2. Digitale Systeme

2.1 Grundlagen der Schaltungstechnik

2.1.1 Entwurfsprobleme und Schaltungslösungen

Es hat sich bewährt, Entwurfsprobleme in Klassen oder Kategorien einzuteilen, um die so erkannten Teilaufgaben mit jeweils geeigneten gedanklichen Ansätzen, Vorgehensweisen und Entwicklungswerkzeugen bearbeiten zu können. Signale aus der Außenwelt werden in Eingabeschaltungen in eine interne Informationsdarstellung gewandelt. Ergebnisse der Informationsverarbeitungsvorgänge werden über Ausgabeschaltungen in der Außenwelt zur Wirkung oder zur wahrnehmbaren Darstellung gebracht (sichtbare Anzeige, akustische Signale, Sprachausgabe usw.). Die Verarbeitungsvorgänge laufen in Verarbeitungsschaltungen (Operationswerken) ab. Nur sehr wenige Aufgaben wird man ausschließlich durch kombinatorische Zuordnungen lösen können. Zumeist werden aufeinander folgende Schritte elementarer Informationswandlungen auszuführen sein. Hierfür müssen Speichermittel und Steuerschaltungen vorgesehen werden.

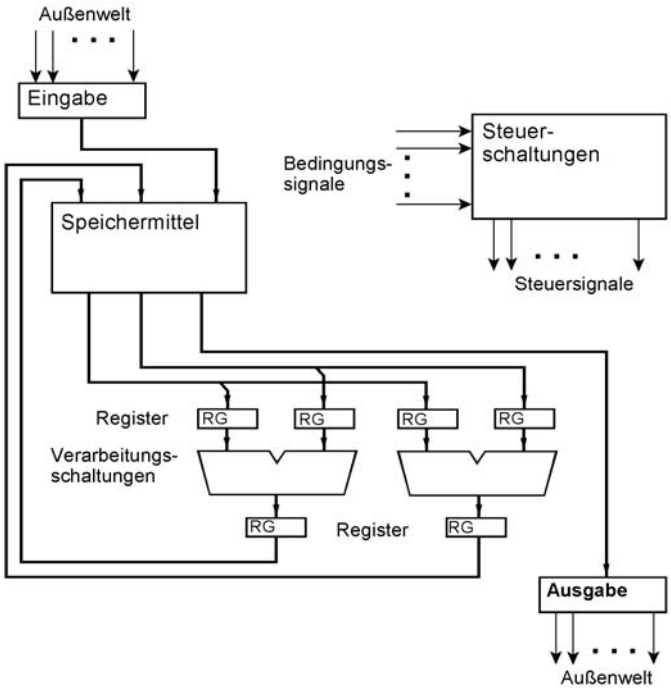


Abb. 2.1 Der grundsätzlich Aufbau einer digitalen informationsverarbeitenden Einrichtung.

Steuerschaltungen

Typische Steuerungsaufgaben bestehen darin, bestimmte Signalfolgen abzugeben (Beispiel: Steuerung eines Schrittmotors) und auf Eingangssignale durch Erregen von Ausgangssignalen zu reagieren (Beispiel: Steuerung einer E-A-Schnittstelle). Steuerschaltungen sind sequentielle Schaltungen. Das Grundmodell der Steuerschaltung ist der Zustandsautomat (Finite State Machine FSM).

Verarbeitungsschaltungen
Typische Verarbeitungsaufgaben bestehen im Auswerten und Umformen von binären Wertangaben (Beispiel: das Vergleichen, Klassifizieren, Filtern usw. von Werten, die von einem Analog-Digital-Wandler geliefert werden). Das Grundmodell der Verarbeitungsschaltung ist eine Anordnung aus Registern (zum Speichern der Operanden und Ergebnisse) und kombinatorischen Netzwerken, die die eigentlichen Verknüpfungsoperationen erledigen (Register-Transfer-Ebene RTL). Beispiel: eine Rechenschaltung, die addieren und subtrahieren kann.

Informationstransport und Kommunikation
Einfachste Signalwege in Form von Punkt-zu-Punkt-Verbindungen (vom Ausgang zum nachfolgenden Eingang) sind nicht immer ausreichend. Typische Entwurfsprobleme betreffen:

- die Nutzung gemeinsamer Signalwege (Bussysteme),
- die Übertragung über vergleichsweise schmale Signalwege (Serialisierung und Deserialisierung),
- das Auswählen von Signalbelegungen (Multiplexing),
- das Verteilen von Signalbelegungen (Demultiplexing).

Informationsspeicherung
Die einfachsten Speicherelemente sind Latches und Flipflops. Sie können direkt mit Gattern verbunden werden. Ein Latch oder Flipflop hat eine Speicherkapazität von einem Bit. Die Speicherkapazität kann durch folgende grundsätzliche Maßnahmen gesteigert werden:

a) Zusammenfassung mehrerer Latches oder Flipflops zu Registern,
b) Zusammenfassung mehrerer Register zu Registerblöcken (Register Files),
c) Nutzung von Speichermatrizen (Memory Arrays),
d) Nutzung von Speicherschaltkreisen,
e) Übergang auf Massenspeicher (NAND-Flash, Festplatten usw.).

Anordnungen gemäß a) und b) gehören noch unmittelbar zur Digitalschaltung. c) und d) sind Grenzfälle. Von einer gewissen Speicherkapazität an aufwärts kann die Speicheranordnung nicht mehr auf dem Digitalschaltkreis angeordnet werden (andere Technologie). Manche Speichermatrizen und Speicherschaltkreise kann man als Ansammlungen von Latches oder Registern betreiben, manche nicht; sie sind dann nur über besondere Speichersteuerschaltungen (Memory Controller) anzusteuern. Massenspeicher können – wenn die Aufwendungen annehmbar bleiben sollen – nur in Umgebungen eingesetzt werden, die programmgesteuerte Einrichtungen aufweisen; die einschlägigen Wirkprinzipen (Dateisysteme usw.) liegen außerhalb des Bereichs der eigentlichen Digitaltechnik.

Signalaufbereitung, Signalformung usw.
Einige Beispiele derartiger Aufgaben:

- Aus schmalen Impulsen sind breite zu machen.
- Es sind nur Impulse von einer gewissen Dauer an weiterzugeben, kürzere aber zu unterdrücken.
- Es sind Impulse einer gewissen Dauer abzugeben.

Im Grunde handelt es sich um sequentielle Schaltungen. Der Theorie nach sind solche Aufgaben zu erledigen, indem man entsprechende Zustandsautomaten entwirft. Nicht selten werden aber Trick- oder Verbundlösungen (analog + digital) gewählt. Das ist manchmal durchaus berechtigt. So kann ein schematisch entworfener Zustandsautomat zu aufwendig sein[1)]. Verbundlösungen sind aber vor allem dann erforderlich, wenn es sich um Schnittstellen zur Außenwelt handelt. An solchen Stellen können Synchronisationsprobleme auftreten, die mit rein digitalen Schaltmitteln nicht zu lösen sind.

Zählfunktionen
Das Zählen ist ein gewisser Grenzfall. So kann man sich einen Zustandsautomaten vorstellen, der aufeinanderfolgende Zustände 1 – 2– 3 usw. durchläuft. Auch könnte man den Zählvorgang als Verarbeitungsoperation ansehen (z. B. Zählen als Addieren einer Eins). Traditionell löst man jedoch viele Zählprobleme durch den Einsatz von Zählern, die als Schaltkreise oder Funktionselemente angeboten werden.

Zeitstufen
In der Digitaltechnik läuft das Darstellen von Zeit im Grunde auf Zählvorgänge hinaus (Abzählen von Taktimpulsen). Manchmal passt die darzustellende Zeit aber nicht in ein Taktraster. Dann muss man auf Verbundlösungen (analog + digital) zurückgreifen.

2.1.2 Kombinatorische und sequentielle Schaltungen

Kombinatorische Schaltungen werden durch Zusammenschalten von Gattern und Negatoren aufgebaut, wobei es nicht vorkommt, dass irgend ein Gatterausgang auf Eingänge irgend eines vorgeordneten Gatters zurückgeführt ist. Kombinatorische Schaltungen (Abb. 2.2a und 2.3a) haben keine Speicherelemente, also kein Gedächtnis.Wenn man irgend eine Kombination von Eingangswerten anlegt, so ergibt sich immer dieselbe Ausgangsbelegung. Die Belegung eines jeden Ausgangssignals a_i (i = 1...m) kann als Boolesche Funktion f_i dargestellt werden, deren Variable Eingangssignale e_1, e_2,...,e_n sind:

$$a_i = f_i (e_1, e_2,...,e_n)$$

Sequentielle Schaltungen sind mit Rückführungen und Speicherelementen erweiterte kombinatorische Schaltungen. Ihr Verhalten wird somit nicht nur von der jeweilige Eingangsbelegung, sondern auch von den Belegungen der Rückführungen und Speicherelemente und damit von der Vorgeschichte abhängen. Die Rückführungen können einfache Verbindungen sein, können aber auch Verzögerungsstufen oder Speicherelemente enthalten.

Asynchrone Schaltwerke
Schaltwerke ohne Taktsteuerung bezeichnet man als asynchron. Ihr Zeitverhalten wird ausschließlich von den internen Verzögerungen bestimmt. Das Grundproblem besteht darin, eine solche Schaltung zum stabilen Arbeiten zu bekommen. Sie arbeitet dann stabil, wenn sich bei einer bestimmten Eingangsbelegung und einem bestimmten aktuellen Zustand stets derselbe

1: Praxistipp: Nicht nur die Flipflops und Gatter zählen, sondern eine solche Vermutung jeweils genau überprüfen (ggf. Probeentwurf) – es kommt immer auf die Technologie an, mit der man arbeitet ...

Folgezustand ergibt. Sind in die Rückführungen keine Verzögerungsstufen eingebaut (freie Rückführungen), so ist dies nicht immer gegeben, da die einzelnen Schaltnetze zur Bildung des Folgezustandes unterschiedliche Durchlaufzeiten haben. Hieraus können sich Wettlauferscheinungen (Races) ergeben. Infolgedessen ist damit zu rechnen, dass die Schaltung in irgend einen beliebigen Zustand fällt oder dass sie auf Dauer ins Schwingen gerät. In der Praxis kommt ein solcher Ansatz nur für Entwurfsaufgaben in Betracht, bei denen die Eingangssignale definierte Schaltfolgen aufweisen (z. B. für nicht allzu komplizierte Interfacesteuerungen).

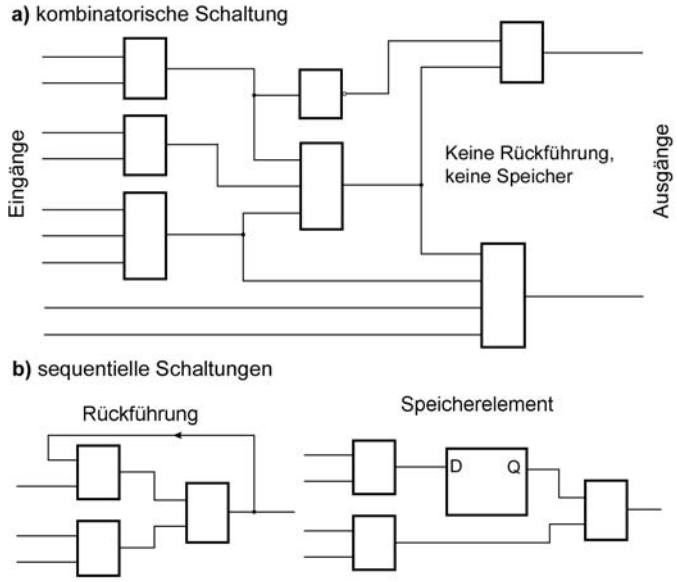


Abb. 2.2 Kombinatorische und sequentielle Schaltungen.

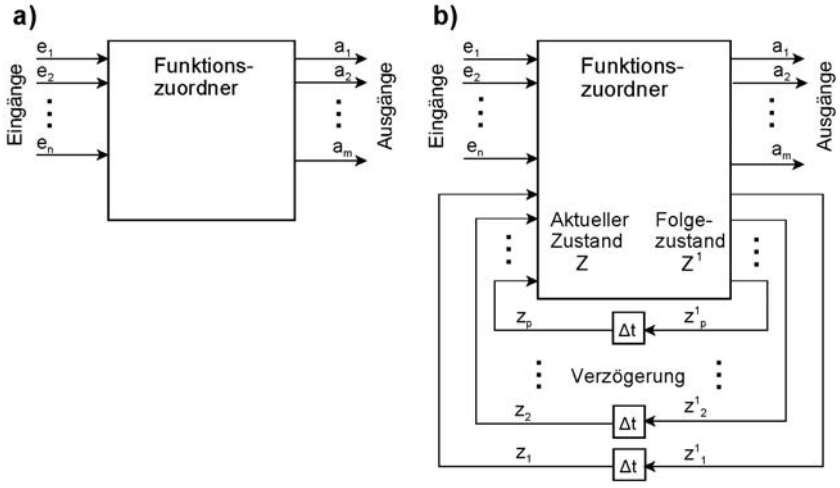


Abb. 2.3 Allgemeine Modellvorstellungen. a) kombinatorische Schaltung; b) sequentielle Schaltung (Modell nach Huffmann ([2.1])).

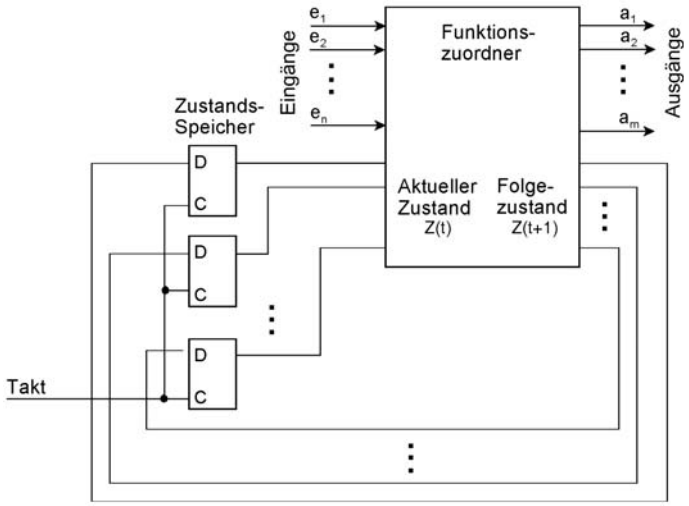


Abb. 2.4 Sequentielle Schaltung mit Zustandsspeicherung.

Synchrone Schaltwerke
Schaltwerke mit Taktsteuerung bezeichnet man als synchron. Ihr Zeitverhalten wird von Taktsignalen bestimmt, die die Zustandswechsel auslösen. In den meisten Fällen ist ein Takt eine kontinuierliche Folge von Impulsen mit konstanter Folgefrequenz. Es gibt aber auch Einzeltakte und Taktimpulse mit veränderlicher Folgefrequenz. Die weitaus meisten sequentiellen Schaltungen werden als synchrone Schaltwerke entworfen.

Synchron und asynchron
Der Sprachgebrauch ist nicht immer so eindeutig wie soeben beschrieben (synchron = mit, asynchron = ohne Takt). Oftmals bezeichnet man nur Schaltungen, die mit kontinuierlich durchlaufenden Taktsignalen betrieben werden, als synchron, Schaltungen aber, die ihre zeitbestimmenden Impulse auf andere Weise bilden (z. B. durch Erkennen von Änderungen) als asynchron.

Zustandsautomaten (Finite State Machines)
Weil die Funktion einer sequentiellen Schaltung durch die Zustände (States) und Zustandsübergänge (State Transitions) entscheidend bestimmt wird, bezeichnet man eine solche Schaltung auch als Zustandsautomat, Finite State Machine (FSM) oder kurz als State Machine.

Latches und Flipflops
Das sind die typischen Fachbegriffe für derartige Speicherschaltungen. Der Begriff "Latch" (= Klinke) bringt das Selbsthalteprinzip bildhaft zum Ausdruck.

Latch
Der Dateneingang ist so lange zum Ausgang durchgeschaltet, wie das Taktsignal aktiv ist. Bei dieser Taktbelegung folgt der Ausgang allen Änderungen des Eingangs nach. Man sagt, das Latch ist *transparent* (durchlässig) für das Signal am Dateneingang. Die Datenübernahme erfolgt mit dem Übergang zum inaktiven Taktpegel. Da das Verhalten des Latches vom Pegel des Taktsignals abhängt, bezeichnet man solche Speicherelemente auch als pegelgesteuert (level-sensitive).

Flipflop
Die Eingangsbelegung wird nur mit der jeweils spezifizierten Taktflanke übernommen. Was zwischen diesen Taktflanken am Dateneingang geschieht, hat keinen Einfluss auf den gespeicherten Zustand. Solche Speicherelemente heißen demgemäß flankengesteuert (edge-sensitive).

Es wird aber nicht immer streng unterschieden. Manche Theoretiker bezeichnen jeden 1-Bit-Speicher als Flipflop, unabhängig davon, wie die Taktsignale wirken. Hingegen findet man in der praxisbezogenen Literatur (u. a. in Datenblättern) häufig die Bezeichnung "Latch" auch für Bauelemente (z. B. Register), die flankengesteuerte Flipflops enthalten. Näheres zu den Latches und Flipflops in Kapitel 4.

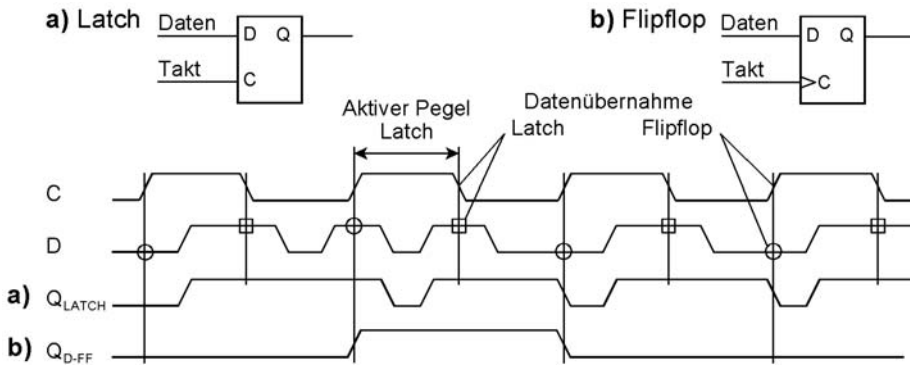


Abb. 2.5 Schaltverhalten im Vergleich. a) Latch; b) Flipflop.

