



Konrad Zuse

(22.Juni 1910; † 18.Dezember 1995)

Er hat 1937(1941) den ersten programmgesteuerten Digitalrechner gebaut (Basis: **mechanische Logik-Gatter**)!

Vordenker der Digitaltechnik waren **Leibniz** (17/18-te Jahrhundert) und **Boole** (19-te Jahrhundert).

Boole ist vor allem dadurch bekannt, dass die für die Computertechnik grundlegende **boolesche Algebra** nach ihm benannt wurde. Boole erkannte als erster, dass die **Aussagenlogik als eine Algebra** aufgefasst werden kann, die zwei Elemente hat (heute als die zwei Wahrheitswerte bezeichnet: **wahr** oder **falsch**).

Die Boolesche Algebra ist das Fundament der modernen Informationstechnologie.

Eine Realisierung eines Digitalrechners gab es von beiden nicht - es fehlten ja die technischen Mittel, es gab noch keine entsprechenden elektrischen oder gar elektronischen Bauelemente, aber durch **Boole** wurde die theoretische Basis, die von **Leibniz** angedacht war, erarbeitet.

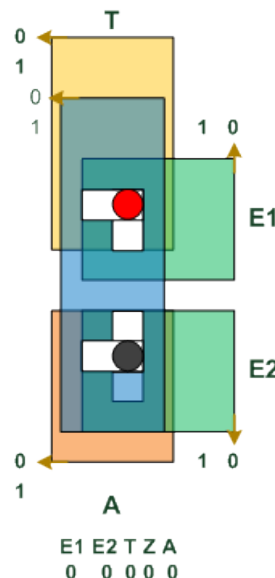
Hier aber kommt Herr Zuse ins Spiel!



AND

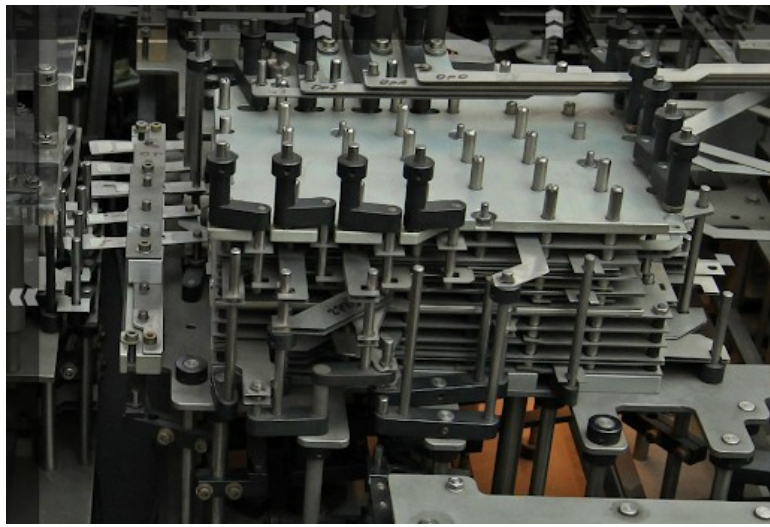
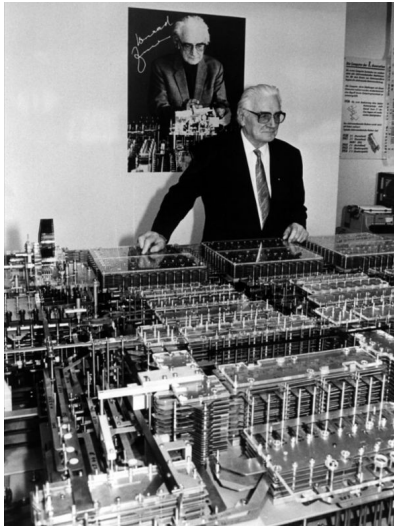


Symbol für AND



Zuse's mechanische Lösung für AND

Letztere **Aussage ist aber nicht ganz richtig**, denn **Herr Zuse** hat mit seiner programmgesteuerten Rechenmaschine Z1 gezeigt, dass auch mechanisch eine Rechenmaschine im Dualsystem realisierbar ist.

