Grundlagen der Technischen Informatik

Thomas Studer

Herbstsemester 2022

Über dieses Dokument

Das ist eine Zusammenfassung der Vorlesung “Grundlagen der Technischen Informatik” von Prof. Dr. Thomas Studer im Herbstsemester 2022. Du darfst sie so verwenden, wie sie dir am meisten beim Verständnis des Materials hilft. Verantwortlich dafür was hier drin steht ist Manuel Flückiger. Beachte, dass der Dozent dieses Dokument nicht geschrieben (und vielleicht auch nicht gelesen) hat. Falls du selbst Fehler entdeckst, oder Verbesserungsvorschläge hast, sind diese unter der Adresse

`manuel.flueckiger@students.unibe.ch`

Bei dieser Mitschrift ist zu beachten, dass der Aufbau und die Einführung ein Plagiat von Levi Ryffel’s Analysis 1 Notizen ist. Es dient mir vorallem beim lernen von LaTeXund beim Verständnis der Vorlesung. Das original Repository von Levi Ryffel findet ihr unter: https://github.com/raw-bacon/ana1-notes

Anerkenntnis

Diese Zusammenfassung wurde von den slides der Vorlesung abgeschrieben und interpretiert und kann viele Fehler enthalten. Unzählige davon sollten entdeckt und behoben werden von euch, den Lesern. Vielen Dank euch! Weitere hilfreiche Beiträge kamen (”von eure Namen, wenn ihr helft”).

Inhaltsverzeichnis

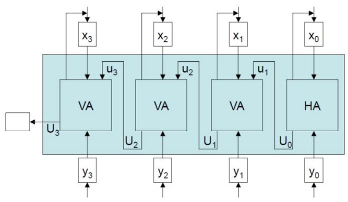
Kapitel I

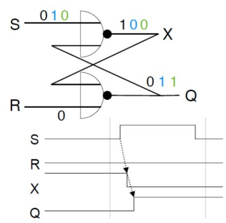
Einführung

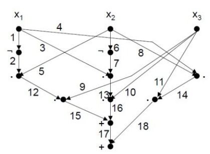
1 Ziele der Vorlesung

Ziele.

- Addieren, Speichern, Berechnen

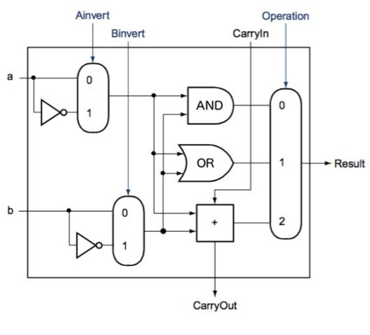


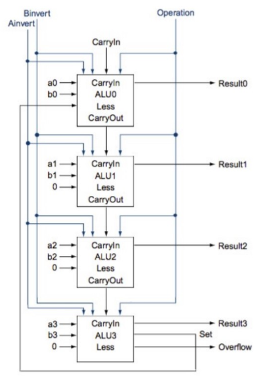




$f(x_1,x_2,x_3) = m_3 + m_5 + m_7 = \neg x_1x_2x_3 + x_1\neg x_2x_3 + x_1x_2x_3$

- Arithmetic Logic Unit





- Literatur

> W. Oberschelp, G. Vossen
  Rechneraufbau und Rechnerstrukturen
  ISBN 3-486-57849-9
  R. Oldenbourg Verlag, München, Wien
  einige Kapitel sind in Ilias verfügbar!

> B. Becker, P. Molitor
  Technische Informatik - Eine einführende Darstellung
  ISBN 3-486-58650-5
  Oldenbourg