Register
Register sind Aneinanderreihungen gleichartiger Speicherelemente (Latches oder Flipflops) mit gemeinsamem Takt.

Speichermatrizen
Speichermatrizen (Memory Arrays) sind reguläre Anordnung von Speicherzellen, die mit Auswahlschaltungen so verbunden sind, dass bei jedem Zugriff eine einzige Speicherzelle ausgewählt wird.

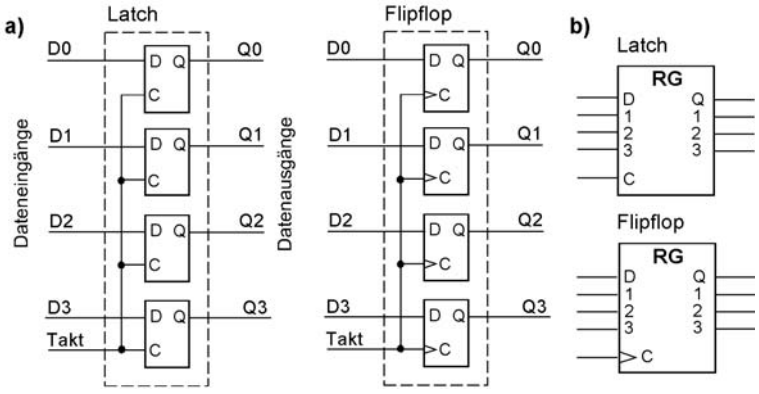


Abb. 2.6 Register. a) Schaltungen; b) Schaltsymbole.

2.1.3 RTL – die Register-Transfer-Ebene

RTL = Register Transfer Level. Auf dieser Ebene kann man eine Digitalschaltung in funktioneller Hinsicht vollständig dokumentieren. Sie besteht aus Speicherelementen (Flipflops und Latches), die gemäß der Anzahl der jeweils zusammengehörenden Signale zu Registern zusammengefasst werden. Alle kombinatorischen Verknüpfungen werden funktionell beschrieben (mit Booleschen Gleichungen, Wahrheitstabellen o. dergl.). Wie diese Verknüpfungen im einzelnen (z. B. als Gatternetze) realisiert werden, ist dabei gleichgültig. In der zeichnerischen Darstellung sind die kombinatorischen Netzwerke lediglich Blocksymbole (Funktionszuordner). Komplizierte Digitalschaltungen – vor allem Verarbeitungsschaltungen und Zustandsautomaten – werden oftmals auf der RTL-Ebene entworfen. Hierbei werden lediglich die Register definiert und die kombinatorischen Funktionen angegeben. Die Umsetzung in Gatterstrukturen ist dann Sache des Entwicklungssystems. In einer exakten RTL-Darstellung sind beschrieben:

- alle Register bis aufs einzelne Flipflop,
- alle Funktionszuordner durch die Schaltfunktionen, die die jeweiligen Beziehungen zwischen Eingangs- und Ausgangsbelegungen angeben (z. B. durch Boolesche Gleichungen oder Wahrheitstabellen).

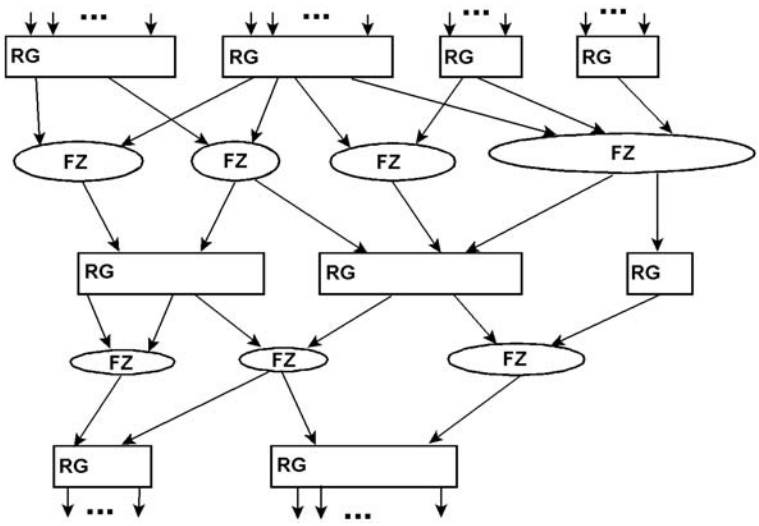


Abb. 2.7 Zum Prinzip der RTL-Darstellung. RG - Register (Speicherelemente); FZ - Funktionszuordner (Kombinatorik).

2.1.4 Wahrheitswerte und elektrische Pegel

Die beiden Wahrheitswerte sollen weiterhin mit 0 (falsch, invertiert, negiert) und 1 (wahr) bezeichnet werden. Beide Wahrheitswerte müssen in der Praxis durch Werte einer elektrischen Kenngröße dargestellt werden. Diese Werte bezeichnet man als Signalpegel oder Logikpegel (Signal / Logical Levels).

Low und High

Die beiden Pegel werden üblicherweise mit Low (L; LO) und High (H; HI) bezeichnet. (Genauer gesagt: es gibt nicht zwei einzelne Werte, sondern zwei Wertebereiche oder Toleranzfelder). Mit dieser Bezeichnungsweise will man die absoluten Werte der betreffenden

(physikalischen) Kenngröße zum Ausdruck bringen: *Low* ist der niedrigere Wert (die geringere Spannung oder der geringere Strom); *High* der höhere (die höhere Spannung, der höhere Strom). Exakt formuliert: der Wertebereich, der näher an $-\infty$ (minus Unendlich) liegt, ist *Low*, jener, der näher an $+\infty$ (plus Unendlich) liegt, ist *High*.

Positive und negative Logik
Die beiden Wahrheitswerte 0 und 1 müssen den beiden Signalpegeln Low und High irgendwie zugeordnet werden. Hierfür gibt es zwei Möglichkeiten. Je nach der Zuordnung spricht man von positiver oder von negativer Logik.

Signalpegel	Positive Logik	Negative Logik
Low	0	1
High	1	0

Tabelle 2.1 Positive und negative Logik.

An sich kann man die Zuordnung beliebig treffen und auch – z. B. zwecks Schaltungsvereinfachung (Aufwandsminimierung) – mitten in der Schaltung wechseln. Beim Wechseln gelten die DeMorganschen Regeln: UND wird zum ODER, ODER zum UND; NAND zum NOR und NOR zum NAND. Ein Bauelement kann somit – je nach Pegelzuordnung – verschiedene Funktionen ausführen. Die meisten Datenblätter enthalten deshalb keine Wahrheitstabellen mit Nullen und Einsen, sondern Funktionstabellen mit Low (L) und High (H). Ob ein bestimmtes Bauelement beispielsweise als NAND-Gatter in positiver oder als NOR-Gatter in negativer Logik eingesetzt wird, ist dem Anwender überlassen.

Heutzutage vorherrschend: die positive Logik
Die positive Logik hat sich weitgehend durchgesetzt. Die Entwicklungsgeschichte in Stichworten:

- Silizum wurde zum vorherrschenden Halbleitermaterial.
- In Siliziumhalbleitern sind Elektronen die Majoritätsträger.
- Somit kommt man – um einen bestimmten Strom durchzuleiten (Treibvermögen) – bei Elektronenleitung mit geringeren Materialquerschnitten bzw. Siliziumfläche aus als bei Löcherleitung.
- Deshalb sind NPN- und N-Kanal-Transistoren die kostengünstigsten Bauelemente.
- Damit aufgebaute Schaltstufen brauchen eine positive Versorgungsspannung.
- NANDs sind kostengünstige Gatterstrukturen.
- NAND-NAND entspricht UND-ODER, also der naheliegenden Realisierung kombinatorischer Verknüpfungen (DNF)).

Diese Tradition wurde – von den 60er Jahren an – bis in die heutige Zeit weitergeführt. Deshalb wird auch in den folgenden Darstellungen ausschließlich die positive Logik verwendet.

Aktiv – inaktiv
Ein Signal ist *aktiv* (erregt, asserted, energized, valid, manchmal auch: true) wenn es seine jeweilige Wirkung ausübt, ansonsten ist es *inaktiv* (nicht erregt, in Ruhe, deasserted, invalid,

manchmal auch: false). Tritt die jeweilige Wirkung ein, wenn das Signal auf High-Pegel liegt, so nennt man es "aktiv High", andernfalls "aktiv Low".

Die Wirkung kommt üblicherweise im Signalbezeichner – mehr oder weniger zutreffend – zum Ausdruck. So wirkt ein Signal MREQ als Speicheranforderung (Memory Request), ein Signal READY als Fertigmeldung, ein Signal GRANT als Bestätigung usw. Aktiv Low wirkende Signale werden in Schaltbild und Signalbezeichnung zumeist durch ein Negationssymbol gekennzeichnet (ein aktiv Low wirkendes Speicheranforderungssignal heißt dann beispielsweise MREQ#).

Manchmal sind beide Signalbelegungen wirksam
Mit anderen Worten: es gibt keine Unterscheidung zwischen aktiv und inaktiv – beide Pegel (sowohl Low als auch High) signalisieren oder bewirken irgend etwas. Das gilt für Datensignale, Adresssignale und für Steuersignale mit auswählender Wirkung. Gelegentlich kommen beide Wirkungen in den Signalbezeichnern zum Ausdruck. In den Handbüchern der Schaltkreishersteller werden die jeweiligen Bezeichnungsregeln (Signal Conventions) erklärt. Beispiel: was bedeutet *M/IO#*? *M* steht für Speicherzugriff (Memory Access), *IO* steht für E-A-Zugriff (I/O Access). Das Negationssymbol nach IO kennzeichnet, dass diese Wirkung bei Low eintritt. Also bewirkt M/IO# = High einen Speicherzugriff und M/IO# = Low einen E-A-Zugriff.

2.1.5 Logik- und Signalspezifikationen

Solche Spezifikationen sind Standards, die Kennwerte digitaler Signale beschreiben. Schaltkreise, die der gleichen Spezifikation entsprechen, kann man typischerweise direkt zusammenschalten (es sind keine Pegelwandlungen o. dergl. erforderlich). So ein Standard enthält meist nur einige Tabellen mit Angaben zu Versorgungsspannungen, Signalpegeln, Strömen und Anstiegszeiten, ergänzt um die Erläuterung der Messbedingungen.

Verknüpfung oder Informationsübertragung?
Die sozusagen klassische Signalleitung verbindet Logikschaltungen (Gatter, Flipflops usw.) untereinander. Derartige Signale werden unmittelbar miteinander verknüpft. Die meisten der neueren Spezifikationen betreffen hingegen die reine Informationsübertragung zwischen hochintegrierten Schaltkreisen. Die Verknüpfungen finden im Innern statt. Es ist also zu unterscheiden zwischen (1) Signalen, die logischen Verknüpfungen unterzogen werden und (2) Signalen, die ausschließlich zur Informationsübertragung vorgesehen sind.

Die Anschlüsse der modernen hochintegrierten Logikschaltkreise (CPLDs, FPGAs, ASICs) können für verschiedene Signalspezifikationen eingerichtet werden (durch entsprechendes Programmieren oder Auswählen aus einer Funktionselementebibliothek). Auch gibt es ein reichhaltiges Angebot an Schaltkreisen zur Signalwandlung. Die folgenden Erläuterungen beschränken sich ausschließlich auf Logiksignale im eigentlichen Sinne. Alle Signalpegel werden auf Masse bezogen (single ended signals), und die Unterscheidung zwischen Low und High ergibt sich im Schaltkreis ohne Bezug auf eine externe Referenzspannung.

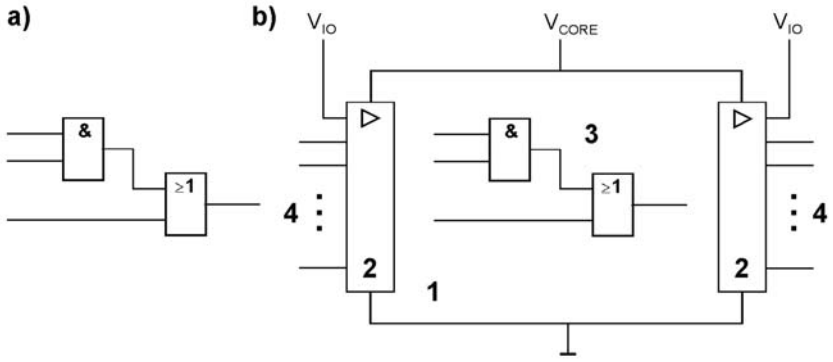


Abb. 2.8 Digitale Signale. a) zwischen Logikschaltungen; b) in moderner Hardware auf Grundlage hochintegrierter Schaltkreise. 1 - Logikschaltkreis (CPLD, FPGA usw.); 2 - E-A-Koppelstufen; 3 - logische Verknüpfungen; 4 - diese Signale dienen vorwiegend zur Informationsübertragung; V_{CORE} - Versorgungsspannung für die logischen Verknüpfungen (Kernlogik); V_{IO} - Versorgungsspannung(en) für die externen Schnittstellen (Ein- und Ausgabe).

2.1.6 Elementare Kennwerte

Signalpegel

Es gibt Pegelspezifikationen für Ausgänge und Eingänge. Low und High werden jeweils durch einen unteren und einen oberen Endwert (Minimalwert, Maximalwert) gekennzeichnet. Der verbotene Bereich liegt zwischen dem Maximalwert des Low-Pegels und dem Minimalwert des High-Pegels. Es kommt darauf an, mit möglichst geringem Aufwand sicher zwischen Nullen und Einsen unterscheiden zu können, und zwar unter den Bedingungen des praktischen Betriebs, d. h. unter dem Einfluss von Fertigungsabweichungen, Versorgungsspannungsschwankungen und überlagerten Störungen. Deshalb dürfen die drei Bereiche (Low, verboten, High) nicht zu schmal sein.

Die üblichen Pegelspezifikationen betreffen Spannungsbereiche zwischen 0 V (Massepotential (GND)) und einer positiven Versorgungsspannung (V_{CC}, V_{DD}). Der minimale Low-Pegel entspricht dem Massepotential, der maximale High-Pegel entspricht der Versorgungsspannung. Damit sind nur die Grenzen zum verbotenen Bereich zwischen Low und High von Interesse, also die maximalen Low- und die minimalen High-Pegel. Der Betrag der Differenz zwischen diesen Eingangs- und Ausgangspegeln heißt Störspannungsabstand oder kurz Störabstand (Noise Margin).

Signalhub

Der Signalhub ist die Differenz zwischen High- und Low-Pegel.

Schwellenspannung

Diese Angabe (Threshold Voltage V_{TH}) kennzeichnet die Eingangsspannung, bei der die Eingangsstufe zwischen Low und High umschaltet.. Als Schwellenspannung V_{TH} ist jener Punkt der Kennlinie definiert, an dem die Ausgangsspannung gleich der Eingangsspannung ist ($V_I = V_O$).

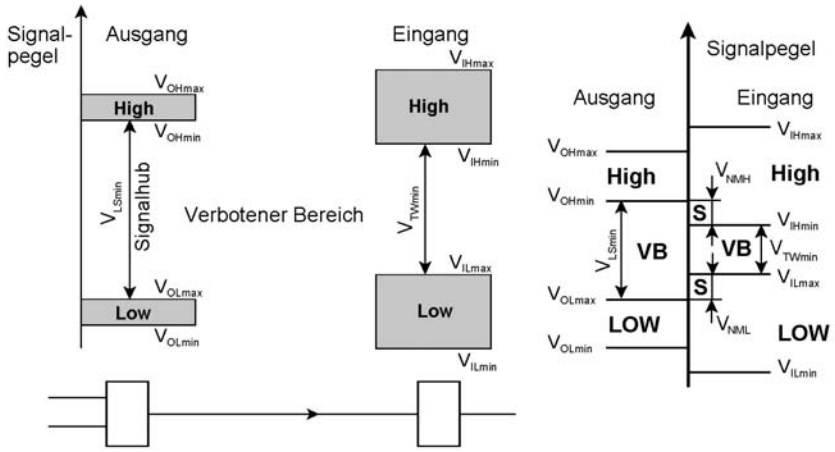


Abb. 2.9 Signalpegel von Digitalschaltungen. VB - verbotener Bereich; S - Störspannungsabstand.

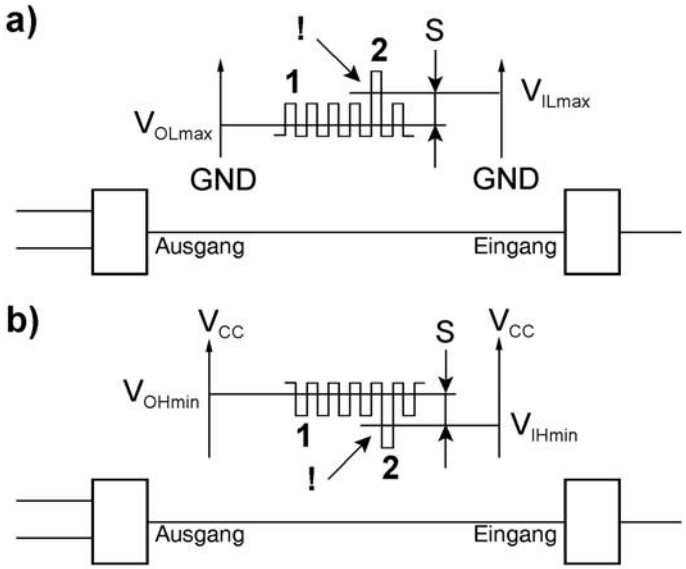


Abb. 2.10 Eine von Störungen betroffene Signalleitung. a) Low-Signal; b) High-Signal. S - Störspannungsabstand; 1 - typische (= zulässige) überlagerte Störungen; 2 - diese Störpegel sind unzulässig. Sie können Fehlschaltungen am Ausgang des empfangenden Gatters verursachen.

Ganz elementare Faustregeln:

- TTL-Kompatibilität: Signalhub 3 V, Schwellenspannung 1,5 V,
- CMOS: der Signalhub entspricht nahezu der Versorgungsspannung (Rail-to-Rail-Schaltverhalten). Richtwerte: Low höchstens 0,1 V; High nicht weniger als V_{CC} – 0,1 V. Schwellenspannung = halbe Versorgungsspannung (0,5 V_{CC}).