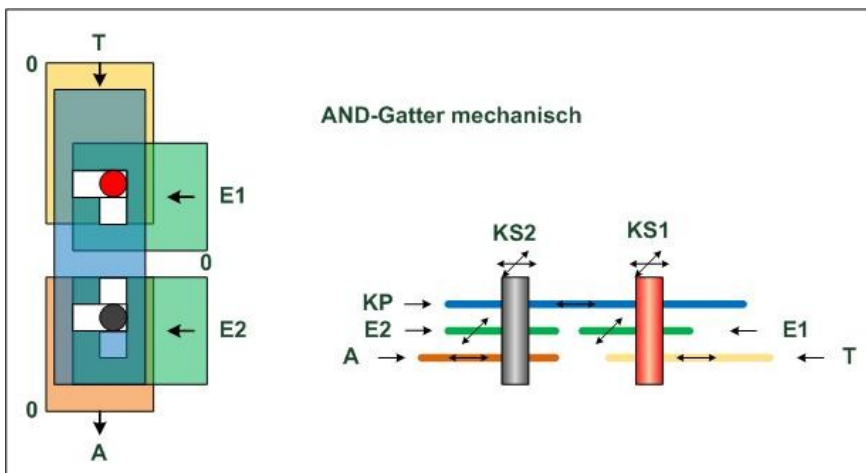


Die Recherei, wie auch die Speicherung der Daten erfolgt rein mechanisch. Die Eingabe der Daten erfolgt durch einen Lochstreifen, die Abarbeitung der Befehle wird durch einen Kurbelantrieb, bzw. durch einen Elektromotor, realisiert. Dazu wird wie in einem modernen Rechner ein Mikroprogramm abgearbeitet

Die erste Maschine wurde durch den Krieg vernichtet. Der Nachbau steht nun auch schon sehr lange im Museum, zwar durch Glas und Plastik geschützt, aber die Materialien der Maschine sind gealtert, so dass Führungen der Bleche nicht mehr die Richtige Position garantieren, so dass dann bei Bewegung mit ausreichender Kraft diese verbiegen und die gesamte Maschine nicht mehr funktioniert.

Z1-Addierermodelle: <https://www2.hs-fulda.de/~grams/mathehilft/Zuse/Z1-Addierermodelle.pdf>

Es erfordert schon eine Menge Vorstellungskraft, die Funktion der Maschine zu verstehen. Es wird alles auf die Grundelemente der Boolesche Algebra zurück geführt, folglich muss es möglich sein diese Elemente zu separieren. Ich möchte deshalb kurz ein mechanisches **AND-Gatter** vorstellen und auch eine Simulation zeigen.

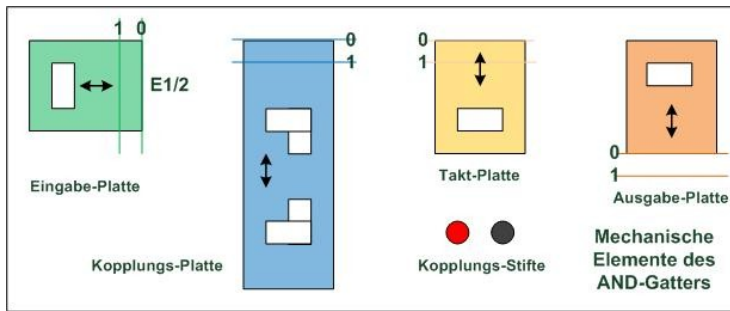


Es besteht aus drei Ebenen:
Takt- und Ausgabebene,
Eingabe-Ebene und
 einer **Kopplungs-Ebene.**

Die Kopplungsebene realisiert die Funktion des Gatters. Wichtig sind auch die **Kopplungsstifte**, sie verbinden die einzelnen Elemente der Ebenen und

bewegen sich in speziellen Ausschnitten in den Elementen. Das Auslösen der Bewegung erfolgt

durch das Takt-Element. Die Stifte sind völlig frei, natürlich in den Ausschnitten, beweglich und verändern zum Teil die Position der Elemente in den Ebenen (Ausgabeelement, Kopplungselement). Darstellung der Elemente:



Die Eingabeplatten bewegen sich waagrecht, die anderen Platten nur senkrecht. Die Kopplungsstifte hingegen können sich sowohl waagrecht wie senkrecht bewegen und übertragen ihre Bewegung auf alle anderen Platten.

