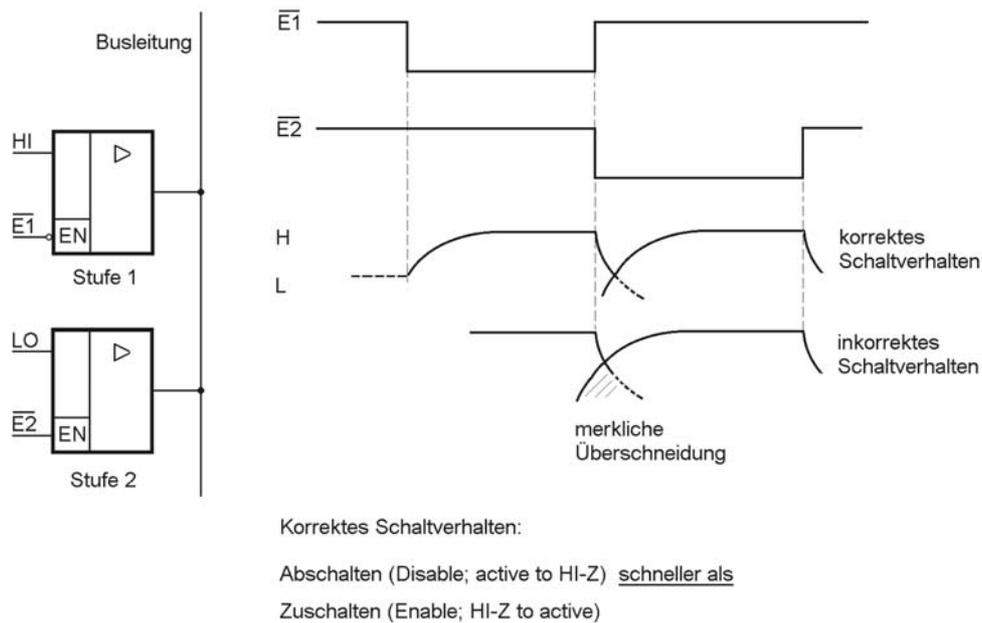
#### *Bus Contention und Tri State*

Es darf jeweils nur eine Tri-State-Stufe den Bus treiben; mehrere Treiber dürfen nie gleichzeitig aktiv sein. Auch dürfen sich die Aktivierungen mehrerer Treiber am Bus nicht überschneiden (es darf nicht sein, daß ein Treiber aufschaltet, während ein anderer noch nicht in den „hochohmigen“ Zustand zurückgeschaltet hat).

Der Konfliktfall ergibt sich, wenn der eine Schaltkreis noch aufgeschaltet hat, der andere Schaltkreis aber schon aufschaltet (Abschalten langsamer als Zuschalten; Abschaltzeit > Aufschaltzeit). Die Dauer des Konfliktfalls entspricht der Differenz von Abschaltzeit und Aufschaltzeit (Konfliktdauer = Abschaltzeit - Aufschaltzeit). Beispiel: Abschaltzeit: 10 ns, Aufschaltzeit: 4 ns, Konfliktdauer = 10 - 4 = 6 ns.

#### *Wann können solche Konflikte vorkommen?*

Sehen wir von groben Fehlern aller Art ab: genau dann, wenn es schnell gehen soll. Beispiel: es wird eine neue Speicheradresse geliefert. Die betrifft aber einen anderen Schaltkreis. Also muß der zur Zeit ausgewählte Speicherschaltkreis deaktiviert und der andere aktiviert werden. Wird das auf die einfachste und schnellste Weise erledigt, so kann es zu Überschneidungen kommen.



Überschneidung beim Umschalten zwischen zwei Busbelegungen. HI, LO - aufzuschaltende Signalbelegungen; E1, E2 - Aufschalterlaubnisignale (Enable-Signale). Ist ein Aufschalterlaubnisignal aktiv (hier: Low), so wird der betreffende Bustreiber gleichsam scharf und schaltet die jeweilige Signalbelegung zur Busleitung durch.

*Sind Buskonflikte auszuhalten?*

Es kommt darauf an. Genauer: es hängt von der Auslegung der Schaltkreise, von den fließenden Strömen und von der Dauer des Konfliktes ab.

Herkömmliche (weniger vornehm: altmodische) Tri-State-Schaltkreise (DIL-Gehäuse, höchstens 8 Ausgänge, Treibfähigkeit um die 20 mA) halten auch einen Dauerkurzschluß aus (die integrierten Widerstände wirken strombegrenzend, und die Erwärmung infolge des starken Stromflusses wirkt widerstandserhöhend, so daß der Strom wiederum begrenzt wird). *Das ist aber reine Vorsorge für den Notfall* (wenn, beispielsweise durch einen Fehler an anderer Stelle, tatsächlich zwei Bustreiber gleichzeitig aktiv werden, gehen die Schaltkreise nicht gleich kaputt). Eine solche „ausfallsichere“ Auslegung ist aber nicht dazu vorgesehen, Bus Contention als sozusagen offiziellen Betriebszustand einzuführen (Stromfluß und übermäßige Erwärmung über längere Zeit verkürzen die Lebensdauer).

Moderne Bustreiber haben hingegen viele Ausgänge (16...36 sind typisch) und eine geradezu extreme Treibfähigkeit (je Ausgang bis zu 60 mA und mehr). Zudem sind sie in kleinen Gehäusen untergebracht. Und hier können sich Überschneidungen wirklich verheerend auswirken - solche Schaltkreise können buchstäblich „den Deckel abheben“.

