

Externe Erweiterung

Weshalb extern erweitern? (Elektrische Anpassung / funktionelle Erweiterung)

Elektrische Probleme an den Schnittstellen zur Außenwelt

Pegel / Signalhub

Flanken / Anstiegszeiten

Zeitlicher Verlauf / Impulsbreiten

Ströme / Treibvermögen

offene Eingänge (z. B. steckbare Funktionseinheiten nicht angeschlossen, Drähte nicht angeklemmt)

Kurzschlüsse

ESD

EMV

Betriebsfall eingeschaltet / ausgeschaltet (Partial Power Down)

Metastabilität

Isolation (galvanische Trennung)

Einzweckanschlüsse

Bussysteme und Interfaces

Grundlagen

Elektrische Auslegungen

Auslegung paralleler Bussysteme

Ringstrukturen

Schieberegisterstrukturen

E-A-Erweiterung

Speichererweiterung

Welche Bauelemente einsetzen?

Wir reden grundsätzlich über vergleichsweise kleine Projekte – alles, was irgendwie kompliziert oder aufwendig aussieht, wollen wir schließlich der Rechenmaschine (vom Mikrocontroller bis zum Industrie-PC) überlassen und dort mit Software erledigen. Bei der Bauelementeauswahl sind heutzutage nicht nur die Kosten von Bedeutung, sondern auch Fragen der praktischen Handhabbarkeit:

- wie aufwendig ist ggf. der rechnergestützte Schaltungsentwurf (Software, Computer, Programmiergeräte usw.)?
- kann man mit einfachen Mitteln einen Laboraufbau zustande bringen?
- kann man die Leiterplatte mit gleichsam hausbackenen Mitteln entwerfen?*)
- läßt sich die Leiterplatte bei jedem x-beliebigen Dienstleister preisgünstig fertigen?
- passen Speisespannung und Signalpegel zum Rest des Systems oder sind in dieser Hinsicht Sonderaufwendungen erforderlich (z. B. zur Pegelwandlung)?

Es geht also nicht nur um die eigentliche Funktion, sondern auch um Gehäusebauformen, um Anschlußabstände und Lötverfahren, um die Ebenenzahl der Leiterplatte, um Speisespannung und Signalpegel (Stichworte: 5-V-Toleranz, 3,3-V-Toleranz) sowie um die anzuwendenden Entwurfs- und Programmierverfahren. Unter diesen Gesichtspunkten wird es sich ab und zu herausstellen, daß mancher – an sich sehr attraktive – Schaltkreis für unsere Zwecke nicht in Frage kommt (das betrifft vor allem ganz moderne Typen, die vor allem zum Einsatz in Mobiltelefonen u. dergl. entwickelt wurden).

*) soll heißen: Placieren und Entflechten nach gängigen Faustregeln und gesundem Menschenverstand.

Herkömmliche Schaltkreise (Off the Shelf)

Naheliegender, wenn es um kleinere Vorhaben geht. Sind aber 20 Stück^{*)} und mehr erforderlich, sollte man sich nach Alternativen umsehen.

Programmierbare Logikschaltkreise

Das Mittel der Wahl für alle auch nur halbwegs komplizierten und/oder umfangreichen Digitalschaltungen (Richtwert: zu erwägen, wenn der Entwurf ansonsten mehr als 5 DIL-Gehäuse (TTL/CMOS) erfordern würde). Der Einstieg erfordert aber einigen Aufwand, nämlich eine passende Entwicklungssoftware, einen hinreichend leistungsfähigen PC und ggf. ein Programmiergerät.

Praxistip:

Eine CPLD- oder FPGA-Familie mit Flash-Programmierung, die in der Anwendungsschaltung programmiert werden kann (In-System Programming (ISP)). Die Programmiervorkehrungen beschränken sich darauf, die zum Programmieren erforderlichen Signale auf geeignete Anschlüsse zu führen (Abb. 1.1). Die Preise der Entwicklungsumgebungen bzw. Starterkits liegen in erschwinglicher Größenordnung (Mindestausstattung: Entwurfssoftware + Download-Kabel). Einige Hersteller bieten kostenlose Entwurfssoftware an. Die einfachste Form der Entwurfseingabe: über Schaltplan. Typische Vorteile:

- viele Projekte kommen mit einem einzigen Schaltkreis aus (vgl. Abb. 1.1),
- man kann immer wieder ändern (= löschen und neu programmieren),
- man kann oft gleichsam naiv entwerfen, ohne sich um die Schaltungsoptimierung, um das Heraussuchen passender 74er oder 4000er Schaltkreise, um die Aufteilung der Funktionen auf die Schaltkreise usw. kümmern zu müssen (mit den DeMorganschen Regeln, mit Karnaugh-Plänen usw. müssen wir uns nicht mehr abgeben -- das erledigt die Entwurfssoftware).

*): Richtwert. Entspricht typischerweise einer mit DIL-Gehäusen voll bestückten Europakarte 100 @160 mm.

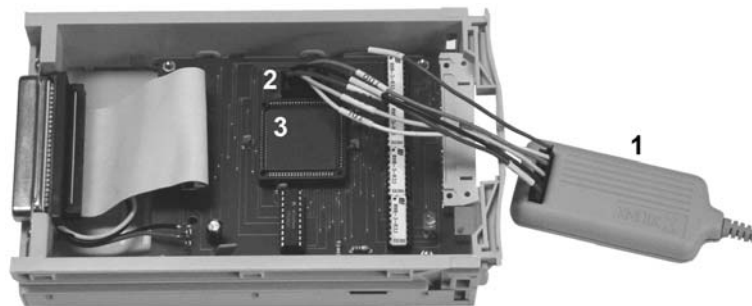


Abb. 1.1 Ein CPLD-Schaltkreis wird programmiert. 1 - Download-Kabel (an die Parallelschnittstelle des Entwicklungs-PCs angeschlossen). 2 - Programmier-Steckverbinder auf der Leiterplatte; 3 - der zu programmierende Schaltkreis

IP: fertige Entwürfe nutzen

IP = Intellectual Property. Es werden Schaltungsentwürfe angeboten, die man in eigene CPLDs und FPGA einbringen kann. Das ist zum einen eine kommerzielle Angelegenheit (kostet richtig Geld). Zum anderen gibt es eine „freie“ bzw. „offene“ Szene. Von derartigen Schaltungen kann man aber keineswegs erwarten, daß sie unter allen Umständen ohne eigenes Zutun auf Anhieb funktionieren – es kommt u. a. darauf an, für welche Schaltkreistypen der Entwurf ursprünglich vorgesehen war, in welcher Form (VHDL, Verilog, Netzliste, Programmierdaten usw.) er übergeben wird und welche weitere Unterstützung (Dokumentation, Testdaten usw.) es gibt.

