

4 Entwurf kombinatorischer Schaltungen an Beispielen

Lernziele: Systematisches Vorgehen zur Festlegung der Schaltungs-Funktion in umsetzungsgerechter Form, sowie zur Analyse und Entwicklung kombinatorischer digitaler Schaltungen, ihre technische Realisierung und Funktionsüberprüfung anhand von Beispielen aus vielen Anwendungsbereichen.

In Kapitel 2 und 3 wurde der Weg zu einer Schaltung mittels formeller Schritte aufgezeigt. Dabei wurden in der Regel Beispiele benutzt, die diese Schritte im Detail nachvollziehbar machen, die aber nicht unbedingt einen tieferen Sinn als Schaltungsfunktion hatten. Im Folgenden soll nun die **Funktion**, und zwar die **Funktionsdefinition** und **-findung**, sowie ihre sinnvolle und technisch effiziente **Realisierung** als digitale Schaltung und schließlich deren **Signalverhalten** im Vordergrund stehen. Folgende Überlegungen sind anzustellen:

- 1) Was soll die Aufgabe der Schaltung sein?
- 2) Was ist von anderer Quelle, von außen bekannt oder verfügbar? In welcher Form, in welchem Format?
- 3) Was ist gesucht, was soll die Schaltung liefern, wer will sie wie benutzen? In welcher Form, in welchem Format?
- 4) Wie können diese Anforderungen erfüllt, wie kann das was gesucht ist, auf das abgebildet oder von dem abgeleitet werden, was bekannt ist?
- 5) Wie kann schließlich überprüft und sichergestellt werden, ob die angestrebte Funktion tatsächlich auch erreicht wurde bzw. wird und wie kann diese auch in Zukunft sichergestellt werden.

Diese Fragen lassen sich übersetzen zu folgenden Verfahrensschritten:

1. Beschreibung der Schaltungsfunktion
2. Eingänge einschließlich Wert- und Funktionszuordnung (Kodierung)
3. Ausgänge einschließlich Wert- und Funktionszuordnung (Kodierung)
4. Zuordnung der Ausgabewerte auf die Eingaben (Wertetabelle, logische Beschreibung)
5. Formulierung der logischen Verknüpfung, ggf. mit Vereinfachung/Optimierung
6. Schaltung (Schaltplanentwurf und Realisierung)
7. Abgleich von Soll und Ist (Simulation und Messtests)

Nach diesem Fahrplan werden im Folgenden alle Schaltungen bearbeitet.

4.1 Lichtschrankenschaltung

Lichtschranken dienen zur Erkennung von Hindernissen oder Gegenständen. Hier soll die Lichtschrankentechnik nicht weiter vertieft werden. Lichtschranken bieten aber eine Reihe von anschaulichen Beispielen für Anwendungen digitaler Schaltungen. Fällt das Licht auf einen Fotodetektor, so kann das Signal dieses Detektors nach einer Schwellwertbestimmung in eine logische 1 oder 0 umgewandelt werden. Festgelegt werden muss natürlich, ob bei Blockierung des Lichtstrahls eine logische 1 oder eine logische 0 resultiert. Wie bei anderen Schaltschnittstellen können dabei auch Sicherheits- und Schaltungsausfallfragen berücksichtigt werden. Also z. B. die Frage: Welche Wertigkeit (Kodierung) ist sinnvoller für den Fall, dass die Lichtschranke einmal ausfallen, also defekt sein könnte. Soll dann die Schaltung bedingungslos freischalten oder sperren?