*Wie rechnen die Hersteller?*

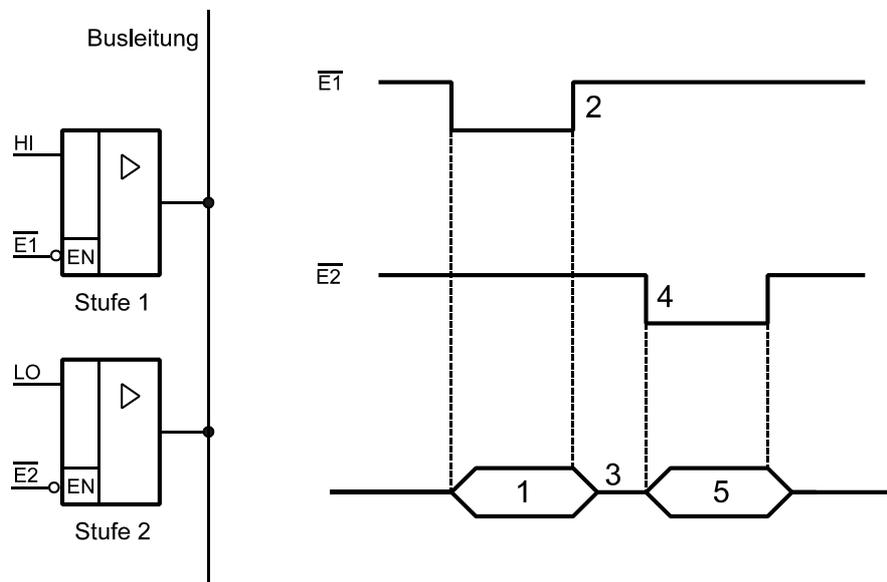
Der Schaltkreis darf nicht zu warm werden; die Kristalltemperatur sollte auf jeden Fall unter 150° C bleiben. Im Konfliktfall ergibt sich ein Temperaturanstieg um 200° je  $\mu\text{s}$ , der sich mit 1 mm je  $\mu\text{s}$  ausbreitet. Die Transistorstrukturen haben aber Abmessungen im Bereich weniger  $\mu\text{m}$ . Daraus ergibt sich, daß Konflikte gerade noch geduldet werden können, sofern sie wesentlich kürzer sind als 1  $\mu\text{s}$  und vergleichsweise selten auftreten. Richtwert: Überschneidungen von ein paar ns (bis hin zu ca. 10...30 ns) schaden nicht, sofern Periodendauer : Konfliktdauer > 10 : 1. Beispiel: Periodendauer = 100 ns, zulässige Konfliktdauer höchstens 10 ns. Beachten Sie aber, daß bei sehr kurzer Periodendauer (= schnellem Bustakt) auch an sich kurzzeitige Konflikte unzulässig sind (da sie zu oft vorkommen). So dürfte bei einer Periodendauer von 10 ns ( $\times$  100 MHz) der Konflikt nicht länger als 1 ns dauern.

*Kann man solche (kurzzeitigen) Konflikte in Kauf nehmen?*

Auch hier kommt es darauf an. Der Schaltkreishersteller sagt nur, daß bei dieser Betriebsweise sein Schaltkreis nicht kaputtgeht; was sonst noch passiert, ist ihm gleichgültig. Konflikte sind aber stets mit Spitzen im Strombedarf verbunden (Kurzschlußströme). Und die wirken sich als Störungen aus (auch wenn sie nur ns dauern ...). Falls diese Störungen innerhalb des Systems nicht schaden und falls die EMV-Anforderungen ohnehin durch andere Maßnahmen erfüllt werden, könnte man mit solchen kurzzeitigen Überschneidungen leben. Andernfalls muß sich der Entwickler schon Mühe geben, auch den kürzesten Konflikt gar nicht erst entstehen zu lassen.

*Konfliktvermeidung*

Das Prinzip: erst abschalten, dann aufschalten. Im Fach-Englisch: Break before Make. Dies kann sowohl durch Schaltungsmaßnahmen als auch durch entsprechende Auslegung der Signalfolgen am Bus (der Busprotokolle) erreicht werden. Manche - auf den ersten Blick merkwürdig anmutende - Schaltfolgen sind nur eingeführt worden, um Buskonflikte zu vermeiden (indem man z. B. einem Schaltkreis fast einen ganzen Taktzyklus lang Gelegenheit gibt, sich vom Bus zu verabschieden).



So sollte ein Tri-State-Bus angesteuert werden. 1 - Stufe 1 ist aufgeschaltet; 2 - Stufe 1 wird deaktiviert; 3 - es wird gewartet, bis Stufe 1 den Bus wirklich freigegeben hat (bis also die Busleitung wieder hochohmig geworden ist); 4 - Stufe 2 wird aktiviert; 5 - Busleitung mit Datenbelegung gemäß Stufe 2.

*Konfliktvermeidung im Schaltkreis*

Manche Schaltkreise gewährleisten von Hause aus, daß es zu keiner Überschneidung kommen kann. Die Voraussetzung dafür: die Ausschaltzeit (Disable Time; Zustandswechsel von aktiv zu hochohmig) muß stets kleiner sein als die Einschaltzeit (Enable Time; von hochohmig zu aktiv). Maßgebliche Kennwerte:

- Abschaltzeit: Turn-off bzw. Disable Time, „Output to High Z“,
- Aufschaltzeit: Turn-on bzw. Enable Time, „Output to Low Z“.

Meist ist aber nicht zu erkennen, daß derartige Vorkehrungen getroffen wurden.

Symbol	Parameter	71024S12	
		Min.	Max.
<b>Read Cycle</b> 			
t <sub>RC</sub>	Read Cycle Time	12	—
t <sub>AA</sub>	Address Access Time	—	12
t <sub>ACS</sub>	Chip Select Access Time	—	12
t <sub>CLZ</sub>	Chip Select to Output in Low-Z <b>2</b>	3	—
t <sub>CHZ</sub>	Chip Deselect to Output in High-Z <b>1</b>	0	6
t <sub>OE</sub>	Output Enable to Output Valid	—	6
t <sub>OLZ</sub>	Output Enable to Output in Low-Z <b>4</b>	0	—
t <sub>OHZ</sub>	Output Disable to Output in High-Z <b>3</b>	0	5

Ein Datenblattauszug (IDT)

Es handelt sich um einen SRAM-Schaltkreis. Uns interessiert das Verhalten des Tri-State-Datenbus. 1 und 3 - Deaktivieren; 2 und 4 - Aktivieren. Schalten wir den Bus über den Chip-Select-Eingang (CS) hochohmig (3), so kann dies bis zu 6 ns dauern (soll heißen: spätestens nach 6 ns ist der Bus frei). Wenn wir aber den Schaltkreis durch Aktivieren von Chip Select auswählen, so belegt er den Bus spätestens nach 3 ns (2). Steuern wir die Datenaufschaltung über Output Enable (OE), so sieht es noch schlimmer aus: beim Deaktivieren (3) kann es bis zu 5 ns dauern, bis der Bus endlich frei wird, beim Aktivieren ist hingegen der Bus sofort (0 ns)<sup>1</sup> belegt. Also dauert das Abschalten sogar länger als das Aufschalten? – Dem ersten Anschein nach ja. Sehen wir aber genauer hin (Pfeil):

- die Abschaltzeiten 1, 3 (High-Z) sind Maximalwerte (soll heißen: es dauert höchstens so lange - kann aber kürzer sein). Maximalwerte gelten für den ungünstigsten Betriebsfall: höchste zulässige Betriebstemperatur und geringste zulässige Betriebsspannung („schlimmer kann es nicht werden“).
- die Aufschaltzeiten 2, 4 (Low-Z) sind Minimalwerte (soll heißen: es dauert mindestens so lange, kann aber länger sein). Minimalwerte gelten für den günstigsten Betriebsfall: geringste zulässige Betriebstemperatur und höchste zulässige Betriebsspannung („auch unter besten Bedingungen dauert es garantiert so lange“).

