

Logische Zustände (LST)

Vorlesung Digitales Design

Hes·so  **VALAIS
WALLIS**



Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

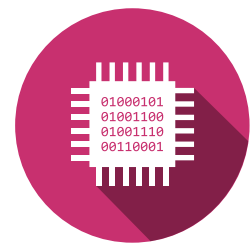
Orientierung: [Systemtechnik \(SYND\)](#)

Kurs: Digitales Design (DiD)

Verfasser: [Christophe Bianchi](#), [François Corthay](#), [Pierre Pompili](#), [Silvan Zahno](#)

Datum: 25. August 2022

Version: v2.1



Inhaltsverzeichnis

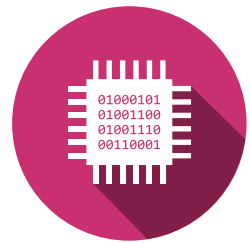
1 Einführung	2
2 Logische Basiszustände	3
2.1 Logischer Zustand '0'	3
2.2 Logischer Zustand '1'	3
2.3 Elektronische Realisierung der logischen Zustände	3
3 Logikgatter, die nur einen Zustand hervorbringen	4
3.1 Verbindung von Ausgängen	4
3.2 Open-Drain-Gatter	4
3.3 Open-Source-Gatter	5
4 Logikgatter mit hochohmigem Ausgang	7
4.1 Tri-State-Puffer	7
5 Logische Zustände für Simulatoren	8
5.1 Unbekannter Zustand	8
5.2 Nichtinitialisierter Zustand	8
6 Logischer Zustand für die Synthese	10
6.1 Nichtspezifizierter Zustand	10
Literatur	11
Akronyme	11



1 Einführung

Die logischen Zustände der Booleschen Algebra sind '0' und '1'. Die Planer von logischen Schaltungen kennen jedoch noch andere logische Zustände, mit denen die Ausgänge verschiedener Schaltungen untereinander verbunden oder die vom Simulator angezeigten Resultate verstanden werden können.

In diesem Kapitel befassen wir uns mit den **beiden logischen Basiszuständen** sowie mit **zusätzlichen Zuständen**, die für die Realisierung von Schaltungen oder für ihre Simulation benutzt werden.



2 Logische Basiszustände

2.1 Logischer Zustand '0'

Der logische Zustand '0' entspricht entweder der Bedingung "falsch" oder der Binärziffer 0.

2.2 Logischer Zustand '1'

Der logische Zustand '1' entspricht entweder der Bedingung "wahr" oder der Binärziffer 1.

2.3 Elektronische Realisierung der logischen Zustände

Für die Realisierung der Zustände '0' und '1' wird in den elektronischen Digitalschaltungen die elektrische Leitung des Ausgangs entweder zum negativen oder zum positiven Potential der Speisung gezogen.

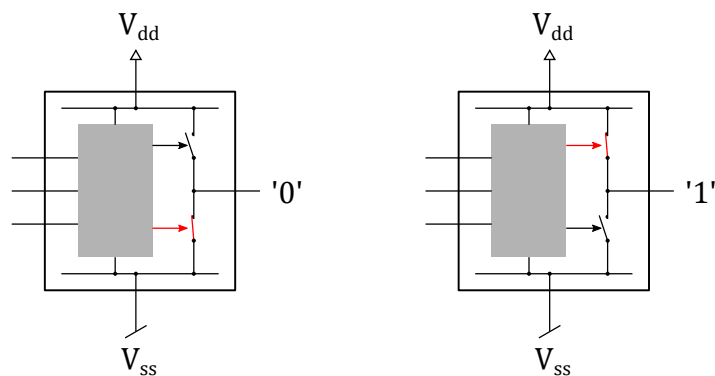
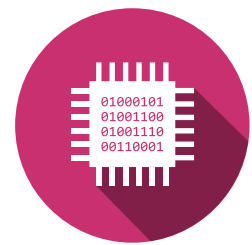


Abbildung 1: Elektronische Realisierung der Logikpegel

Die Schalter werden für die **Transistor Transistor logic** (TTL)-Schaltungen elektronisch mit Bipolartransistoren und für die **Complementary metal-oxide-semiconductor** (CMOS)-Schaltungen mit komplementären **metal-oxide-semiconductor** (MOS)-Transistoren realisiert.



3 Logikgatter, die nur einen Zustand hervorbringen

3.1 Verbindung von Ausgängen

Zwei oder mehrere Ausgänge einer Standardschaltung dürfen nicht verbunden werden, denn wenn eine Schaltung eine '0' verlangt und die andere eine '1', verursachen die beiden Schaltungen auf der Speisung einen Kurzschluss (siehe Abbildung 2).