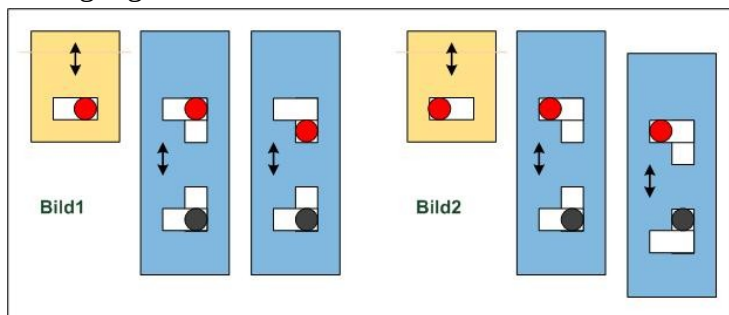


Bild1:

Die Eingabeplatte steht auf 0, somit steht der rote Kopplungsstift rechts in der Kopplungsplatte (auch der schwarze). Wird nun der Takt betätigt (auf 1 nach unten geschoben) rutscht der rote Kopplungsstift im Schlitz der Kopplungsplatte nach unten – die

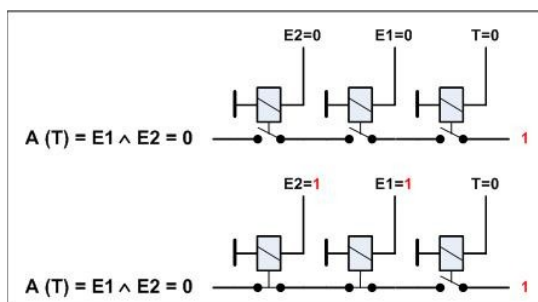
Kopplungsplatte wird nicht bewegt! Der schwarze Stift bleibt deshalb an seiner alten Position.

Bild2:

Die Eingabeplatte steht auf 1, also wurde der rote Stift nach links geschoben. Jetzt bewegt der Takt 1 die Kopplungsplatte nach unten, da es an dieser Stelle keinen Schlitz nach unten gibt. Aber der schwarze Stift kann in dem Schlitz nach oben rutschen – er bewegt sich nicht von der Stelle, d.h. die Ausgabe bleibt unverändert.

Nur wenn die Eingabe2 den schwarzen Stift nach links geschoben hatte, würde auch der schwarze Stift nach unten geschoben (andere Ausgabe).

Genau so soll ein AND-Gatter funktionieren, nur wenn beide Eingänge auf 1 stehen wird auch die Ausgabe auf 1 gesetzt!



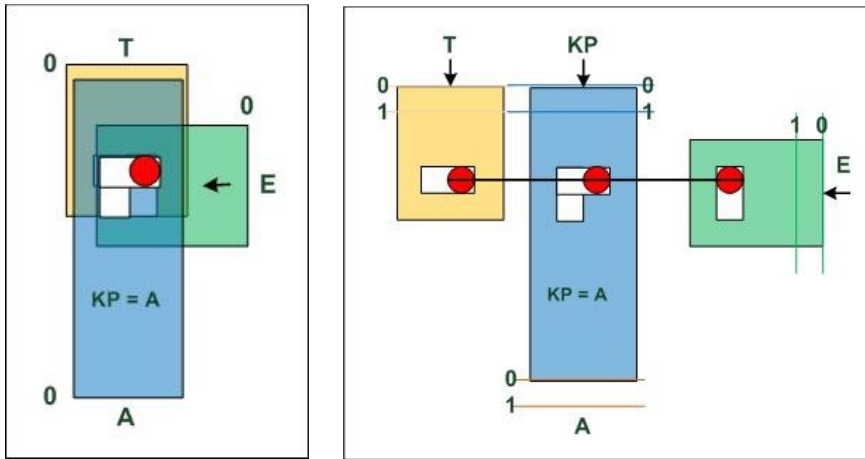
Das Bild soll das logische mechanische AND mit Relais simulieren.

Werden die Eingänge E1 und E2 auf 1 gesetzt, schließen zwar die Relais, aber es fließt kein Strom, da der Takt noch offen ist.

So funktioniert das mechanische Element eben auch. Durch Einschieben von E1 und E2 wird das AND vorbereitet, ausgelöst aber erst wenn T auf 1 geschoben wird

In ähnlicher Weise kann auch ein Negator gestaltet werden.

