

Grundidee

Die Idee des Projektes ist es, die Funktionsweise eines Taschenrechners anhand einer Schaltung zu veranschaulichen. Dabei wird soweit möglich auf gängige mächtige, aber komplexe Bauteile verzichtet, um die Schaltung möglichst anschaulich zu machen.

Die Schaltung soll am Ende in einem Kasten gut sichtbar gemacht werden, indem mit LEDs Zustände in der Schaltung markiert werden, und mit Beschreibungstafeln die Theorie hinter der Schaltung erklärt wird.

Binäre Logikoperationen

Zum Verständnis der Funktionsweise des Taschenrechners werden in diesem Abschnitt die grundlegenden binären Logikoperationen erklärt. Das sind bitweise Operationen zwischen zwei Eingaben, die je nach Operator ein entsprechendes Ergebnis liefern.

AND

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

OR

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

NAND

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

NOR

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

XOR

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

Flip-Flop

Neben den Logikgattern benötigt es sogenannte Flip-Flops, die zur Speicherung der Zustände zuständig ist. Je nach Ausführung, haben sie unterschiedliche Funktionen. Grundsätzlich lassen sich Flip-Flops in zustandsgesteuerte und flankengesteuerte Flip-Flops klassifizieren. Allerdings gibt es Flip-Flops, die gar nicht taktgesteuert sind oder sowohl zustandsgesteuert und auch flankengesteuert arbeiten.

Bei den zustandsgesteuerten Flip-Flops werden die Eingaben nach einem bestimmten Logikpegel als *HIGH* erkannt und das Ergebnis entsprechend in ein Register geschrieben. Die flankengesteuerten Flip-Flops reagieren bei einer bestimmten Steilheit des Steuersignals und aktualisieren anschließend das Ergebnis. Unter den verschiedenen Logikgattern gibt es verschiedene Eingaberegister, die als Steuerregister arbeiten.

D-Flip-Flop

Das D-Flip-Flop wurde in diesem Projekt größtenteils dazu verwendet, um die Eingabe oder Ergebnisse zwischenzuspeichern.

D-Flip-Flop als svg: [Flipflop](#)

T-Flip-Flop

Das T-Flip-Flop wurde aus einem D-Flip-Flop entworfen und dient zum Wechseln zwischen den ausgewählten Registern. Es ändert das Ergebnis, wenn sich die Eingabe ändert und wechselt somit nach jeder Eingabe.

T-Flip-Flop als svg: [Flipflop](#)

Addierer

Um den n-bit Addierer des Rechners umzusetzen, müssen mehrere Volladdierer hintereinander geschaltet werden. Ein Volladdierer besteht aus zwei Halbaddierern, welche jeweils aus einem AND und XOR Gatter bestehen. Als Eingang des Volladdierers werden jeweils ein Bit der zu addierenden Zahl verwendet und ggf. ein Carry-In Bit. Als Ausgang ergibt sich dann die Summe, also das Ergebnis, sowie ggf. ein Carry-Out Bit. Für den Volladdierer bedarf es neben den Halbaddierern zusätzlich noch ein OR Gatter.

Der Aufbau eines Volladdierers muss n-mal hintereinander aufgebaut werden, um einen n-bit Addierer zu erhalten.