Hochintegrierte Schaltkreise der Massenfertigung

Ehe man damit beginnt, eine komplizierte Schaltung selbst zu entwerfen, liegt es nahe, sich erst einmal danach umzusehen, was es so gibt. Für unsere Zwecke kommen vor allem E-A-Schaltkreise in Betracht:

- universelle Typen, z. B. der alte, aber immer noch gern verwendete 8255,
- die Super-IO-Schaltkreise der PCs,
- USB-Schnittstellenschaltkreise,
- PCI-Schnittstellenschaltkreise,
- Schaltkreise mit Einfachinterfaces (I²C, SPI o. dergl.),
- Leistungselektronikschaltkreise mit digitaler Schnittstelle (Logikteil + Leistungsteil), z. B. als Relaisstreiber, zum Ansteuern von Schrittmotoren usw.

Mikrocontroller

Kleine Mikrocontroller sind gelegentlich eine Alternative zum programmierbaren Logikschaltkreis oder zum fertig bezogenen E-A-Schaltkreis. Der Mikrocontroller ist zu programmieren. Das kann ein Vorteil sein (mehr Freiheit, eigene Vorstellungen zu verwirklichen, Entlastung des PCs von bestimmten Funktionen), aber auch ein Nachteil (Arbeitszeit, Kosten).

Mikrocontroller oder CPLD?

Ein kleiner CPLD-Schaltkreis mit typischerweise 32...36 Makrozellen (= Flipflops) kostet ungefähr dasselbe wie ein PIC oder AVR oder 8051 usw. mit vergleichbarer Anzahl an Signalanschlüssen. Man hat also gelegentlich die Qual der Wahl. Ein naheliegendes Entscheidungskriterium ist die Reaktionszeit an den zu steuernden Schnittstellen:

- wenn es auf Mikrosekunden oder gar Nanosekunden ankommt: CPLD,
- wenn die Millisekunde kaum eine Rolle spielt: Mikrocontroller.

Hinweise:

1. Der Mikrocontroller bietet mehr fürs Geld, nämlich nahezu unbeschränkte Funktionsvielfalt (freie Programmierbarkeit), die CPLD hingegen nur das, was sich mit den wenigen Flipflops und einigen hundert Gattern realisieren läßt. Zudem haben viele Mikrocontroller eingebaute Schnittstellen (die man andernfalls selbst entwerfen müßte).
2. Die CPLD ist viel schneller. 100 MHz Takt heißt 10 ns Reaktionszeit vom Eingang zum Ausgang. Ein 100-MHz-Mikrocontroller hätte da bestenfalls einen einzigen Befehl ausgeführt -- und der leistet nicht eben viel...
3. Extrem schnelle Mikrocontroller (von 25 MHz an aufwärts) werden angeboten. Sie sind deutlich teurer als die Massenfabrikate (mit typischerweise 5...20 MHz). Trotz der hohen Taktfrequenz sind die Reaktionszeiten in nichttrivialen Anwendungen enttäuschend, ja manchmal geradezu lausig.

Die im folgenden kurz beschriebenen Probleme ergeben sich, weil es nicht um Verbindungen innerhalb einer einzigen Schaltungsanordnung handelt, sondern um Verbindungen zwischen verschiedenen Einrichtungen oder zwischen elektronischer Einrichtung und Außenwelt. Die Vernachlässigung dieser Probleme äußert sich zumeist in überdurchschnittlich vielen Funktionsfehlern, die oft den Charakter von immer wieder auftretenden flüchtigen Fehlern (Aussetzfehlern) haben.

Infolge der Verbindung mit der Außenwelt ist Schnittstellen-Hardware in besonderem Maße fehleranfällig. Eine Unachtsamkeit – z. B. das Stecken eines fehlerhaften Kabels – kann ausreichen, die Hardware zu entschärfen. Wichtige Gefahrenpunkte im Überblick:

- elektrostatische Entladung (ESD) z. B. beim Berühren von Kontaktstiften. Moderne Hardware sollte so etwas wegstecken (EMV-Vorschriften). Trotzdem Vorsicht. Zudem: die Gesetzgebung betrifft nur komplette Geräte und damit nur die ESD-Festigkeit der außen zugänglichen Anschlüsse. Für Busanschlüsse von Steckkarten und für interne Interfaces gilt sie nicht!
- Stecken bei anliegender Speisespannung (Hot Plugging). Serielle Schnittstellen müssen das aushalten, desgleichen alle Interfaces, die nach neueren Standards ausgelegt sind.
- Kurzschlüsse. Kritisch sind vor allem Kurzschlüsse zwischen Speisespannung und Masse. Hiervon können alle Interfaces mit herausgeführter Speisespannung betroffen sein*).
- Kurzschlüsse zwischen Signalleitungen. Die meisten Interfaces dürften diese überleben (bitte nicht allzu sehr darauf verlassen...). *Serielle Schnittstellen* (RS-232) sind grundsätzlich kurzschlußfest.

*) : moderne Interfacestandards schreiben typischerweise den Einsatz selbststrückstellender Sicherungen vor.

Hot Plugging, Hot Docking, Hot Swapping

Die Begriffe werden häufig verwendet, aber nicht immer einheitlich. Es ist wichtig, zwei Merkmale genau zu unterscheiden:

- die Schnittstelle hält das Stecken und Trennen bei anliegender Betriebsspannung aus, geht also – als Hardware – davon nicht kaputt. Das System wird aber typischerweise abstürzen (beim Trennen) oder ein neu angestecktes Gerät nicht erkennen.
- das Stecken und Trennen bei laufendem Betrieb wird auch vom System unterstützt; ein abgetrenntes Gerät wird aus dem System gleichsam verabschiedet, ein neu angestecktes in das System aufgenommen*).

*) : weitere Fachbegriffe mit gleicher Bedeutung: Hot Insertion, Dynamic Attach/Deattach o. dergl.

Hinweise:

1. Fehlt die Hot-Docking-Unterstützung, so wird die Plug&Play-Software nicht automatisch durchlaufen. Somit wird das Hinzufügen oder Abtrennen von Geräten nicht erkannt. Solche Systeme sind, wenn an den betreffenden Schnittstellen herumgestöpselt wird, stets herunterzufahren und neu zu starten.
2. Im PC-Bereich werden manche Schnittstellen traditionell unterstützt – also auch von älteren Systemversionen (z. B. das PC-Card-Interface). Oft funktioniert das Hot Docking aber erst mit zeitgemäßen Betriebssystemen wirklich befriedigend (Beispiel: die Hot-Docking-Unterstützung in Windows ab 2000/XP).
3. Hot Docking in Eigenentwicklungen: von Grund auf kaum durchführbar (Entwicklungsaufwand, Kosten). Behelf: die eigene Lösung auf entsprechende Angebote der einschlägigen Industrie stützen, z. B. auf USB-Interfacewandler oder auf PC-Karten.
4. Viele Embedded Systems sind gleichsam abgeschlossene Anwendungslösungen, in denen niemand wahllos herumstöpselt. Hier ist Hot Docking nicht erforderlich.

Pegel / Signalhub

Signalhub zu niedrig: Transistorschaltstufe, Komparator

Signalhub zu hoch: Spannungsteiler, Begrenzer- oder Klammerschaltung, Komparator

Unterschiedliche Bezugspegel (Massepotentiale): Spannungsteiler, Komparator, rechnende Operationsverstärkerschaltungen, Differenzmeßverstärker (Instrumentation Amplifier), Optokoppler, transformatorische oder kapazitive Kopplung. Für Impulse geringer Wiederholfrequenz auch Relais (halten eingangsseitig was aus).