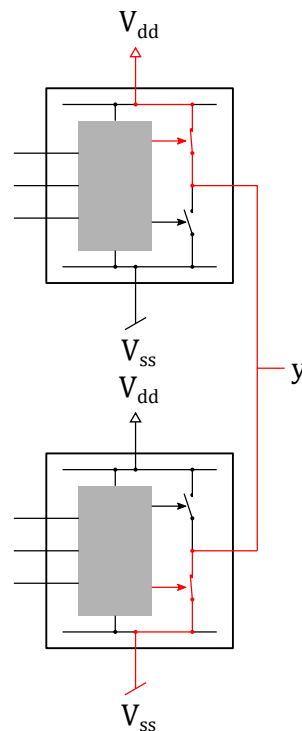


Abbildung 2: Verbindung von Ausgängen, die zu einem Kurzschluss führt

3.2 Open-Drain-Gatter

Die Open-Drain-Gatter oder die Open-Kollektor-Gatter (in der CMOS- bzw. in der TTL-Technik) zeichnen sich durch eine Ausgangsstufe aus, die nur einen Schalter zum negativen Speisepotential hin aufweist.

Diese Gatter können somit nur einen Zustand '0' hervorbringen. Damit sie benutzt werden können, muss ein Widerstand zwischen den Ausgang und die positive Klemme der Versorgung geschaltet werden, um das Ausgangspotential auf den Zustand '1' zu setzen, wenn das Gatter es nicht in den Zustand '0' zwingt. Dank dieser Eigenschaft können mehrere Ausgänge von Open-Drain-Gatter verbunden werden. In diesem Fall nimmt die entsprechende Leitung den logischen Wert an, welcher der UND-Funktion der Ausgänge entspricht, welche die Gatter erzwingen wollen. Diese Montage wird **wired- AND** (**verdrahtetes UND**) genannt. Es ist in der Abbildung 3 dargestellt.



Die Open-Drain-Gatter werden benutzt, um mehrere Schaltungen mit einer einzigen elektrischen Leitung zu verbinden, ohne diese durch ein UND-Gatter zu unterbrechen.

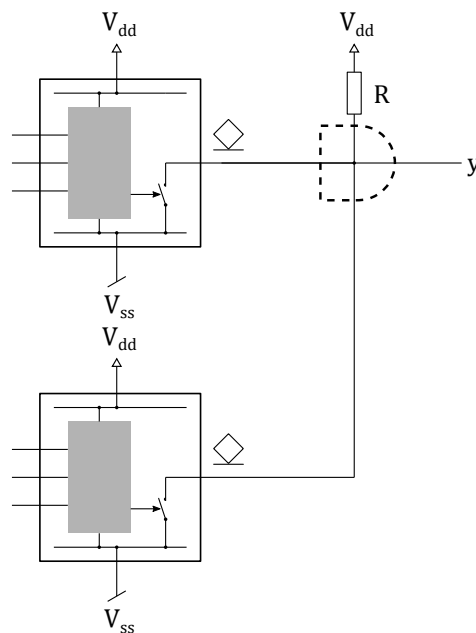
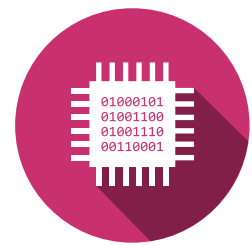
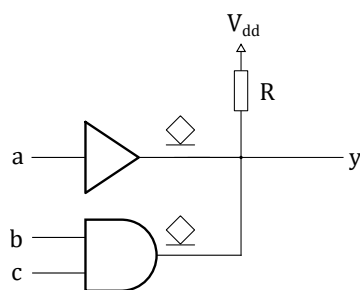


Abbildung 3: Open-Drain-Gatter

Beispiel

Abbildung 1 zeigt ein Schaltung mit Open-Drain-Gattern und gibt die Wahrheitstabelle der somit erzeugten Funktion.



<i>a</i>	<i>b</i>	<i>c</i>	<i>y</i>	<i>driver</i>
0	0	0	0	buffer, and
0	0	1	0	buffer, and
0	1	0	0	buffer, and
0	1	1	0	buffer
1	0	0	0	and
1	0	1	0	and
1	1	0	0	and
1	1	1	1	R

Tabelle 1: Schaltung mit Open-Drain-Gattern

Der Ausgang weist den Zustand '1' nur dann, wenn keines der Gatter einen Zustand '0' durchbringt.

3.3 Open-Source-Gatter

Die Open-Source-Gatter oder die Open-Emitter-Gatter (in der CMOS- bzw. in der TTL-Technik) zeichnen sich durch eine Ausgangsstufe aus, die nur einen Schalter zum positiven Speisepotential hin aufweist.