Kennwert	TTL-Kompatibilität	5-V-CMOS (HC, AC)*	Moderne CMOS-Spezifikationen
Ausgangskennwerte			
Minimum-High am Ausgang (V_{OHmin})	2,4 V	4,44 V (typisch 4,9 V)	0,75 V_{CC}
Maximum-Low am Ausgang (V_{OLmax})	0,4 V	0,44 V (typisch 0,1 V)	0,25 V_{CC}
Eingangskennwerte			
Minimum-High am Eingang (V_{IHmin})	2 V	0,7 V_{CC} (3,5 V)	0,65 V_{CC}
Maximum-Low am Eingang (V_{ILmax})	0,8 V	0,3 V_{CC} (1,5 V)**	0,35 V_{CC}
Schwellenspannung (V_{TH})	1,5 V	0,5 V_{CC} (2,5 V)	0,5 V_{CC}
Störspannungsabstände			
Low (V_{NML})	0,4 V	1,06 V (typisch 1,4 V)	0,1 V_{CC}
High V_{NMH})	0,4 V	0,94 V (typisch 1,4 V)	0,1 V_{CC}

*: Betriebsspannungsbereich 2...6 V. Werte in Klammern für V_{CC} = 5 V. **: Nach JEDEC nur 0,2 V_{CC} (1 V).

Tabelle 2.2 Logikpegel, Störspannungsabstände und Schwellenspannungen typischer Signalspezifikationen.

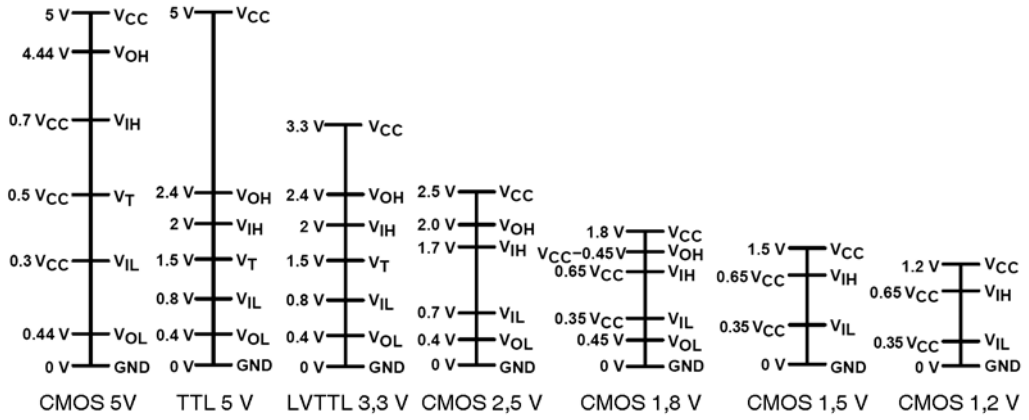


Abb. 2.11 Typische Signalspezifikationen im Überblick (nach [2.8]).

Verzögerungszeit

Die Zeit, die vergeht, bis aufgrund einer Signaländerung an einem Eingang die entsprechende Signaländerung am Ausgang wirksam wird, bezeichnet man als Verzögerungszeit oder Durchlaufverzögerung (Propagation Time oder Propagation Delay). Die Zeiten werden typischerweise dann gemessen, wenn der Signalhub der Schwellenspannung entspricht (die genauen Messbedingungen stehen im Datenmaterial).

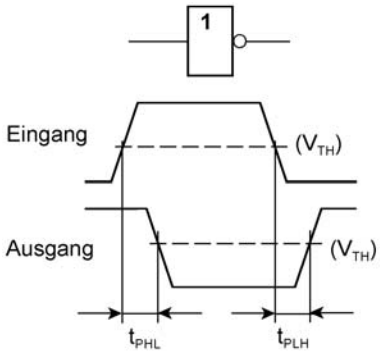


Abb. 2.12 Verzögerungszeiten zwischen Eingangs- und Ausgangsänderung.

Schaltungstiefe

Die Schaltungstiefe s ist eine Zahl, die angibt, wieviele Gatter nacheinander durchlaufen werden müssen, bis sich eine Eingangsänderung am Ausgang auswirkt. Dabei wird der ungünstigste Fall betrachtet. Bestimmung: abzählen, wieviele Gatter zwischen den Ein- und Ausgängen hintereinandergeschaltet sind. Die größte dieser Zahlen ergibt die Schaltungstiefe.

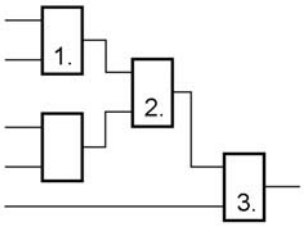


Abb. 2.13 Zur Definition der Schaltungstiefe. Im Beispiel ist s = 3.

Durchlaufverzögerung

Ein Gatternetz der Schaltungstiefe s hat – roh gerechnet – eine Durchlaufverzögerung von $s \cdot t_{Pd}$. Die Verbindungen zwischen den Gattern kann man zunächst vernachlässigen (Überschlagsrechnung). Geht es um sehr kurze Zeiten (Richtwert: $\leq$ 1 ns)[2)] sind solche Vereinfachungen allerdings nicht mehr zulässig.

Signalflanken

Die Eingangssignale müssen den Bereich zwischen Low und High möglichst schnell durchlaufen. Verharrt ein Eingangssignal zu lange im Bereich der Schwellenspannung, kann der Ausgang keinen eindeutigen Logikpegel annehmen. In solchen Betriebszuständen ergibt sich zudem eine höhere Stromaufnahme. Deshalb werden in den Logikspezifikationen Maximalwerte der Anstiegszeit (oder Minimalwerte der Flankensteilheit) vorgegeben. Falls die ankomenden Signale diesen Anforderungen nicht genügen, können z. B. Schaltkreise mit Schmitt-Trigger-Eingängen eingesetzt werden.

2: Zur Veranschaulichung: 1 ns ist die Signallaufzeit über eine 20 cm lange Punkt-zu-Punkt-Verbindung oder eine 5 cm lange Busleitung.

Viel hilft nicht immer viel ...
Je steiler die Signalflanken, desto intensiver die Störstrahlung und die Störungen zwischen den einzelnen Signalen. Neuere Spezifikationen nennen deshalb auch Minimalwerte der Anstiegszeit (oder Maximalwerte der Flankensteilheit), die nicht unterschritten werden dürfen. Viele programmierbare Schaltkreise haben Ausgänge mit programmierbarer Flankensteilheit (zumeist im Sinne einer Auswahl zwischen schnell und langsam).

2.1.7 Halbleitertechnologien

Digitalschaltkreise werden u. a. nach der Technologie unterschieden, in der sie gefertigt werden. Die Begriffe bezeichnen das Ausgangsmaterial (z. B. Silizium oder Galliumarsenid), die grundlegenden Transistorstrukturen (bipolare oder Feldeffekt- (MOS-) Transistoren) und die jeweilige Schaltungstechnik (z. B. TTL, ECL, CMOS).

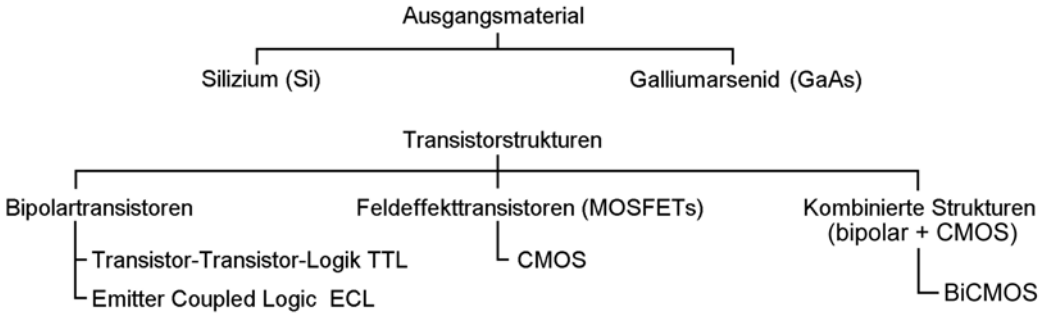


Abb. 2.14 Technologien digitaler Schaltkreise.

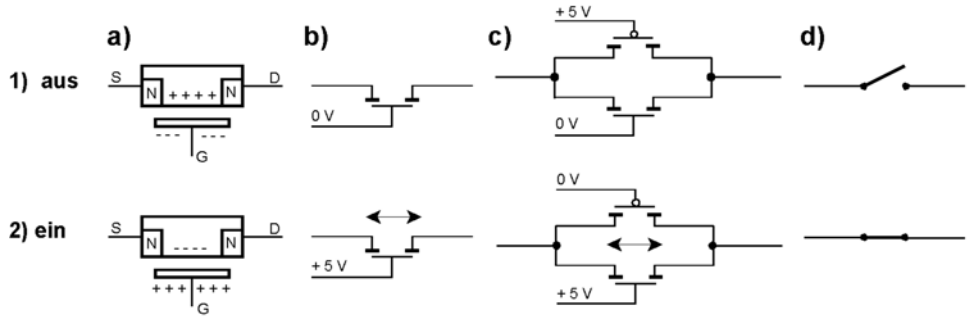
Bipolar	CMOS
Vergleichsweise geringer Signalhub (TTL eingangsseitig ca. 2 V, ausgangsseitig 2,4 V... typisch 3,5 V; ECL etwa 0,8 V).	Der Signalhub entspricht praktisch der Versorgungsspannung (Rail-to-Rail). Low ist fast 0 V; High entspricht bei geringer Belastung etwa V_{CC} - 0,1 V; bei voller Belastung sind es 0,2...0,3 V weniger.
Die Stromaufnahme ist nur in geringem Maße frequenzabhängig.	Die Stromaufnahme wächst linear mit der Taktfrequenz (s (2.10)).
Hohe Anforderungen an die Spannungsversorgung (typische Spannungstoleranz ± 5 %)	Spannungsversorgung vergleichsweise unkritisch (z. B. zwischen 3 und 15 V oder zwischen 2 und 6 V). Allerdings sind die Betriebskennwerte spannungsabhängig: je höher die Speisespannung, um so höher die zulässige Betriebsfrequenz, um so kürzer die Schaltzeiten, um so besser die Treibfähigkeit.

Tabelle 2.3 Bipolare und CMOS-Logikschaltkreise im Vergleich.

2.1.8 Schalter und Übertragungsgatter

Gatter gelten üblicherweise als die einfachsten Funktionselemente zum Aufbau von Digitalschaltungen. Gelegentlich kommen aber auch Bauelemente zum Einsatz, die als Schalter wirken. Es sind vor allem zwei Grundtypen von Bedeutung:

- Analogschalter. Die Source-Drain-Strecke eines N-Kanal-Feldeffekttransistors (FETs) wird leitend, wenn man an das Gate eine positive Spannung legt. Es gibt ein umfangreiches Angebot entsprechender Bauelemente, darunter auch Typen, die für den Einsatz in Digitalschaltungen optimiert sind (Busschalter, Multiplexer usw.). Zu den typischen Anwendungen gehören das Stromsparen, das Austauschen von Funktionseinheiten bei laufendem Betrieb (Hot Plugging) sowie das Abschalten zeitweise nicht genutzter Signalwege (zwecks Verminderung der kapazitiven Belastung).
- CMOS-Übertragungsgatter (Transfer Gate). Dem N-Kanal-FET ist ein P-Kanal-FET parallelgeschaltet, der invertiert angesteuert wird. So sind beide Transistoren gleichzeitig entweder gesperrt oder leitend. Übertragungsgatter werden vor allem im Innern von Schaltkreisen eingesetzt.



a) Der Zustand der Source-Drain-Strecke wird durch die Ladung auf dem Gate bestimmt: negative Ladung = aus, positive Ladung = ein.

b) Feldeffekttranistor als Schaltsymbol. Liegt eine positive Spannung am Gate, so wird die Source-Drain-Strecke leitend.

c) Übertragungsgatter (Transfer Gate) in CMOS-Technologie. Wirkung ähnlich b), nur mit komplementärem Transistorpaar.

d) mechanischer Kontakt (zum Vergleich).

Abb. 2.15 Der Feldeffekttransistor (FET) als Schalter. S - Source, D - Drain, G - Gate. 1) Signalweg gesperrt (entspricht einem geöffneten Kontakt); 2) Signalweg durchgeschaltet (entspricht einem geschlossenem Kontakt).

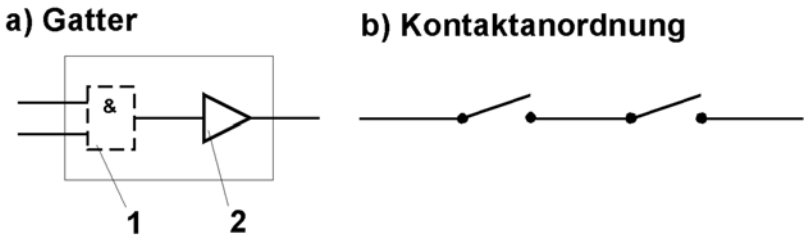


Abb. 2.16 Gatter und Kontaktanordnung. 1 - Verknüpfungschaltung; 2 - Treiberstufe.

Beide Anordnungenleisten auf den ersten Blick das Gleiche; in beiden Fällen handelt es sich um eine UND-Verknüpfung. Die Unterschiede ergeben sich aus dem grundsätzlich anderen Aufbau:

a) Gatter. Es handelt sich um elektronische Schaltungen, die aus der jeweiligen Belegung der Eingänge die entsprechende Ausgangsbelegung bilden. Gatter bestehen typischerweise aus der eigentlichen Verknüpfungsschaltung 1 und einer nachgeordneten Treiberstufe (Ausgangsstufe) 2. Sind beide Eingänge der UND-Verknüpfung 1 mit Signalen belegt, die dem Wahrheitswert Eins entsprechen, so wird die Treiberstufe 2 derart erregt, dass sie ein Ausgangssignal mit Wahrheitswert Eins abgibt. Signalflüsse in die Gegenrichtung (vom Ausgang zu den Eingängen) gibt es offensichtlich nicht; das Gatter ist rückwirkungsfrei (unidirektionaler Signalfluss).

b) Kontaktanordnung. Kontakte stellen Stromwege her oder trennen Stromwege. Es ist die Struktur (Topologie) der Kontaktanordnung, die die logische Funktion bestimmt. Der Stromweg durch die UND-Verknüpfung ist nur dann geschaltet, wenn beide Kontakte geschlossen sind – der erste UND der zweite. In welche Richtung der Strom tatsächlich fließt, ist offensichtlich gleichgültig (bidirektionaler Signalfluss).

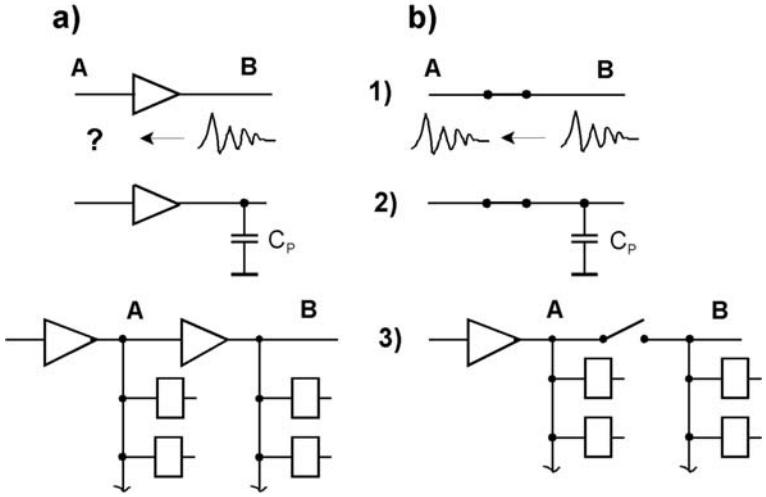


Abb. 2.17 Gatter und Schalter im Vergleich. a) Gatter (Treiberstufe); b) Schalter; 1) Rückwirkung; 2) parasitäre Kapazitäten; 3) ausgangsseitige Belastung.

a) Eine Treiberstufe (z. B. die Ausgangsstufe eines Gatters) entkoppelt beide Seiten elektrisch voneinander. Die Seite B wird aktiv getrieben und wirkt somit nicht auf Seite A zurück.

b) Ein geschlossener Schalter ist hingegen praktisch das Gleiche wie ein Stück Draht:
   1) Seite A sieht alle Störungen auf Seite B.
   2) Seite A sieht die parasitäre Kapazität C_p der Seite B.
   3) Wird der Schalter geschlossen, so muss die Treiberstufe auf Seite A auch die Lasten von Seite B mit treiben.

Gatter	Schalter
• Aktive Ausgangsstufe (Treiberstufe). • Rückwirkungsfrei. Nur Signalfluss in eine Richtung – von den Eingängen zum Ausgang (unidirektional). • Längere Verzögerungszeiten (Gatterschaltkreise typischerweise > 1 ns). • Benötigt vergleichsweise viel Siliziumfläche.	• Passive Durchreiche. • Nicht rückwirkungsfrei. Signalfluss in beide Richtungen möglich (bidirektional). • Vernachlässigbare Durchlassverzögerung (CBT-Schaltkreise typischerweise < 250 ps). • Benötigt sehr wenig Siliziumfläche.

Tabelle 2.4 Gatter- und Schalterbauelemente.

Schalter haben eine praktisch vernachlässigbare Durchlassverzögerung. Dass der Signalfluss in beide Richtungen gehen kann, ist in manchen Anwendungsfällen bedeutungslos. Gelegentlich ist es sogar ein ein Vorteil. Sie haben keine Treibfähigkeit, sind also nicht in der Lage, ein durchgeleitetes Signal zu regenerieren. Schalter sind deshalb keine Alternative zum Gatter, sondern Bauelemente für Sonderzwecke. Als Anwendungsbeispiele vgl. *****.

2.1.9 Digitale Signale

Ungenutzte Eingänge

Eingänge von Digitalschaltkreisen dürfen nie offen bleiben. Ein offener Eingang führt ein undefiniertes Potential. Er ist empfindlich gegen Störeinflüsse aus der Umgebung. Offene CMOS-Eingänge bewirken, daß Querströme durch die Eingangsstufen fließen. Ungenutzte Eingänge sind deshalb mit Festwerten zu beschalten. Die anzulegenden Pegel sind so zu wählen, dass der Schaltkreis die gewünschten Funktionen ausführen kann.