Gar kein gemeinsamer Bezugspegel: Wechselspannungskopplung (transformatorisch oder kapazitiv), Gleichstromwiederherstellung, Optokoppler

Flanken / Anstiegszeiten

Flanken zu steil: Flankenverschleifung (Kondensatoren, RC-Glieder)

Flanken nicht steil genug: Schmitt-Trigger, Komparator

Zeitlicher Verlauf / Impulsbreiten

Kurze (Stör-) Impulse ausblenden: digitale Signalbewertung, Tießpaßfilter

Kurze Impulse erkennen und auswerten: Impulsdehnung (z. B. mit monostabilem Multivibrator), Fangschaltungen.

Ströme / Treibvermögen

Eingangsströme zu gering: Verstärker

Hohe Ausgangsströme erforderlich: Treiberstufen, Leistungsstufen, Relais.

Zur Treibfähigkeit der E-A-Ports:

- TTL-Baureihe, herkömmliche Controller usw.: Treibvermögen nach Low viel größer als nach High (Standard TTL 16 mA vs. 0,4 mA, Atmel AVR: 20 mA vs. 3 mA).
- neuere Logikbaureihen (z. B. AC / ACT) und Controller: Treibvermögen nach beiden Seiten gleich groß (z. B. 25 mA (AC /ACT) oder 20 mA (Atmel MegaAT)).

Achtung: Gesamtstrom nicht größer als maximaler Strom durch die Masse- und Speisespannungsanschlüsse.

Offene Eingänge

(z. B. steckbare Funktionseinheiten nicht angeschlossen, Drähte nicht angeklemt)

So etwas darf eigentlich nicht vorkommen – jeder Schaltkreiseingang sollte stets mit einem definierten Signalpegel belegt sein (das betrifft vor allem CMOS-Schaltkreise). Wann kann es zu offenen Eingängen kommen?

- wenn der Eingang nicht genutzt wird,
- wenn an ein Interface nichts angeschlossen ist (Kabel abgezogen o. dergl.),
- wenn es sich um Busleitungen handelt und wenn der Bus gerade von keiner der angeschlossenen Einrichtungen getrieben wird.

Abhilfe:

- ungenutzte Eingänge: fest beschalten, so daß sich definierte Pegel ergeben.
- Kabelanschlüsse: Widerstandsbeschaltung (Pull-up, Pull-down, Spannungsteiler).
- Busleitungen: Widerstandsbeschaltung oder Bus parken oder Bushalteschaltkreise vorsehen (halten die letzte Belegung auf dem Bus) .

Hinweise:

1. Pegel ggf. so festlegen, daß der betreffenden Eingang nicht aktiviert wird (z. B. keine Interruptauslösung).
2. Pegel so festlegen, daß der Betriebszustand als Fehler erkannt wird.
3. Manche Schaltkreise (Mikrocontroller, CPLDs usw.) haben an ihren Anschlüssen eigens durch Programmierung schaltbare Pull-up-Widerstände). Ausnutzen!

Kurzschlüsse

Kurzschlüsse zwischen Speisespannung und Masse: Strombegrenzung, Sicherungselemente (z. B. selbstrückstellende Halbleitersicherungen), Erkennung des Betriebsspannungsabfalls (Brownout), softwareseitiger Wiederanlauf (Recovery).

Kurzschlüsse zwischen Signalleitungen: Strombegrenzung, kurzschlußfeste Treiberstufen (z. B. Open Collector statt Gegentakt), diagnostische Rückführungen, um diesen Betriebsfall als Fehler zu erkennen

Schutz gegen elektrostatische Entladung (ESD)

Daß die Hardware elektrostatische Entladungen unbeschadet überlebt, ist eine Grundforderung der EMV-Gesetzgebung. Signalwege, die über Steckverbinder mit berührbaren Kontakten verlaufen, sind besonders gefährdet. Abhilfe: Schutzbeschaltung, Berührungsschutz.

Schutzbeschaltung:

- mit Metalloxid-Varistoren (MOVs),
- Suppressordioden (Transient Voltage Suppressors, TSVs) oder
- Suppressorbauelemente auf Polymerbasis.

Jede Signalleitung wird mit einem solchen Bauelement beschaltet.

Varistoren sind kostengünstig und höher belastbar, schalten aber langsamer. Sie stellen eine vergleichsweise große kapazitive Last dar (von einigen hundert pf bis zu mehreren nF). TSVs schalten in kürzester Zeit (Größenordnung: einige ps... ca. 5 ns). Sie können hohe impulsförmige Ströme aushalten (Größenordnung: < 10... > 100 A). Die Dauerbelastbarkeit ist aber deutlich geringer (einige zehn mA...wenige A). Ist ein Schutz gegen länger dauernde Überlastung gewünscht, müssen Bauelemente vorgeschaltet werden, die den Stromweg trennen (Schmelzsicherung, PolySwitch-Sicherung oder ähnliches). Suppressorbauelemente auf Polymerbasis haben eine sehr geringe Kapazität (unter 1 pF), vertragen aber nur die typischen ESD-Entladungen und keine anderweitig eingekoppelten Störungen.

Schaltkreise ohne Zusatzbeschaltung

Man kann typischerweise mit einer ESD-Festigkeit von 2000 V rechnen. Aber schon beim Gang über einen Teppich können sich bis zu 15 000 V aufbauen...

Es gibt Schnittstellenschaltkreise mit garantierter ESD-Festigkeit. Typische Werte: ± 2 kV, ± 5 kV, ± 10 kV, ± 15 kV (jeweils gemäß dem sog. Human Body Model (HBM) der ESD-Entladung).

In der Massenfabrikation ist man bestrebt, die Anforderungen der EMV-Gesetzgebung mit geringstem Aufwand zu erfüllen (also möglichst ohne Zusatzbeschaltung). Man bewegt sich hierbei oft scharf an der Grenze (Motto: nicht mehr tun, als unbedingt nötig...). Wer wirkliche Qualität anbieten will, sollte aber nicht um jeden Preis sparen (Abb. 1.2).

