Zusammenfassung meines Vortrages vom 26. Jänner 2017

Digitale Systeme und Schaltungen

Andreas Grimmer
Pro Scientia Linz
Johannes Kepler Universität Linz, Austria
andreas.grimmer@jku.at

In dieser Zusammenfassung wird die Funktionsweise von digitalen Systemen und Schaltungen anhand eines konkreten Beispiels, nämlich eines Addierers für zwei Binärzahlen, hergeleitet. Für diese Herleitung werden die dazu notwendigen Logikgatter eingeführt und erklärt, welche dann zur Realisierung einer Addiererschaltung verwendet werden.

1 Einführung

Digitale Systeme und Schaltungen werden in allen Bereichen unseres täglichen Lebens verwendet und beeinflussen somit unseren Alltag. Schon im frühen Kindesalter kommen wir in Berührung mit digitalen Systemen. Ganz selbstverständlich werden hier Computer und Tablets in der Ausbildung eingesetzt.

Was bedeutet jedoch "digital"? Oder wie stellt ein Computer Zahlen dar und wie kann er damit rechnen? Viele Benutzer dieser Systeme würden wahrscheinlich nur sehr vereinfachte Antworten auf diese Fragen geben können.

Dies ist ein Grund, warum ich mich bei meinem Semestervortrag für Pro Scientia entschieden habe, Grundkenntnisse in digitalen Systemen und Schaltungen zu erklären. Dazu wird im Folgenden die Funktionsweise eines binären Addierers schrittweise hergeleitet.

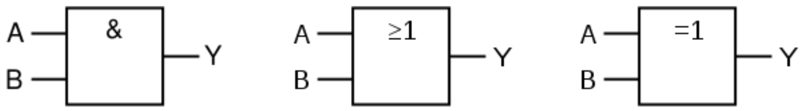
2 Logikgatter

In der digitalen Schaltungstechnik nehmen Signale nur zwei unterschiedliche Werte an, nämlich logisch 1 und logisch 0. Diese zwei Booleschen Zustände, können mit unterschiedlichen elektrischen Spannungen realisiert werden. Zum Beispiel kann eine 1 mit einer positiven Spannung und eine 0 mit Masse assoziiert werden.

Mittels Logikgattern (d.h. mit elektronischen Bauelementen) können Signale verknüpft werden. Dabei implementiert ein Gatter eine logische Verknüpfung der Eingänge und das Resultat ist wieder ein Boolesches Signal am Gatterausgang. Im Folgenden werden jene Gattertypen eingeführt, die für die Realisierung eines Addierers benötigt werden.

2.1 Und-Gatter

Ein Und-Gatter hat zwei (oder mehrere) Eingänge und einen Ausgang. Am Ausgang liegt genau dann eine 1 an, wenn alle Eingänge mit 1 belegt sind. Abb. 1(a) zeigt das Schaltsymbol eines Und-Gatters



(a) Und-Gatter (b) Oder-Gatter (c) Oder-Gatter

Abbildung 1: Logikgatter

mit den Eingängen A und B. Sowohl A als auch B sind logische Werte (entweder 1 oder 0) und werden durch das Und-Gatter verknüpft und das Resultat wird am Ausgang Y ausgegeben. In der folgenden Wahrheitstabelle werden alle vier möglichen Kombinationen der Eingangsbelegung aufgelistet und der dazugehörige Ausgangswert angegeben.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2.2 Oder-Gatter

Ein Oder-Gatter hat zwei (oder mehrere) Eingänge und einen Ausgang. Am Ausgang liegt genau dann eine 1 an, wenn mindestens ein Eingang mit 1 belegt ist. Abb. 1(b) zeigt das Schaltsymbol eines Oder-Gatters mit den Eingängen A und B. Die Eingänge und der Ausgang stellen logische Werte dar und werden folgendermaßen verknüpft:

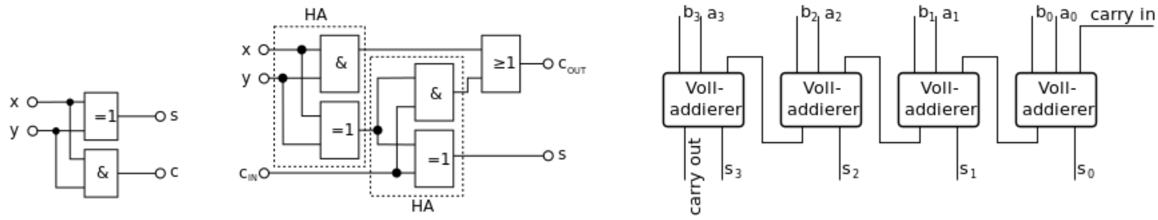
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

2.3 Exklusives-Oder-Gatter

Ein Exklusives-Oder-Gatter (kurz Xor-Gatter) ist ein Gatter mit zwei Eingängen und einem Ausgang. Der Ausgang ist dann 1, wenn genau einer der beiden Eingänge 1 ist (die Funktion lässt sich auch für mehrere Eingänge definieren). Abb. 1(c) zeigt das Schaltsymbol eines Xor-Gatters mit den Eingängen A und B. Wie die Wahrheitstabelle zeigt, ist der Ausgang Y genau dann gleich 1, wenn einer der beiden Eingänge A oder B mit 1 belegt ist.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Diese vorgestellten Logikgatter werden nun verwendet, um einen Addierer für Binärzahlen zu realisieren.