Nun werden die gegeneinander kämpfenden Schaltkreise in der Praxis typischerweise unter gleichen Bedingungen betrieben (gleiche Betriebstemperatur, gleiche Versorgungsspannung). Eventuelle Konflikte sind deshalb so kurz (1...2 ns oder weniger), daß sie nicht schaden.

*Aber Achtung:* das gilt nur, wenn es sich um Schaltkreise gleichen Typs handelt. Ist vorgesehen, an einem solchen Bus Steckkarten, Speichermoduln usw. an sich x-beliebiger Herkunft zu betreiben, so ist Vorsicht geboten.

#### *Konflikte in Fehler- und Sonderfällen*

Daß zwei Einrichtungen gleichzeitig den Bus belegen, ist ein vergleichsweise häufiger Fehler, der vielfältige Ursachen haben kann:

---

<sup>1</sup>„0 ns“ sind eine physikalische Unmöglichkeit. Eine solche Angabe besagt einfach, daß der Hersteller den Wert nicht nachprüft.

### *Ausfälle der Hardware*

Typische Anzeichen: Häufung von Funktionsstörungen, offensichtlich zu warme Schaltkreise. Bei herkömmlicher Hardware könnte man den Fehler im einzelnen suchen und durch Schaltkreistausch beseitigen. Das Auffinden des tatsächlich schuldigen Schaltkreises ist aber ein Kunststück für sich. Auch wirklich gewiefte Praktiker wissen sich oftmals nicht anders zu helfen als auf Verdacht hin zu tauschen.

### *Inkorrekte Adressen- bzw. Konfigurationseinstellungen*

Besonders fehleranfällig: das manuelle Konfigurieren von Steckkarten. Offensichtlich darf es nicht sein, daß zwei Steckkarten den gleichen Adreßbereich zugewiesen bekommen (oder einen Bereich, der bereits anderweitig belegt ist). Grundlage des Handwerks: eine genaue Buchführung über die tatsächliche Konfiguration.

### *Zu langsame Funktionseinheiten an einem schnellen Bus*

An einen schnellen Bus sind Funktionseinheiten mit langsamen Bustreibern angeschlossen. Auch wenn die zeitweiligen Bukonflikte den Schaltkreisen nicht schaden: wir müssen mit *funktionellen* Fehlern rechnen. Nehmen wir an, ein schneller Prozessor arbeitet mit einer E-A-Einrichtung zusammen, die mit ausgesprochen langsamen Bustreibern bestückt ist. Bei einem Lesezugriff zu dieser Einrichtung sind deren Bustreiber aktiv. Ist dieser Zugriff beendet, startet ein schneller Prozessor sofort (z. B. nach 10...20 ns) den nächsten Zyklus. Haben dann die Bustreiber der zuvor angesprochenen Einrichtung den Bus noch nicht freigegeben, stellt sich auf dem Bus eine fehlerhafte Belegung (als „Mischung aus alt und neu“) ein.

### *Ein- und Ausschalten der Betriebsspannung*

Betriebsspannungen können nicht in wenigen Nanosekunden ein- oder ausgeschaltet werden. Es handelt sich vielmehr um ein geradezu allmähliches Hochlaufen und Abklingen im Bereich von vielen Millisekunden. Das Problem: in diesen Betriebszuständen können auch jene Schaltungen nicht richtig arbeiten, die an sich vorgesehen sind, Buskonflikte zu vermeiden. Und bei modernen Schaltkreisen können schon Konflikte von wenigen ms ausreichen... Abhilfe: (1) entsprechende Schaltungsauslegung; (2) Vorkehrungen in den Schaltkreisen (um die Ausgänge bei zu niedriger Betriebsspannung hochohmig zu halten).

### **Open Collector oder Tri State?**

Will man bei Tri State Überschneidungen sicher vermeiden, muß man zwischen den einzelnen Buszyklen Lücken lassen. Das begrenzt die Gesamt-Datenrate (den Durchsatz) des Bussystems. Ist deshalb Open Collector vorzuziehen?

Gibt es bei Tri State wegen der besagten Lücken Leistungsverlust, so bei Open Collector wegen des Arbeitswiderstandes. Wurden die ersten Bussysteme (der 60er und 70er Jahre) vielfach mit Open-Collector-Schaltkreisen verwirklicht, ging man später – insbesondere aus Geschwindigkeitsgründen – auf Tri State über. Derzeit sind die meisten Buskoppelstufen Tri-State-Bauelemente. Nur Busleitungen, bei denen die Wired-Or-Eigenschaft funktionell notwendig ist, werden mit Open-Collector-Koppelstufen angesteuert. Zwischenzeitlich hat man aber, zumindest im obersten Leistungsbereich, die Nachteile der Open-Collector-Ansteuerung durch Weiterentwicklungen der Schaltkreistechnologie überwunden, so daß, wenn es um höchste Datenraten geht, dieses Prinzip wieder an Bedeutung gewinnt (Beispiele: Rambus (RSL), Prozessorbussysteme (auf GTL-Grundlage)).

	<b>Open Collector</b>	<b>Tri State</b>
externe Beschaltung	1 Widerstand je Leitung	nicht erforderlich
Strombedarf	groß (niederohmige Dimensionierung für hohe Geschwindigkeit)	vergleichsweise geringer
Geschwindigkeit	nur HI-LO-Flanke an sich schnell; LO-HI-Flanke durch Widerstand bestimmt	beide Flanken gleich schnell
Umschaltung, Überlappung (Buskonflikte)	Überlappung unproblematisch; zwischen zwei Stufen darf daher überlappend umgeschaltet werden	Überlappung (Contention) ist unbedingt zu vermeiden; Umschaltung zwischen zwei Stufen daher langsamer

### Open Collector und Tri State im Vergleich

#### Der Bus in Ruhe

Was tun, wenn ein Bus nicht genutzt wird? Open-Collector-Signale verharren auf High-Potential; besondere Probleme gibt es hier nicht. Wie verhält sich aber eine Busleitung, die ausschließlich mit Tri-State-Koppelstufen angesteuert wird? Eine solche Leitung hängt gewissermaßen in der Luft. In der Praxis wird sie das jeweils letzte aktive Potential (High oder Low) eine gewisse Zeit noch halten, dann wird sich nach und nach irgendein Zwischenwert einstellen. Eine solche Belegung bezeichnet man als „schwimmendes“ (floating) Potential. Sie führt zu folgenden Nachteilen:

- nachgeschaltete Schaltkreise (Empfänger am Bus) werden inkorrekt angesteuert. Es liegt keiner der zulässigen Pegel (Low oder High) an. Das führt insbesondere bei CMOS-Eingängen zu einer übermäßigen Stromaufnahme.
- es ist mit EMV-Problemen (Störstrahlung) zu rechnen.