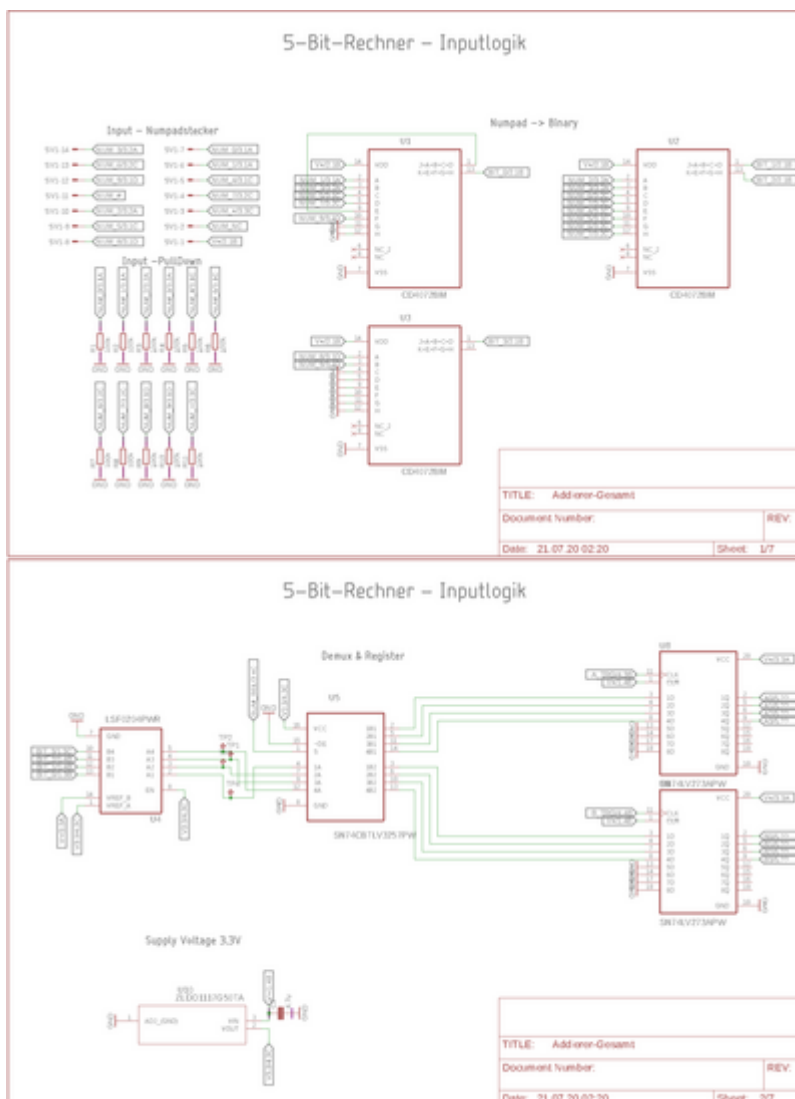
Aufbau eines Volladdierers: [Volladdierer](#)