(a) Halbaddierer (b) Volladdierer (c) Carry-Ripple-Addierer für zwei 3-Bit Binärzahlen

Abbildung 2: Addierer

3 Addierer

Das Binärsystem ist ein Zahlensystem, das zur Darstellung von Zahlen nur zwei verschiedene Ziffern benutzt[1]. Oft werden für diese Ziffern, genauso wie in der digitalen Schaltungstechnik, die Symbole 1 und 0 verwendet. Durch diese Zuordnung von Ziffern zu Signalen können somit Zahlen in digitalen Systemen dargestellt werden.

Folgend wird die Addition von zwei einstelligen Binärzahlen erklärt. Dabei wird zuerst der Halbaddierer eingeführt, welcher zwei einstellige Binärzahlen ohne Übertrag addiert. Basierend auf dem Halbaddierer wird dann ein Volladdierer eingeführt, der zusätzlich einen Übertrag berücksichtigt. Diese Volladdierer werden schließlich zur Implementierung eines Carry-Rippl-Addierers verwendet.

3.1 Halbaddierer

Der Halbaddierer hat zwei Eingänge x und y, welche die zwei zu addierenden Binärzahlen darstellen. Addiert man zwei Binärzahlen ergibt dies eine einstellige Summe und einen möglichen Übertrag, falls das Ergebnis nicht in einer Stelle ausgedrückt werden kann. Zum Beispiel wenn die Eingänge mit $x = 1$ und $y = 1$ belegt sind, ist das Ergebnis gleich 2 in Dezimaldarstellung. Um die Dezimalzahl 2 als Binärzahl dazustellen, werden zwei Bits benötigt. Die Addition liefert also einen Übertrag an die nächste Stelle. Der Halbaddierer hat deswegen genau zwei Ausgänge: einen Ausgang s für die Summe und einen Ausgang c (engl. Carry) für den Übertrag. Die Funktionsweise eines Halbaddierers lässt sich mit folgender Wahrheitstabelle beschreiben:

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Der Ausgang c kann durch ein Und-Gatter und der Ausgang s durch ein Xor-Gatter realisiert werden. Dadurch ergibt sich die Schaltung in Abb. 2(a).

Diese Schaltung kann jedoch keinen möglichen Übertrag aus einer niederwertigeren Stelle berücksichtigen. Daher benötigen wir einen Addierer, der drei einstellige Binärzahlen addieren kann – einen sogenannten Volladdierer.

[1]Folgend wird die Kenntnis des Binärsystems vorausgesetzt.

3.2 Volladdierer

Ein Volladdierer addiert drei einstellige Binärzahlen und hat dafür drei Eingänge x, y und c_{in} und zwei Ausgänge s und c_{out}. Wie beim Halbaddierer liefert der Ausgang s die Summe (d.h. die niederwertigere Stelle des Ergebnisses) und der Ausgang c_{out} den Übertrag (d.h. die höherwertige Stelle des Ergebnisses). Die Funktionsbeschreibung eines Volladdierers ist durch folgende Wahrheitstabelle gegeben:

x	y	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Diese Funktion des Volladdierers kann mittels zwei Halbaddierern implementiert werden (siehe Abb. 2(b)). Zusätzlich müssen die beiden Carry-Ausgänge der beiden Halbaddierer mit einem Oder-Gatter verknüpft werden.

Dieser Volladdierer kann nun zwei einstellige Binärzahlen inklusive einen möglichen Übertrag aus einer niederwertigeren Stelle summieren. Als Nächstes werden mehrere Volladdierer in einer Schaltung kombiniert, um mehrstellige Binärzahlen addieren zu können.

3.3 Carry-Ripple-Addierer

Ein Carry-Ripple-Addierer ist eine Schaltung zur Addition von mehrstelligen Binärzahlen. Die Funktionsweise eines Carry-Riple-Addierers gleicht jener der Schulmethodik: Es wird jede Stelle der zwei Zahlen einzeln summiert und ein möglicher Übertrag wird an die nächste Stelle weitergegeben. Daher verwendet diese Schaltung pro Stelle einen Volladdierer und der Übertragsausgang wird jeweils an den Übertragseingang des Volladdierers der nächsten Stelle angeschlossen. Somit hat diese Schaltung $2 \cdot n + 1$ Eingänge und $n + 1$ Ausgänge und ist in Abb. 2(c) abgebildet.

4 Konklusion

In dieser Zusammenfassung wurde die Funktionsweise von Digitalen Schaltungen anhand einer Addierer-Schaltung für zwei mehrstellige Binärzahlen erklärt. Dazu wurden die benötigten Logik-Gatter eingeführt und schrittweise zu einem Carry-Ripple-Addierer kombiniert. Die resultierende Schaltung ist sehr einfach, jedoch verursacht die Weitergabe des Übertragsbits von der niederwertigsten zur höchstwertigen Stelle eine lange Laufzeit. Daher wurden effizientere Addierernetzwerke, wie etwa der Carry-Skip, der Carry-Look-Ahead oder der Carry-Select-Addierer entwickelt.

Literatur

[1] S. Higgins, Z. Xiao, and M. Katsipataki, "The impact of digital technology on learning: A summary for the education endowment foundation," *Durham, UK: Education Endowment Foundation and Durham University*, 2012.

[2] D. Harris and S. Harris, *Digital design and computer architecture*. Elsevier, 2012.

[3] U. Tietze and C. Schenk, *Halbleiter-Schaltungstechnik*. Springer, 2002.