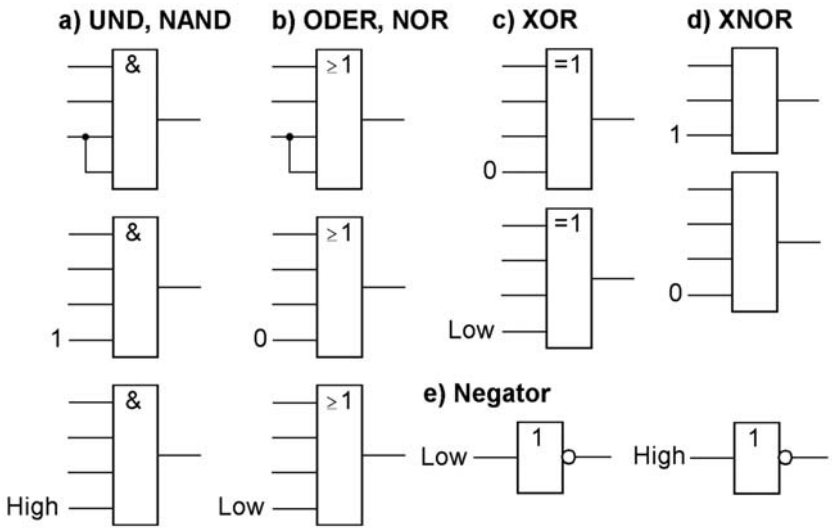


Abb. 2.18 Beschaltung ungenutzter Gattereingänge.

a) UND und NAND. Ungenutzte Eingänge können mit genutzten verbunden oder mit einem 1-Pegel belegt werden.
b) ODER und NOR. Ungenutzte Eingänge können mit genutzten verbunden oder mit einem 0-Pegel belegt werden.
c) XOR (Antivalenz). Ungenutzte Eingänge müssen mit einem 0-Pegel belegt werden.
d) XNOR (Äquivalenz). Handelt es sich um eine assoziative Verknüpfung, so kommt es auf die Anzahl der Eingänge an. Ist sie ungerade, Belegung mit 0-Pegel, ist sie gerade, Belegung mit 1-Pegel. Handelt es sich um eine invertierte XOR-Verknüpfung, Belegung mit 0-Pegel.
e) Negator. Der Einsatzfall ergibt sich, wenn das Funktionselement nicht genutzt wird. Welcher feste Pegel angelegt wird, ist gleichgültig.

Der Eingang soll...	**Der Eingang wirkt...**	**Anzulegender Pegel**
Unwirksam sein	Aktiv Low	High
	Aktiv High	Low
Wirksam sein	Aktiv Low	Low
	Aktiv High	High

Tabelle 2.5 Festbeschaltung von Eingängen.

Der Low-Pegel ergibt sich auf einfachste Weise durch Anschließen an Masse (üblicherweise mit GND oder V_{SS} bezeichnet), der High-Pegel durch Anschließen an die Speisespannung (üblicherweise mit V_{CC} oder V_{DD} bezeichnet).

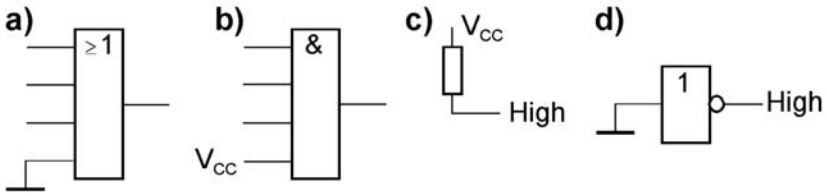


Abb. 2.19 Bildung von Festwerten. a) Low-Pegel; b), c), d) High-Pegel.

Binäre Ausgänge

Die typische Digitalschaltung arbeitet mit zwei Signalwerten. Demgemäß verhalten sich auch die Ausgänge: von Umschaltvorgängen (Signalflanken) abgesehen, führen sie entweder einen Low-Pegel oder einen High-Pegel. Die typische binäre Ausgangsstufe ist eine Gegentakttreiberstufe.

Mehrere binäre Ausgänge auf eine gemeinsame Leitung arbeiten lassen

Es ist ersichtlich, dass es nicht funktioniert – wenn der eine Ausgang auf High-Pegel geschaltet ist und der andere auf Low-Pegel, fließt ein Kurzschlussstrom von der Speisespannung nach Masse.

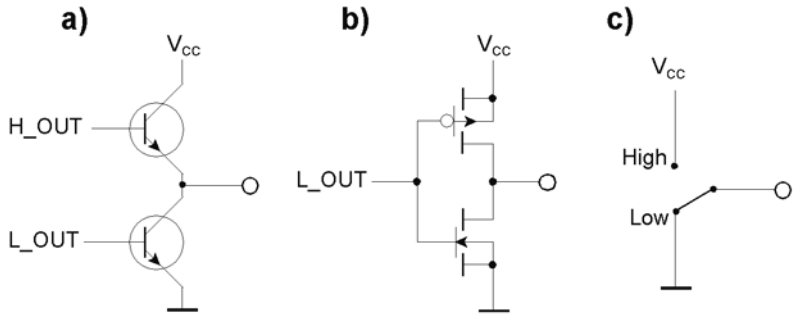


Abb. 2.20 Binärer Ausgang als Gegentakttreiberstufe. a) Grundschaltung mit Bipolartransistoren; b) CMOS-Grundschaltung; c) Ersatzschaltung zur Veranschaulichung der Funktionsweise. Je nachdem, ob ein Low- oder ein High-Pegel auszugeben ist, wird der Anschluss mit Masse oder mit der Versorgungsspannung verbunden. L_OUT und H_OUT sind Ansteuersignale zum Ausgeben eines Low- oder High-Pegels.

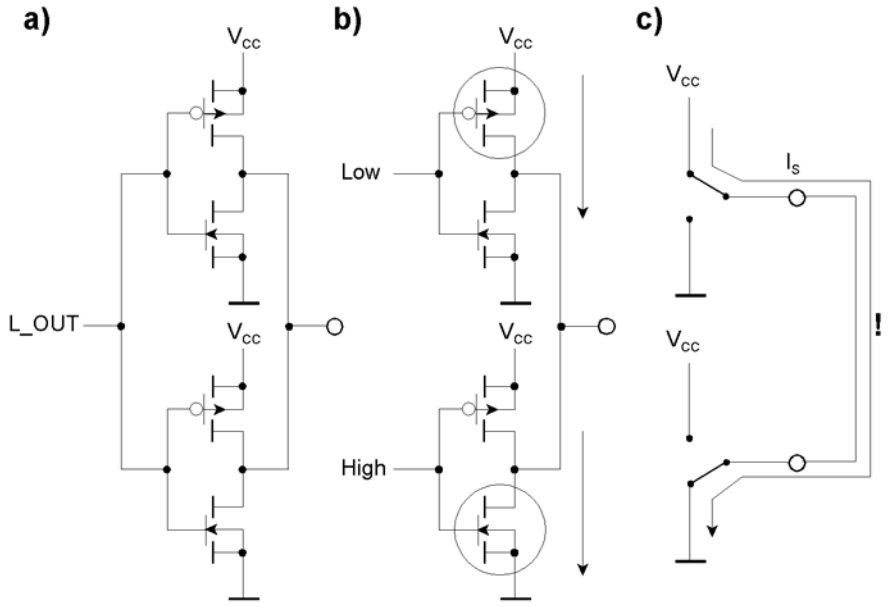


Abb. 2.21 Parallelschaltung binärer Gegentaktausgänge. a) grundsätzlich zulässig (Erhöhung des Treibvermögens); b) unzulässig (Kurzschlussstrom bei unterschiedlicher Erregung); c) Ersatzschaltung zu b).

Transistorstufen mit Arbeitswiderstand: Open Collector, Open Emitter, Open Drain

Diese binären Ausgänge sind mit einem einzigen Transistor bestückt. Nur bei einem der beiden Logikpegel ist der Transistor aktiv; der jeweils andere Pegel ergibt sich über einen (typischerweise außen anzuschließenden) Widerstand (Arbeitswiderstand). Bei Open Collector und Open Drain ist der aktive Pegel der Low-Pegel, bei Open Emitter ist es der High-Pegel. Derartige Ausgangsstufen vertragen das Zusammenschalten, da der fließende Strom vom Arbeitswiderstand begrenzt wird.

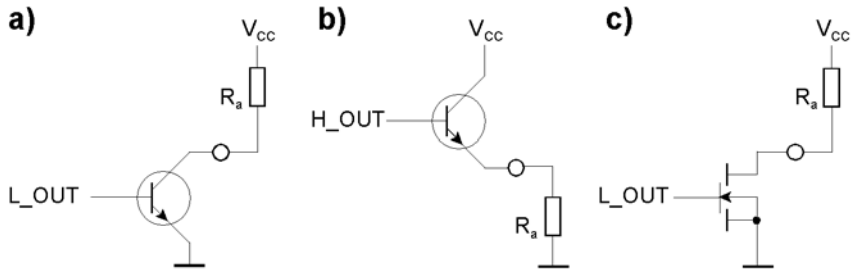


Abb. 2.22 Binärer Ausgänge mit nur einem Transistor. a) Open Collector; b) Open Emitter; c) Open Drain.

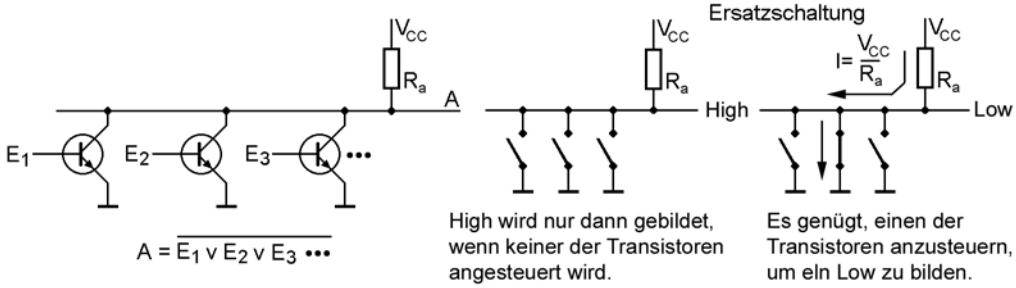


Abb. 2.23 Parallelschaltung von Open-Collector-Stufen.

Jeder erregte Transistor zieht die gemeinsame Ausgangsleitung auf Low. Sie führt einen High-Pegel nur dann, wenn kein Ausgang auf Low geschaltet ist. Aus Sicht der positiven Logik entspricht dies einer UND-Funktion, aus Sicht der negativen Logik einer ODER-Funktion (High nur dann, wenn alle Ausgänge auf High geschaltet sind, Low bereits dann, wenn nur ein Ausgang auf Low geschaltet ist). Weil diese Verknüpfungen praktisch über den Draht zustande kommen, werden sie als Wired AND und Wired OR bezeichnet. Ein wesentlicher Nachteil dieser Einfachschaltungen besteht darin, dass nur der Übergang in einen Zustand (z. B. Low) ein aktives Durchschalten ist, der Übergang in den anderen Zustand (z. B. High) hingegen ein mehr oder weniger allmähliches Hochlaufen, weil die parasitären Kapazitäten nur über den Arbeitswiderstand umgeladen werden können.

Tri-State-Ausgänge

Dieses Prinzip wurde entwickelt, um Busleitungen schnell zwischen beiden Signalpegeln umzuschalten, dabei aber ohne Arbeitswiderstand auszukommen und so den zusätzlichen Stromfluss durch die jeweils aktive Treiberstufe zu vermeiden. Die Lösung: eine Gegentaktstufe, die einen weiteren (den dritten) Zustand einnehmen kann. Eine solche Tri-State-Treiberstufe hat zwei Eingänge:

- Das eigentliche (aufzuschaltende) Datensignal.
- Ein Aufschalterlaubnissignal (Enable-Signal). Ist dieses Signal inaktiv, so befindet sich die Stufe ausganggsseitig im dritten Zustand, ist es aktiv, wird die Datensignalbelegung zur Busleitung durchgeschaltet; die Stufe verhält sich wie ein binärer Gegentakttreiber.

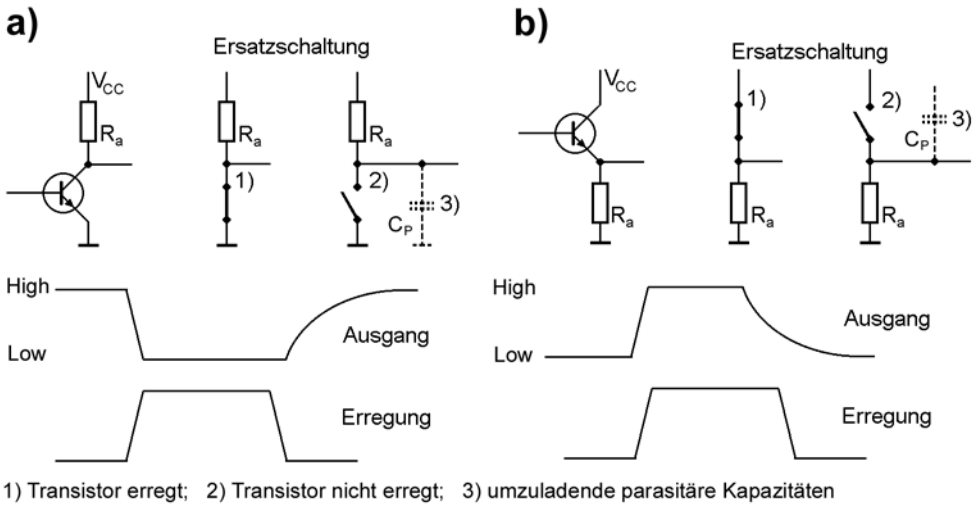


Abb. 2.24 Der geschaltete Arbeitswiderstand führt zum Verschleifen der Signalflanke beim Übergang in den inaktiven Pegel. a) Open Collector / Open Drain; b) Open Emitter. 1 - Transistor erregt; 2 - Transistor gesperrt; 3 - zu entladende parasitäre Kapazitäten.

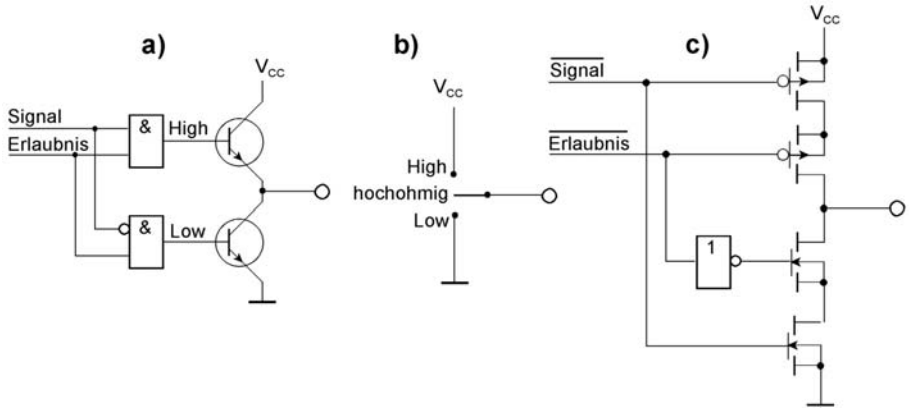


Abb. 2.25 Tri-State-Ausgang. a) Prinzipschaltung; b) Ersatzschaltung; c) Tri-State-Stufe in CMOS-Technologie.

Im dritten Zustand liefert die Stufe praktisch nichts – sie schaltet überhaupt keinen Signalpegel auf. Somit gibt sie anderen Stufen Gelegenheit zum Aufschalten. Auf diese Weise können mehrere Ausgänge auf einen einzigen Signalweg arbeiten (Busbetrieb).

Die typischen Vorteile:

- hohe Flankensteilheit in beiden Richtungen (Low– High und High – Low),
- geringe Stromaufnahme.

Übliche Fachbegriffe: der Ausgang ist hochohmig oder hat eine hohe Impedanz (High Impedance, High Z, tristated, Z = ∞ u. dergl.).

Die meisten Schaltkreise, die zum Anschließen an Busleitungen vorgesehen sind, haben Tri-State-Ausgänge.

2.2 Schaltkreise

2.2.1 Standardisierte elementare Schaltfunktionen

Bei den hier in Rede stehenden Schaltkreisen unterscheidet man folgende Stufen der Schaltungsintegration:

- SSI (Small Scale Integration): ein Schaltkreis, der nur wenige Gatterfunktionen enthält (Beispiel: maximal 12 Gatterfunktionen).
- MSI (Medium Scale Integration): ein Schaltkreis, der eine mittlere Anzahl an Gatterfunktionen enthält (Beispiel: zwischen 13 und 99 Gatterfunktionen). Der Schaltkreis stellt typischerweise eine kompliziertere Funktion dar (Zähler, Decoder, Multiplexer usw.).
- LSI (Large Scale Integration): damit bezeichnet man alle noch komplizierteren Schaltkreise (von 100 Gatterfunktionen an aufwärts).

Solche Bauelemente können aus dem Regal entnommen werden (Off-the-Shelf-Schaltkreise). Elementare digitale Schaltkreise gibt es seit den 60er Jahren; es handelt sich also um einen gesättigten Stand der Technik. Die Hersteller fertigen die Bauelemente, die nachgefragt werden. Es gibt Typen, die Jahrzehnte alt sind, aber noch in riesigen Stückzahlen hergestellt werden, weil sie für bestimmte Aufgaben unentbehrlich sind. Moderne Entwicklungen betreffen:

- Verringerte Versorgungsspannungen (von 3,3 V an abwärts bis hin zu Werten unter 1 V).
- Miniaturisierte Gehäuse.
- Die Ausrichtung auf zwei Typensortimente: (1) Bustreiber und andere Interface-Koppelstufen, (2) Grundfunktionen zum Aufbauen der sog. Restlogik (Glue Logic). Diese besteht heutzutage zumeist aus sehr einfachen Schaltungen, die an verschiedenen Stellen auf der Leiterplatte benötigt werden (z. B. eine NAND-Verknüpfung vor einem Speicherschaltkreis oder ein Flipflop vor einem Mikrocontroller). Schaltkreise mit mehreren Funktionselementen kann man oftmals gar nicht richtig ausnutzen. Deshalb hat man den Integrationsgrad verringert (einzelne Gatter und Flipflops) und die Gehäuse extrem verkleinert.

Ein wichtiger Vorteil der Standardschaltkreise
Im Gegensatz zu den meisten der modernen programmierbaren Schaltkreise sind sie sofort nach dem Einschalten betriebsfähig; die Signale an ihren Ausgängen laufen mit der Versorgungsspannung hoch. Sie eignen sich also gut für Schaltungen der Initialisierung, der Stromversorgungs- und Einschaltsteuerung usw.