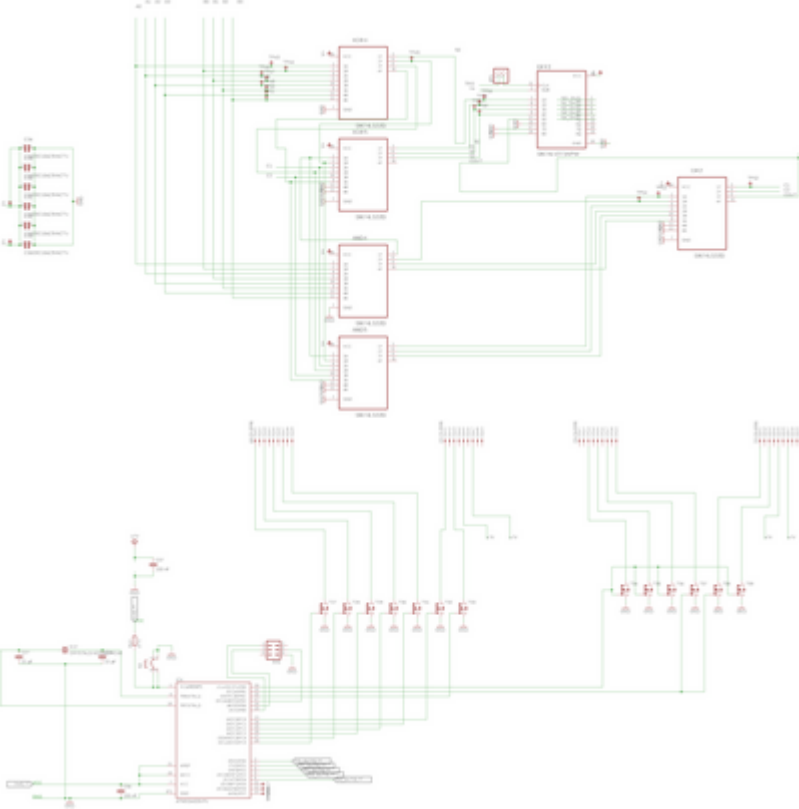
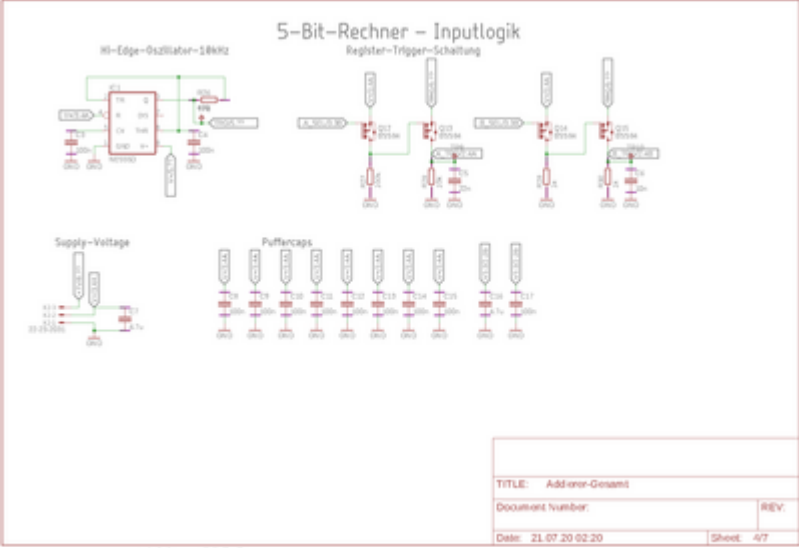
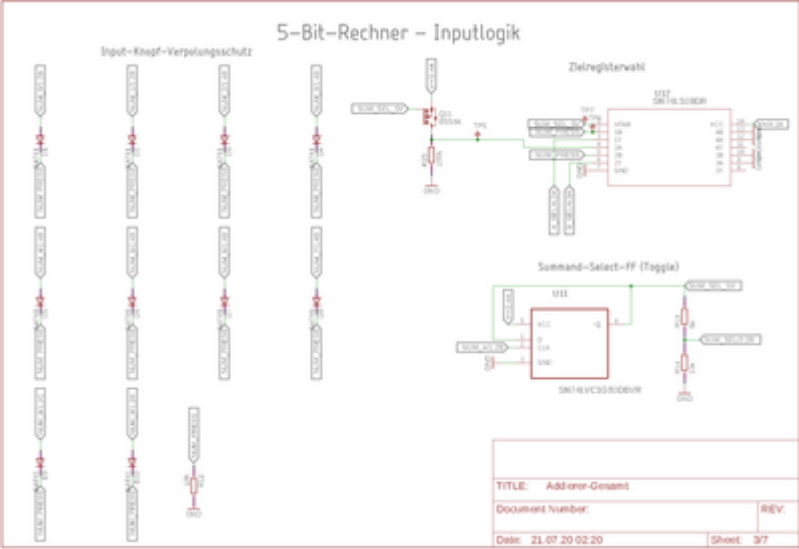
Anzeige

Zur Darstellung des Endergebnisses wurden zwei 7-Segment verwendet. Eine einzelne 7-Segment Anzeige kann die Ziffer 0-9 anzeigen. Da jedoch die Endergebnisse zweistellig werden können bedarf es zwei Anzeigen. Das führte zu dem Problem, dass die Logik der Übersetzung von 5 Bit zu zwei 7-Segment Anzeigen zu komplex wurde und zur Berechnung deshalb ein Mikrocontroller, dem Atmega328P, verwendet wurde. Dieser hat die 5 Bit als Eingang über die Pins erhalten und daraus die zweistellige Zahl berechnet und die 7-Segment Anzeigen angesteuert.