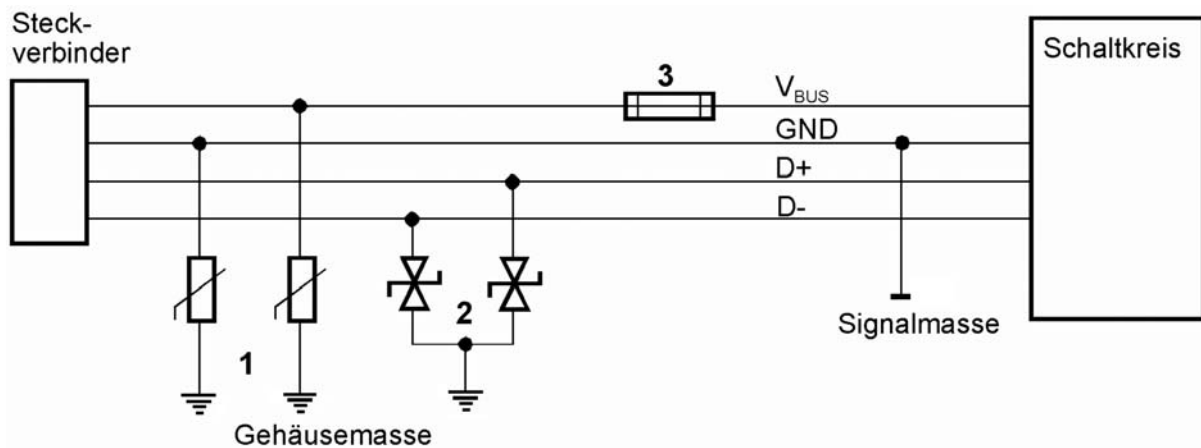


Abb.1.2 Eine typische Schutzbeschaltung am Beispiel eines USB-Anschlusses (nach Littelfuse)

Erklärung:

1 - Schutz der Stromversorgungsleitungen durch Mehrschichtvaristoren. Die vergleichsweise hohe Kapazität ist hier sogar erwünscht (Glättungswirkung). 2 - Schutz der Datenleitungen durch bidirektionale Suppressordioden (USB 2: Suppressorbauelemente auf Polymerbasis – bei der hohen Datenrate kommt es auf geringste kapazitive Belastung an). 3 - selbstrückstellende Halbleitersicherung im Stromweg (Vorkehrung gegen Kurzschluß).

EMV

Filterschaltungen, Abschirmung

ESD + EMV: Überwachung des Betriebszustandes (z. B. Watchdog) und softwareseitiger Wiederanlauf (Recovery).

Betriebsfall eingeschaltet / ausgeschaltet (Partial Power Down)

Wir betrachten miteinander verbundene Einrichtungen (PC und Interfaceadapter, Interfaceadapter und Anwendungshardware). Werden alle Einrichtungen aus der gleichen Spannungsquelle gespeist (gemeinsame Batterie oder gemeinsames Netzteil), gibt es das hier in Rede stehende Problem nicht. Ansonsten ist mit einem besonderen Betriebsfall zu rechnen: nämlich damit, daß die eine von zwei miteinander verbundene Einrichtungen eingeschaltet und die andere ausgeschaltet ist (im Fach-Englisch: Partial Power Down). In einem solchen Fall muß zweierlei gewährleistet sein: (1) die ausgeschaltete Hardware darf den Betrieb der eingeschalteten nicht beeinträchtigen, (2) die ausgeschaltete Hardware darf nicht durch Erregen seitens der eingeschalteten Schaden nehmen. Das Problem ist gelöst, wenn die ausgeschaltete Hardware gegenüber der eingeschalteten eine hohe Impedanz darstellt.

Um festzustellen, ob ein Anschluß eines ausgeschalteten Schaltkreises hoch- oder niederohmig ist, kann man sich die jeweilige Innenschaltung ansehen und annehmen, daß bei ausgeschalteter Betriebsspannung ($V_{CC}=0$) ein Kurzschluß zwischen Masse (GND) und Betriebsspannung (V_{CC}) besteht. Auf dieser Grundlage ist dann zu untersuchen, welche Stromwege sich bilden können, wenn an den Schaltkreisanschluß eine Spannung gelegt wird. Wesentlichen Einfluß hierauf haben die Diodenstrukturen an den Ein- und Ausgängen der Schaltkreise (Abb. 1.3 und 1.4 Tabelle 1.1).

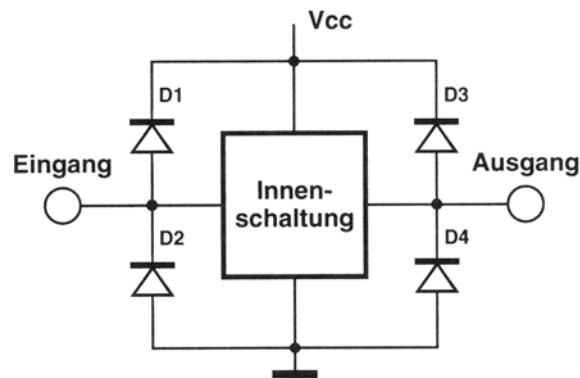


Abb.1.3 Diodenstrukturen an den Ein- und Ausgängen von Schaltkreisen (nach Texas Instruments)

Diode	Erklärung	vorhanden in...	nicht in...
D1	ESD-Schutz (Ableitung positiver Spannungsspitzen)	CMOS	TTL und BiMOS
D2	parasitäre Struktur; bipolare Schaltkreise haben parallel dazu eine absichtlich eingebaute Diode, um negative Spitzen (Unterschwinger) zu kappen	CMOS, TTL, BiMOS	
D3	parasitäre Struktur*); manche CMOS-Typen haben parallel dazu eine absichtlich eingebaute Diode als ESD-Schutz	bipolaren Gegentaktausgängen und CMOS	bipolaren Open-Collector- und Tri-State-Ausgängen
D4	parasitäre Struktur; manche Schaltkreise haben parallel dazu eine absichtlich eingebaute Diode als ESD-Schutz oder zum Kappen negativer Spitzen	CMOS, TTL, BiMOS	

*) in CMOS-Schaltungen bildet der "obere" p-Kanal-FET der Ausgangsstufe eine parasitäre Diode, die in Flußrichtung geschaltet ist, sobald die Ausgangsspannung größer ist als V_{CC}