Analog zu den Open-Drain-Gatter können diese Gatter nur einen Zustand '1' hervorbringen. Zwischen den Ausgang und die negative Klemme der Versorgung muss ein Widerstand geschaltet werden. Diese Verbindung mehrerer Ausgänge von Open-Source-Gattern ergibt ein **wired-OR** (verdrahtetes ODER). Dies ist in der Abbildung 4 dargestellt.

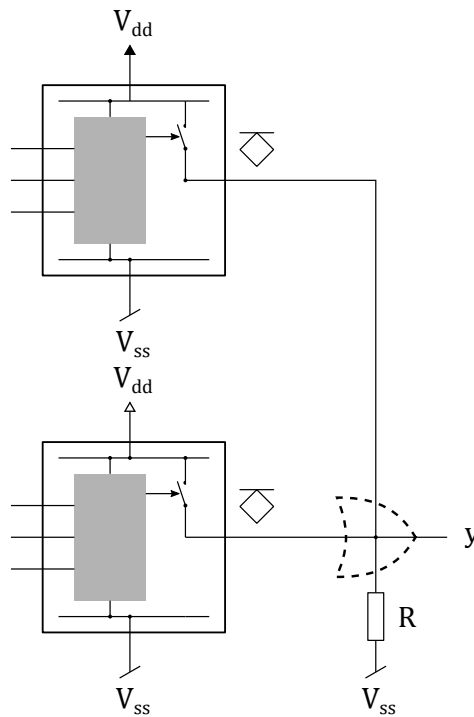
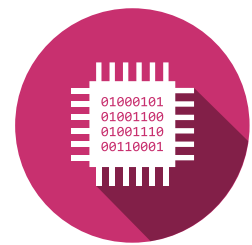


Abbildung 4: Open-Source-Gatter



Aus historischen Gründen sind die **TTL**-Gatter eher in der Lage, eine Leitung auf '0' als auf '1' zu ziehen. Es gibt viel weniger Open-Source-Schaltungen als Open-Drain-Schaltungen.

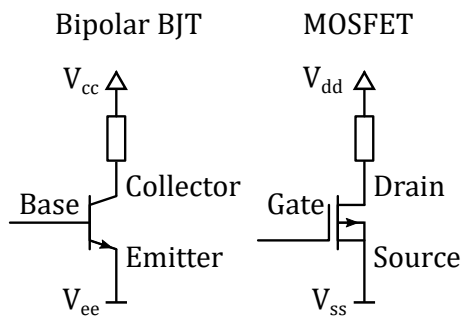
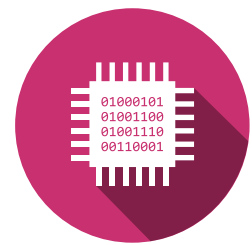


Abbildung 5: Bezeichnungen **BJT** gegenüber **MOSFET**



4 Logikgatter mit hochohmigem Ausgang

Für die Realisierung der Zustände '0' und '1' wird in den elektronischen Digitalschaltungen die elektrische Leitung des Ausgangs entweder zum negativen oder zum positiven Speisepotential gezogen. Für gewisse Logikgatter besteht zusätzlich die Möglichkeit **floating** (**gleitender**) Ausgänge, d.h. sie sind nicht an eines der Speisepotentiale gebunden (siehe Abbildung 6). Es ist dies ein sog. **high impedance**, **hi-Z** (**hochohmiger Zustand**). Er wird mit dem Buchstaben 'Z' bezeichnet.

Da diese Gatter drei möglich Ausgänge aufweisen, spricht man auch von **tri-state output** (**Tri-State-Ausgängen**).