4.1.1 Zufahrtskontrolle

Als einfaches Beispiel betrachten wir nun eine Zufahrtskontrolle zu einem Werksgelände. Ein Pförtner überwacht die Zufahrt, die mit einer Rot-Grün-Ampel geregelt wird. Zur Arbeiterleichterung sollen PKW auch ohne weitere Kontrolle des Pförtners Grünes Licht erhalten. LKW dagegen sollen erst dann einfahren dürfen, wenn dies vom Pförtner genehmigt wurde, vgl. Abb. 4.1.1.1:

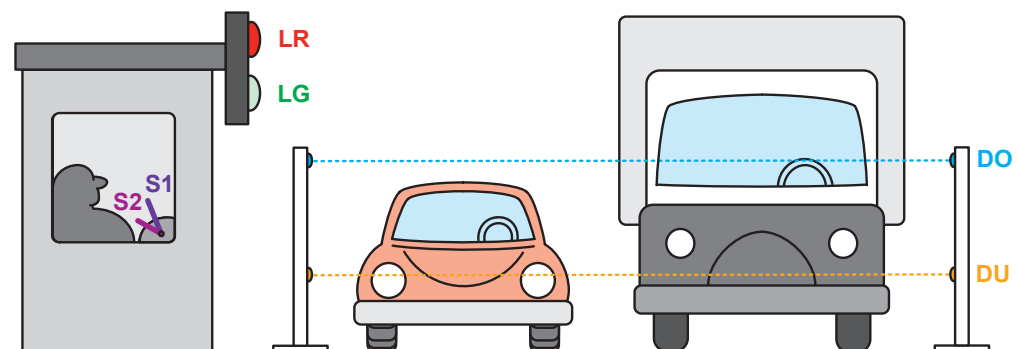


Abb. 4.1.1.1 Zufahrtskontrolle mittels Lichtschranke

Gemäß Abb. 4.1.1.1 wird mittels zweier, in verschiedenen Höhen angebrachter Lichtschranken unterschieden, ob ein PKW oder ein LKW vorfährt. Die Situation ist natürlich vereinfacht, um in einer Übungsaufgabe behandelt werden zu können. Von einer realistischeren Lösung wollen wir also absehen. Wir betrachten vorerst auch nur den Fall, dass der Pförtner (nur) einen Schalthebel hat, um die Ampel auf Grün zu schalten.

Folgende Anforderungen seien festgelegt: Ist keine der beiden Lichtschranken unterbrochen, dann bleibt die Ampel vorsorglich auf Rot, wie auch in allen Zweifelsfällen. Fährt nun ein PKW vor und unterbricht die Lichtschranke **DU**, nicht aber **DO**, so soll die Ampel solange auf Grün geschaltet werden. Werden jedoch beide Lichtschranken unterbrochen, wenn ein LKW einfährt, wird die Ampel nur dann auf Grün geschaltet, wenn der Pförtner den Schalter **S1** umlegt. Auch wenn der Pförtner vergessen sollte, den Schalter **S1** wieder zurückzulegen, so schaltet die Ampel dennoch nach Durchfahrt des LKW wieder auf Rot. Der Schalter **S2** bleibt vorerst unberücksichtigt.

- (1) Damit haben wir die erforderliche **Funktionsbeschreibung** der Schaltung.
- (2) Welche **Eingänge** liegen nun vor? Es sind die beiden Lichtschranken **DU** und **DO** und der Schalthebel **S1** im Pförtnerhäuschen. Wir müssen diesen Eingängen noch eine Wertigkeit zuordnen bzw. deren Kodierung festlegen: Sind die Lichtschranken unterbrochen, dann soll eine logische 0 am Schaltungseingang angelegt werden. Möchte der Pförtner auf Grün schalten, dann soll dies mit einem Eingabewert **S1 = 1** geschehen. Es liegen also **3 Eingänge** vor.
- (3) Welche **Ausgänge** sind erforderlich? Es sind die beiden Lampen, Rot (**LR**) und Grün (**LG**) in der Ampel. Soll eine Lampe leuchten, dann soll die Schaltung das zugehörige Ausgangssignal auf 1 setzen. Es liegen also **2 Ausgänge** vor.
- (4) Für eine **Zuordnung der Ausgabesignale auf die Eingabesignale** stellen wir fest:

Die Ampel darf nur auf Grün (**LG=1**) stehen, wenn (a) für eine LKW-Zufahrt beide Lichtschranken blockiert sind, **DU = 0**, **DO = 0**, und gleichzeitig aber der Pförtnerschalter auf Grün gelegt ist, **S1 = 1**, oder wenn (b) für eine PKW-Zufahrt nur die untere Lichtschranke blockiert ist, also **DU = 0**, unabhängig davon wie der Pförtnerschalter steht. In allen anderen Fällen steht die Ampel auf Rot (**LR=1**).