Mit Standardschaltkreisen entwerfen
Das traditionelle Entwurfsziel: so wenige Schaltkreisgehäuse wie möglich – gleichgültig, was sie enthalten (Minimum Package Count). Die Ausnutzung der Schaltkreise ist wichtiger als eine akademische Schaltungsminimierung. Entwirft man mit dieser Zielstellung, so zerlegt

man das Problem zunächst soweit, dass sich erkennen lässt, an welchen Stellen Schaltkreise mittleren Integrationsgrades einzusetzen sind, also Decoder, Multiplexer, Register, Zähler usw. Die Notwendigkeit, Gatterschaltkreise einzusetzen, ergibt sich nur dann, wenn die höher integrierten Schaltkreise nicht ohne weiteres aneinander passen oder wenn sie ohne Zusatzbeschaltung nicht alle Funktionen ausführen. Wer in dieser Tradition aufgewachsen ist, entwirft typischerweise einen Zähler oder einen Decoder nicht von Grund auf, sondern wird sofort mit der Überlegung beginnen, ob beispielsweise der 74161 oder der 74193 besser geeignet ist, um das aktuelle Entwurfsproblem zu lösen. Manche Entwicklungssysteme unterstützen diese Vorgehensweise, indem sie entsprechende Funktionselemente in ihren Bibliotheken vorhalten.

Ältere Schaltkreistypen einsetzen
Beim heutigen Stand der Technik realisiert man alles, was über ein paar Gatter hinausgeht, mit programmierbaren Schaltkreisen. Die Halbleiterindustrie hat sich darauf eingestellt. Manche traditionellen Typen werden nur noch gefertigt, um die laufende Produktion der Gerätehersteller beliefern zu können. Es wird ausdrücklich davon abgeraten, sie in neuen Vorhaben einzusetzen – auch wenn die Bauelemente noch in den Katalogen angeboten werden. Bei den Herstellern nachsehen (Internet), welchen Status diese Bauelemente haben und ob sie noch für Neuentwicklungen empfohlen werden. Ein weiteres Anzeichen: Werden diese Funktionen auch in neueren Baureihen weitergeführt oder nicht?

2.2.2 Anwendungsspezifische Schaltkreise

Geht es um höchsten Integrationsgrad, um höchste Geschwindigkeit und – vor allem – um sehr hohe Stückzahlen, so sind anwendungs- oder kundenspezifische Schaltkreise (Application Specific Integrated Circuits; ASICs) in Betracht zu ziehen.

Mit der Weiterentwicklung der programmierbaren Schaltkreise verschieben sich die Stückzahlen, bei denen sich der Übergang zum ASIC lohnt, weiter nach oben. Es gibt Hersteller, die auch für Stückzahlen von 500 000 und mehr programmierbare Schaltkreise bevorzugen, vor allem Typen, die man in der Anwendungsschaltung programmieren kann. Die Vorteile:

- Programmierbare Schaltkreise ermöglichen die Fertigung verschiedener Einrichtungen auf Grundlage einer einzigen “Plattform” (ein einziger Grund-Entwurf, womöglich sogar ein einziger Leiterplatten-Typ, der selektiv bestückt und programmiert wird).
- Man kann am Markt schnell reagieren (Änderungen, Modernisierungen, Ausbügeln von Fehlern, Anpassungen an neue Standards).

Gattermatrix (Gate Array)
Der Schaltkreis hat eine Grundstruktur aus einzelnen Gattern, die in Form einer Matrix angeordnet sind. Diese Matrix ist am Schaltkreisrand von Ein- und Ausgangsstufen umgeben. Zwischen den Gatter-Reihen sind Verdrahtungskanäle freigelassen, in denen die Gatter auf anwendungsspezifische Weise untereinander verbunden werden.

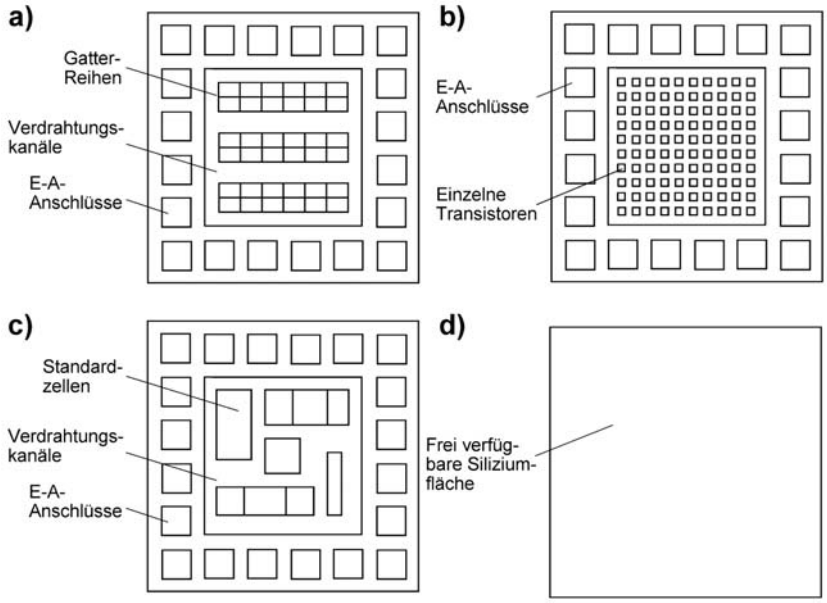


Abb. 2.26 Gate-Array-Schaltkreise. a) Gattermatrix (Gate Array); b) Transistorfeld (Sea of Gates); c) Standardzellen (Standard Cells); d) voll anwendungsspezifisch (Full Custom).

	Gate Array	**Sea of Gates**	**Standard Cell**	**Full Custom**
Chip-Abmessungen	Festgelegt (als Baureihe mit unterschiedlicher Gatteranzahl vorgefertigt)	Festgelegt (als Baureihe mit unterschiedlicher Transistoranzahl vorgefertigt)	Je nach Schaltung	Je nach Schaltung
Anzahl der kundenspezifischen Masken	1 oder 2	2	Voller Maskensatz (11...14); keine Vorfertigung möglich	Voller Maskensatz; keine Vorfertigung möglich
Zeit für Fertigungsanlauf	Serie nach ca. 6 Monaten	Serie nach ca. 6 Monaten	Serie nach 8...12 Monaten	Serien nach 1½ Jahren
Sinnvolle Stückzahlen*)	Unter 30 000	Unter 30 000	Über 30 000	Über 0,5 Mio
Bemerkungen		Erlaubt bei gleicher Chipgröße höhere Komplexität als ein Gate Array	Flächenausnutzung besser (ca. 20%) als bei Gate Array; preisgünstiger	Bestmögliche Flächenausnutzung (50 % besser im Vergleich zu Gate Array); bei hohen Stückzahlen am preisgünstigsten

*): S. Text.

Tabelle 2.6 ASIC-Prinzipien (nach [2.53]).

Transistorfeld (Sea of Gates)
Die wörtliche Übersetzung gibt die Besonderheit nicht genau wieder: es handelt sich nicht um Gatter, sondern um eine große Anzahl von Transistoren, die am Schaltkreisrand von Ein- und Ausgangsstufen umgeben sind. Besondere Verdrahtungskanäle sind nicht vorgesehen. Die Transistoren werden nach Bedarf zu Gattern und anderen Funktionseinheiten verschaltet. Verbindungsleitungen werden über Transistoren hinweggeführt (derart überdeckte Transistoren können nicht funktionell ausgenutzt werden).

Typische Zellen
Einfache Zellen bestehen aus wenigen Bauelementen, beispielsweise aus vier Feldeffekttransistoren, die ein NAND oder ein NOR mit zwei Eingängen bilden. Daraus lassen sich alle komplexeren Funktionselemente aufbauen. Ein typisches Sortiment an Zellen umfasst:

- NAND-Gatter mit 2, 3, 4, 6 und 8 Eingängen,
- NOR-Gatter mit 2, 3 und 4 Eingängen,
- elementare Gatternetze,
- Inverter,
- Ein- und Ausgangsstufen (u. a. Schmitt-Trigger, mit Widerständen, Open Drain, Tri State),
- Multiplexer 2 zu 1, 4 zu 1 und 8 zu 1,
- Latches und Flipflops.

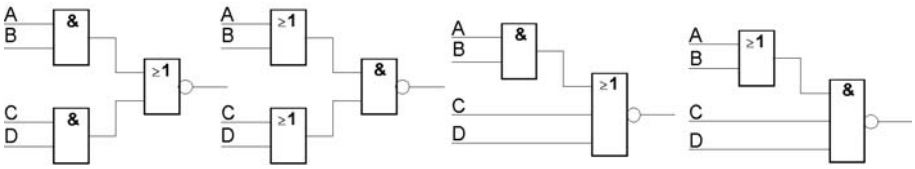


Abb. 2.27 Elementare Gatternetze nach JEDEC 12-3.

Standardzellenschaltkreise (Standard Cells)
Diese Schaltkreise enthalten Funktionseinheiten (Makro- und Megazellen), die der Entwickler aus einer Zellenbibliothek auswählt. Die Ein- und Ausgangsstufen sowie die Verdrahtungskanäle ergeben sich aus der Zellenauswahl. Als Zellen sind Gatter, Auswahlschaltungen (Multiplexer), Decoder, Zähler usw. bis hin zu kompletten Mikrocontrollern und Prozessoren aller Leistungsklassen nutzbar.

Voll anwendungsspezifische Schaltkreise (Full Custom)
Bei solchen Schaltkreisen ist nichts vorgefertigt oder vordefiniert. Vielmehr wird der gesamte Schaltkreis von Grund auf für die jeweils gewünschte Funktion entwickelt.

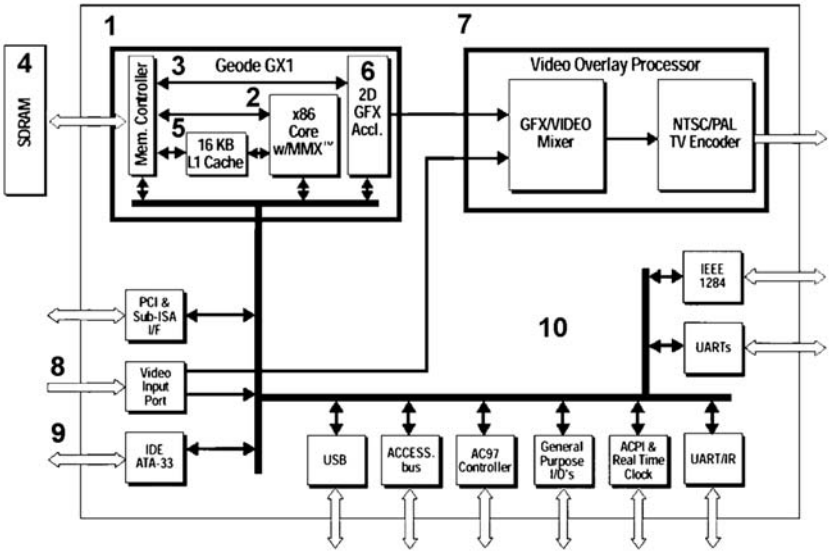


Abb. 2.28 Ein Standardzellenschaltkreis (National Semiconductor / AMD). Es handelt sich um ein (nahezu) komplettes System auf einem Schaltkreis (Geode SC1200). Anwendung: u. a. in Fernseh-Vorsatzgeräten (Set-top Boxes). 1 - Prozessormodul Geode GX1 (besteht seinerseits aus mehreren Standardzellen); 2 - der eigentliche Prozessorkern (Pentium P5-Klasse mit MMX, 266 MHz); 3 - Speichersteuerung; 4 - Arbeitsspeicher (SDRAMs); 5 - Cache; 6 - Graphikbeschleuniger; 7 - Videoüberlagerung und fernsehgerechte Codierung (NTSC/PAL); 8 - Videoeingang; 9 - Festplatteninterface; 10 - Anschlusssteuerungen für PC-typische Schnittstellen.

2.2.3 Programmierbare Schaltkreise

Ein programmierbarer Schaltkreis enthält keine bestimmten Funktionen, sondern vielseitig nutzbare Elementarstrukturen (Zellen, schaltbare Verbindungen usw.). Erst durch Programmieren erhält man die jeweils gewünschten Funktionen. Mit programmierbarer Logik wollen die Schaltkreishersteller gleichsam einen Mittelweg anbieten: Verfügbarkeit aus dem Regal, einfache Entwicklungsabläufe, leichte Änderbarkeit, Wirtschaftlichkeit selbst bei Einzelfertigung vereinigt mit den Vorzügen von ASIC-Lösungen. Hierfür ist zweierlei notwendig:

- Die Festlegung der endgültigen Funktion des Schaltkreises darf keine Fertigungsschritte der Halbleitertechnologie mehr erfordern.
- Der Schaltkreis muss sich kostengünstig fertigen lassen. Das betrifft sowohl die Ausnutzung der Siliziumfläche als auch die Gehäusekosten.

Die verschiedenen Typen der programmierbaren Schaltkreise unterscheiden sich hinsichtlich des Aufbaus der Zellen, der Anzahl der Zellen, der Auslegung der Verbindungen zwischen den Zellen und hinsichtlich des Programmierverfahrens.

Programmierverfahren	Ausführung	Änderbarkeit	Bemerkungen
Maskenprogrammierung	Beim Halb-leiterhersteller	Im einzelnen Schaltkreis nicht mehr änderbar	Nur bei extrem hohen Stückzahlen von praktischer Bedeutung**)
Durchschmelzprinzip (Fuse)	Beim Anwender*)	Im einzelnen Schaltkreis praktisch nicht mehr änderbar	Alle Verbindungen sind vor-gefertigt, die nicht benötigten werden beim Programmieren getrennt
Aufschmelzprinzip (Antifuse)	Beim Anwender*)	Im einzelnen Schaltkreis praktisch nicht mehr änderbar	Alle Verbindungsstellen sind zunächst getrennt, benötigte Verbindungen werden beim Programmieren hergestellt
Ladungsspeicherung mit UV-Löschung	Beim Anwender*)	Durch Löschen und Neuprogrammieren	Löschen durch UV-Licht; erfordert Quarzglasfenster im Schaltkreis . Es gibt auch preisgünstige Ausführungen ohne Fenster; diese kann man nicht mehr löschen (One Time Programmable; OTP).
Ladungsspeicherung mit elektrischer Löschung (EEPROM, Flash)	Beim Anwender*)	Durch Löschen und Neuprogrammieren (auch: in der Anwendungsschaltung (In System Programming)	Löschen durch elektrische Impulse
RAM-Zellen	Beim Anwender*) bzw. während des Betriebs	Durch Umladen; auch während des normalen Betriebs beliebig oft mög-lich	Halten der Information in Flipflops oder Latches; nach jedem Einschalten ist erneutes Laden erforderlich

*): Anwender = Hersteller des Gerätes oder der Funktionseinheit; **): Maskenprogrammierte Schaltkreise sind im Grunde anwendungsspezifische Schaltkreise (nicht änderbar). Manche Hersteller bieten als Dienstleistung an, Entwürfe für programmierbare Schaltkreise auf Maskenprogrammierung umzusetzen.

Tabelle 2.7 Programmierverfahren (Übersicht).

Beim aktuellen Stand der Technik lassen sich zwei grundsätzliche Auslegungen (Architekturen) der Logikzellen und drei Architekturen programmierbarer Schaltkreise unterscheiden: Makrozellen (Produkttermzellen) und Universalzellen (Zuordnerzellen) einerseits sowie GAL, CPLD und FPGA andererseits.

Makrozellen (Produkttermzellen)

Eine solche Zelle besteht aus einer UND-ODER-Struktur mit einer nachgeschalteten Ausgangszelle, die ein Flipflop und eine Ausgangsstufe enthält. Das Flipflop kann auf verschiedene Betriebsarten eingestellt oder umgangen werden. Die Anzahl der Eingänge ist vergleichsweise hoch (Richtwerte: 20 bis über 50). Jeder Eingang kann direkt oder invertiert

an jedes UND-Gatter angeschaltet werden. Jedes UND-Gatter ist somit in der Lage, einen beliebigen Produktterm über eine beliebige Auswahl von Eingangssignalen zu bilden. Eine Zelle enthält beispielsweise fünf solcher UND-Gatter, die disjunktiv verknüpft sind. Mit einer solchen Zelle kann man jede beliebige Schaltfunktion darstellen, die als disjunktive Normalform nicht mehr als fünf Produktterme aufweist. Die Anzahl der Variablen je Produktterm ist dabei gleichgültig (von Null (= nicht vorhanden) bis zur maximalen Anzahl der Eingänge).

Universalzellen (Zuordnerzellen)
Diese Zellen enthalten einen Funktionszuordner mit vergleichsweise wenigen Eingängen (Richtwert: 4...8), der jede beliebige Schaltfunktion darstellen kann. Eine typische Lösung: eine kleine RAM-Anordnung, die die Wahrheitstabelle der Schaltfunktion aufnimmt (Lookup Table LUT). Dem Ausgang des Funktionszuordners ist ein Flipflop nachgeschaltet, das auf verschiedene Betriebsarten eingestellt oder umgangen werden kann.

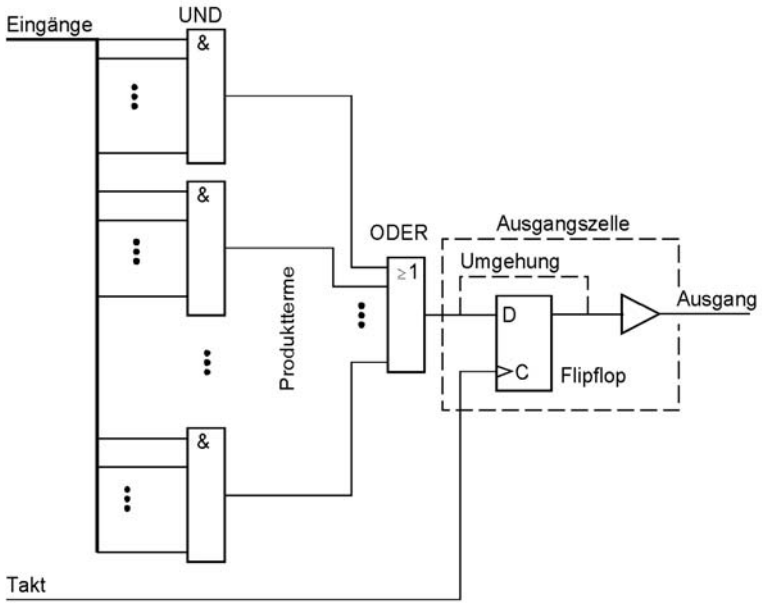


Abb. 2.29 Eine Makrozelle (Produkttermzelle) – eine dicke UND-ODER-Struktur mit nachgeschaltetem Flipflop (vereinfachte Prinzipschaltung) . Viele Eingänge (Richtwert: 20... > 50), die in mehreren UND-Gattern (Richtwert: 4...8) konjunktiv verknüpft werden können.