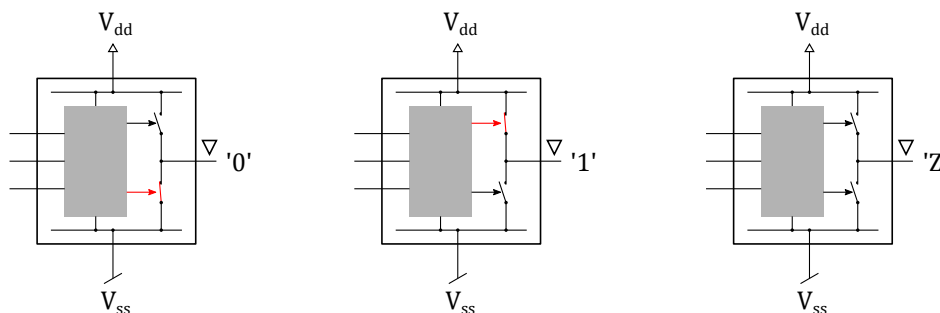


Abbildung 6: Gatter mit hochohmigem Ausgang

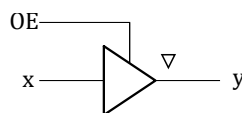


Die hochohmigen Ausgänge werden benutzt, damit mehrere Schaltungen Werte auf dieselbe Leitung oder denselben Bus schreiben können. Ein typisches Beispiel ist ein Datenbus, wo Prozessoren, Festspeicher, Lese-Schreib-Speicher und Peripheriegeräte untereinander Informationen austauschen.

Im Gegensatz zu den mit Open-Drain-Gatter gesteuerten Bussen müssen für die hochohmigen Busse die Schaltungen untereinander koordiniert werden, damit nicht zwei oder mehrere von ihnen unterschiedliche Werte auf dieselbe Leitung schreiben.

4.1 Tri-State-Puffer

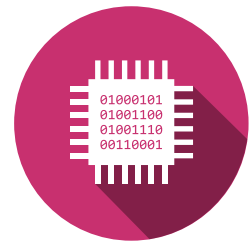
Abbildung 2 zeigt einen **tri-state buffer** (**Tri-State-Puffer**) und dessen Wahrheitstabelle.



OE	x	y
0	0	Z
0	1	Z
1	0	0
1	1	1

Tabelle 2: Tri-State-Puffer

Diese Schaltung leitet den Eingang durch, wenn der Steuersignal **Output Enable** (OE) aktiv ist.



5 Logische Zustände für Simulatoren

5.1 Unbekannter Zustand

Unter bestimmten Bedingungen, die in einer korrekt aufgebauten Schaltung nicht auftreten sollten, kann es vorkommen, dass der Simulator den logischen Zustand eines Signals nicht bestimmen kann. Dies ist z.B. der Fall, wenn ein Eingang eines Logikgatters mit nichts verbunden ist. Der Simulator gibt dies mit der Anzeige **unknown** (**unbekannter**) Zustand an. Dieser wird mit dem Buchstaben 'X' bezeichnet.

Eine Leitung, die an nichts angeschlossen ist, liefert ein hochohmiges Signal, das wie bereits gesehen mit einem 'Z' bezeichnet wird.



Abbildung 7: Unbekannter Zustand



In einer physischen Schaltung hängt das elektrische Potential einer Leitung im Zustand 'X' von den elektronischen Eigenschaften der betreffenden Logikschaltungen ab. Dieses Potential muss nicht unbedingt im Bereich sein, der die logischen Zustände '0' oder '1' definiert, was zu Problemen bei der elektronischen Funktion führen kann.

5.2 Nichtinitialisierter Zustand

Wenn man eine Schaltung unter Spannung setzt, können die von den Speicherelementen gespeicherten Werte einen beliebigen Zustand aufweisen. Die Simulatoren definieren daher in den meisten Fällen einen **uninitialized** (**nichtinitialisierten**) Zustand. Dieser Zustand wird mit dem Buchstaben 'U' bezeichnet.

Beispiel: Speicherelement

Abbildung 8 zeigt den Beispiel eines Speicherelementes. Beim Start der Schaltung und solange die Setzung auf Eins nicht aktiviert ist, kann der Ausgangszustand vom Simulator nicht bestimmt werden. Dieser zeigt daher einen Zustand 'U' an. Die physische Schaltung würde in diesem Moment bereits funktionieren, obwohl es unmöglich ist, ihren Zustand theoretisch zu bestimmen.

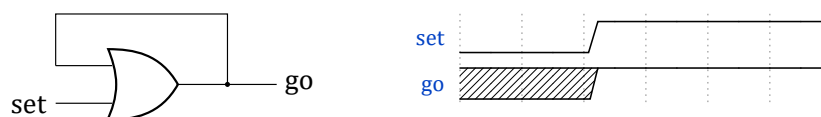
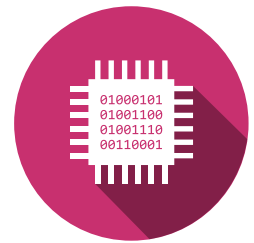


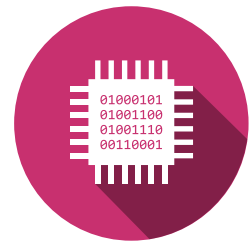
Abbildung 8: Nichtinitialisierter Zustand



Das Auftreten von Signalen im Zustand 'U', die mit einer Sequenz der Eingänge des Systems nicht in einen anderen Zustand versetzt werden können, gibt an, dass diese Schaltung, selbst wenn sie funktionell ist, nicht getestet werden kann.