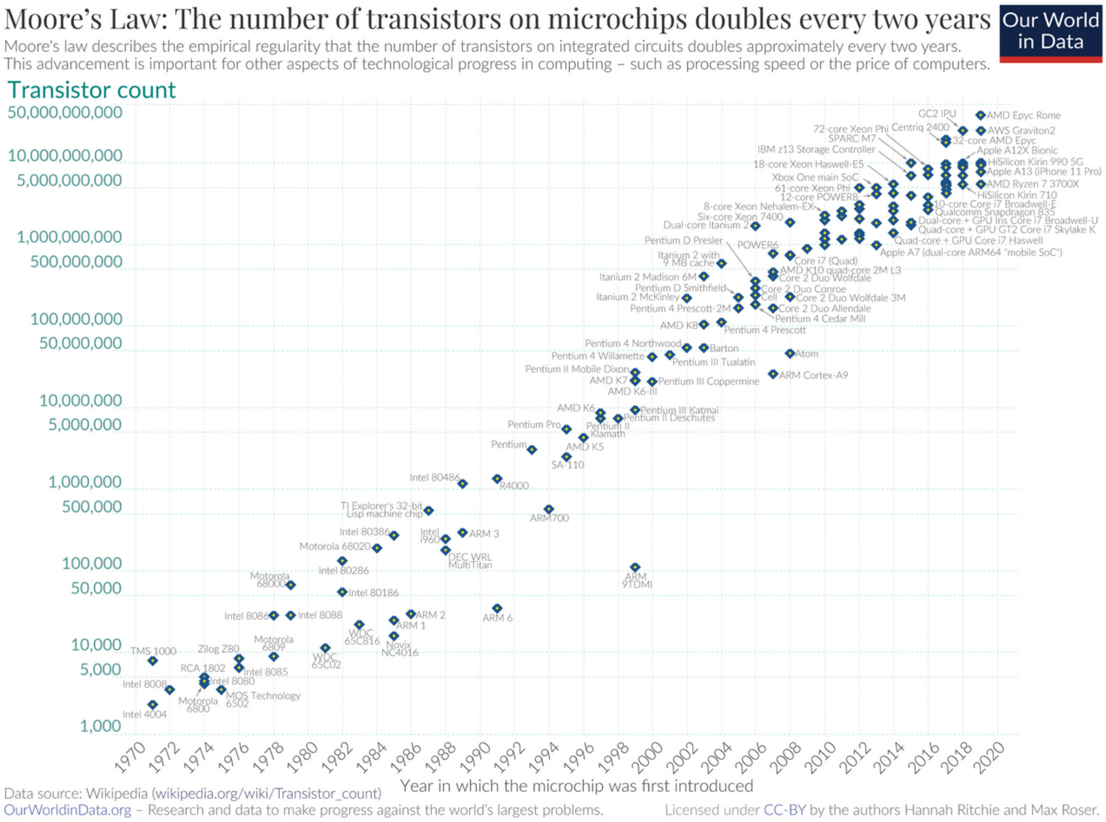
- Diskrete Mathematik
  - ≫ Es wird ein paar inhaltliche Überschneidungen zwischen GTI und der Vorlesung Diskrete Mathematik geben
  - ≫ Dies betrifft vor allem die nächsten beiden Kapitel
  - ≫ Verwenden Sie in GTI die Definitionen aus der GTI Vorlesung und in DM diejenigen aus der DM Vorlesung

2 Geschichte

Wen's interessiert soll die slides durchschauen oder eine Zusammenfassung davon schreiben, da sie aber wahrscheinlich nicht Prüfungsrelevant ist, behandle ich die hier kaum.

2.1 Moore's Law

Mooresche Gesetz. Das Mooresche Gesetz (englisch Moore's law; deutsch „Gesetz" im Sinne von „Gesetzmäßigkeit") besagt, dass sich die Komplexität integrierter Schaltkreise mit minimalen Komponentenkosten regelmäßig verdoppelt; je nach Quelle werden 12, 18 oder 24 Monate als Zeitraum genannt. Unter Komplexität verstand Gordon Moore, der das Gesetz 1965 formulierte, die Anzahl der Schaltkreiskomponenten auf einem integrierten Schaltkreis. Gelegentlich ist auch von einer Verdoppelung der Integrationsdichte die Rede, also der Anzahl an Transistoren pro Flächeneinheit. Diese technische Entwicklung bildet eine wesentliche Grundlage der „digitalen Revolution".