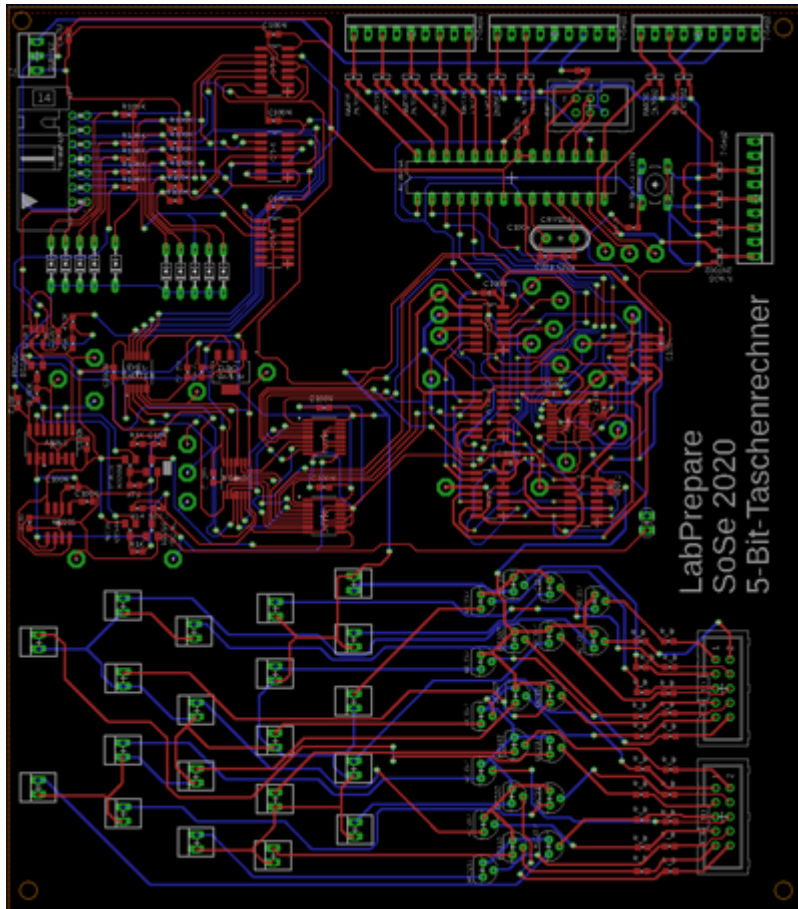
Entwurf der Platine

Eagle Schematics

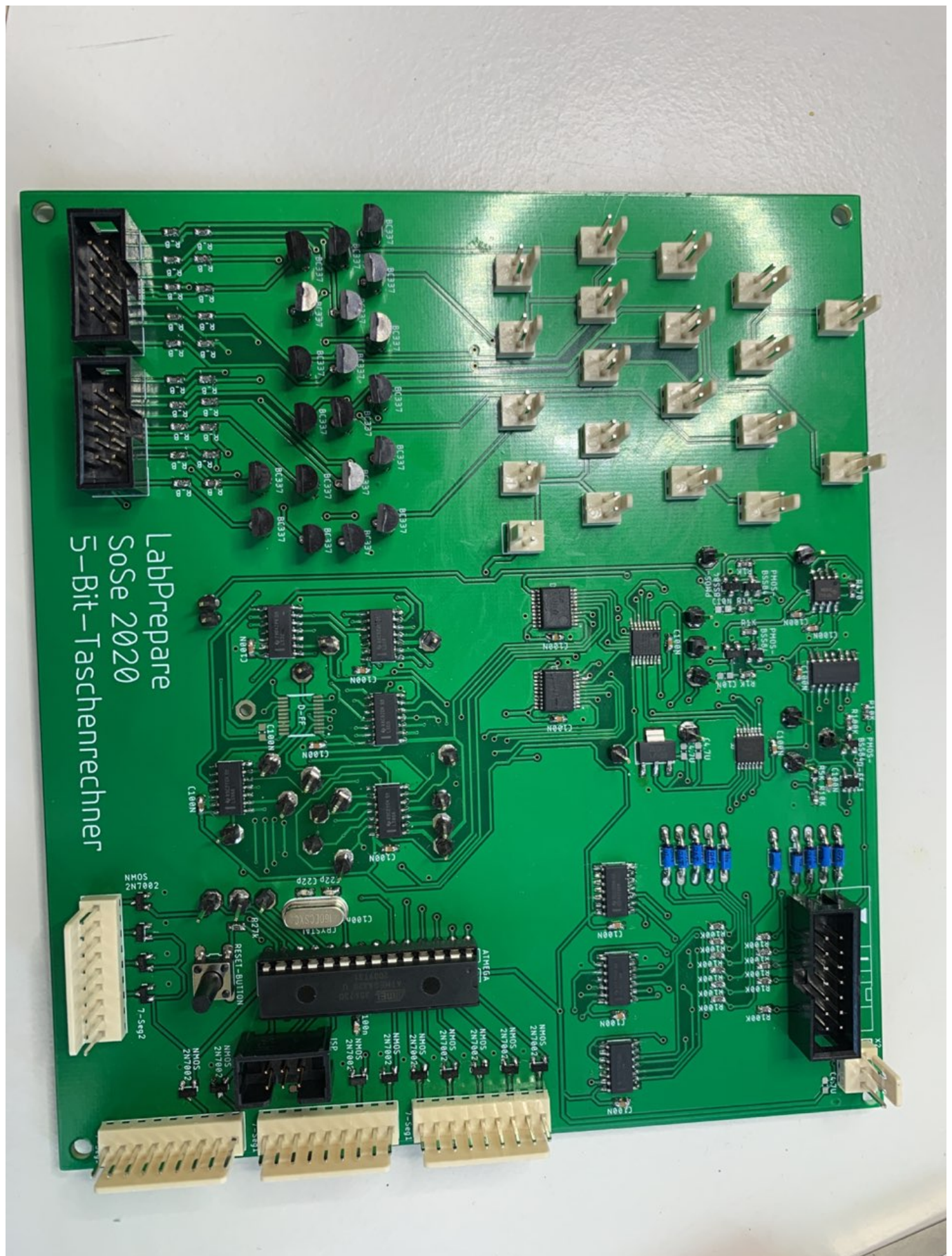




Eagle Layout

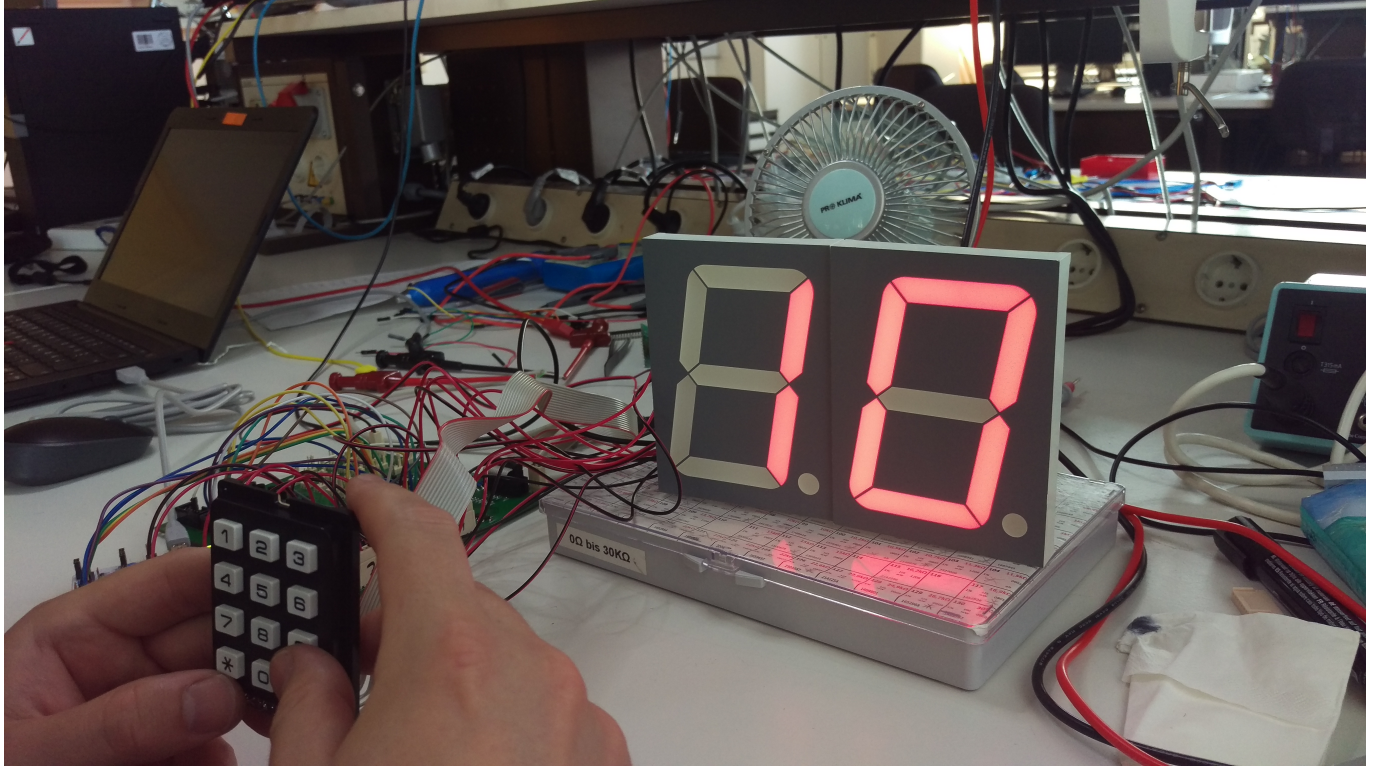


Bestückte Platine



Endergebnis

In diesem Projekt wurde erfolgreich ein 5-Bit Addierer auf Logikebene umgesetzt. Die Eingabe der zwei zu addierenden Zahlen erfolgt über ein Numpad. Diese werden in Register gespeichert und mit den vorher beschriebenen Logikgattern addiert. Anschließend wird das Endergebnis auf zwei 7-Segment Anzeigen dargestellt.



Anstehende Aufgaben

Es werden noch für die Darstellung der Bits LEDs vorbereitet und zusammen mit der Platine in ein anschauliches Gehäuse verbaut. Die LED's spiegeln hierbei die internen binären Zustände der Logikgatter des n-Bit-Addierers wider. Anhand eines Blockschaltbildes auf dem Schaukasten soll es so möglich sein einfach und interaktiv die Funktionsweise eines Addierwerkes nachzuvollziehen. Erklärtexte und Wahrheitstabellen sollen hierbei unterstützend Verwendung finden.

Quellen

- [Volladdierer](#)
- [Demultiplexer](#)
- [Segmentanzeige](#)
- [Flip-Flops](#)

Projektdokumentation

[Projektplan](#)

From:

<https://labprepare.wiki.tu-berlin.de/> - **Project Sci.Com Wiki**

Permanent link:

https://labprepare.wiki.tu-berlin.de/doku.php?id=ss20:n-bit_taschenrechner

Last update: **2021/10/26 22:55**