GAL (Generic Array Logic)
GAL-Schaltkreise enthalten eine vergleichsweise geringe Anzahl an Makrozellen (Richtwert: 8...16). Deren Ein- und Ausgänge sind auf Schaltkreisanschlüsse geführt.

CPLD (Complex Programmable Logic Device)
CPLD-Schaltkreise enthalten eine größeren Anzahl an Makrozellen (Richtwert: 32...> 200). Üblicherweise fasst man unter diesem Begriff all das zusammen, was komplexer ist als eine einfache Zusammenstellung einiger Makrozellen, aber nicht so komplex wie ein FPGA. Die

Ein- und Ausgänge sind von den Makrozellen getrennt. Makrozellen und E-A-Blöcke können über ein programmierbares Koppelfeld untereinander verbunden werden.

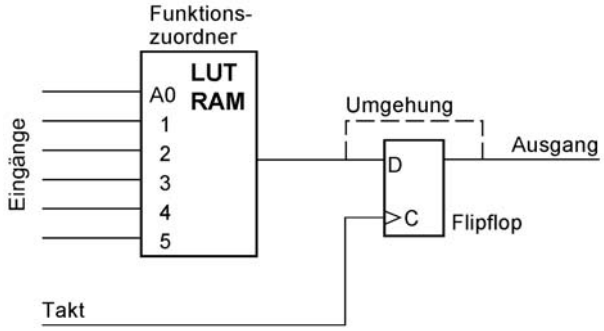


Abb. 2.30 Eine Universalzelle – ein Funktionszuordner (Lookup Table LUT) mit nachgeschaltetem Flipflop (vereinfachte Prinzipschaltung). Vergleichsweise wenige Eingänge. Die Kompliziertheit der Schaltfunktion spielt keine Rolle (es sind beliebige Verknüpfung möglich).

FPGA (Field Programmable Gate Array)

FPGA-Schaltkreise enthalten eine vergleichsweise große Zahl an Universalzellen (Richtwert: 2000 bis einige hunderttausend (nach oben offen)). Die Schaltkreisanschlüsse sind mit programmierbaren E-A-Zellen verbunden. Universal- und E-A-Zellen sind in ein Netzwerk von Verbindungswegen eingebettet. FPGA ist ein Sammelbegriff für Schaltkreise, die es ermöglichen, eine Vielzahl programmierbarer Zellen freizügig untereinander zu verbinden, so dass – wenigstens näherungsweise – ein ähnlicher Grad der Schaltungsintegration erreicht werden kann wie bei Nutzung kundenspezifischer Gate-Array-Schaltkreise. Größere FPGA-Schaltkreise können zusätzliche Funktionseinheiten enthalten, von Multplizierern und Speicherblöcken bis hin zu kompletten Prozessoren.

Die Übergänge sind fließend

Ob ein Schaltkreis als CPLD oder als FPGA bezeichnet wird, ist manchmal lediglich eine Sache der Marketing-Abteilung des Herstellers. Offensichtlich kann man die Ressourcen des Schaltkreises besser ausnutzen, wenn die Zellen kleiner sind und flexibel zusammengeschaltet werden können. Beim herkömmlichen CPLD belegt ein NAND mit zwei Eingängen ebensoviel Platz wie eine Schaltfunktion mit fünf Produkttermen von 28 Variablen. Andererseits ist die zweistufige UND-ODER-Struktur mit nachgeschaltetem Flipflop viel leichter zu überschauen, vor allem in Hinsicht auf die Schaltzeitberechnung. Kurz gesagt, die Nutzung eines CPLDs bereitet deutlich weniger Mühe als die eines FPGAs. Mittlerweile ist man aber in der Lage, ein (kleineres) FPGA so darzustellen, als handele es sich um ein CPLD (vergleichsweise einfaches Entwerfen, geringe Kosten, Betriebsbereitschaft unmittelbar nach dem Einschalten, feste Programmierung[3)]. Deshalb werden solche Schaltkreisfamilien bisweilen als CPLDs bezeichnet.

3: Zumindest anscheinend – manche Typen beruhen auf RAM-Zuordnern, die beim Einschalten aus einem eingebauten Flash-ROM so geladen werden, daß es praktisch nicht auffällt ...

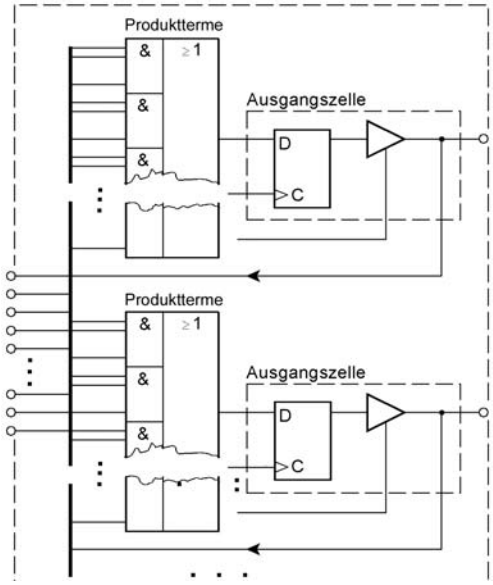


Abb. 2.31 Mehrere von außen zugängliche Makrozellen bilden einen GAL-Schaltkreis.

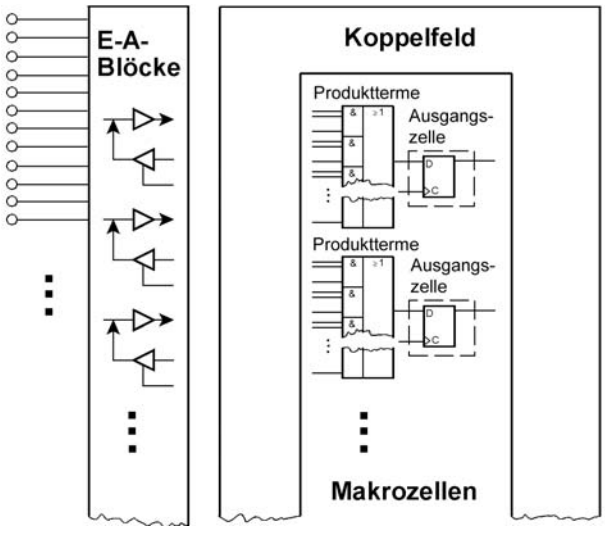


Abb. 2.32 CPLDs enthalten viele Makrozellen, die im Innern des Schaltkreises untereinander verbunden werden können.

Mit programmierbaren Schaltkreisen entwerfen

Von Hand ausgeführten Entwurfsschritte können mit Fehlern, von Programmen ausgeführte zudem mit Überraschungen[4)] behaftet sein. Der entscheidende Ansatz zur Verringerung der

4: Beispiele: die Schaltung braucht zuviel Platz oder passt gar nicht in den Schaltkreis, es wurden unnötige Funktionen implementiert, die Schaltung läuft nicht mit der vorgesehenen Taktfrequenz. Zudem können die Programmsysteme und Bibliotheken Fehler aller Art enthalten.

Fehlerrate besteht darin, die eigentliche Entwurfsabsicht so genau und zutreffend zu erfassen wie nur irgend möglich. Viele Entwurfsaufgaben sind im Grunde gar nicht so kompliziert. Man muss sie nur in Teile zerlegen. Dabei ergeben sich meist Teilaufgaben, die mit bekannten Schaltungen zu lösen sind. Hierbei versteht es sich von selbst, fertige Schaltungslösungen einzusetzen, wenn immer dies zweckmäßig ist.

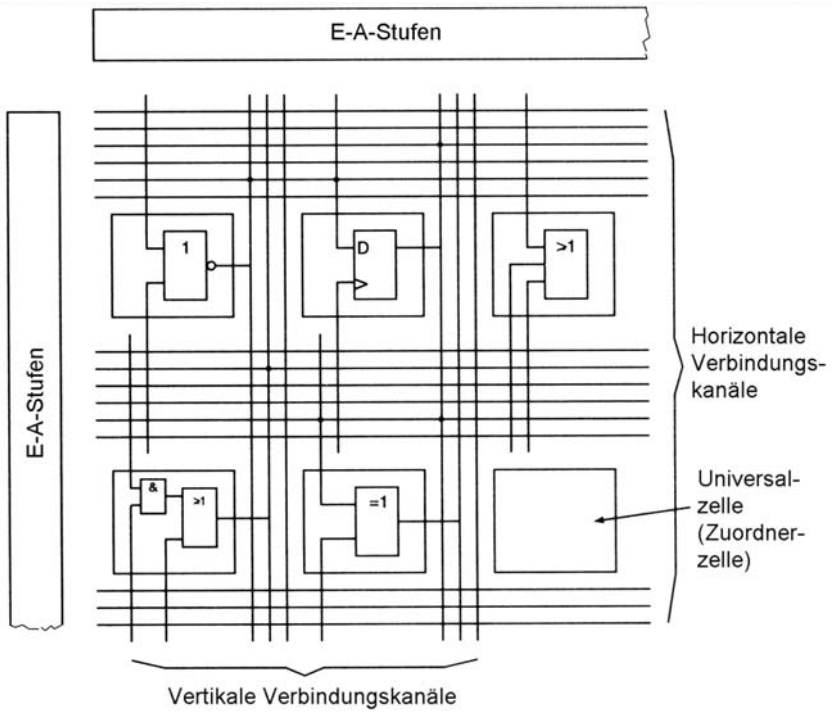


Abb. 2.33 FPGAs enthalten viele kleine Universalzellen, die in ein Netzwerk programmierbarer Verbindungskanäle eingebettet sind. Eine typische FPGA-Zelle entspricht einer kombinatorischen Verknüpfung von wenigen Eingängen und einem nachgeschalteten Flipflop.

Entwerfen über Schaltplan

Der Schaltplan wird am Bildschirm eingegeben. Die Implementierung bis hin zur Erzeugung der Programmierdaten erledigt das Entwicklungssystem. Das ist der typische Entwurfsablauf bei kleineren Projekten mit CPLDs und FPGAs. Nutzt man moderne Entwicklungssysteme, kann man es sich leisten, die Schaltung so einzugeben, dass sie die Entwurfsabsicht so klar und verständlich wie möglich darstellt – gleichgültig wie aufwendig die Lösung erscheinen mag. Tricksen mit Gattern und Optimieren von Hand lohnt sich nicht – die Maschine kann das viel besser. Auf die Innenschaltung des Schaltkreises muss man zunächst keine Rücksicht nehmen. Manche Aufgaben – beispielsweise das Entwerfen von Zustandsautomaten – lassen sich mit speziellen Entwicklungswerkzeugen erledigen.

Entwerfen auf Grundlage von Hardwarebeschreibungssprachen

Das ist das typische Entwurfsverfahren bei größeren Projekten. Zumeist wird man die Entwurfsabsicht nicht von Hand in eine (mit den Ausdrucksmitteln der Sprache

darzustellende) Schaltungsstruktur umsetzen, sondern formal beschreiben, wie die Schaltung funktionieren soll. Die Umsetzung dieser Verhaltensbeschreibung (Behavioral Description) in eine Schaltungsstruktur ist Sache des Entwicklungssystems. Derartige formale Beschreibungen werden vor allem zu zwei Zwecken verwendet:

- Zum Simulieren der Entwurfslösung.
- Zum Implementieren einer entsprechenden Schaltung (Schaltungssynthese).

Sollen die Ergebnisse der Schaltungssynthese wirklich brauchbar sein, heißt es, beim Eingeben des Entwurfs mitzudenken. Man kann an einen Schaltungsentwurf keineswegs so herangehen wie an ein Anwendungsprogramm für Personalcomputer. Die Sprache ist nur ein Werkzeug zum Formulieren. Was wird wohl das System aus den eingegebenen Anweisungen erzeugen? Nicht selten muss man dem Entwicklungssystem klar machen, was man eigentlich haben will. Die Hersteller geben hierzu Richtinien und Musterformulierungen an (Coding Styles). Um eine effektive Implementierung zu unterstützen, ist es gelegentlich besser, komplexe Bibliotheksfunktionen nicht zu nutzen und stattdessen die Entwurfsabsicht mit elementaren Zuweisungen auszudrücken.

2.3 Impulse

2.3.1 Kennwerte

Ein Impuls ist ein Signalverlauf, der von einem logischen Wert (einem Signalpegel) zum jeweils anderen logischen Wert übergeht und nach einer gewissen Zeit (der Impulsdauer) zum ursprünglichen Logikpegel zurückkehrt.

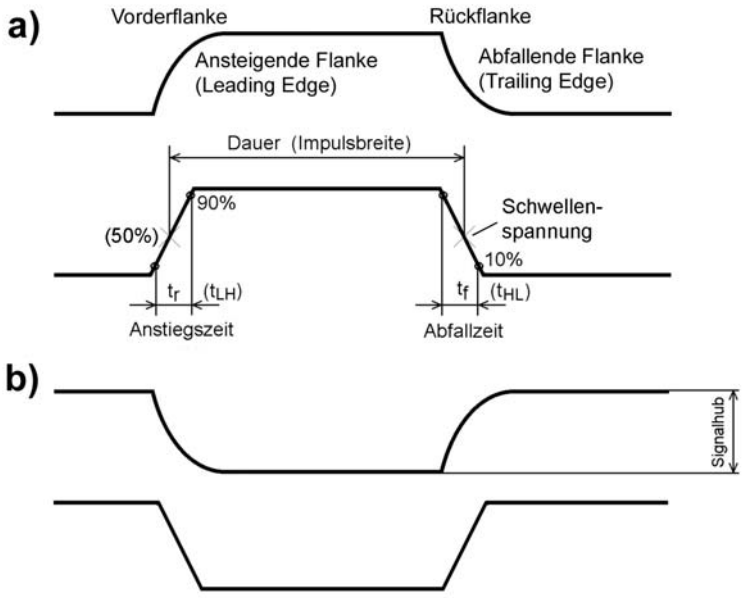


Abb. 2.34 Impulsverläufe. a) positiver Impuls (High-Impuls); b) negativer Impuls (Low-Impuls). Jeweils oben ein näherungsweise tatsächlicher, darunter ein linearisierter Signalverlauf.

Impulsflanken

Die Flanken (Edges) begrenzen den Impuls; sie bilden die Übergänge zwischen den beiden logischen Pegeln. Die Bezeichnungen Anstiegs- und Abfallflanke, steigende bzw. fallende Flanke oder Low-High- bzw. High-Low-Flanke beziehen sich direkt auf die Richtung der Pegeländerung. Hingegen bezeichnen die Begriffe Vorderflanke (Leading Edge) und Rückflanke (Trailing Edge) die Signalwechsel am Anfang und am Ende des Impulses ohne direkten Bezug zur Richtung der Signaländerung. In der Praxis kann es keine echten Sprünge geben, sondern nur stetige Übergänge. Steile Impulse können zudem vor dem Übergang in den jeweiligen Logikpegel Über- und Unterschwinger aufweisen (ein geringfügiges Über- und Unterschwingen ist kein Fehler). Solche Einzelheiten kann man aber typischerweise vernachlässigen; die Übergänge zwischen den logischen Pegeln werden durch einen linearisierten Verlauf mit hinreichender Genauigkeit beschrieben.

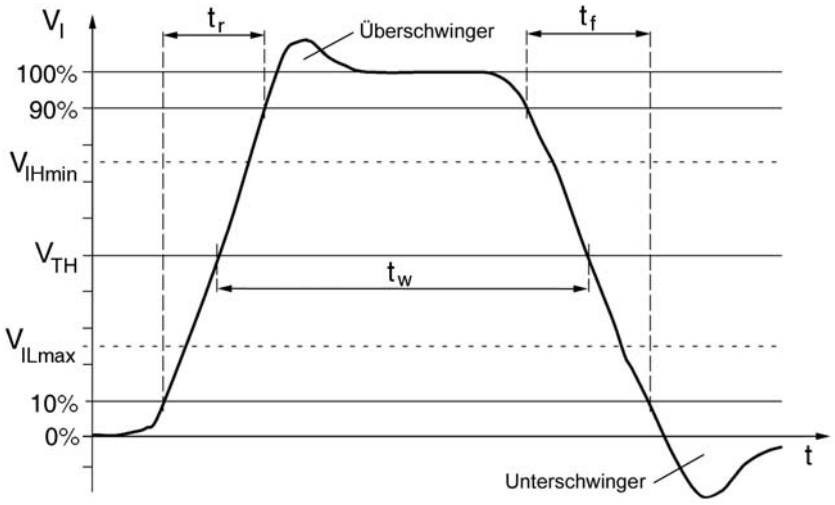


Abb. 2.35 Zur Definition der Impulskennwerte.

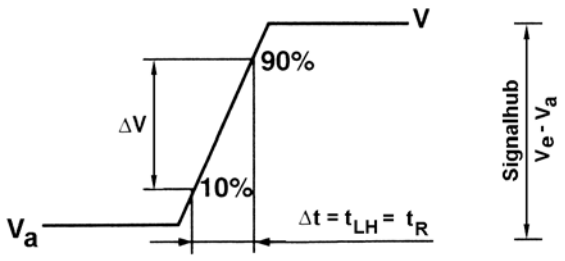


Abb. 2.36 Kennwerte von Signalflanken. V_a - Anfangswert; V_e - Endwert.

Der Signalhub

Impulskennwerte im Datenmaterial beziehen sich auf die Schaltkreiseingänge. Der Signalhub oder Spannungshub (Signal oder Voltage Swing) ist die Differenz zwischen Endwert V_e und Anfangswert V_a, also praktisch zwischen High-Pegel und Low-Pegel. Über- und Unterschwinger werden nicht berücksichtigt.

Anstiegs- und Abfallzeiten
Diese Zeitangaben kennzeichnen die Dauer der jeweiligen Flanke:

- Die Anstiegszeit betrifft die Low-High-Flanke (Rise Time t_r oder t_{LH}).
- Die Abfallzeit betrifft die High-Low-Flanke (Fall Time t_f oder t_{HL}).

Die Zeitkennwerte werden typischerweise zwischen 10% und 90% des Signalhubs gemessen ($t_{10\text{-}90}$), manchmal auch zwischen 20% und 80% ($t_{20\text{-}80}$).

Typische Logikspezifikationen geben höchstzulässige Anstiegszeiten (als Maximalwerte) vor (der Bereich der Schwellenspanung soll möglichst schnell durchlaufen werden). Viel hilft aber nicht immer viel: extrem steile Flanken sorgen für Störungen im System und für eine erhöhte Störstrahlung. Deshalb fordern neuere Spezifikationen auch Minimalwerte (die Flanken dürfen nicht steiler sein als unbedingt nötig).