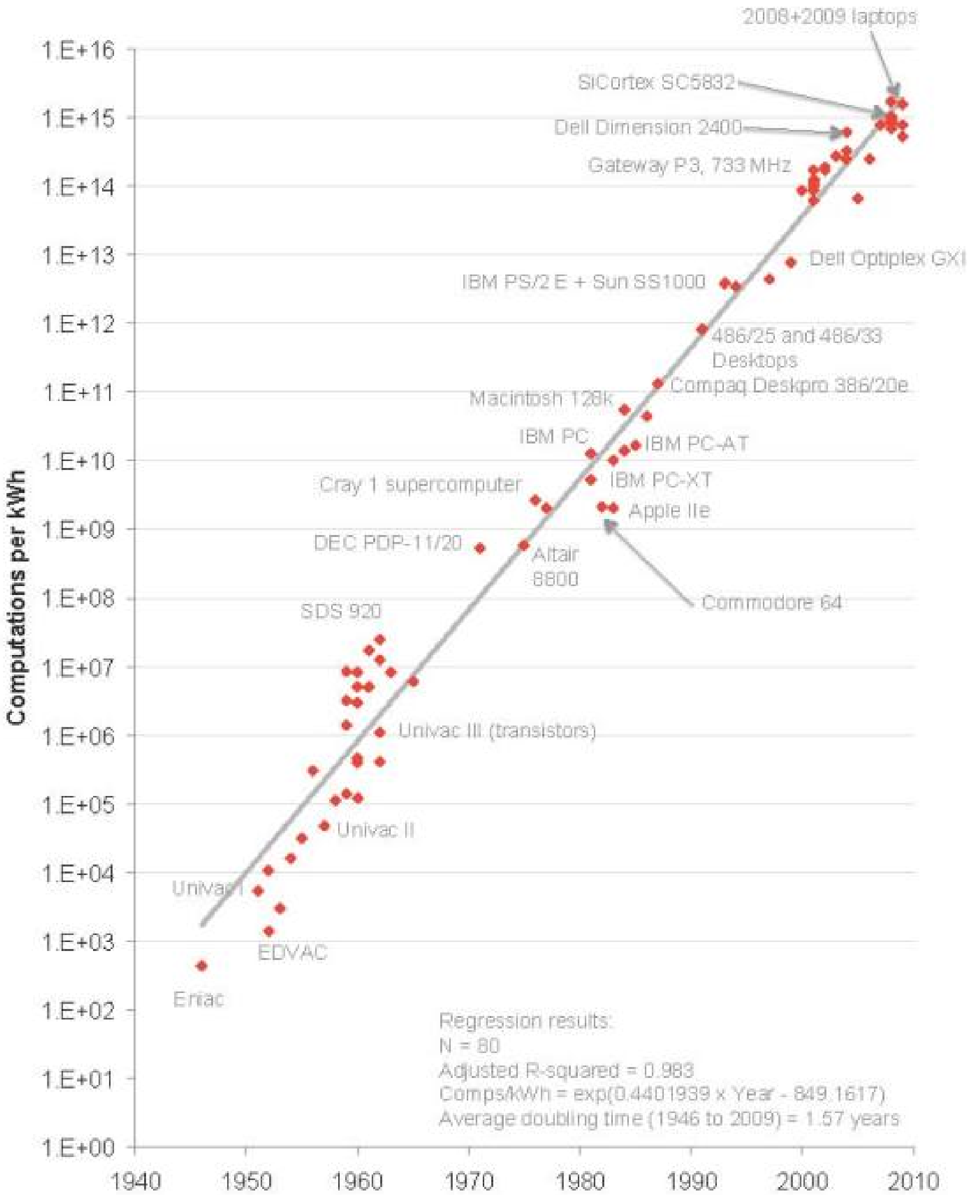
Quelle: https://de.wikipedia.org/wiki/Mooresches_Gesetz

2.2 Koomey's Law

Koomeys Gesetz. Das Koomey-Gesetz beschreibt einen Trend in der Geschichte der Computerhardware: Etwa ein halbes Jahrhundert lang verdoppelte sich die Zahl der Berechnungen pro Joule verbrauchter Energie etwa alle 1,57 Jahre. Professor Jonathan Koomey beschrieb diesen Trend in einem Papier aus dem Jahr 2010, in dem er schrieb, dass "bei einer festen Rechenlast die benötigte Batteriemenge alle anderthalb Jahre um den Faktor zwei abnimmt".

Dieser Trend war seit den 1950er Jahren bemerkenswert stabil gewesen (R^2 von über 98 %) Im Jahr 2011 untersuchte Koomey diese Daten jedoch erneut und stellte fest, dass sich die Verdopplung nach dem Jahr 2000 auf etwa alle 2,6 Jahre verlangsamte. Dies hängt mit der Verlangsamung des Mooreschen Gesetzes[2.1] zusammen, also der Möglichkeit, kleinere Transistoren zu bauen, und mit dem Ende der Dennard-Skalierung um 2005, also der Möglichkeit, kleinere Transistoren mit konstanter Leistungsdichte zu bauen.

"Der Unterschied zwischen diesen beiden Wachstumsraten ist beträchtlich. Eine Verdopplung alle anderthalb Jahre führt zu einer 100-fachen Steigerung der Effizienz pro Jahrzehnt. Eine Verdopplung alle zweieinhalb Jahre ergibt nur eine 16-fache Steigerung", schrieb Koomey.

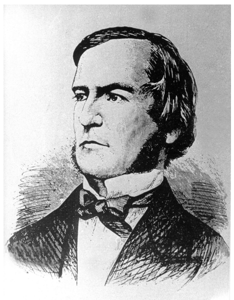


Quelle: https://en.wikipedia.org/wiki/Koomey%27s_law

3 Boolsche Logik

3.1 George Boole (1815-1864)

The Laws of Thought (1854):
Logisches Denken wird auf das Lösen von Gleichungen reduziert.