Die Bussysteme einfacher Mikroprozessorkonfigurationen sind höchstens im Mikrosekundenbereich inaktiv (schließlich muß der Prozessor immer wieder über den Bus auf den Speicher zugreifen, um Befehle zu holen). In einem solchen Fall braucht man nichts zu tun.

Bei Prozessoren mit Caches ist eine pausenlose Aktivität keineswegs mehr zu erwarten, und der Systembus eines Multiprozessorsystems kann unter Umständen lange unbenutzt bleiben (wenn alle Prozessoren mit lokal gespeicherten Daten intensiv rechnen). In solchen Fällen ist es notwendig, durch Zusatzbeschaltung den Bus in einen definierten Zustand zu versetzen.

#### Widerstände

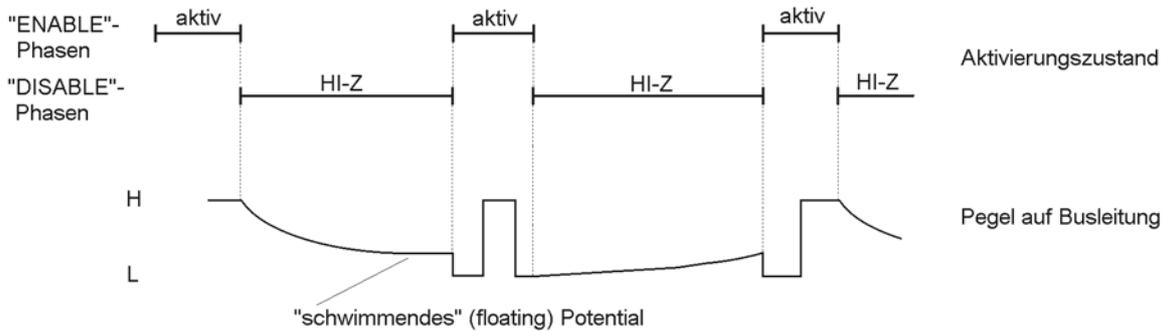
Es liegt nahe, ein inaktives Signal mittels Pull-up- oder Pull-down-Widerstand auf High oder auf Low zu ziehen. Manche Bussysteme erfordern ohnehin Widerstände oder Spannungsteiler als Leitungsabschluß.

#### Widerstände an CMOS-Bussystemen

Man setzt CMOS u. a. deshalb ein, um Strom zu sparen. Durch den Widerstand fließt aber Strom, wenn das Signal auf den jeweils anderen Pegel getrieben wird (z. B. auf Low bei Beschaltung mit Pull-up-Widerstand). Macht man den Widerstand deswegen ausgesprochen hochohmig, so ergeben sich beim Übergang in den hochohmigen Zustand vergleichsweise lange Anstiegszeiten (da die parasitären Kapazitäten nur über den Widerstand umgeladen werden können). Ein Ausweg: das Busprotokoll unterstützt den Widerstand. In einem solchen Fall wird vorgeschrieben, daß vor dem Deaktivieren des Bustreibers der jeweilige Pegel aktiv einzustellen ist: mit Pull-up-Widerständen beschaltete Signale sind

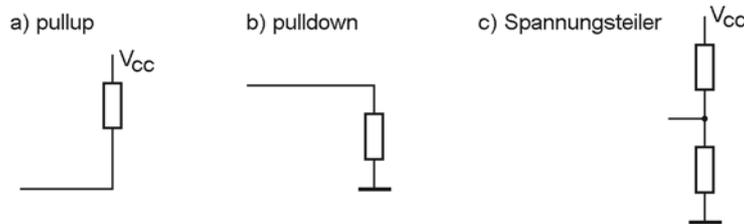
vor dem Deaktivieren auf High zu treiben, mit Pull-down-Widerständen beschaltete Signale auf Low. Der Widerstand hat dann lediglich die Aufgabe, das Signal weiterhin auf dem betreffenden Pegel zu halten. Dieser Trick ermöglicht es, mit hochohmigen Widerständen auszukommen. Beispiel: PCI.

Ein neuerer Trick: die Widerstände werden nur dann angeschaltet, wenn sich das Bussystem in Ruhe befindet. Am arbeitenden Bus sind hingegen keine Widerstände wirksam (somit können auch keine zusätzlichen Ströme fließen). Hierfür gibt es eigens Schaltkreise mit eingebauten Widerständen.

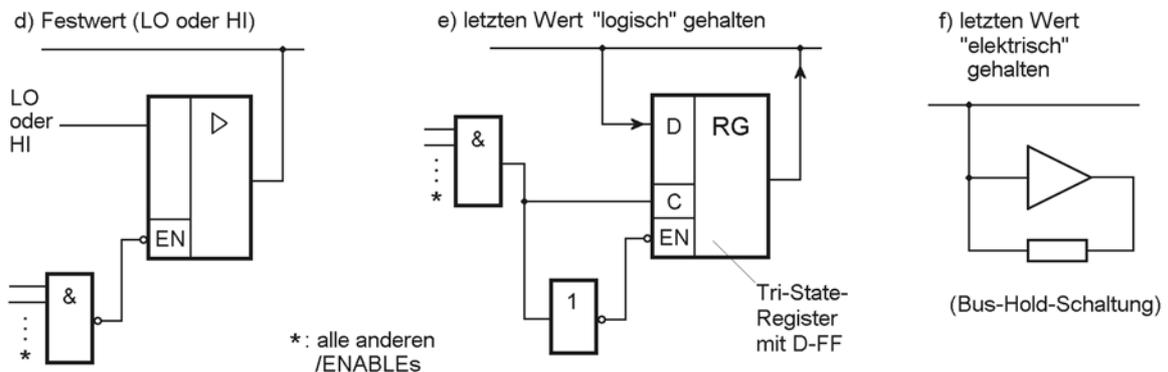


**Auswege:**

1. Widerstände



2. Sonderaufschaltung



**Der Tri-State-Bus in Ruhe: Problem und Lösungen**

*Aktives Halten (Parken)*

Wird der Bus nicht genutzt, so werden die Leitungen mit einem Festwert belegt, oder es wird die jeweils letzte Signalbelegung aktiv gehalten. Beispiel: PCI. Im Ruhezustand muß eine der angeschlossenen Einrichtungen als sog. Park Master wirksam werden.