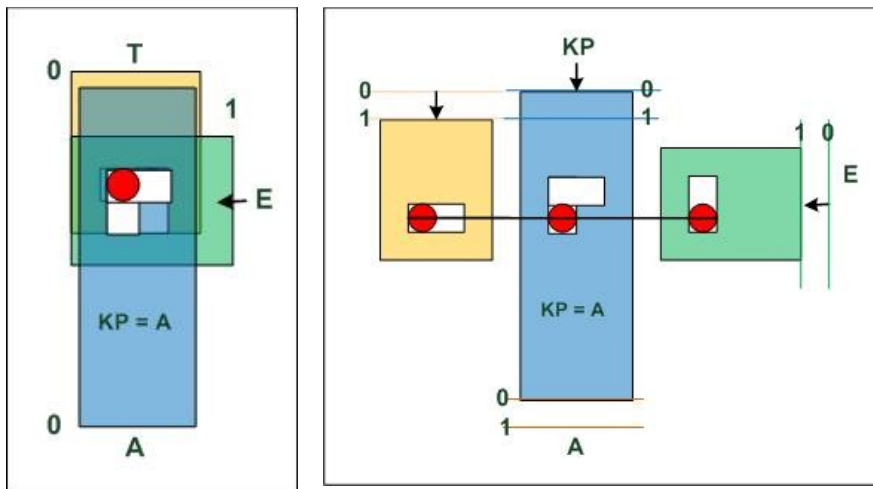
Es gibt natürlich nur einen Eingang, eine Takt und Kopplungsplatte, die in diesem Fall zugleich Ausgabe ist:



Ist der Eingang 0 steht der Kopplungsstift rechts in dem Ausschnitt der Kopplungsplatte.
 Wird nun der Takt auf 1 gesetzt, wird auch die Kopplungsplatte über den Kopplungsstift nach unten geschoben, da keine andere Möglichkeit besteht.
 Am Ausgang erscheint eine 1, das ist richtig weil für den Negator gilt:

$$A = \neg E \quad (1 = \neg 0)$$

Anders hingegen die Situation $E = 1$:



Der Kopplungsstift steht nun links.
 Wird nun der Takt auf 1 gesetzt, rutscht der Stift in den Schlitz in der Kopplungsplatte nach unten, die Platte bewegt sich nicht, am Ausgang bleibt 0. Das ist richtig, denn es gilt:

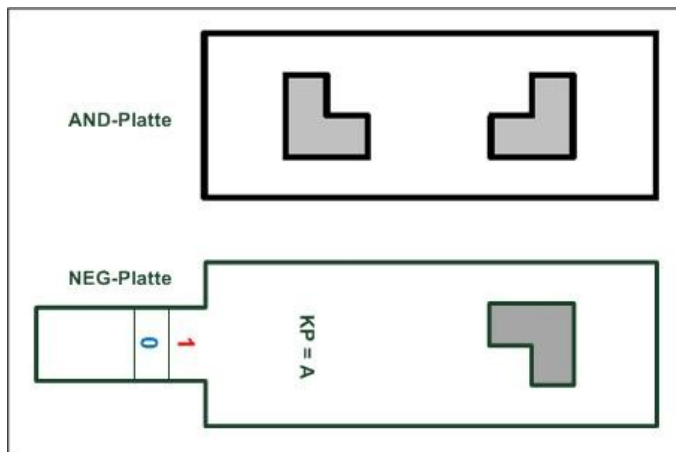
$$A = \neg E \quad (0 = \neg 1)$$

Im Internet kann man zu der Problematik mehrere Beiträge finden, sehr interessant ist der Betrag:

Bastelbogen für ein mechanisches Schaltgliedmodell

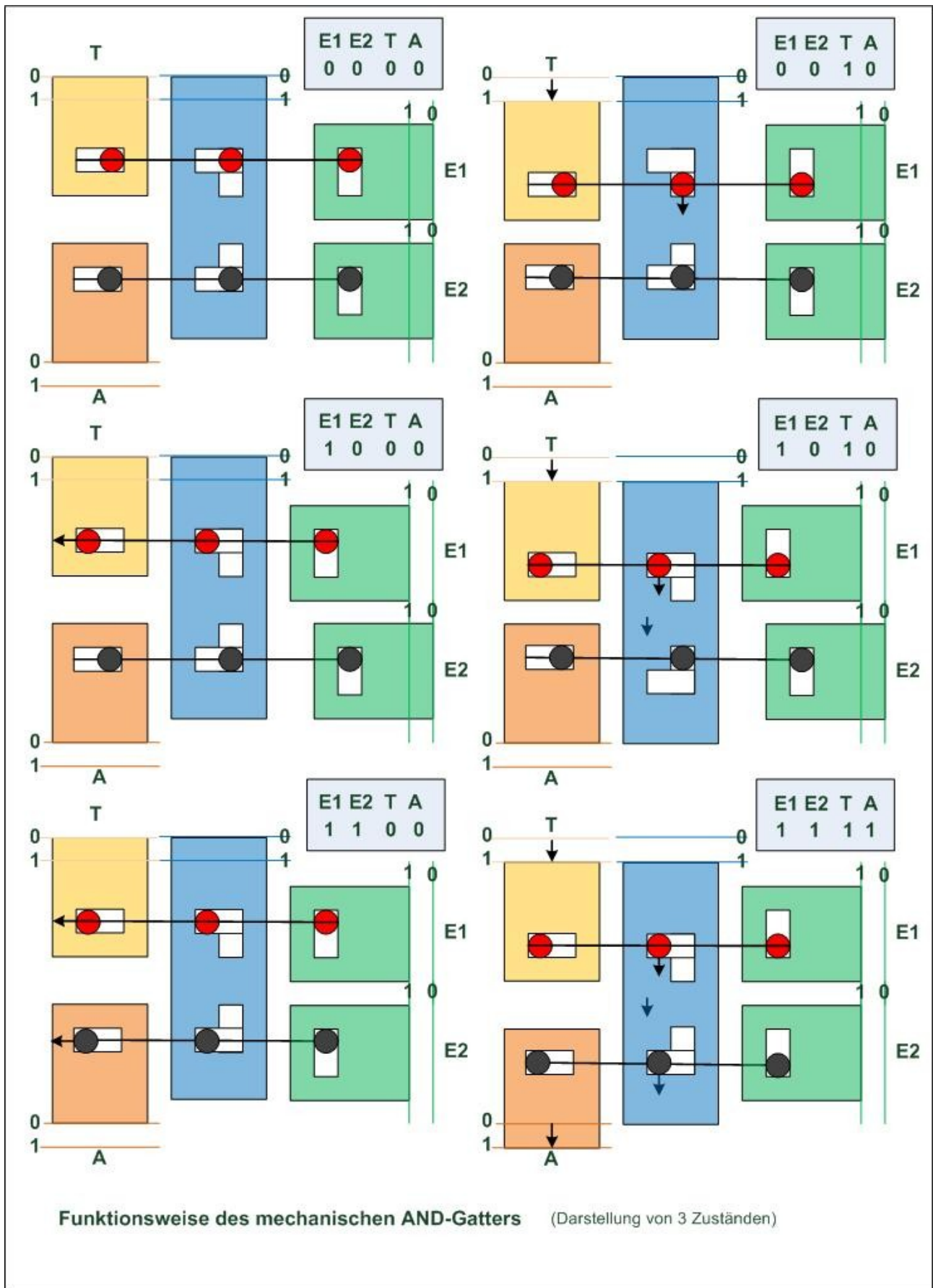
Timm Grams, Fulda, 17. März 2010 (aktualisiert: 26.07.10)

Neben einer kleinen Beschreibung wird für ein **AND-Gatter** eine Möglichkeit angeboten, dieses Teil aus Papier zu gestalten. Ich habe es nachgebaut und es funktioniert gut. Weiterhin habe ich auch einen Negator mit diesem Teil realisiert, es wird nur ein Eingang genutzt und die Kopplungsplatte so gestaltet, dass sie die Funktion NEG realisiert und zugleich das Ergebnis ausgibt:



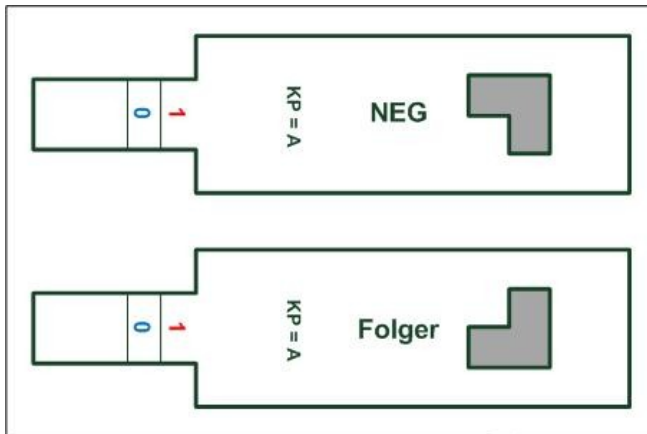
Die Grundplatte kann man natürlich auch halbieren, ich wollte jedoch das gleiche Teil benutzen und nur die Kopplungsplatte austauschen.