3.2 boolsche Logik

- x, y seien Kollektionen von Objekten
- $x+y$ ist die Kollektion von allen Objekten, die zu x und zu y gehören (Vereinigung)
- xy ist die Kollektion von allen Objekten, die zu x und zu y gehören (Schnitt)
- 0 ist die leere Kollektion
- Alle s sind p $s = sp$
- Kein s ist p $sp = 0$
- Einige s sind p $sp \neq 0$
- Einige s sind nicht p $s \neq sp$
- Rechenregeln:

  $$(xy)z = x(yz)$$
  $$x0 = 0$$

  Beispiel. b: Berner
  s: Schweizer
  m: auf dem Mond gewesen

  Alle b sind $s = b = bs$
  Kein s ist $m = sm = 0$

$bm = (bs)m = b(sm) = b0 = 0$
Also $bm = 0$, also kein b ist m

Kapitel II

Boolesche Funktionen

1 Zahlendarstellung

- $\sum$: ein fest gewähltes Input Alphabet
- $\sum^n$: ein Wort der Länge n über $\sum$
- $\sum_b := \{0, \dots, b-1\}$: Alphabet des b-adischen Zahlensystems ($b < 1$)

Beispiele. - $\sum_2 = \{0, 1\}$: Dual oder Binäralphabet
- $\sum_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$: Oktalalphabet
- $\sum_1 0 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$: Dezimalalphabet

1.1 b-adische Darstellung natürlicher Zahlen

Definition. Sei $b \in \mathbb{N}$ mit $b > 1$. Dann ist jede natürliche Zahl z mit $0 \leq z \geq b^n - 1$ (und $n \in \mathbb{N}$) eindeutig als Wort der Länge n über $\sum_b$ darstellbar durch:

$$z = \sum_{i=0}^{n-1} z_i b^i = z_0 b^0 + z_1 b^1 + \ldots + z_{n-1} b^{n-1}$$

mit $z_i \in \sum_b = \{0, \dots, b-1\}$ für $i = 0, \dots, n-1 \dots$

Als vereinfachende Schreibweise ist dabei folgende $Ziffernschreibweise$ üblich;

$$z = z_{n-1} zn - 2 \ldots z_1 z_{0b}$$

Wichtiger Spezialfall: $b = 2$ ("Dualdarstellungnatürlicher Zahlen)

Beispiele.

$$26_1 0 = 2 * 10^1 + 6 * 10^0 = 2 * 10 + 6 * 1 = 20 + 6$$

$$11010_2 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^2 = 1 * 16 + 1 * 8 + 0 * 4 + 1 * 2 = 16 + 8 + 2$$

$$1A_1 6 = 1 * 16^1 + 10 * 16^0 = 1 * 16 + 10 * 1 = 16 + 10$$

2 Addition

-Dezimaladdition -Binäraddition

```
  183           10110111
 +197          +11000101
  11           1     111
  ---          ---------
  380          101111100
```

3 Schaltfunktion

Definition. Seien $n, m \in \mathbb{N}, n, m \leq 1$. Dann heisst eine Funktion $F : B^n \to B^m$ Schaltfunktion.

Beispiele.

- Addition von zwei 16-stelligen Dualzahlen

$$A : B^{32} \to B^{17}$$

- Multiplikation von zwei 16-stelligen Dualzahlen

$$M : B^{32} \to B^{32}$$

- Sortieren von 30 16-stelligen Dualzahlen

$$S : B^{480} \to B^{480}$$

- Primzahltest

$$P : B^{32} \to B,\ mit\ P(x) = 1,\ falls\ x\ Primzahl\ ist,\ 0\ sonnst.$$

4 Boolesche Funktionen

Definition. Eine Schaltfunktion $f : B^n \to B$ heisst (n-stellige) Boolesche Funktion.
$Zusammenhang\ zu\ Schaltfunktionen:$
Sei $F : B^n \to B^m$ mit $F(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$. Setzt man für jedes $i \in \{1, \ldots, m\}$

$$f_i : B^n \to B,$$

def. durch

$$f_i(x_1, \ldots, x_n) = y_i,$$

so ist F wie folgt darstellbar:

$$F(x_1, \ldots, x_n) = (f_1(x_1 \ldots x_n), f_2(x_1 \ldots x_n), \ldots, f_m(x_1, \ldots, x_n))$$

für alle $x_1, \ldots, x_n \in B$.