*Halten über Rückführung (Bus-Hold-Schaltung)*

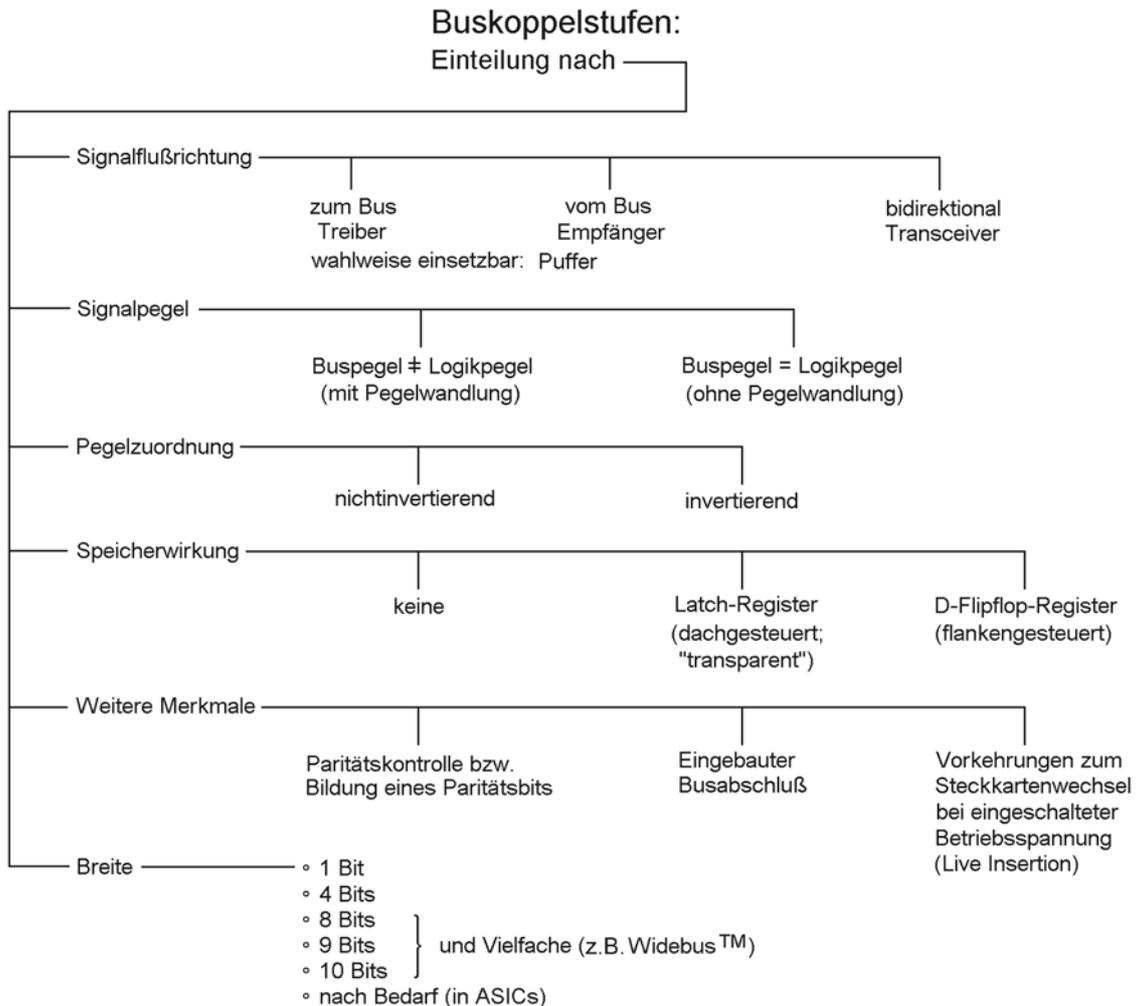
Die jeweils letzte Busbelegung wird gehalten. Viele Schaltkreise haben eingebaute Bushalteschaltungen. Die Vorteile: (1) kein zusätzlicher Strombedarf bei Schaltvorgängen, (2) Busbelegung wird gehalten, ohne daß einschlägige Vorkehrungen in den Busprotokollen erforderlich sind.

*Hinweise:*

1. Das Halten der jeweils letzten Belegung hat - gegenüber der Belegung mit einem Festwert - den Vorteil, daß Schaltvorgänge entfallen (Stromersparnis, verminderte Störstrahlung).
2. Nicht alle Signale können einfach mit dem letzten Wert belegt bleiben. So müssen Steuersignale, die die einzelnen Buszyklen und Busphasen kennzeichnen, im Ruhezustand inaktiv gehalten werden. (Beispiel: PCI. Die Steuersignale werden über Pull-up-Widerstände inaktiv gehalten, nachdem sie zuvor in den inaktiven Zustand getrieben wurden. Jene Signale, deren Belegung bedeutungslos ist, werden hingegen vom Park Master durch aktives Treiben geparkt.)

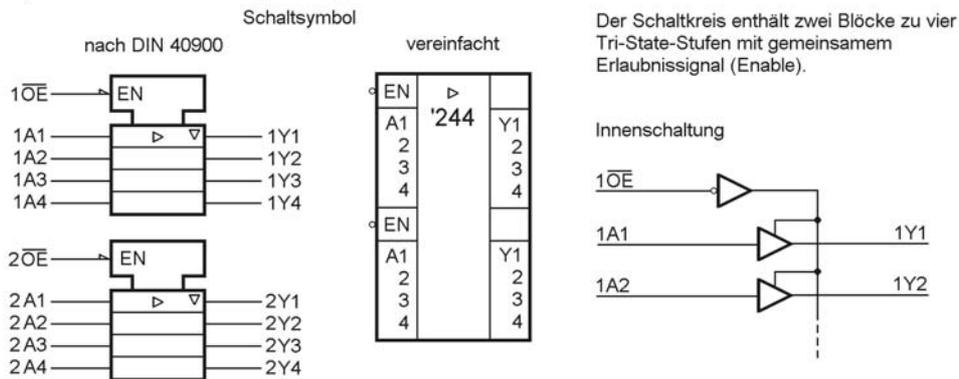
**Buskoppelstufen**

Koppelstufen, die Busleitungen erregen, also Signale zum Bus senden, heißen *Bustreiber* (Bus Driver bzw. Transmitter) oder kurz Treiber. Koppelstufen, die Busleitungen nachgeschaltet sind, also Signale vom Bus empfangen, heißen *Empfänger* (Receiver). Koppelstufen für beide Signalflußrichtungen (die sowohl senden als auch empfangen können) heißen *bidirektionale* Koppelstufen oder *Transceiver* (Transmitter + Receiver). Es gibt auch universell (wahlweise als Treiber oder Empfänger) einsetzbare Koppelstufen..