- (5) Dies führt zu der **Wertetabelle**:

(a)	DO	DU	S1	LR	LG	(b)	DO	DU	S1	LR	LG
	0	0	0	1	0		0	0	0	1	0
	0	0	1	0	1		0	0	1	0	1
	0	1	0	1	0	→	0	1	X	1	0
	0	1	1	1	0	→	1	0	X	0	1
	1	0	0	0	1	→	1	1	X	1	0
	1	0	1	0	1						
	1	1	0	1	0						
	1	1	1	1	0						

Abb. 4.1.1.2 (a) Vollständige Wertetabelle und (b) mittels Don't Cares verkürzte Wertetabelle

In Abb. 4.1.1.2 ist die vollständige Wertetabelle, die alle kombinatorisch möglichen Eingabevariablen-Kombinationen auflistet, angegeben. Diese kann reduziert werden, wenn Don't Cares eingeführt werden und zwar hier für die

Eingaben. Don't Cares für die *Ausgaben* wurden dagegen in Kap. 2.4.2.6 genutzt. Die Don't Cares für **S1** resultieren daraus, dass die Ampel auf Grün bzw. auf Rot geschaltet ist, *unabhängig* davon, wie der Pfortnerschalter **S1** steht. Zuerst ist zu bemerken, dass $LG = \overline{LR}$.

(6) Aus der Wertetabelle (a) ergibt sich die **boolesche Bedingung** für **LG** zuerst in ODER-Normalform, dann mittels Boolescher Algebra vereinfacht:

$$\begin{aligned}
 LG = \overline{LR} &= \overline{DO} \wedge \overline{DU} \wedge S1 \vee DO \wedge \overline{DU} \wedge \overline{S1} \vee DO \wedge \overline{DU} \wedge S1 \\
 &\stackrel{(1)}{=} \overline{DU} \wedge (\overline{DO} \wedge S1 \vee DO \wedge \overline{S1} \vee DO \wedge S1) \\
 &\stackrel{(1)}{=} \overline{DU} \wedge (\overline{DO} \wedge S1 \vee DO \wedge [\overline{S1} \vee S1]) \\
 &\stackrel{(2)}{=} \overline{DU} \wedge (\overline{DO} \wedge S1 \vee DO \wedge 1) \\
 &\stackrel{(3)}{=} \overline{DU} \wedge (\overline{DO} \wedge S1 \vee DO) \\
 &\stackrel{(4)}{=} \overline{DU} \wedge (S1 \vee DO)
 \end{aligned}$$

Das Resultat lässt sich direkter aus der Wertetabelle (b) ableiten. Hier tritt für **LG** nur zweimal die 1 auf, die Variable mit X entfällt jeweils, entsprechend:

$$\begin{aligned}
 LG = \overline{LR} &= \overline{DO} \wedge \overline{DU} \wedge S1 \vee DO \wedge \overline{DU} \\
 &\stackrel{(1)}{=} \overline{DU} \wedge (\overline{DO} \wedge S1 \vee DO) \\
 &\stackrel{(4)}{=} \overline{DU} \wedge (S1 \vee DO)
 \end{aligned}$$

Hierbei wurden folgende Vereinfachungen genutzt: ⁽¹⁾R6.1.1, ⁽²⁾R3.2.4, ⁽³⁾R3.1.2, ⁽⁴⁾R6.2.2.