4.1 1-stellige Boolesche Funktionen

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

Es gilt: $f_0(x) = 0, f_1(x) = x, f_2(x) = \bar{x}, f_3(x) = 1$
Die Funktion $f_2(x)$ wird auch als NOT(x) und ¬x bezeichnet.

4.2 2-stellige Boolesche Funktionen

	Funktion	x y	1 1	1 0	0 1	0 0		
1	Konstant 1	1	1	1	1	1	Allaussage	(TRUE)
2	x oder y	$x \vee y$	1	1	1	0	Disjunktion	(OR)
3	wenn y,dann x	$y \rightarrow x$	1	1	0	1	Subjunktion	
4	x	x	1	1	0	0		
5	wenn x, dann y	$x \rightarrow y$	1	0	1	1	Subjunktion	
6	y	y	1	0	1	0		
7	Äquivalenz	$x \leftrightarrow y$	1	0	0	1	Bijunktion	(XNOR)
8	x und y	$x \wedge y$	1	0	0	0	Konjunktion	(AND)
9	negiertes Und	$x \uparrow y$	0	1	1	1	Negierte Konjunktion	(NAND)
10	exklusives Oder	$x \oplus y$	0	1	1	0	Kontravalenz	(XOR)
11	nicht y	$\neg y$	0	1	0	1	Negation	(NOT)
12	x und nicht y	$x \wedge \neg y$	0	1	0	0		
13	nicht x	$\neg x$	0	0	1	1	Negation	(NOT)
14	nicht x und y	$\neg x \wedge y$	0	0	1	0		
15	negiertes Oder	$x \downarrow y$	0	0	0	1	Negierte Disjunktion	(NOR)
16	konstant 0	0	0	0	0	0	Nullaussage	(FALSE)

4.3 Gesetze einer Booleschen Algebra

- Kommuntativgesetze: $x \vee y = y \vee x, x \wedge y = y \wedge x$
- Assoziativgesetze: $(x \vee y) \vee z = x \vee (y \vee z), (x \wedge y) \wedge z = x \wedge (y \wedge z)$
- Verschmelzungsgesetze: $(x \vee y) \wedge x = x, (x \wedge y) \vee x = x$
- Distributivgesetze: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
- Komplementgesetze: $x \vee (y \wedge y) = x, x \wedge (y \vee y) = x$
- Neutrale Elemente 0, 1: $x \vee 0 = x, x \wedge 0 = 0, x \wedge 1 = x, x \vee 1 = 1$
- de Morgansche Regeln: $\overline{x \vee y} = \bar{x} \wedge \bar{y}, \overline{x \wedge y} = \bar{x} \vee \bar{y}$

- Idempotenz: $x = x \vee x = x \wedge x = \bar{\bar{x}}$

Beispiel.

$$
\begin{aligned}
((a * c) + (a * d)) a\ \mathit{ausklammern}\ (\mathit{Distributivgesetz}) \\
= (a * (C + d))\ \mathit{de\ Morgan} \\
= a * (c + d)\ \mathit{Idempotenz} \\
= a + (C + d)\ \mathit{de\ Morgan} \\
= a + (c * d)
\end{aligned}
$$

4.4 Bedeutung von Booleschen Algebren

Boolesche Algebren kommen in verschiedenen mathematischen Teilgebieten vor. Ihre Bedeutung ist nicht nur auf die technische Informatik beschränkt.

Beispiele.

- Boolesche Ringe (Algebra)
- Verbände (Ordnungstheorie)
- Kompakte, total unzusammenhängende Hausdorfräume (Topologie)
- Semantik für Logiken
- Boolean-valued models (Modelltheorie, gängiges Mittel um die Unabhängigkeit der Kontinuumshypothese von ZFC zu zeigen)

4.5 Boolesche Körper

Die Menge B = {0,1} mit den Operationen XOR und AND bildet einen Körper mit dem Nullelement 0 und dem Einselement 1. In dem Zusammenhang wird XOR als Addition (+) und AND als Multiplikation (*) betrachtet.

Ziele.

1. Sie können Zahlen zwischen verschiedenen Zahldarstellungen umrechnen
2. Sie kennen die Begriffe Schaltfunktion und Boolesche Funktion
3. Sie kennen alle ein- und zweistellige Boolesche Funktionen
4. Sie können für eine zusammengesetzte Boolesche Funktion die Wahrheitstabelle berechnen
5. Sie kennen die Gesetze der Boolesche Algebra und können diese verwenden, um Boolesche Ausdrücke umzuformen
6. Sie können überprüfen, ob zwei Boolesche Ausdrücke dieselbe Funktion beschreiben (sowohl durch Ausrechnen als auch durch Umformen).