Ich habe diesen [Bastelbogen](#) ebenfalls hier per Link verbunden, **es ist nicht mein Eigentum, es bleibt natürlich Eigentum von Herrn Grams!**



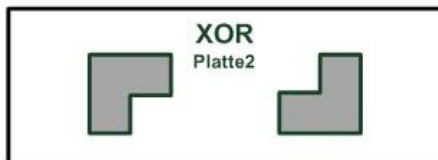
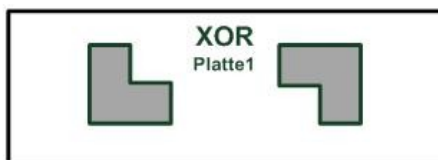
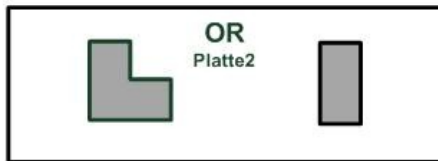
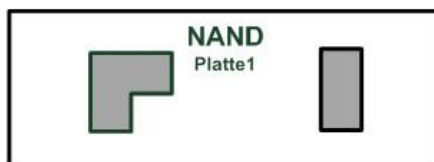
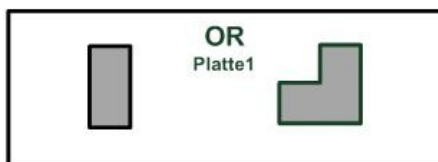
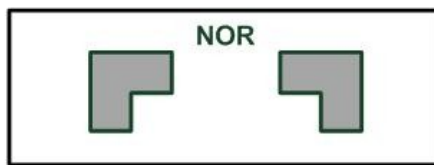
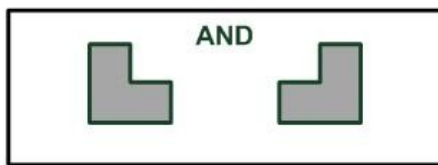
Das sind so etwa die logischen Schalt-Elemente im Zuse-Rechner Z1.
(hier gibt es ein Beispiel der AND-Simulation)

Weitere logisch, mechanische Grundelemente



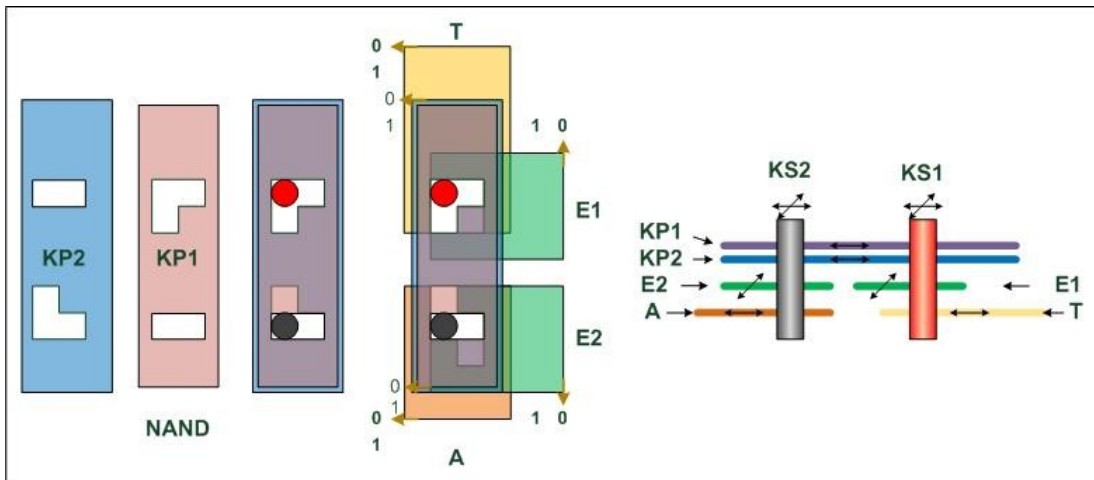
Kopplungsplatten für verschiedene logische Funktionen

Dreht die Platte, wie neben dargestellt, wird aus der Negation ein Folger bzw. ein **Einschalter**, z.B. um eine Übernahme von Daten zu steuern.