Einige Simulatoren kennen den Zustand 'U' nicht; sie ersetzen ihn durch den Zustand 'X'.



6 Logischer Zustand fpr die Synthese

6.1 Nichtspezifizierter Zustand

Bei der Planung von Logikschaltungen treten gewisse Bedingungen nie auf (Bedingung never happens“) oder, wenn sie auftreten, können sie gewissen Signalen einen beliebigen Wert zuordnen (Bedingung "don't care"), da sie nicht berücksichtigt werden.

Ein nichtspezifizierter Zustand wird mit dem Buchstaben ' - ' bezeichnet.

Beispiel: Unvollständig definierte Funktion

Die 7-Segment-Anzeige der Abbildung 9 wird für die Darstellung von Dezimalzahlen benutzt. Die Funktion, die eine mit 4 Bit codierte Binärzahl in die 7 Signale umwandelt, die das Einschalten der 7 Segmente steuern, ist für die Werte definiert, die den Dezimalzahlen zwischen 0_d und 9_d entsprechen. Wenn die Zahlen 10_d bis 15_d nie auftreten, kann man die Schaltung, die für die Umwandlungsfunktion verantwortlich ist, vereinfachen.

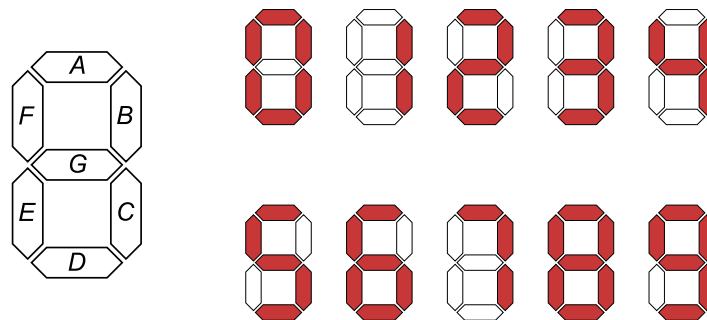


Abbildung 9: Unvollständig definierte Funktion

Die in der Abbildung 10 dargestellte Vereinfachung beruht auf der Summe der Produkte der Funktion des Segments B .

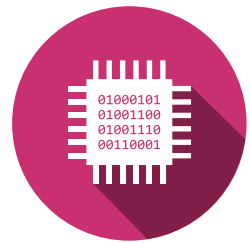
	$n_2 \quad n_3$		
	1	1	-
n_0	1	0	-
n_1	1	1	-
	1	0	-

$$\bar{n}_2 + \bar{n}_1 \bar{n}_0 + n_1 n_0$$

Abbildung 10: Vereinfachung der Funktion des Segments B



Das Resultat dieser Vereinfachung ist eine Logikschaltung, die einfacher ist als jene, für welche die nichtspezifizierten Zustände durch einen gegebenen logischen Zustand, '0' oder '1', ersetzt sind.



Literatur

- [1] Suhail Almani. *Electronic Logic Systems*. second edition. New-Jersey: Prentice-Hall, 1989.
- [2] Jean Michel Bernard und Jean Hugon. *Pratique Des Circuits Logiques*. quatrième édition. Paris: Eyrolles, 1987.
- [3] Michael D. Ciletti und M. Morris Mano. *Digital Design*. second edition. New-Jersey: Prentice-Hall, 2007.
- [4] David J. Comer. *Digital Logic and State Machine Design*. Saunders College Publishing, 1995.
- [5] Marcel Gindre und Denis Roux. *Electronique Numérique, Logique Combinatoire et Technologie*. Paris: McGraw-Hill, 1987.
- [6] Ronald J. Tocci und André Lebel. *Circuits Numériques: Théorie et Applications*. deuxième édition. Ottawa: Editions Reynald Goulet inc. / Dunod, 1996.

Akronyme

BJT bipolar junction transistor. 6

CMOS Complementary metal-oxide-semiconductor. 3–5

gleitender floating. 7

hochohmiger Zustand high impedance, hi-Z. 7

MOS metal-oxide-semiconductor. 3

MOSFET metal-oxide-semiconductor field-effect transistor. 6

nichtinitialisierten uninitialized. 8

OE Output Enable. 7

Tri-State-Ausgängen tri-state output. 7

Tri-State-Puffer tri-state buffer. 7

TTL Transistor Transistor logic. 3–6

unbekannter unknown. 8

verdrahtetes ODER wired-OR. 5

verdrahtetes UND wired- AND. 4