Tabelle 1.1 Erläuterungen zu den Diodenstrukturen in Abb. 1.3

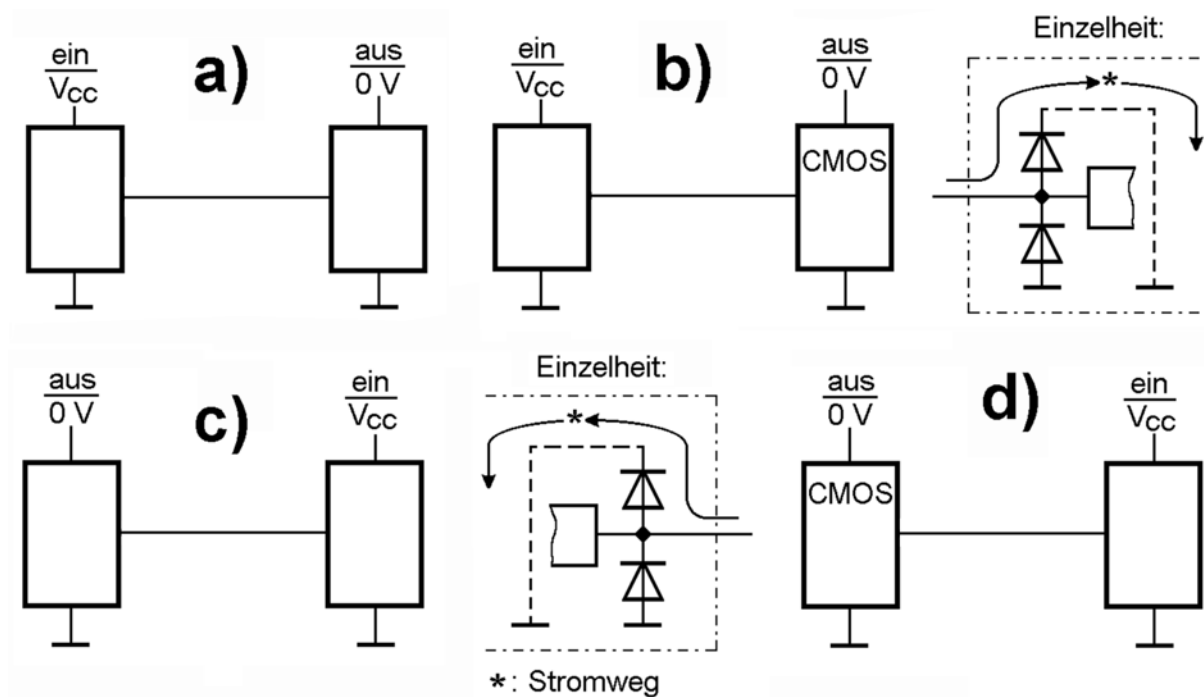


Abb.1.4 Betriebsfälle in teilweise ausgeschalteter Hardware (Partial Power Down)

Erklärung:

- ein aktiver Ausgang wirkt auf einen inaktiven bipolaren Eingang. Unproblematisch, da bipolare Eingangsstufen bei ausgeschalteter Speisespannung hinreichend hochohmig sind.
- ein aktiver Ausgang wirkt auf einen inaktiven CMOS-Eingang. Ein inaktiver CMOS-Eingang ist -- infolge der Dioden D1, D2 (vgl. Abb. 1.3) – grundsätzlich niederohmig. Es ist an sich der ungünstigste Betriebsfall: die gegen V_{CC} geschaltete Diode D1 am Eingang liegt bei $V_{CC} = 0$ in Flußrichtung, und es gibt nichts, was den Strom begrenzt. Der inaktive Schaltkreis kann somit zerstört werden.
- ein inaktiver bipolarer Ausgang wirkt auf einen aktiven Eingang. Bipolare Tri-State-Ausgänge (z. B. jene von Bustreibern) sind im ausgeschalteten Zustand hochohmig, normale bipolare Ausgangsstufen hingegen niederohmig.
- ein inaktiver CMOS-Ausgang wirkt auf einen aktiven Eingang. Ein inaktiver CMOS-Ausgang ist – infolge der Dioden D3, D4 (vgl. Abb. 1.3) – niederohmig. CMOS-Stufen mit nach V_{CC} geschalteten Dioden können also nicht an Schnittstellen verwendet werden, die Hochohmigkeit erfordern.

Schaltungslösungen:

Gemeinsame Spannungsversorgung. Damit werden Betriebsfälle gemäß Abb. 1.4 sicher vermieden. Die Lösung ergibt sich bei entsprechend kompakter Auslegung des gesamten Systems (alles in einem Gehäuse, alles aus einem gemeinsamen Netzteil gespeist).

Bestückung der Schnittstellen mit geeigneten Schaltkreisen. Diese müssen die kritische Betriebsfälle ausschließen oder aushalten. Beispiele: bipolare und BiMOS-Digitalschaltkreise, CMOS-Schaltkreise mit entsprechenden Vorkehrungen (LVT, ALVT, LVC, AVC usw.). bipolare Comparatoren, fehlergeschützte (failure protected) Analogschalter und -multiplexer.

Isolation durch FET-Schalterbauelemente. Solche Schaltkreise gewährleisten der erforderliche Isolation zwischen ein- und ausgeschalteter Hardware (Abb. 1.5).

Geeignete Mikrocontroller und CPLDs. Es gibt Typen, die ausdrücklich so entworfen wurden, daß sie den Betriebsfall aushalten (z. B. durch eingebaute FET-Schalter ähnlich Abb. 1.5).

Zusatzbeschaltung mit Widerständen und Dioden. Solche Lösungen aus der Frühzeit der integrierten Digitalschaltkreise sind auch heutzutage noch zweckmäßig -- vor allem dann, wenn es um nur wenige Anschlüsse geht, die vor dem Betriebsfall Partial Power Down zu schützen sind (Abb. 1.6).

Hinweise:

1. Kann die gemeinsame Speisung aller Funktionseinheiten gewährleistet werden, tritt das Problem nicht auf.
2. Nutzt man die externen Schnittstellen der PCs und setzt man auf der PC-Seite des Interfaceadapters jeweils standardkonforme Schaltkreise ein, löst sich das Problem dort von selbst (Schaltkreise sind für Partial Power Down ausgelegt).
3. Auf der Anwendungsseite tritt das Problem nicht auf, wenn Anwendungshardware und Interfaceadapter eine gemeinsame Stromversorgung haben.

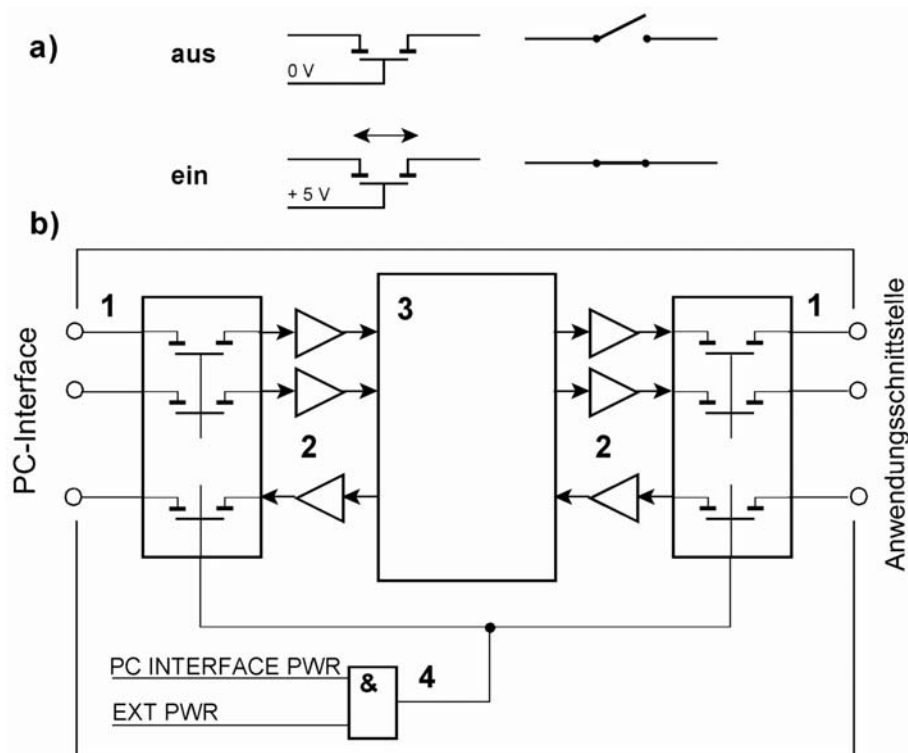


Abb. 1.5 Isolation durch FET-Schalterbauelemente

Erklärung:

- a) Prinzip des FET-Schalters. Ist das Gate nicht geladen (0 V), so ist der FET gesperrt (offener Schalter). Liegt eine positive Spannung am Gate, so ist der FET leitend (geschlossener Schalter; Durchgangswiderstand typisch 5Ω , Verzögerungszeit typisch $< 250 \text{ ps}$). Der Schalter wirkt in beiden Richtungen (bidirektional).