Impulsfolgen
Digitalsignale sind zumeist Impulsfolgen. Neben der Impulsdauer ist der Abstand zwischen den einzelnen Impulsen von Bedeutung. Es gibt unregelmäßige Impulsfolgen, bei denen der Abstand zwischen den Einzelimpulsen sich ständig ändert (vielfach ändert sich auch die Impulsdauer). Demgegenüber bleiben bei regelmäßigen Impulsfolgen Impulsdauer und Impulsabstand jeweils gleich.

Taktimpulse
Die in der Praxis wichtigsten regelmäßigen Impulsfolgen sind die Takte. Takte bestimmen den Arbeitsrhythmus sequentieller Schaltungen.

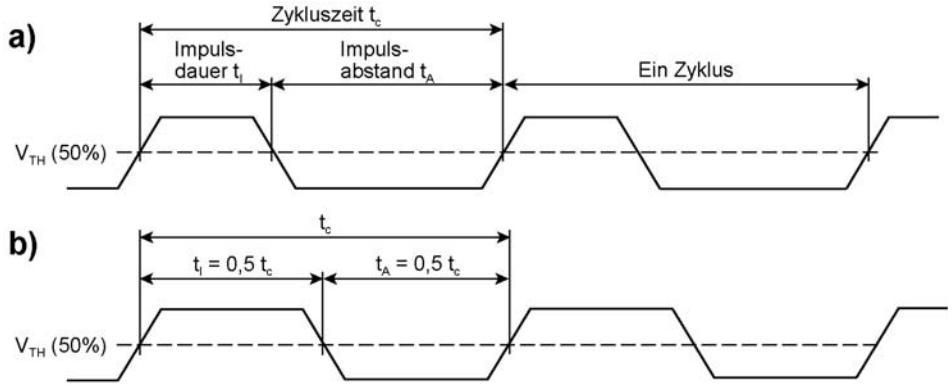


Abb. 2.37 Parameter regelmäßiger Impulsfolgen. a) allgemein; b) symmetrisch.

Takt und Daten
Es ist offensichtlich, dass es Probleme geben wird, wenn alle Signale gleichzeitig umschalten. Takt- und Strobe-Impulse müssen also einen bestimmten Zeitbezug zu jenen Signalen haben, deren Übernahme, Auswertung usw. sie steuern. Die einschlägigen Kennwerte betreffen Zeitabschnitte, in denen die Datenbelegung stabil anliegen muss. Hierbei bezieht man sich

typischerweise auf die Schwellenspannung V_{TH} (bei CMOS-Pegeln also auf 50 % des Signalhubs).

Vorhaltezeit (Setup Time t_{su})
Die Vorhaltezeit ist die Dauer des Zeitabschnitts, in der die Signale, die auf den Gültigkeitsimpuls bezogen werden (z. B. Datenbelegungen), vor der jeweiligen Flanke des Gültigkeitsimpulses gültig sein, also stabil anliegen müssen.

Haltezeit (Hold Time t_h)
Nach der betreffenden Flanke des Gültigkeitsimpulses müssen die auf ihn bezogenen Signale weiterhin eine gewisse Zeit stabil gehalten werden; sie dürfen sich gegenüber dem Wert, den sie zur Vorhaltezeit hatten, nicht ändern.

Die Minimalwerte der Vorhalte- und Haltezeit sind im Datenblatt vermerkt. Für viele Flipfloptypen ist eine Haltezeit von Null spezifiziert. Das bedeutet, die Eingangsbelegung darf sich sofort ändern, nachdem die jeweilige Taktflanke die Schwellenspannung durchlaufen hat.

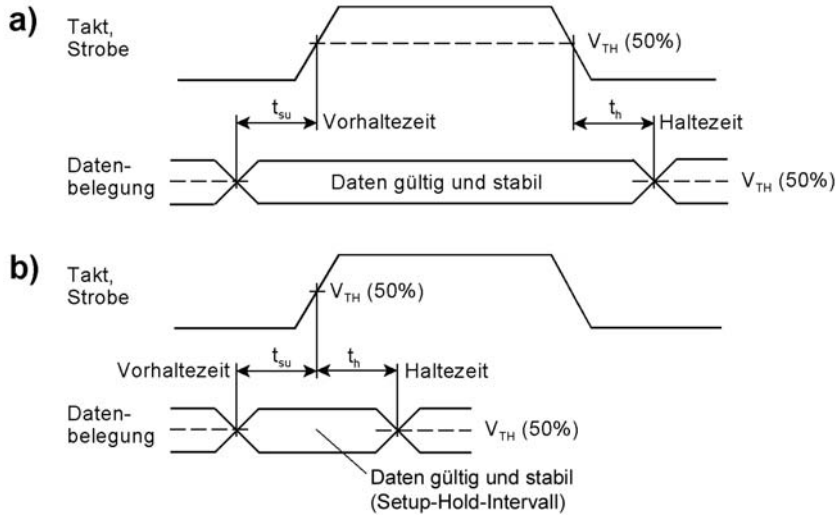


Abb. 2.38 Zeitliche Beziehungen zwischen einem Gültigkeitsimpuls (Takt, Strobe) und den zugehörigen Datensignalen. a) Pegelsteuerung; b) Flankensteuerung.

Wirkprinzipien der Datenübernahme
Die Wirkprinzipien der Datenübernahme unterscheiden sich danach, ob sich die Zeitkennwerte auf beide Flanken oder auf nur eine Flanke des Gültigkeitsimpulses beziehen.

Pegelsteuerung
Jeder der beiden Zeitkennwerte bezieht sich auf eine andere Flanke. Die Datenbelegung muss vor der Aktivierung des Takt- oder Strobe-Impulses stabil anliegen. Sie darf sich während der gesamten Impulsdauer nicht ändern. Beispiele: (1) die herkömmliche parallele Schnittstelle der PCs; (2) die ursprünglichen Master-Slave-Flipflops.

Hinweis: Beim Latch beziehen sich die Zeitkennwerte auf nur eine Flanke. Das Latch gilt jedoch als pegelgesteuert, weil sich bei aktivem Taktpegel die Eingangsbelegung direkt auf die Ausgangsbelegung auswirkt (transparentes Verhalten).

Zweiflankensteuerung
Die eine Taktflanke bewirkt die Datenübernahme, die andere die Datenausgabe Beide Zeitkennwerte beziehen sich auf die Flanke der Datenübernahme; die übernommenen Daten erscheinen jedoch erst mit der anderen Taktflanke am Ausgang. Beispiele: Master-Slave-Flipflops mit Datenverriegelung.

Einflankensteuerung (Flankensteuerung)
Die Datenübernahme wird von nur einer Flanke des Takt- oder Strobesignals gesteuert. Die Datenbelegung muss nur während des Anstiegs dieser Flanke stabil anliegen (Vorhaltezeit + Haltezeit). Beispiele: alle modernen Flipfloptypen. Diese Auslegung wird zumeist nur kurz als Flankensteuerung bezeichnet.

Doppelte Datenrate (DDR)
Beide Flanken werden zur Datenübernahme ausgenutzt. Es handelt sich aber um eine Einflankensteuerung (eine Taktflanke = ein Datenübernahmevorgang). Ein DDR-Flipflop ist ein Verbund aus zwei flankengesteuerten Flipflops.

2.3.2 Unerwünschte Impulse

Spikes und Glitches
Beide Fachbegriffe bezeichnen Störimpulse, also zeitweilige Übergänge von Null nach Eins oder umgekehrt, die eigentlich nicht vorkommen sollten. Dafür, dass sie oft vorkommen, gibt es vielfältige Ursachen, die teils elektrischer, teils logisch-funktioneller Natur sind. Die folgenden Betrachtungen beschränken sich auf störende Impulse, die aufgrund logisch-funktioneller Zusammenhänge auftreten.

Races und Hazards
Die grundsätzliche Ursache besteht darin, dass die Durchlaufzeiten durch die Gatter eines Schaltnetzes nicht genau gleich sind. Störende Impulse können dann auftreten, wenn sich Signalbelegungen ändern. Diese sog. Wettlauferscheinungen werden als Races und Hazards bezeichnet. Ein Race ist ein Wettlauf zwischen Signalflanken auf verschiedenen Leitungen (sie sollten gleichzeitig auftreten, treten aber versetzt auf). Hazards sind Wettlauferscheinungen, die Störimpulse auf einer einzigen Signalleitung hervorrufen. Hierbei sind zwei Fälle zu unterscheiden:

a) Statische Hazards. Ein statischer Hazard liegt vor, wenn die neue Eingangsbelegung die gleiche Ausgangsbelegung ergibt wie bisher. Im Idealfall bleibt der Ausgangspegel konstant. Infolge der Wettlauferscheinung ergeben sich jedoch Störimpulse.

b) Dynamische Hazards. Ein dynamischer Hazard liegt vor, wenn die neue Eingangsbelegung eine entgegengesetzte Ausgangsbelegung ergibt. Im Idealfall weist das Ausgangssignal eine einzige Flanke auf. Infolge der Wettlauferscheinung ergeben sich jedoch mehrere Signalflanken (ähnlich dem Prellen von Kontakten).

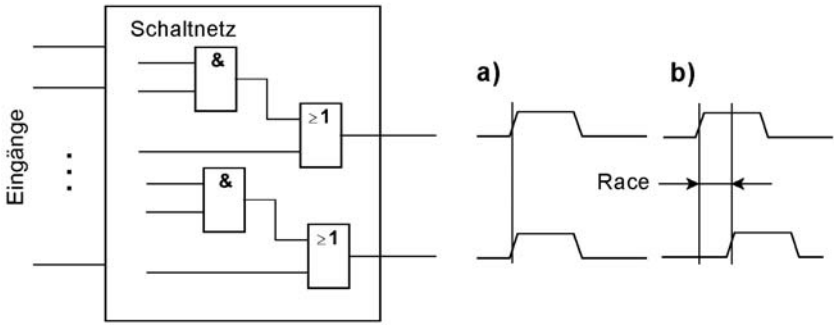


Abb. 2.39 Race – eine Wettlauferscheinung zwischen den Belegungen verschiedener Signalleitungen. a) gleichzeitige Flanken – der Idealfall. b) in der Praxis treten jedoch die Flanken gegeneinander versetzt auf.

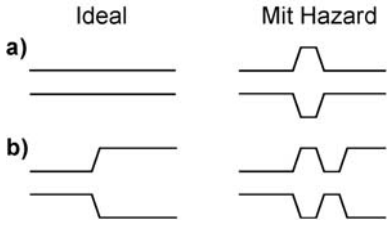


Abb. 2.40 Hazards. a) statisch. Signalpegel sollte gleichbleiben (Low der High); es entstehen aber Impulse; b) dynamisch. Eigentlich sollte nur eine Signalflanke auftreten. Es treten aber mehrere auf (Prelleffekt)

Races vermeiden?

Sie sind grundsätzlich nur dann zu vermeiden, wenn sich zu einer Zeit nur die Belegung einer einzigen Signalleitung ändert. Man müsste also die Signalcodierung passend festlegen (einschrittige Codierung). Das gelingt aber nicht immer.

Hazards vermeiden?

Die Theorie untersucht, welche Variablen sich gleichzeitig ändern. Handelt es sich um einen statischen Hazard und bilden die sich ändernden k Variablen einen Unterraum, dessen 2^k Punkten derselbe Funktionswert zugeordnet ist, so spricht man von einem logischen oder Strukturhazard. Das Gegenteil – den Punkten dieses Unterraums sind verschiedene Funktionswerte zugeordnet – heißt Funktionshazard. Ob es sich um einen Struktur- oder einen Funktionshazard handelt, hängt sowohl von den zugeordneten Funktionswerten als auch davon ab, welche Signalbelegungen sich ändern und welche nicht.

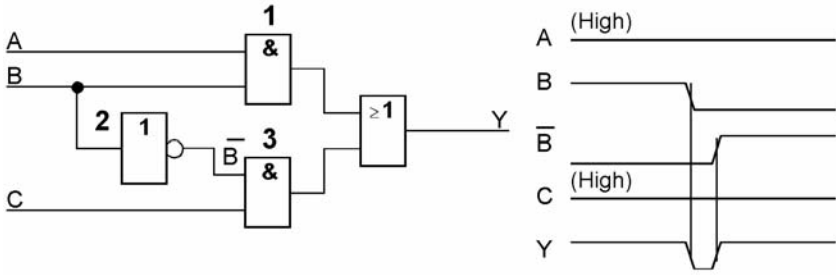


Abb. 2.41 Ein typischer Strukturhazard. Die anfängliche Belegung: A = B = C = High. B ändert sich von High auf Low. Das wirkt sich sofort auf Gatter 1 aus. Dessen Ausgang schaltet nach Low. An Gatter 3 ist infolge der Verzögerungszeit des Negators 2 $\overline{B}$ noch Low. Deshalb fällt der Ausgang Y so lange auf Low, bis $\overline{B}$ High wird.

a)

dez.	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

b)

dez.	A	B	C	U	V
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	0	0
3	0	1	1	0	0
4	1	0	0	1	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Abb. 2.42 Struktur- oder Funktionshazard? a) die Wahrheitstabelle der Schaltfunktion von Abb. 2.41. b) zwei weitere Schaltfunktionen. Nähere Erläuterung im Text.

a) Im Beispiel von Abb. 2.41 ändert sich nur die Variable B. Die Wahrheitstabelle zeigt, dass dabei der Funktionswert Y gleich bleibt. Es handelt sich also um einen Strukturhazard.

b) Die Belegung B, C ändert sich von 0, 0 auf 1, 1. Dabei können – bedingt durch die unterschiedlichen Schaltverzögerungen – kurzzeitig auch die Belegungen 0, 1 und 1, 0 anliegen. In allen vier Fällen – mit anderen Worten, im gesamten Unterraum (A = 1, B, C) – weist die Schaltfunktion U den Funktionswert 1 auf. Somit kann sich ein Strukturhazard ergeben. Die Schaltfunktion V hat hingegen im gleichen Unterraum verschiedene Funktionswerte. Das kann zu einem Funktionshazard führen.

Wettlauferscheinungen in der Praxis

Die Wunderwaffe des Praktikers der alten Schule ist der Kondensator (z. B. 1...10 nF nach Masse oder V_{CC}). Gegen richtige Hazards hilft er aber auch nicht immer (vor allem dann nicht, wenn der betreffende Schaltkreis eine etwas höhere Treibfähigkeit hat). Im hochintegrierten Schaltkreis geht es ohnehin nicht. Der Trick hat also in einer fertig

durchentwickelten Schaltung nichts zu suchen. Er ist höchstens als Hilfsmittel im Entwicklungslabor annehmbar (wenn es mit Kondensator funktioniert und ohne nicht, muss es an Glitches liegen oder ein Laufzeitprobem sein).

Hazards erkennen

Um sie messtechnisch nachzuweisen, braucht man ein Oszilloskop oder einen Logikanalysator mit Glitcherkennung. Das eigentliche Praxisproblem: Hazards können auftreten, müssen aber nicht (exemplarabhängig). Es kann sein – und es ist nicht selten vorgekommen –, dass die Erprobungsmuster perfekt funktionieren und die Störimpulse erst in Seriengeräten auftreten. Es liegt also nahe, Verfahren der Schaltalgebra und der Schaltungssimulation anzuwenden, um zu erkennen, ob Hazards überhaupt auftreten können. Der Rechenaufwand ist aber beachtlich.

Entwurfslösungen, in denen die Störeffekte keine Rolle spielen

Bei synchronem, taktgesteuertem Betrieb schaden derartige Störeffekte nicht, sofern nur der Taktzyklus lang genug ist. An kritischen Stellen (z. B. wenn Taktsignale, Rücksetzsignale o. dergl. zu bilden sind) muss man ggf. zu Fuß nachsehen:

- Decodierung. Das Ausgangssignal einer UND-Verknüpfung ist dann glitchfrei, wenn sich zu einer Zeit nur eine Signalbelegung ändert. Ändern sich mehrere gleichzeitig, kann es zu Störimpulsen kommen.
- Kritische Signale (z. B. Takte). Solche Signale am besten gar nicht mit kombinatorischen Netzwerken bilden oder nur mit solchen, die nicht von Races betroffen sind und deren Schaltverhalten nachweislich frei von Hazards ist.

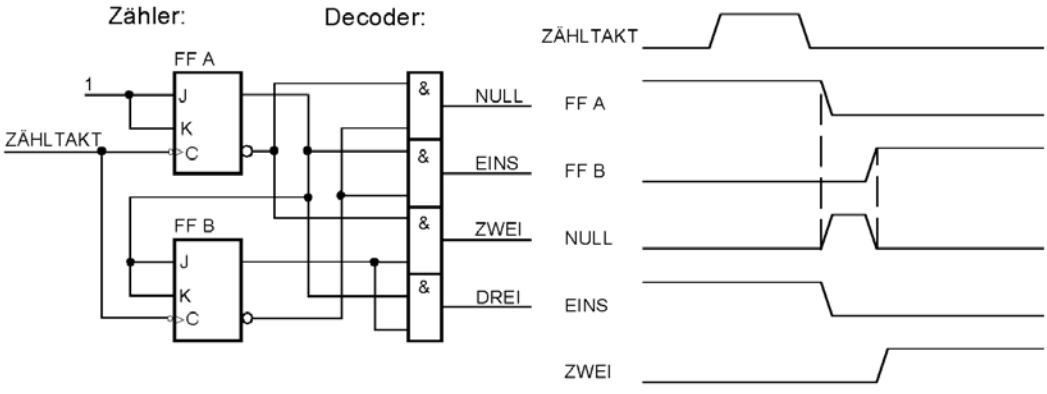


Abb. 2.43 Störimpulse in der Praxis. Es ist ein synchroner Binärzähler mit nachgeschaltetem Decoder dargestellt. Der Zähler befinde sich in Stellung EINS (Belegung 0, 1). Jetzt tritt ein Zählimpuls auf. Der Zähler gelangt in Stellung ZWEI (Belegung 1, 0), Nun schalten beide Flipflops nie mit absolut gleichen Verzögerungszeiten. Schaltet Flipflop A eher als Flipflop B, so tritt kurzzeitig die Belegung 0, 0 auf – und damit ein Störimpuls am Decoderausgang NULL. Sinngemäß wird ein Störimpuls am Decoderausgang DREI auftreten, wenn Flipflop B eher schaltet als Flipflop A (kurzzeitige Belegung 1, 1).