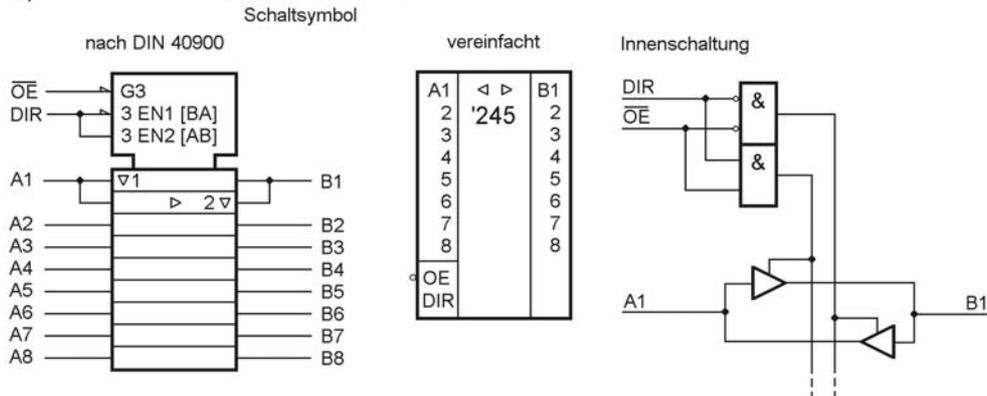


Buskoppelstufen: eine Übersicht

a) Bustreiber (Puffer) '244



b) Transceiver '245

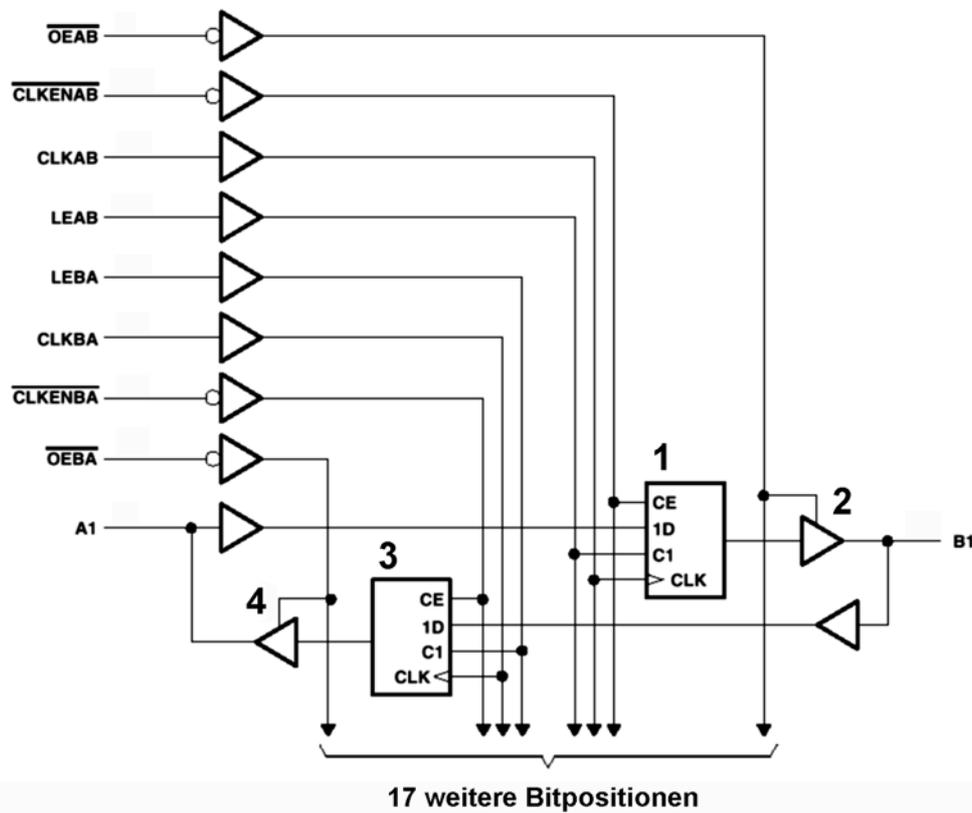


- a) Bustreiber (Puffer). Eine Signalfußrichtung (von A nach Y). A - Dateneingänge; Y - Ausgänge (Busanschlüsse); OE bzw. EN - Erlaubniseingänge (betreffen je 4 Datensignale). Bei aktivem Erlaubnissignal werden die zugehörigen Datensignale zum Bus durchgeschaltet. Ist das Erlaubnissignal inaktiv, sind die zugehörigen Ausgänge (Y) hochohmig.
- b) bidirektionale Koppelstufe (Transceiver). Eine von zwei Signalfußrichtungen wählbar. Beide Seiten A, B, haben Tri-State-Anschlüsse. OE - Erlaubniseingang; DIR - Richtungssteuereingang (Direction Control Input). Zur Wirkungsweise siehe Tabelle 1.3.

Herkömmliche Buskoppelschaltkreise (Beispiele)

DIR	OE	Wirkung
0	0	Richtung von B nach A; A-Ausgänge aktiv
0	1	Richtung von B nach A; A-Ausgänge hochohmig
1	0	Richtung von A nach B; B-Ausgänge aktiv
1	1	Richtung von A nach B; B-Ausgänge hochohmig

Zur Wirkungsweise der Steuersignale des Transceivers '245



Eine Stufe eines Universal Bus Transceivers (74LVT16601; Texas Instruments)

Ein Schaltkreis des angegebenen Typs enthält insgesamt 18 solcher Stufen. Die Steuereingänge sind allen Stufen gemeinsam. 1 - Datenspeicher für Richtung A - B; 2 - Tri-State-Treiber für B-Anschluß; 3 - Datenspeicher für Richtung B - A; 4 - Tri-State-Treiber für A-Anschluß.

Die Funktionsweise wollen wir zunächst anhand der Signalflußrichtung von links nach rechts (von A nach B) beschreiben (A ist Eingang, B Ausgang). OEAB - Output Enable Richtung A-B; CLKENAB - Clock Enable Richtung A-B; CLKAB - Takt Richtung A-B; LEAB - Latch Enable Richtung A-B. Sinngemäß erklären sich die Steuersignale der Gegenrichtung (B-A).

*Aktivieren und Deaktivieren des Ausgangs (B):* mittels OEAB (Low: Ausgang aktiv, High: Ausgang hochohmig).

*Verhindern, daß der Eingang (A) über den Weg B - A beeinflusst wird:* OEBA auf High (deaktiviert Treiber 4).