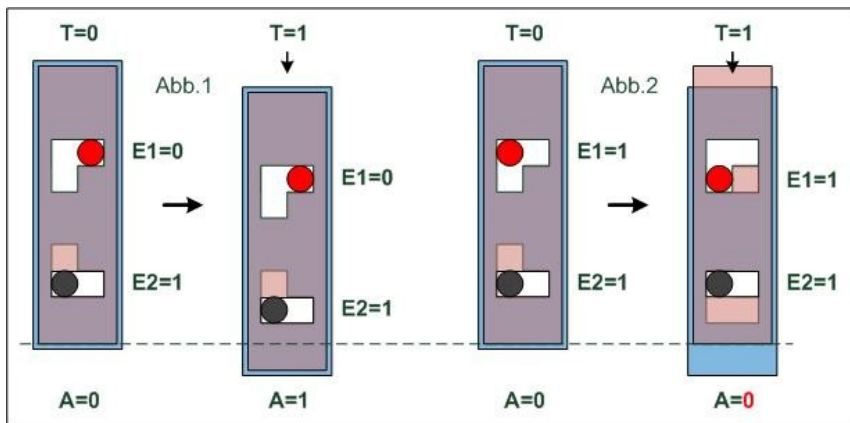


Für die Funktionen **OR, XOR und NAND** Braucht man jeweils 2 Kopplungs-Platten. Diese werden übereinander in die Simulationsbox gelegt
AND und NOR wird mit einer Platte realisiert. Die **Negation** wurde durch eine größere Platte als erforderlich (es gibt nur einen Eingang) realisiert, da wieder die gleiche Simulationsbox genutzt werden sollte.

Unter Einsatz dieser Koppelplatten sollte es nun möglich sein alle notwendigen mechanische logische Elemente zu schaffen, z.B. ein **NAND**-Element



Es werden die oben gezeigten Koppelplatten KP1 und KP2 gebraucht. Die Reihenfolge der Platten spielt keine Rolle, jedoch müssen sie wie dargestellt, zu den Eingabplatten platziert werden. Die Kopplung aller Platten erfolgt ohnehin nur über die Kopplungsstifte.



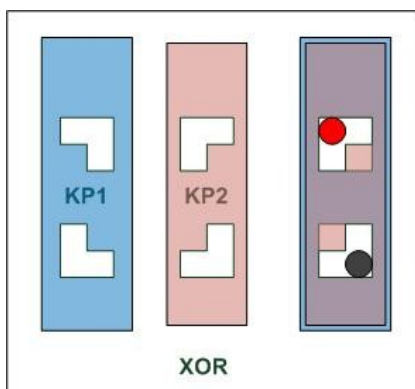
Das nebenstehende Bild zeigt die Kopplungsplatten und die Kopplungsstifte. Der schwarze Stift ist zuständig für die Ausgabe. Die Abb.1 zeigt die Situation, dass nur ein Stift nach links geschoben ist. Bei T=1 wird der Stift herausgeschoben, A ist 1. Das trifft auch für die anderen beiden Fälle (E1=0, E2=1; E1=1, E2=0) zu.

Nur für den in Abb.2 dargestellten Fall (E1=1, E2=1 und T=1) wird der schwarze Stift nicht bewegt, der Ausgang ist 0. Das entspricht den Regeln für **NAND**:

E1	E2	A
0	0	1
1	0	1
0	1	1
1	1	0

([NAND-Simulation](#) zwei Fälle)

In ähnlicher Weise lässt sich auch die logische Funktion **XOR** (bzw. Antivalenz) realisieren. Oben sind bereits die Kopplungsplatten, es sind zwei, dargestellt:



Beide werden übereinander gelegt, die Reihenfolge spielt keine Rolle.