- b) Aufbau eines Interfaceadapters. 1 - FET-Schalterbauelemente; 2 - Treiber- und Empfängerstufen; 3 - die interne Logik; 4 - Steuerung der Schalterbauelemente 1. Hier werden die Schalter geschlossen, wenn beide Versorgungsspannungen (PC- und Anwendungsschnittstelle) anliegen.

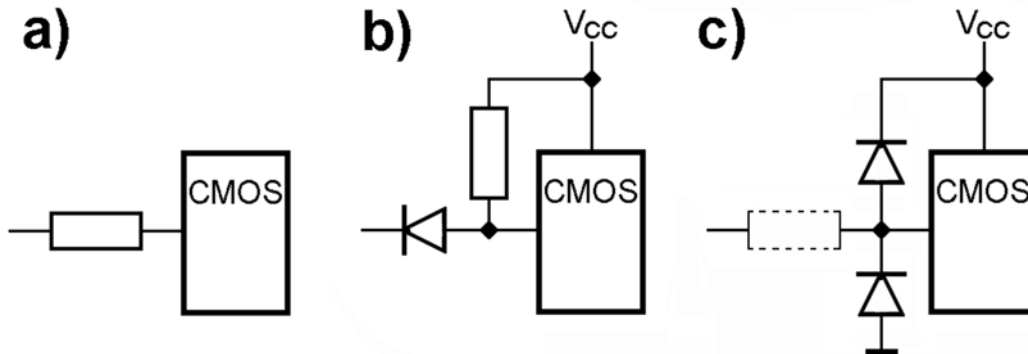


Abb.1.6 Zusatzbeschaltung mit Widerständen und Dioden

Erklärung:

- Strombegrenzung durch Serienwiderstand. Der Widerstand ist so zu dimensionieren, daß bei größter Spannungsdifferenz nur ein unschädlicher Strom (Richtwert: wenige mA) fließen kann.
- Verhindern unerwünschter Stromflüsse durch Einfügen von Sperrdioden. Bei ausgeschalteter Speisespannung liegt die Diode in Sperrichtung, so daß praktisch kein Strom fließen kann. Es ist allerdings ein Pull-up-Widerstand nötig, um den High-Pegel am Eingang zu halten. Somit steigt der Strom bei Low. Zudem wird der Low-Pegel infolge der Flußspannung der Diode angehoben (Schottky-Dioden einsetzen).
- Externe Schutzdioden. Die Schottky-Dioden kappen zu hohe positive oder negative Spannungen auf einen Wert $V_{CC} + 0,5 \text{ V}$ bzw. Massepotential $- 0,5 \text{ V}$. Die Schaltkreise werden damit geschützt; die Kurzschlußströme fließen nun aber durch die externen Dioden, so daß erforderlichenfalls für eine Strombegrenzung zu sorgen ist. Diodenanordnungen gemäß Abb. 1.6c sind marktgängig. Sie halten relativ hohe Ströme aus (Richtwert: 50 mA), so daß Strombegrenzungswiderstände niederohmiger dimensioniert werden können.

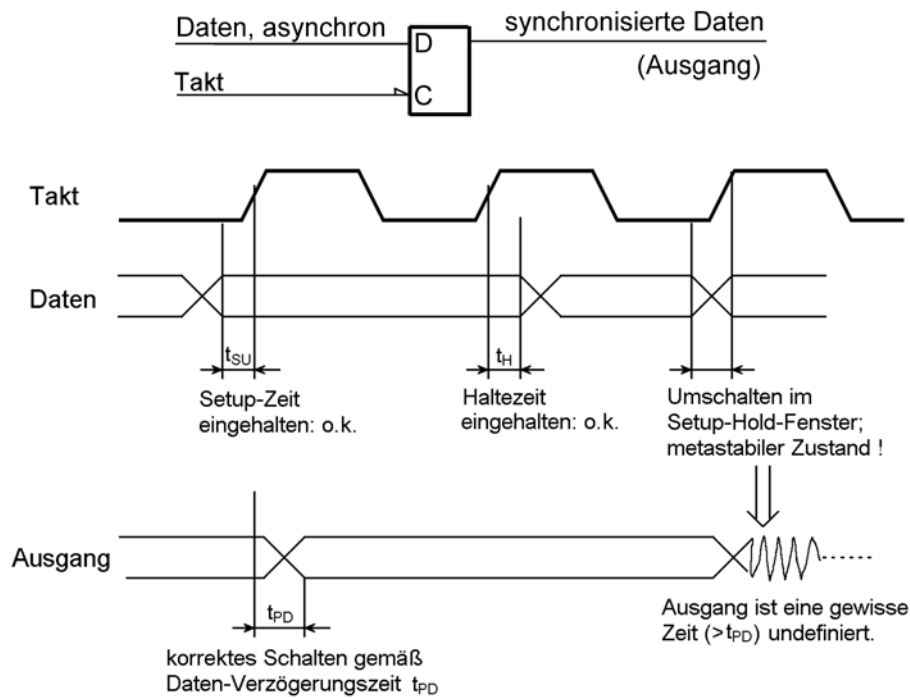
Metastabilität

Moderne Computer sind taktgesteuerte (synchrone) Schaltwerke. Schnittstellen zur Außenwelt hingegen sind naturgemäß immer asynchron (Schalter, Tasten, Sensoren, Interfaces anderer Funktionseinheiten usw. können nun einmal nichts vom Takt im Computer wissen). An diesen Schnittstellen ist also stets das Problem der *Synchronisation* (Eintaktierung) zu lösen. Die naheliegende Lösung: das asynchrone Signal an den Dateneingang eines D-Flipflops legen, der mit dem jeweils passenden Takt angesteuert wird.

Wenn eine Taktflanke mit einer Schaltflanke des asynchronen Signals zusammentrifft, kann es zu sog. metastabilen Zuständen kommen. Kennzeichnend für einen metastabilen Zustand ist, daß das Flipflop Ausgangssignale abgibt, die nicht eindeutig einem der beiden Logikpegel entsprechen (Abb. 1.7). Die Wahrscheinlichkeit hierfür hängt von der Schalthäufigkeit des asynchronen Signals und von der Taktfrequenz ab. Mit Ereignissen, die zu metastabilen Zuständen führen können, ist typischerweise in Abständen von Sekunden...Minuten zu rechnen.