*Steuerung der Betriebsweise des Datenspeichers 1:* über LEAB:

- LEAB = Low: Datenspeicher wirkt als D-Flipflop. Übernimmt mit der Low-High-Flanke des Taktes CLKAB die Belegung des A-Eingangs (Registerfunktion). Damit CLKAB wirksam werden kann, muß CLKENAB = Low sein (Übernahmesteuerfunktion).
- LEAB = High: Datenspeicher wirkt als Latch oder als einfache Durchreiche.

*Durchreiche (keine Speicherfunktion):* Solange LEAB = High ist, wirkt der Datenspeicher als Durchreiche.

*Latch-Funktion:* Wird LEAB von High auf Low geschaltet, so wirkt der Datenspeicher als Speicherglied und hält die Eingangsbelegung zum Zeitpunkt der High-Low-Flanke von LEAB (Verhalten eines transparenten Latches). In dieser Betriebsart darf CLKAB bei LEAB = Low nicht wirksam werden (sonst: Informationsübernahme mit Low-High-Flanke).

Durch entsprechendes Beschalten der Steuereingänge können viele der gängigen Buskoppelstufen nachgebildet werden. Beispiele:

*1. Verhalten ähnlich '244 (einfache Durchreiche mit Tri-State-Ausgängen)*

OEAB wirkt als OE. LEAB und OEBA fest auf High. Restliche Steuersignale auf beliebige Festwerte.

*2. Verhalten ähnlich '245 (bidirektionaler Treiber)*

OEAB und OEBA wirken als Erlaubnissignale auf der B- bzw. auf der A-Seite. LEAB und OEBA fest auf High. Restliche Steuersignale auf beliebige Festwerte. Wird eine zum '245 kompatible Steuerung mit DIR und OE gewünscht, sind OEAB und OEBA über UND-Gatter gemäß Abb. 1.20b anzusteuern.

*3. Einsatz in SDRAM-Speichermoduln mit Registerpufferung*

Richtung: von A nach B. Dazu OEAB fest auf Low und OEBA fest auf High. Datenübernahme mittels Takt an Takteingang CLKAB. Übernahmesteuerung muß aktiv sein (dazu CLKENAB fest auf Low). Betriebsartensteuerung über LEAB (mit REGE-Signal des Speichermoduls belegt). Restliche Steuersignale auf beliebige Festwerte.

- Registerfunktion: LEAB = Low,
- Pufferfunktion: LEAB = High.

### **Praxisbeispiel: Elementare Speichersubsysteme**

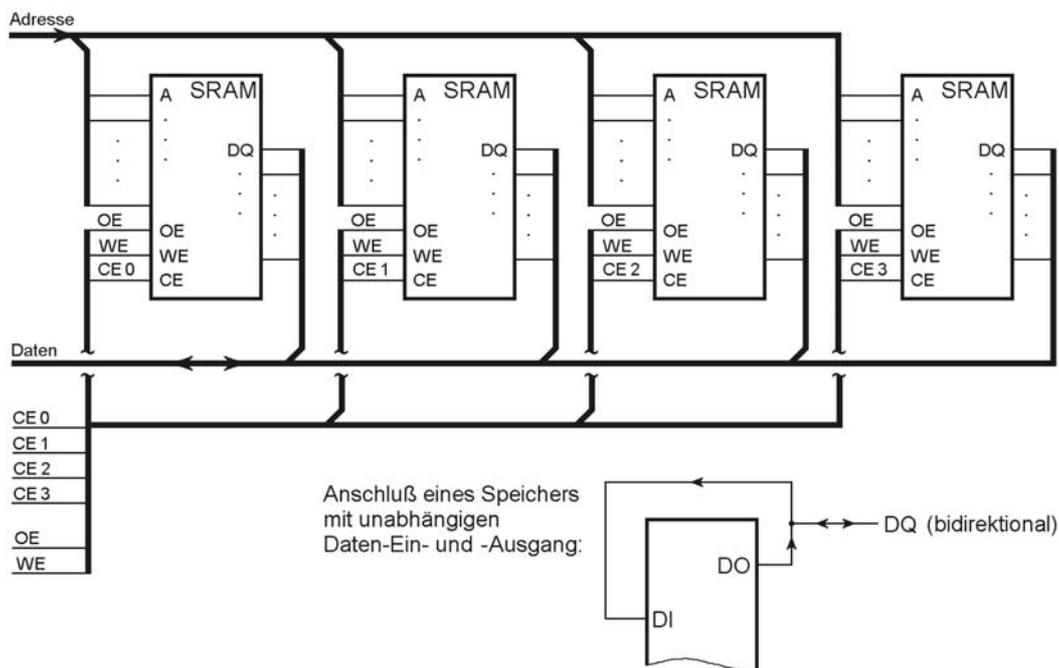
#### *Buskonflikte*

Die Datenwege der typischen Speichersubsysteme sind bidirektionale Busstrukturen, und die einzelnen Schaltkreise haben Tri-State-Ausgangsstufen. In einer solchen Anordnung darf jeweils nur ein Schaltkreis Daten auf den Bus legen. Ist diese Bedingung nicht erfüllt – versuchen also zwei oder mehr Schaltkreise gleichzeitig, den Bus zu treiben – so gibt es Ärger (und zwar genau dann, wenn der eine Ausgang eine Null aufschalten will und der andere eine Eins).

Wirklich harte (länger andauernde) Buskonflikte wollen wir ausschließen (obwohl sie durchaus vorkommen können – und zwar als Folge von Entwurfsfehlern oder von Defekten der Hardware). Es gibt aber zwei Betriebsfälle, in denen mit zeitweiligen Buskonflikten zu rechnen ist: (1) beim Deaktivieren des einen und Aktivieren eines anderen Schaltkreises (Schaltkreisauswahl) und (2) beim Umsteuern zwischen Lesen und Schreiben (Bus Turnaround).

*Überschneidung bei der Schaltkreisauswahl.* Nehmen wir an, wir lesen zunächst Daten aus dem ersten Schaltkreis und adressieren beim darauffolgenden Zugriff den zweiten Schaltkreis<sup>\*)</sup>. Die Datenausgänge des ersten Schaltkreises werden nach dem Deaktivieren zunächst noch einige ns lang den Bus belegen. Wenn während dieser Zeit die Ausgangsstufen des zweiten Schaltkreises aktiv werden, kommt es kurzzeitig zu einem Buskonflikt.

\*) dabei bleibt OE aktiv (Low), während beispielsweise CE0 inaktiv wird und CE1 aktiv (CE0 => High, CE1=> Low).