Dasselbe Resultat ergibt sich aus einer KV-Vereinfachung. Es treten die drei Vollkonjunktionen ⁽¹⁾ $DO \wedge \overline{DU} \wedge \overline{S1}$, ⁽²⁾ $DO \wedge \overline{DU} \wedge S1$ und ⁽³⁾ $\overline{DO} \wedge \overline{DU} \wedge S1$ auf, die im nachfolgenden KV-Diagramm eingetragen sind:

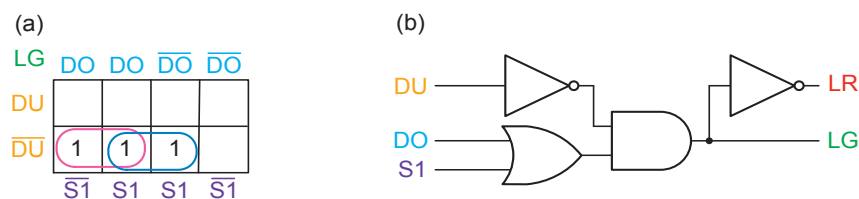


Abb. 4.1.1.3 (a) KV-Diagramm mit den Päckchen \square , \square und (b) Schaltung

Aus der booleschen Vereinfachung ebenso wie aus KV folgt die boolesche Bedingung

$$LG = \overline{LR} = [\overline{DU} \wedge S1] \vee [\overline{DU} \wedge DO] = \overline{DU} \wedge (S1 \vee DO)$$

(7) Die resultierende **Schaltung** ist in Abb. 4.1.1.3 (b) angegeben.

Übung 4.1.1

4.1.1 (a) Leite aus der Wertetabelle (a) bzw. (b) nach Abb. 4.1.1.2 eine boolesche Bestimmungsgleichung und eine Schaltung unmittelbar für **LR** ab, ohne den Umweg über $LG = \overline{LR}$.

4.1.1 (b) ([†]) Erweitere die Schaltung durch einen zusätzlichen Schalthebel **S2** im Pförtnerhaus, siehe Abb. 4.1.1.1, um eine Hauptschalterfunktion: Mit **S2** = 1 soll die bisherige Funktion auch weiterhin so realisiert sein. Die Schalterpositionen **S1** = **S2** = 0 schalten dagegen die Ampel komplett ab. Die Schalterpositionen **S1** = 1, **S2** = 0 stellen die Ampel dauernd auf Rot.

4.1.2 Füllstandanzeige

Als weiteres Beispiel betrachten wir die Bestimmung des Flüssigkeitspegels in einem mit einem Tank verbundenen L-Rohr zur Füllstandmessung. Die Füllstandmessung geschehe durch Lichtschranken. Als ersten Schritt betrachten wir ein Modul von vier Lichtschranken. Hat der Flüssigkeitspegel eine Lichtschranke überschritten, dann ist diese blockiert. Das Detektorsignal wird 0.

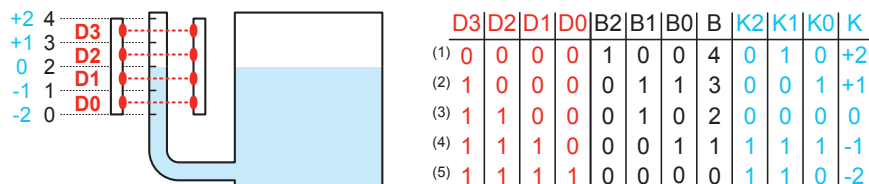


Abb. 4.1.2.1 Füllstandanzeige: (a) Funktionsschema und (b) Wertetabelle (Thermometer-Code, Dual, Zweierkomplement mit Vorzeichen, 2KmVZ)

- (1) Die **Funktionsbeschreibung** folgt aus dem Schema Abb. 4.1.2.1 (a).
- (2) Als **Eingänge** liegen die Detektor-Signale **D0**, **D1**, **D2** und **D3** vor. Hat der Flüssigkeitsstand die Höhe eines bestimmten Detektors überschritten, so geben dieser Detektor und alle darunter liegenden Detektoren den Wert 0 aus, alle darüber liegenden den Wert 1. Diese Kodierung wird anschaulich auch als *Thermometer-Code*, auf Englisch auch als *Unary Coding*, bezeichnet.
- (3) An den **Ausgängen** soll eine geeignete Höhenangabe ausgegeben werden. Wir können diese in **Dual** kodieren, also 0, 1, 2, 3, 4, wie in Abb. 4.1.2.1 (a) angegeben. Falls wir es mit einer Sollhöhe und deren Abweichung nach oben oder unten zu tun haben, so eignet sich besser eine Kodierung in **Zweierkom-**