Metastabile Zustände sind grundsätzlich unvermeidbar. Was man lediglich tun kann: von der übernehmenden Taktflanke an eine gewisse Zeit warten – in der Hoffnung, daß dann ein metastabiler Zustand, der sich womöglich eingestellt hat, wieder abgeklungen sein wird. Der Abklingvorgang verläuft gemäß einer Exponentialfunktion. Demgemäß nimmt mit wachsender Abklingzeit die Wahrscheinlichkeit dafür, daß sich das Flipflop immer noch im metastabilen Zustand befindet, exponentiell ab.

Eine einfache Synchronisationsschaltung:



Beispiele von Signalverläufen:



1 - korrekt; 2, 3 - metastabile Zustände

Abb.1.7 Das metastabile Verhalten eines Flipflops

Gegenmaßnahmen:

- Flipflops auswählen, die von Natur aus nur selten in metastabile Zustände übergehen.
- Ausgänge solcher Synchronisations-Flipflops dürfen nach der schaltenden Taktflanke erst dann ausgewertet werden, wenn mögliche metastabile Zustände abgeklungen sind. Je länger die Wartezeit, um so höher die Funktionssicherheit.
- durch Hintereinanderschalten von zwei Flipflops wird die Wartezeit gleichsam automatisch gewährleistet (Doppelsynchronisation; Abb. 1.8). Das Signal wird dadurch um eine weitere Taktperiode verzögert.

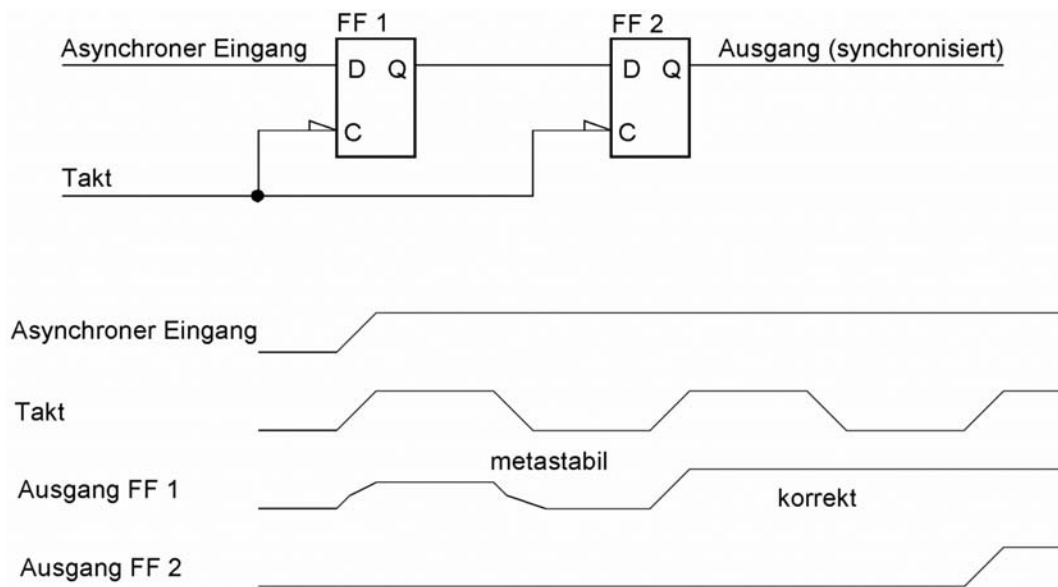


Abb.1.8 Doppelsynchronisation (nach Texas Instruments)

Welche Vorkehrungen gibt es?

- in Off-the-Shelf-Logik (LSI, MSI): nichts; der Entwickler muß aufpassen,
- in CPLDs und FPGAs: desgleichen,
- in Mikrocontrollern: die E-A-Ports enthalten Synchronisationsvorkehrungen (Abb. 1.9). Anschließen asynchroner Signale unbedenklich möglich.
- in Hochleistungsprozessoren: es gibt nur wenige Eingänge, die mit asynchronen Signalen beschaltet werden dürfen (z. B. Reset, Interrupt, NMI). Ansonsten sind die Setup- und Haltezeitpezifikationen einzuhalten.
- im PC: externe Interfaces müssen Synchronisationsvorkehrungen enthalten, interne (ISA, PCI, IDE/ATA usw) hingegen nicht. Die Synchronisation mit den Takt- oder Gültigkeitssignalen des jeweiligen Interfaces ist Sache der angeschlossenen Hardware.

Metastabile Zustände an typischen Schnittstellen:

- alle Schnittstellen mit serieller Datenübertragung (RS-232, USB, Ethernet usw.): kein Problem. Wird in den Steuerschaltkreisen erledigt.
- alle Schnittstellen mit paralleler Übertragung (Bussysteme im PC, IDE/ATA, parallele Schnittstelle, IEEE-488 usw.): aufpassen. Die Setup- und Haltezeitvorgaben der Datenbusbelegung in Bezug auf die jeweiligen Takt-, Strobe- oder Handshake-Signale sind einzuhalten. Einzugebende Daten sind hierzu ggf. in Halteregeistern zwischenzupuffern. Hierzu D-Flipflops nehmen, keine Latches.

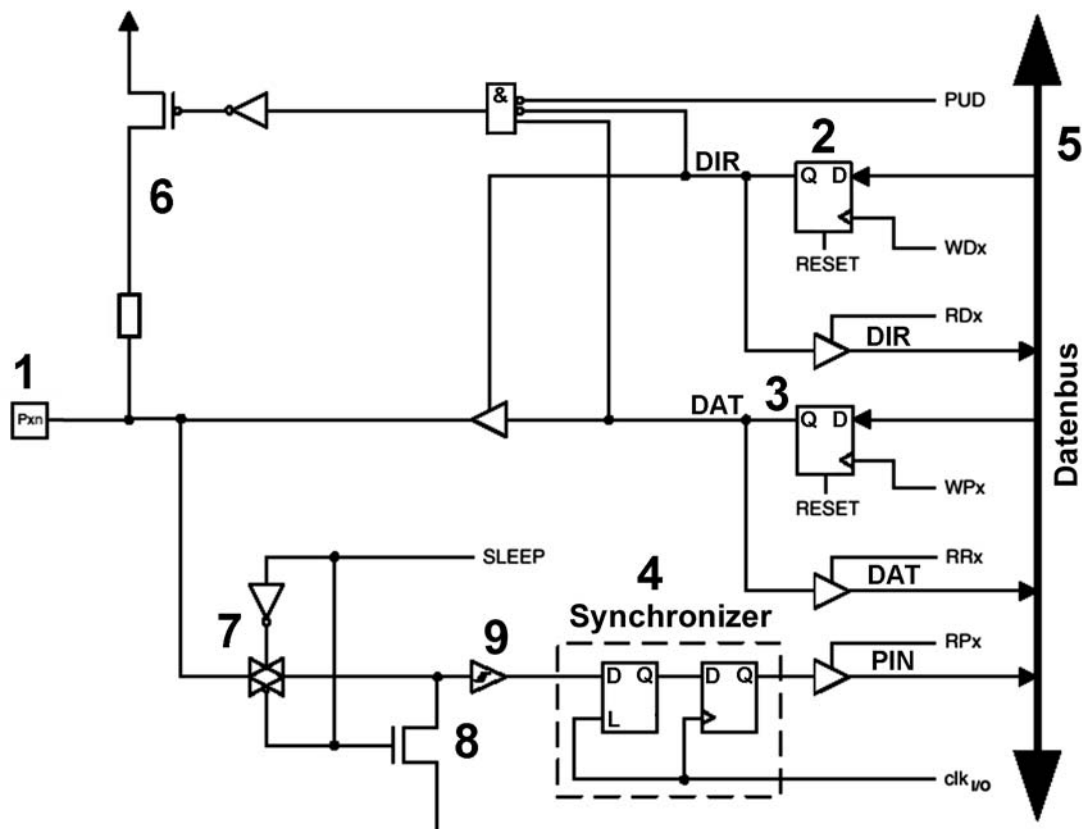


Abb.1.9 Mikrocontroller-Port. Eine einzelne Bitposition (nach Atmel)