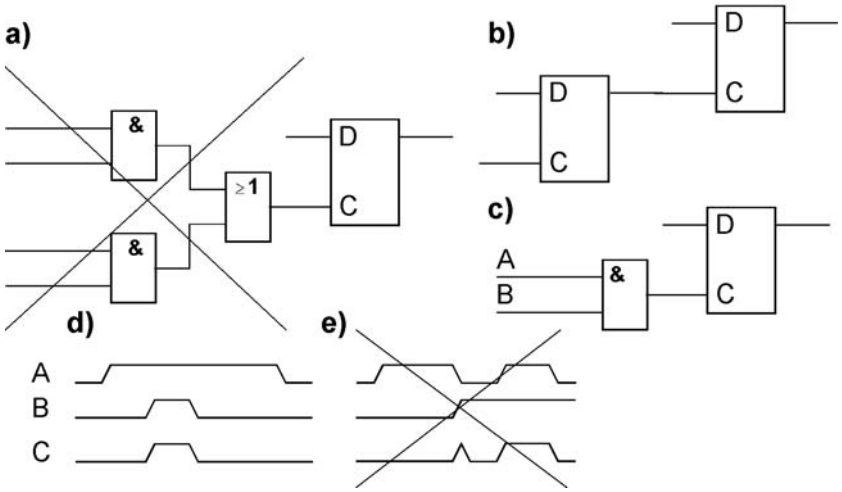


Abb. 2.44 Taktsignalbildung. a) Taktsignale nicht kombinatorisch bilden – zumindest nicht mit beliebigen Schaltnetzen. b) stets richtig. Das Taktsignal wird direkt von einem Flipflop gebildet. c) durchaus möglich – es kommt aber darauf an, wie dieSignale an den Eingängen schalten. d) zulässige Schaltfolge. Es ändert sich nur eine Signalbelegung zu einer Zeit. e) unzulässige Schaltfolge. A schaltet auf 0, B schaltet gleichzeitig auf 1. Kommt die Low-High-Flanke an B etwas eher an als die High-Low-Flanke an A, gibt es einen Glitch.

2.4 Taktsysteme

2.4.1 Zeitbestimmende Impulse

Zeitbestimmende Impulse veranlassen die Zustandsübergänge in sequentiellen Schaltungen. Sie wirken auf die Takt-, Setz- und Rücksetzeingänge der Latches und Flipflops. Die Auslegung einer sequentiellen Schaltung hängt entscheidend davon ab, welcher Art diese Impulse sind; mit anderen Worten, welche Betriebsweise dem Entwurf zugrunde gelegt wird.

Synchron und asynchron

Jedes Flipflop braucht einen Takt – es ist nur die Frage, wie er gebildet wird:

- Vollsynchroner Betrieb. Der Takt ist eine fortlaufende Folge von Impulsen, die ständig an den Takteingängen der Flipflops anliegt. Die Steuerung erfolgt allein über die Dateneingänge (Erlaubnissteuerung).
- Synchroner Betrieb mit gesteuertem Takt. Taktsignale sind fortlaufende Folgen von Impulsen. Sie liegen aber nicht ständig an den Takteingängen der Flipflops an, sondern werden dort nur wirksam, wenn die betreffenden Flipflops schalten sollen (Taktsteuerung).
- Asynchroner Betrieb. Die Taktimpulse werden nur dann gebildet, wenn Bedingungen aufgetreten sind, die Zustandübergänge oder Verarbeitungsschritte zur Folge haben, also u. a. dann, wenn ein Verarbeitungsergebnis gebildet wurde oder wenn sich Signalbelegungen geändert haben.

In der Literatur bezeichnet man oft alle Arten von Schaltwerken, die nicht von einem zentral erzeugten Takt gesteuert werden, als asynchron. In der Praxis kann man aber nur ganz einfache Schaltungen im wörtlichen Sinne asynchron auslegen, also ohne jegliche Schaltmittel, die Zeitintervalle vorgeben (mit anderen Worten: mit freien Rückführungen). Sobald es etwas komplizierter wird, braucht man zeitbestimmende Signale. "Asynchron" ist also nicht immer mit "völlig ungetaktet" gleichzusetzen.

Asynchronbetrieb
Der Zeitrahmen ist variabel; die zeitbestimmenden Signale werden in Abhängigkeit vom jeweiligen Verarbeitungsablauf gebildet.

Synchronbetrieb
Taktsignale geben einen festen Zeitrahmen für die Zustandsübergänge im System vor; die zeitbestimmenden Impulsfolgen haben feste Impulsfolgefrequenzen (Taktfrequenzen).

Asynchron oder synchron?
Die asynchrone Betriebsweise hat offensichtliche Vorteile: jede Schaltung liefert ihr Ergebnis dann, wenn sie mit dessen Bildung fertig ist, man muss dem Zeitraster keine ungünstigsten Fälle (die nur selten vorkommen) zugrunde legen, die Anpassungen an ein starres Taktschema (Synchronisation) ist nicht notwendig, wenn sich nichts ändert, schaltet auch nichts (= (nahezu) keine Störstrahlung und kein Stromverbrauch) usw. Die Aufwendungen sind aber beachtlich, ebenso die Entwurfsschwierigkeiten. Deshalb hat sich die synchrone Arbeitsweise durchgesetzt (Grundgedanke: die Schaltungen werden so einfach wie möglich ausgelegt und so schnell wie möglich betrieben). Beim derzeitigen Stand der Technik überwiegen die Vorteile der synchronen Auslegung die Nachteile bei weitem.

Funktionsentscheidend: saubere Taktsignale
Es ist offensichtlich, dass Taktsignale (nahezu) vollkommen frei von Störungen sein müssen; jeder Glitch an einem Takteingang (C, CLK) kann Schaden anrichten (auch Nadeln von nur wenigen ns Breite bringen Flipflops zum Schalten oder veranlassen die Datenübernahme in ein Latch – wenn sie es nicht gar veranlassen, Schwingungen abzugeben ...).

Taktsignale sind kritische Signale
Beim Verlegen der Taktsignalleitungen ist darauf zu achten, dass keine Störungen eingekoppelt werden und dass es keine Signalreflexionen gibt. Oftmals sind auch Vorgaben zur Signallaufzeit einzuhalten (z. B. alle Taktsignalwege mit gleicher Laufzeit). Bei der Leiterplattenentwicklung werden deshalb die Taktsignale typischerweise zuerst verlegt. In programmierbaren Schaltkreisen gibt es Taktsignalwege, die den Anforderungen an die Zuführung der Taktsignale entsprechen.

Taktsignale und Steuersignale
In den meisten sequentiellen Schaltungen hängt es von Steuersignalen ab, ob die Taktsignale auf Speicherelemente einwirken oder nicht. Es gibt zwei Wirkprinzipien: die Taktsteuerung

(Clock Gating) und die Erlaubnissteuerung (Clock Enable). Die Taktsteuerung ist bei Flipflops und Latches einsetzbar, die Erlaubnissteuerung nur bei Flipflops.

Taktsteuerung

Soll ein Taktsignal den Inhalt eines Speicherelements zeitweise nicht beeinflussen, so liegt es nahe, das Taktsignal am Takteingang einfach abzuschalten, beispielsweise über ein UND-Gatter. Steuersignale geben den Takt entweder frei oder sperren ihn. Diese Steuersignale dürfen nur dann umschalten, wenn das Taktsignal inaktiv ist; es darf nicht vorkommen, dass solche Signale sozusagen mitten im Takt ihren Pegel ändern. Zum einen ist das im Entwurf zu berücksichtigen. Zum anderen ist aber auch an Störungen zu denken, die von außen eingekoppelt werden können. Das heißt: die Steuerleitungen sind ebenso sorgfältig zu verlegen (auf der Leiterplatte oder im Schaltkreis) wie die Taktleitungen.

Erlaubnissteuerung

Die Taktimpulse liegen ständig an den Takteingängen der Flipflops an (ungesteuerter Takt). Die Steuersignale wirken nicht auf Takteingänge, sondern auf kombinatorische Auswahlschaltung, die den Dateneingängen der Flipflops vorgeschaltet sind. Somit genügt es, nur die eigentlichen Taktleitungen ausgesprochen störsicher zu verlegen.

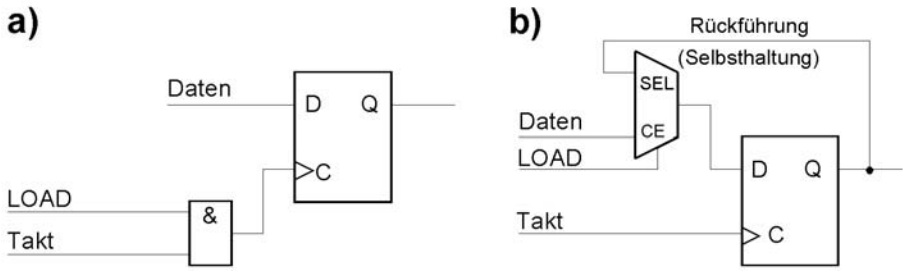


Abb. 2.45 Prinzipien der Taktsteuerung am Beispiel der Datenübernahme in ein Flipflop. Die Datenübernahme wird durch ein Taktsignal bewirkt und soll durch ein Signal LOAD gesteuert werden. Ist LOAD aktiv, so werden die Daten in dasFlipflop übernommen, ist es inaktiv, so bleibt die bisherige Belegung erhalten. a) Taktsteuerung; b) Erlaubnissteuerung. Das Erlaubnissignal (CE = Clock Enable) ist im Grunde ein Umschaltsignal zwischen dem Übernehmen der Eingangsbelegung und dem Halten der bisherigen Belegung.

Vollsynchroner Betrieb

Alle Flipflops des Schaltwerks werden mit einem einzigen, ständig anliegendem Taktsignal betrieben und ggf. über Erlaubnissignale gesteuert – mit anderen Worten, alle Steuerwirkungen werden ausschließlich über kombinatorische Verknüpfungen vor den Dateneingängen der Flipflops erbracht. Die in einem bestimmten Taktzyklus t gebildeten Signale kommen erst im folgenden Taktzyklus t + 1 an den Ausgängen der Flipflops zur Wirkung.

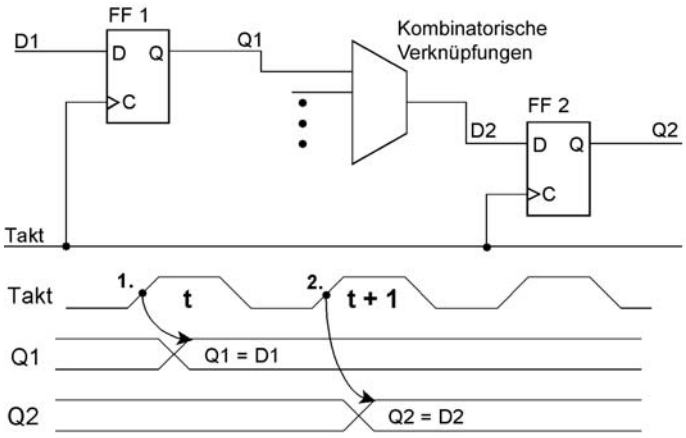


Abb. 2.46 So schalten Flipflops bei vollsynchroner Betriebsweise.

Dies ist ein einfaches Modell einer Register-Transfer-Struktur: Register (= Flipflops) – Kombinatorik – Register. Eine bestimmte Datenbelegung D1 wird im ersten Taktzyklus (= zum Zeitpunkt t) in das erste Flipflop (FF 1) übernommen. Nun wird die nachgeschaltete Kombinatorik durchlaufen. Die sich dadurch ergebende Signalbelegung wird erst mit Beginn des zweiten Taktzyklus (= zum Zeitpunkt t+1) in das zweite Flipflop (FF 2) übernommen. Am Ausgang Q2 zeigt sich die Reaktion auf D1 also erst nach der Low-High-Flanke des zweiten Taktzyklus.

Takt und Daten

Auf den Takt bezogene Datensignale an den Eingängen der Flipflops müssen nur im zeitlichen Umfeld der wirksamen Taktflanken gültig sein und stabil anliegen. Dieses Zeitfenster wird durch die Vorhalte- und Haltezeiten definiert (Setup-Hold-Intervall). Wettlauferscheinungen (Races und Hazards) und anderweitig bedingte Störimpulse (Glitches) schaden nicht, sofern sie – in Hinsicht auf die Dauer des Taktzyklus – zeitig genug abklingen.

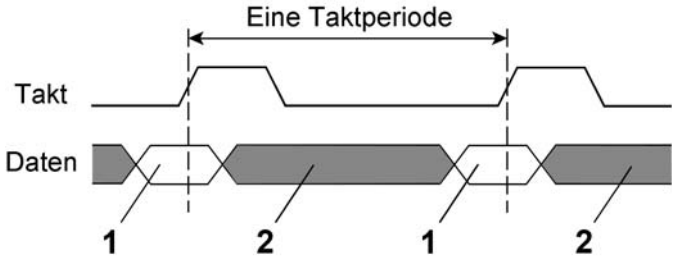


Abb. 2.47 Takt und Daten. 1 - die Datenbelegung muss gültig sein und stabil anliegen (Setup-Hold-Intervall); 2 - die Datenbelegung darf beliebig schalten.

Signallaufzeiten und Taktzyklen

Um die minimale Taktperiode (und damit die maximale Taktfrequenz) festzulegen, ist die Signalübertragung von sendenden Flipflop (Quellflipflop) 1 zum empfangenden Flipflop

(Zielflipflop) 2 im Einzelnen zu betrachten. Es sind folgende Zeitanteile von Bedeutung:

- t_{pd}: die Verzögerungszeit des Quellflipflops 1 und der nachgeschalteten kombinatorischen Verknüpfungen.
- t_{prop}:die Signallaufzeit über die Signalwege (Flight Time).
- t_{su}: am Eingang des Zielflipflops 2 muss das Signal wenigstens für die Dauer der Mindest-Vorhaltezeit t_{su} anliegen.
- t_{skew}: der Zeitversatz der Taktsignale zwischen beiden Flipflops 1,2 (Clock Skew).

$$t_{cyc} = t_{pd} + t_{prop} + t_{su} + t_{skew}$$

Alle diese Zeitanteile müssen in die Taktzykluszeit t_{cyc} hineinpassen. Jede synchrone Schaltung ist zum sicheren Funktionieren zu bringen, sofern man nur genügend Zeit einräumt, also die Taktperiode lang genug macht. Ist t_{cyc} vorgegeben, muss die Schaltung so ausgelegt werden, dass sich entsprechend kurze Zeitanteile ergeben (geringe Schaltungstiefe, kurze Signalwege, kleine Takttoleranzen). Oftmals sind Zeitbereiche spezifiziert, mit denen man auskommen muss.

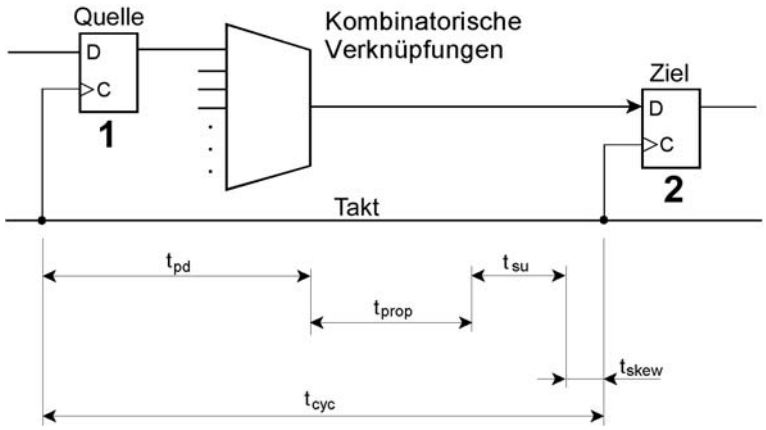


Abb. 2.48 Die Zeitanteile eines Taktzyklus. 1 - sendendes Flipflop (Quellflipflop), 2 - empfangendes Flipflop (Zielflipflop).

Die Haltezeit

Um den Haltezeitkennwert t_h des Zielflipflops auf einfachste Weise einzuhalten, muss der Minimalwert der Verzögerungszeiten größer sein als die minimale Haltezeit, und zwar im ungünstigsten Fall des Versatzes der Taktflanken (d. h. wenn der Takt am Quellflipflop um t_{skew} eher wirksam wird als am Zielflipflop). Viele Flipflops haben eine Haltezeit von Null. Wird hingegen eine bestimmte minimale Haltezeit gefordert, so muss sie auch eingehalten werden. Ein typischer Fehler ergibt sich, wenn Schaltungsteile, für die längere Haltezeitkennwerte gelten, von schnelleren Schaltmitteln angesteuert werden. Dann kann es sein, dass die Datenbelegung schon verschwunden ist, noch ehe das Taktsignal in der

langsameren Schaltung zur Wirkung kommt[5]. Wenn die Haltezeit nicht durch die sozusagen natürliche Verzögerung gewährleistet werden kann, müssen zusätzliche Verzögerungs- oder Flipflopstufen vorgesehen werden.

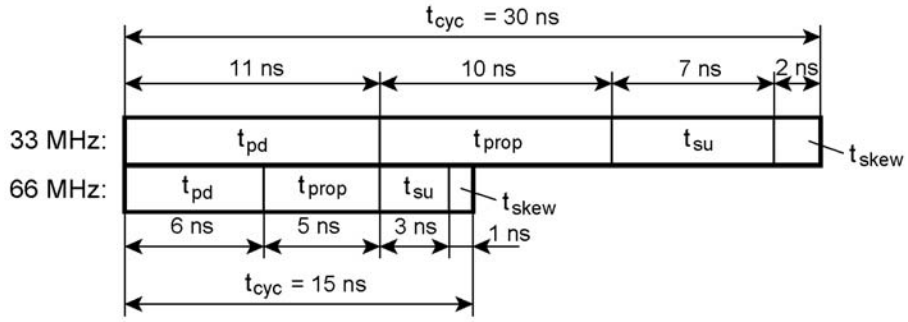


Abb. 2.49 Vorgegebene Zeitanteile (Timing Budget) am Beispiel des herkömmlichen PCI-Bus. Die Verzögerungszeit t_{pd} und die Vorhaltezeit t_{su} sind Schaltkreis-Kennwerte, die Buslaufzeit t_{prop} und der Taktversatz t_{skew} ergeben sich hingegen aus den Leitungslängen der PCI-Konfiguration. Bei 66 MHz sind die Vorgaben offensichtlich sehr knapp ...

5: Hierbei können schon einige wenige ns entscheidend sein. Wenn man nicht weiß, wonach zu suchen ist, sind solche Fehler auch messtechnisch kaum zu finden.