Einfaches Speichersubsystem mit bidirektionalem Datenbus

*Umsteuerung zwischen Lesen und Schreiben (Bus Turnaround).* Beim Lesen liefert der Speicherschaltkreis die Daten, beim Schreiben der Prozessor bzw. der Steuerschaltkreis. Beim Umschalten zum Schreiben sind die Ausgangsstufen des Speicherschaltkreises zu deaktivieren und jene des Prozessors bzw. Steuerschaltkreises zu aktivieren. Auch hierbei kann es – wenn beides gleichzeitig geschieht – zu Überschneidungen kommen.

*Grundsätzliches zur Abhilfe.* Es treten dann *keine* Buskonflikte im Sinne von Überschneidungen auf, wenn das Abschalten (Deaktivieren) schneller abläuft als das Aufschalten (Aktivieren). Die perfekte Lösung besteht also darin, diese Betriebsweise zu verwirklichen (Fachbegriffe: Disable Time < Enable Time, Break before Make). Das wird oft getan<sup>\*)</sup>, kostet aber Zeit und Schaltungsaufwand. Deshalb entscheidet man sich gelegentlich dafür, keine einschlägigen Vorkehrungen (Schaltungsmaßnahmen, Gestaltung der Signalspiele) zu treffen, sondern sich darauf zu verlassen, daß in der Praxis schlimmstenfalls gelegentliche Überschneidungen im Bereich weniger ns auftreten werden, die auf die Funktion des Speichersubsystems keinen Einfluß haben.

<sup>\*)</sup>: beispielsweise sind die Bussysteme der meisten Mikroprozessoren so ausgelegt, daß, wenn man die Speicherschaltkreise „nach Kochbuch“ anschließt, keine Buskonflikte auftreten.

### *Kapazitive Belastung*

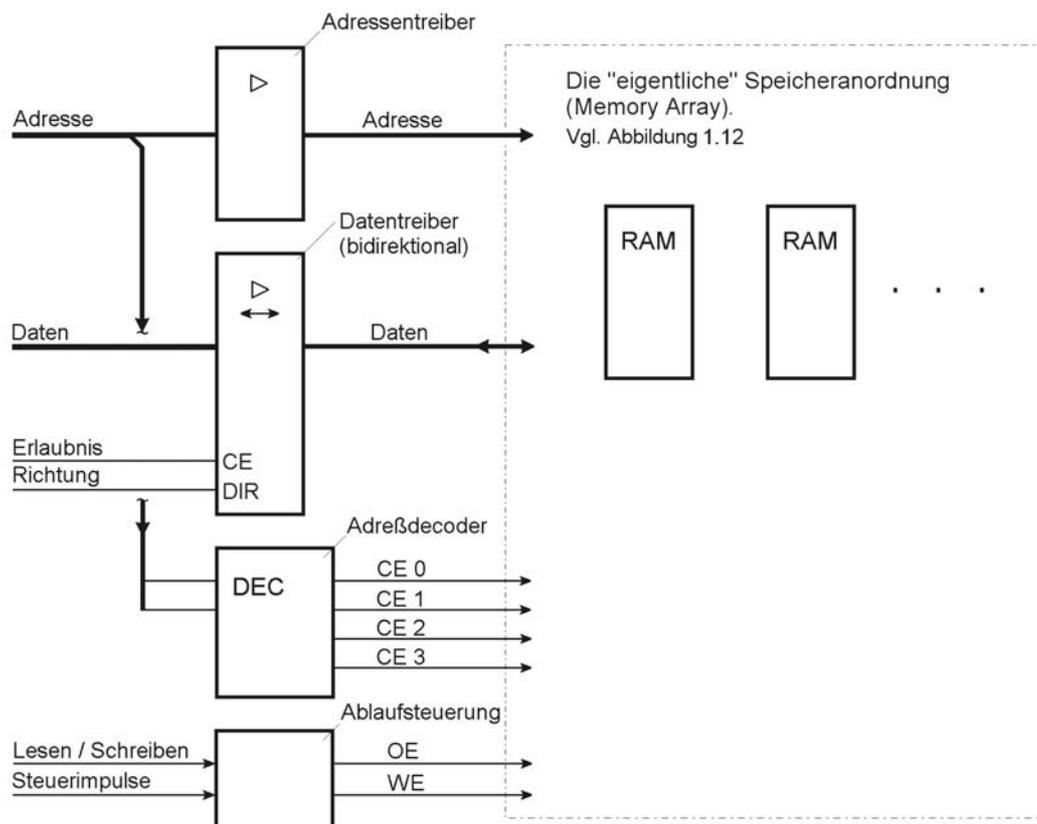
Moderne Speicherschaltkreise beruhen vorwiegend auf CMOS-Technologien. Die Eingänge derartiger Schaltkreise brauchen – bei statischer Ansteuerung – praktisch kaum Eingangsstrom. Das ändert sich aber dann, wenn die Signale schalten. Die Schaltgeschwindigkeit wird maßgeblich von der *kapazitiven Belastung* mitbestimmt.

*Hinweise:*

1. Jeder Schaltkreisanschluß stellt eine bestimmte kapazitive Belastung dar. Hinzu kommen die Leiterzüge. Faustregel: je Anschluß pauschal 5...8 pF, bei bidirektionalen Anschlüssen 10 pF.
2. Zu den Zeitangaben im Datenblatt gehört auch die maximale kapazitive Belastung, unter der die Angaben noch gültig sind. Die Kennwerte moderner Speicherschaltkreise sind typischerweise für eine kapazitive Belastung von 30 pF spezifiziert.
3. Ist die kapazitive Belastung höher als die Angabe im Datenblatt, so kann man den Schaltkreis trotzdem einsetzen – man kann ihn aber nicht mehr mit der maximal zulässigen Geschwindigkeit betreiben. Um wieviel sich die Zeitwerte verlängern, wird durch sog. Derating-Angaben in Form von Kurven, Tabellen oder Formeln beschrieben.

**Grundsätzliches zur Abhilfe.** Es gibt zwei Ansätze:

- viel Strom – das einzige, was hilft, um Kapazitäten schnell umzuladen. Die herkömmliche Lösung: Speichersubsysteme, die mehrere Speicherschaltkreise umfassen und schnell sein sollen (im Sinne geringster Zugriffs- und Zykluszeiten), werden über Bustreiber angesteuert. Das lohnt sich dann, wenn die Durchlaufverzögerung der Bustreiber kürzer als die Verlangsamung infolge der ursprünglichen kapazitiven Belastung (Derating).
- überflüssige kapazitive Lasten abschalten. Eine Lösung auf Grundlage moderner Busschalter-Bauelemente. Die Speicherschaltkreise (bzw. Speichermoduln), auf die jeweils nicht zugegriffen wird, werden gleichsam vom Bus abgetrennt, so daß sich deren kapazitive Belastung nicht auswirkt.



Ein typisches (herkömmliches) Speichersubsystem