Die Abbildung veranschaulicht mehrere der zuvor erläuterten Schaltungsvorkehrungen. 1- E-A-Anschluß (Pin); 2 - Richtungsregister; 3 - Datenregister; 4 - Synchronisationsstufe^{*)} mit Latch und D-Flipflop; 5 - der interne Datenbus; 6 - schaltbarer Pull-up-Widerstand. Kann aktiviert werden, wenn die Bitposition als Eingang geschaltet, aber nicht belegt ist (offener Eingang). 7- Schalterelement; 8 - Pull-down-Transistor; 9 - Eingangspuffer. In Schlafzuständen (SLEEP) wird der Anschluß von der Innenschaltung abgetrennt (7) und der Eingangspuffer 9 mit dem Festwert Low belegt (8).

*) : in vielen Datenblättern ist diese Stufe nicht dargestellt.

Isolation (galvanische Trennung)

Optokoppler. Kann auch Gleichspannungssignale übertragen. Rückwirkungsfrei. Keine Leistungsübertragung.

Transformator

Kondensator

Nicht vergessen: das Rücksetzen

Nach dem Einschalten muß die gesamte Elektronik in einen definierten Anfangszustand gelangen -- sonst ergeben sich womöglich sehr unschöne Nebenwirkungen. Beispiel: wir haben ein Register, daran angeschlossen einen Treibertransistor mit Relais. Dieses steuert einen Elektromotor. Wenn nichts getan wird, steht die Chance 50:50, daß das betreffende Flipflop nach dem Einschalten in die Stellung „1“ kippt (und so das Relais zum Anziehen bringt...).

Wenn das Initialisieren zu lange dauert

Einfache Mikrocontroller-Anwendungen verzeihen manche Sünden in Hinsicht auf eine saubere Initialisierung, weil nach dem Einschalten sofort das Anwendungsprogramm gestartet wird, die Unsicherheit also schlimmstenfalls nur wenige Millisekunden dauert. Das Hochfahren der PCs zieht sich aber hin (Sekunden...Minuten). Das gilt sinngemäß für andere Systeme, die auf eine komplexe Betriebssoftware angewiesen sind (Linux ist übrigens um keinen Deut besser...). Der Anfangszustand muß somit auf direktem Wege erzwungen werden (also ohne auf Software angewiesen sein). Typische Maßnahmen:

- eigene Rücksetzsaltungen im Interfaceadapter (Abb. 1.10),
- Einsatz von Flipflops und Registern mit Rücksetzeingang,
- Anordnung von Pull-up- und Pull-down-Widerständen,
- Nutzung moderner Leistungselektronik-Schaltkreise, die entsprechende Vorkehrungen haben.

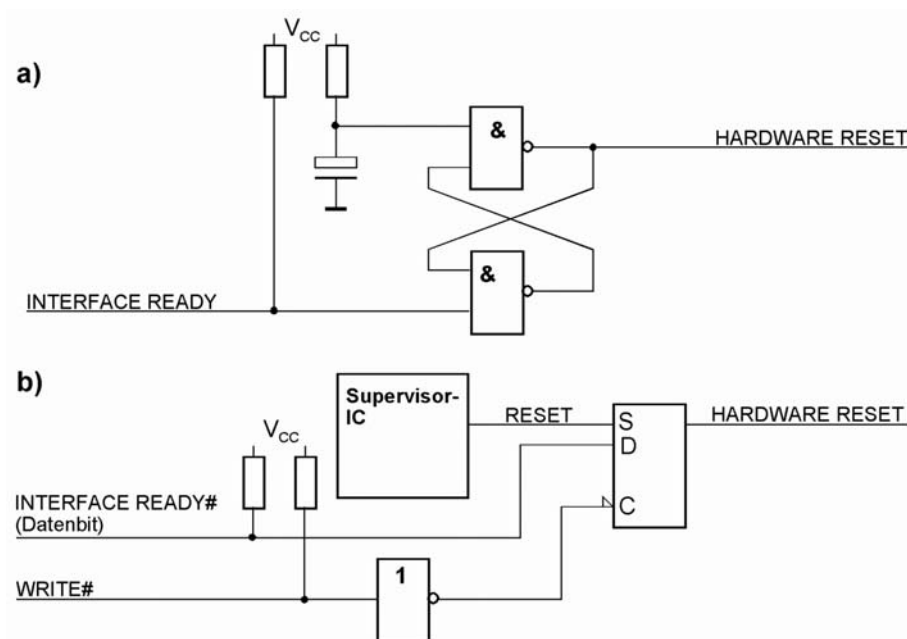


Abb.1.10 Bewährte Rücksetzsaltungen

a) - eine uralte Praxisschaltung. Erzeugen des Rücksetzimpulses über ein RC-Glied. b) - es gibt eigene Schaltkreise, die das Ansteigen und Abfallen der Speisespannung auswerten, um entsprechende Rücksetzimpulse zu bilden (Voltage Supervisors). Beide Schaltungen erzeugen ein Signal zum Hardware-Rücksetzen, das vom Einschalten an solange aktiv bleibt, bis über das Interface eine entsprechende Bereitschaftsmeldung (INTERFACE READY) kommt. Die Abbildung zeigt zwei Auslegungen: a) - mit eigenem Signal zur Bereitschaftsanzeige; b) - mit Übernahme einer Datensignalbelegung (das Rücksetz-Flipflop wird wie ein Ausgaberegister über einen Datenbus angesprochen).

Manche Interfaces haben eine Pegelzuordnung, die es ermöglicht, den Betriebszustand zu erkennen (Beispiel: USB). Gelegentlich kann man Ähnliches selbst verwirklichen, z. B. eine Signalleitung mit einem Pull-up-Widerstand beschalten und den ersten Low-Impuls als Bereitschaftsanzeige definieren.

Übers Interface rücksetzen

Interfaces mit höherentwickelten Signalprotokollen erlauben es, die angeschlossenen Geräte softwareseitig zurückzusetzen oder bestimmte Initialisierungs- und Wiederanlaufvorgänge auszulösen. Es ist sinnvoll, auch in privaten Protokollen und Kommandosätzen so etwas vorzusehen.

Typische Schutzmaßnahmen im Überblick (nach Vishay):

