

# Im Herzen eines Computers

## Student Group

First Name	Surname	Matrikel Nr.

## Table of Contents

- Im Herzen eines Computers** ..... 2
  - Einführung ..... 2
  - zwei Transistor(typ)en reichen ..... 2
  - Gatter logisch? ..... 3
  - nicht-UND, oder? ..... 3
  - eine einfache Rechnung ..... 4
  - Nano-Lego ..... 4
- weiterführende Links** ..... 4
- Bildreferenzen** ..... 5

# Im Herzen eines Computers

## Einführung

Fig. 1: Ein Microcontroller oben ohne 

Der Kern eines Computer ist der Prozessor in dem die Befehle ausgeführt werden. Diese Zentrale Prozessoreinheit (CPU) wird auch in Microcontrollern genutzt, die um uns herum in fast jedem Gerät zu finden sind: Mobiltelefone, PKWs, Bankkarten, Waschmaschinen... Häufig sind in den Geräten sogar mehrere Microcontroller verbaut.

Im Microcontroller ist neben dem befehlsausführenden Microprozessor (genauer der [arithmetisch-logischen Einheit](#)) auch weitere Peripherie wie Speicher, Taktgeneration, Analog-Digital-Wandler und vieles mehr verbaut. Dies macht ihn zum kompakten Werkzeug für viele Anwendungen. Betrachtet man den Microcontroller unter dem optischen Mikroskop, so ergibt sich nebenstehendes Bild ([figure 1](#)). Darin sind die verschiedenen Peripheriekomponenten zu sehen.

Fig. 2: Microcontroller unter dem Mikroskop  Fig. 3: Microcontroller schematisch 

Wir wollen uns aber nun den Aufbau des Prozessors ansehen. Der in [figure 2](#) und [figure 3](#) dargestellte Chip wurde 1990 von [zwei Studenten](#) entwickelt und besteht aus mehreren Zehntausend Transistoren. Dieser Chip hat den Weg zu günstigen, schnellen und dennoch leicht programmierbaren Controllern, vom Faxgerät bis in den Hobbykeller, geebnet und ist u.a. auf den Arduinoboards zu finden. Den ATmega328 - einen entfernten Nachfolger mit mehreren hunderttausend Transistoren - werden Sie in höheren Semestern kennen und programmieren lernen.

Im Folgenden soll nun betrachtet werden, wie eine einfache Rechnung wie  $\$1+1=2\$$  im Rechner abläuft.

## zwei Transistor(typ)en reichen

Fig. 4: n- und p-Kanal MOSFETs

Ok, nun wissen wir zumindest, dass er Chip besteht aus vielen Transistoren besteht. Aber wie funktionieren diese und wie kann man daraus so etwas komplexes wie einen Prozessor aufbauen? Die genaue Funktion ist Inhalt eines Kurses im 2. und 3. Semester. Für die digitale Anwendung ist es ausreichend ein einfaches Bild eines bestimmten Transistortyps - dem MOSFET - im Kopf zu haben. Dieser hat die drei Anschlüsse **S**ource ("Quelle"), **D**rain ("Senke") und **G**ate ("Tor"). Gate agiert dabei als Türsteher, der den Durchgang zwischen Source und Drain regelt: Liegt am Anschluss Gate die richtige Spannung<sup>1)</sup>, so werden die Anschlüsse Source und Drain kurzgeschlossen, dass heißt ein Strom kann fließen und der Spannungsabfall dazwischen wird klein. Vom MOSFET sind bei den folgenden digitalen Schaltungen zwei Typen wichtig:

- einer, der bei niedrigen Spannungen ( $\$0V$ , logisch 0, Low, L oder Falsch) am Gate nichtleitend ist (n-Kanal MOSFET) und
- einer, der bei hohen Spannungen ( $\$5V$ , logisch 1, High, H oder Wahr) am Gate nichtleitend ist (p-Kanal MOSFET).

Im Bild rechts (figure 4) sehen Sie die beiden Varianten in Aktion, wenn die Spannung am Gate gerade die digitalen Spannungswerte annimmt (vgl. Video 1).

Interessant ist nun, dass diese zwei Arten von Transistoren ausreichen, um alle Varianten an logischen Funktionen aufzubauen. Logische Funktionen kombinieren einen oder mehrere Ausgänge in der Art, dass jede Art von Eingabe eindeutig zu einer Ausgabe führt. In Schaltungen entsprechen sogenannte Gatter den logischen Funktionen. Dabei muss es sich nicht nur um so einfache Funktionen wie ein AND-Gatter handeln (logisches UND, Konjunktion, "nur wenn beide Eingänge wahr sind, wird der Ausgang wahr."). Es können so auch mathematische Operationen abgebildet werden. Dazu müssten Gatter geschickt miteinander kombiniert werden. In einer Übung zu Binärer Logik wird gezeigt, dass sich alle Gatter durch NAND- oder NOR-Gatter aufbauen lassen. Wenn wir also herausfinden könnten, wie man ein NAND oder NOR Gatter aus Transistoren aufbauen könnten, so könnte man daraus wiederum alle Gatter bis hin zur Addition aufbauen.