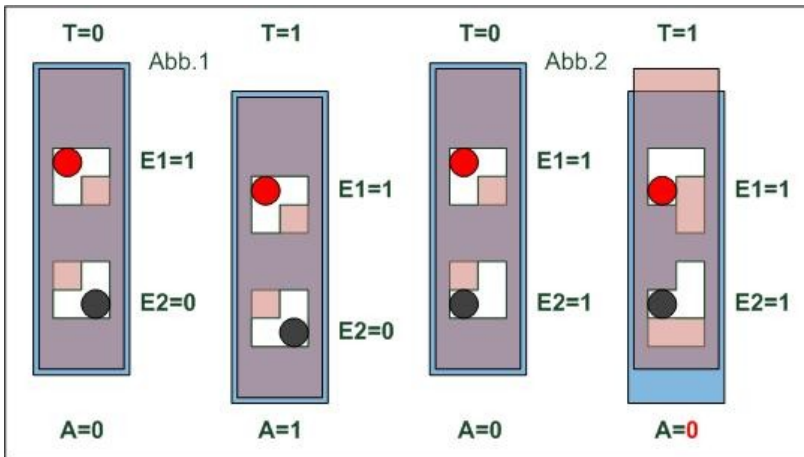


Abb.1 zeigt in dem Bild die Ungleichheit der Eingänge ($E1=1, E2=0$). Als Ergebnis stellt sich am Ausgang 1 ein. Für den anderen nicht dargestellten Fall ($E1=0, E2=1$) entsteht das gleiche Ergebnis. Abb.2 zeigt den Fall der Gleichheit ($E1=1, E2=1$). Dadurch das eine Kopplungsplatte herausgeschoben wird, die andere aber nicht,

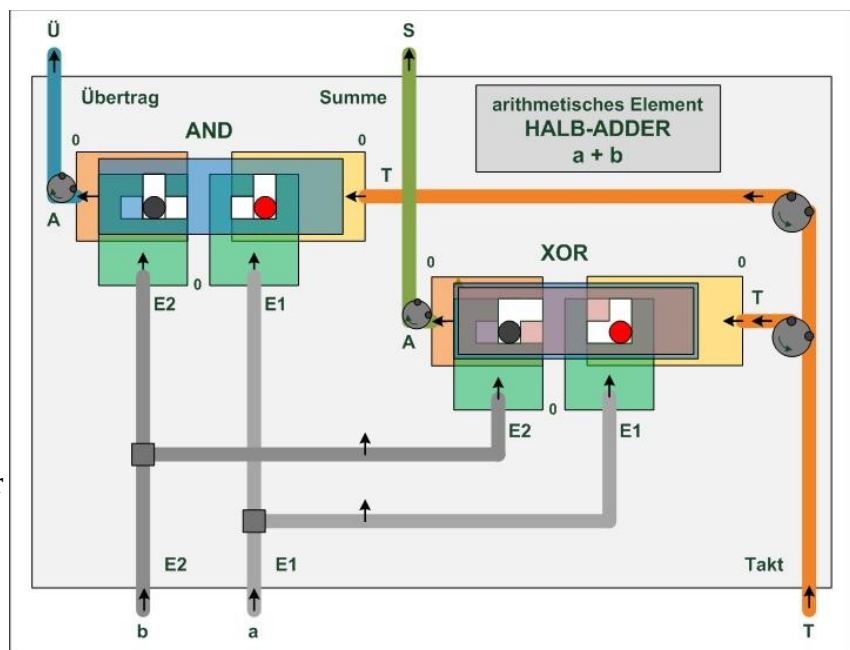
bleibt der schwarze Kopplungsstift an der Stelle, das Ergebnis ist (wie auch bei $E1=0, E2=0$), gleich 0.

(hier kann man eine Simulation der XOR-Funktion sehen, nur die eben beschriebenen 2 Fälle))

Arithmetisches Grundelement PLUS (+)

Nun ist es auch möglich mit den logischen Elementen **XOR** und **AND** ein **HALB-ADDER** zu erstellen (siehe oben), ein arithmetisches Element:

Genau diese Erkenntnis hatte schon **Leibniz**, er hätte es auch realisieren können!



Weitere Information zur Simulation und Test logischer Funktionen, einschließlich der Bauanleitung des Simulators kann man unter den oben bereits genannten Beitrag lesen:

Timm Grams, Fulda; [Bastelbogen für ein mechanisches Schaltgliedmodell](#)

Wir haben nun ein **AND-Gatter** und ein **XOR-Gatter**, somit ist es möglich das arithmetische **PLUS** zu realisieren.

Und ich wollte nur einfach wissen, wie es geht – **und es geht!**