## Gatter logisch?

Fig. 6: Inverter-Gatter in CMOS auf Chip

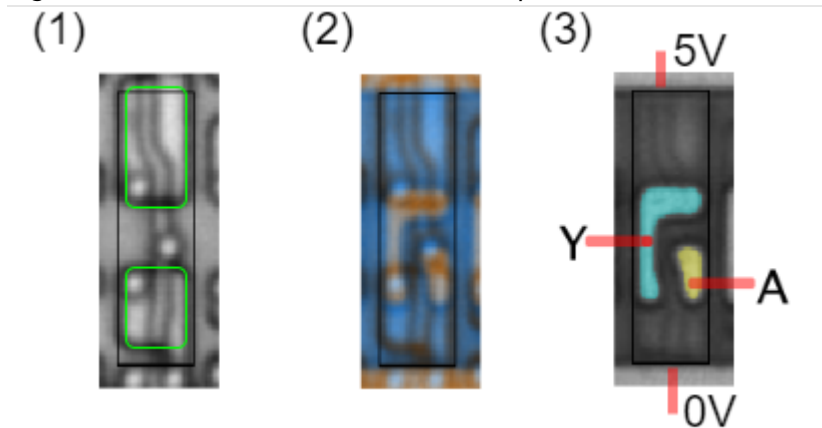


Fig. 5: Falstad Simulation eines Inverters

Zunächst wollen wir eine Schaltung aus MOSFETs aufbauen, welche einen digitalen Eingangswert  $X=0$  in den den Wert  $Y=1$  erzeugt und für  $X=1 \rightarrow Y=0$ . Dieser Schaltung wird Inverter genannt. Zu diesem Zweck sollte die Schaltung mit einer Spannungsquelle (Spannung für logisch 1) und Masse (Spannung für logisch 0) ausgestattet sein.

## nicht-UND, oder?

Fig. 7: Falstad Simulation eines CMOS NAND-Gatters

Fig. 10: Falstad Simulation eines CMOS NAND-Gatters (Struktur ähnlich Si-Die)

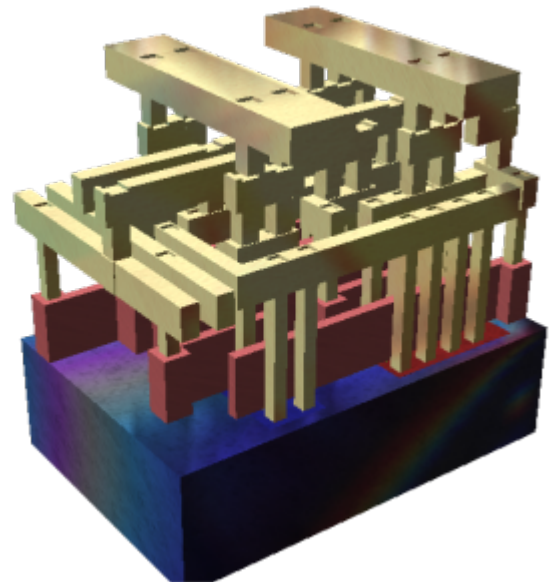
## eine einfache Rechnung

Fig. 11: Simulation eines Addierers

Werden viele Eingänge oder Ausgänge zusammengefasst, können größere Zahlenwerte umgesetzt werden.

## Nano-Lego

Fig. 9



---

## weiterführende Links

- Eine schöne Übersicht der Kernideen zum [Rechnen mit Strom](#) ist vom Gymnasium Kirchenfeld (CH) zusammengestellt worden
- diverse Mikroskopaufnahmen auf [SiliconZoo.org](http://SiliconZoo.org)
- [http://www.righto.com/2016/12/die-photos-and-analysis-of\\_24.html](http://www.righto.com/2016/12/die-photos-and-analysis-of_24.html)
- Erklärung zu [CMOS](#) in der englischen Wikipedia
- Wikiseite zu [Integrierte Schaltkreise](#)

## Bildreferenzen

figure 1: TravisGoodspeed@Flickr CC BY 2.0

figure 2: ZeptoBars@Wikimedia,CC BY 3.0

figure 6, figure ##: SiliconZoo.org, Lizenz unbekannt

figure 9: David Carron@Wikimedia,public domain

1)

tatsächlich kommt es auf die Spannung zwischen Gate und Source an, nicht nur auf die Spannung vom Gate. Aber dazu in [Elektronische Schaltungstechnik](#) mehr

From:  
<https://wiki.mexle.org/> - **MEXLE Wiki**

Permanent link:  
[https://wiki.mexle.org/grundlagen\\_der\\_digitaltechnik/im\\_herzen\\_eines\\_computers?rev=1603638560](https://wiki.mexle.org/grundlagen_der_digitaltechnik/im_herzen_eines_computers?rev=1603638560)

Last update: **2021/05/09 09:59**

