

Der Logikus-Simulator

Werner Lauff, www.lauff.org, lauff@lauff.org

Bevor Sie den Logikus-Simulator in Excel laden, empfehlen wir Ihnen, die Datei in einem neuen Ordner auf Ihrer Festplatte zu speichern (zum Beispiel C:\Logikus) und dieses Verzeichnis in Excel unter "Datei" / "Optionen" / "Trust-Center" / "Einstellungen für das Trust Center" / "Vertrauenswürdige Speicherorte" als "vertrauenswürdigen Speicherort" hinzuzufügen. Dann können Sie die in der Datei enthaltenen Makros nutzen, ohne dass Dateien aus anderen Verzeichnissen oder dem Internet Makros ausführen können.

Was ist der Logikus?

Der Spielcomputer Logikus ist ein Bausatz, den der KOSMOS-Lehrmittelverlag im Jahr 1968 für den damals beachtlich hohen Preis von 68 DM auf den Markt gebracht hat. Trotz seines Namens verwendet er keine Halbleiter wie Transistoren, sondern setzt auf mechanische Elemente und einfache Elektrik.

kosmos SPIELCOMPUTER LOGIKUS

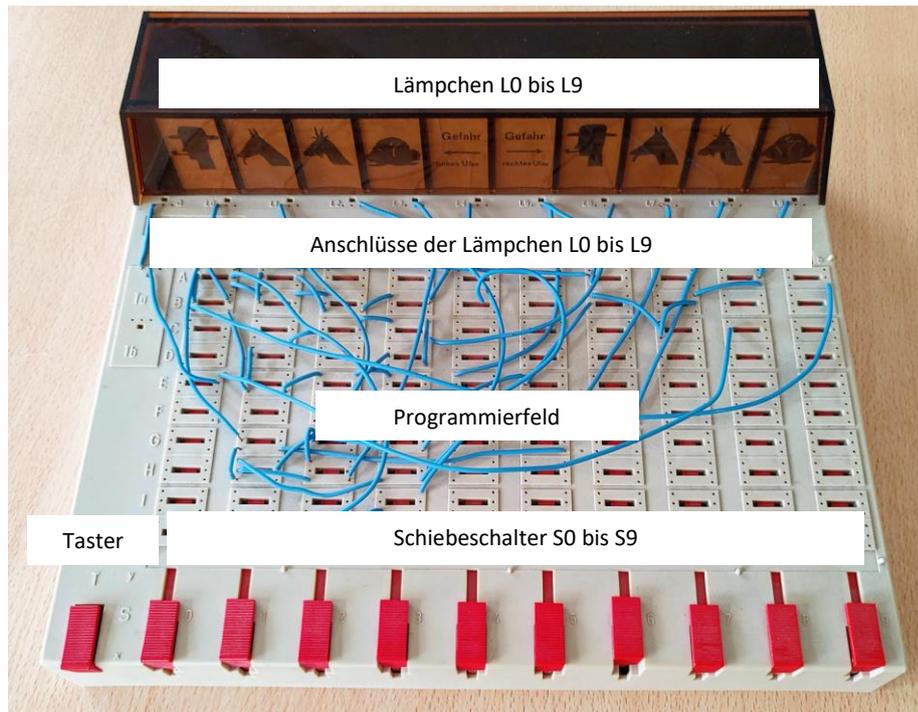
Jeder kann ohne Schwierigkeit ein elektrisches Schaltgerät zusammensetzen, mit dem man das wesentliche Prinzip kennenlernt, nach dem auch die größten Computer arbeiten. Durch steckbare Drahtverbindungen auf dem Programmierfeld seines Schaltpults läßt sich der LOGIKUS für die verschiedensten Aufgaben programmieren. So lernt man z. B. ein Anzeigergerät für Wettervorhersage, einen einfachen Tischrechner, Schaltungen für Rätsel, Probleme und Denksportaufgaben aus Alltag und Technik, Diagnosecomputer, Geheimschriftübersetzer und Intelligenztestgerät im praktischen Versuch kennen. Die Schaltmöglichkeiten sind fast unbegrenzt.

Die Beschäftigung mit dem LOGIKUS schult die logische Denkfähigkeit und vermittelt, dank des methodischen Arbeitsbuches, fundierte Kenntnisse, die heute — im Zeitalter der Automation — jeder gebrauchen kann.

Für Jugendliche wie für Erwachsene ist der LOGIKUS gleichermaßen geeignet.

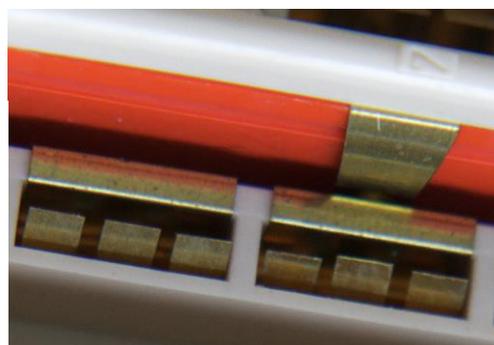
Für die Versuche wird zusätzlich eine 4,5 Volt Taschenlampenbatterie benötigt.

Das Gerät besteht aus zehn Schiebeschaltern (S0 bis S9), mit denen sich - jeweils für die ganze Spalte einheitlich - der Schaltzustand 1 oder 0 einstellen lässt. Außerdem gibt es zehn (durch Folien beschriftbare) Lämpchen L0 bis L9, deren Verhalten (leuchtet / leuchtet nicht) "programmiert" werden kann. Die Schiebeschalter und die Lämpchen haben zunächst nichts miteinander zu tun. Es könnte also auch fünf Schalter und 20 Lämpchen geben oder 20 Schalter und fünf Lämpchen.



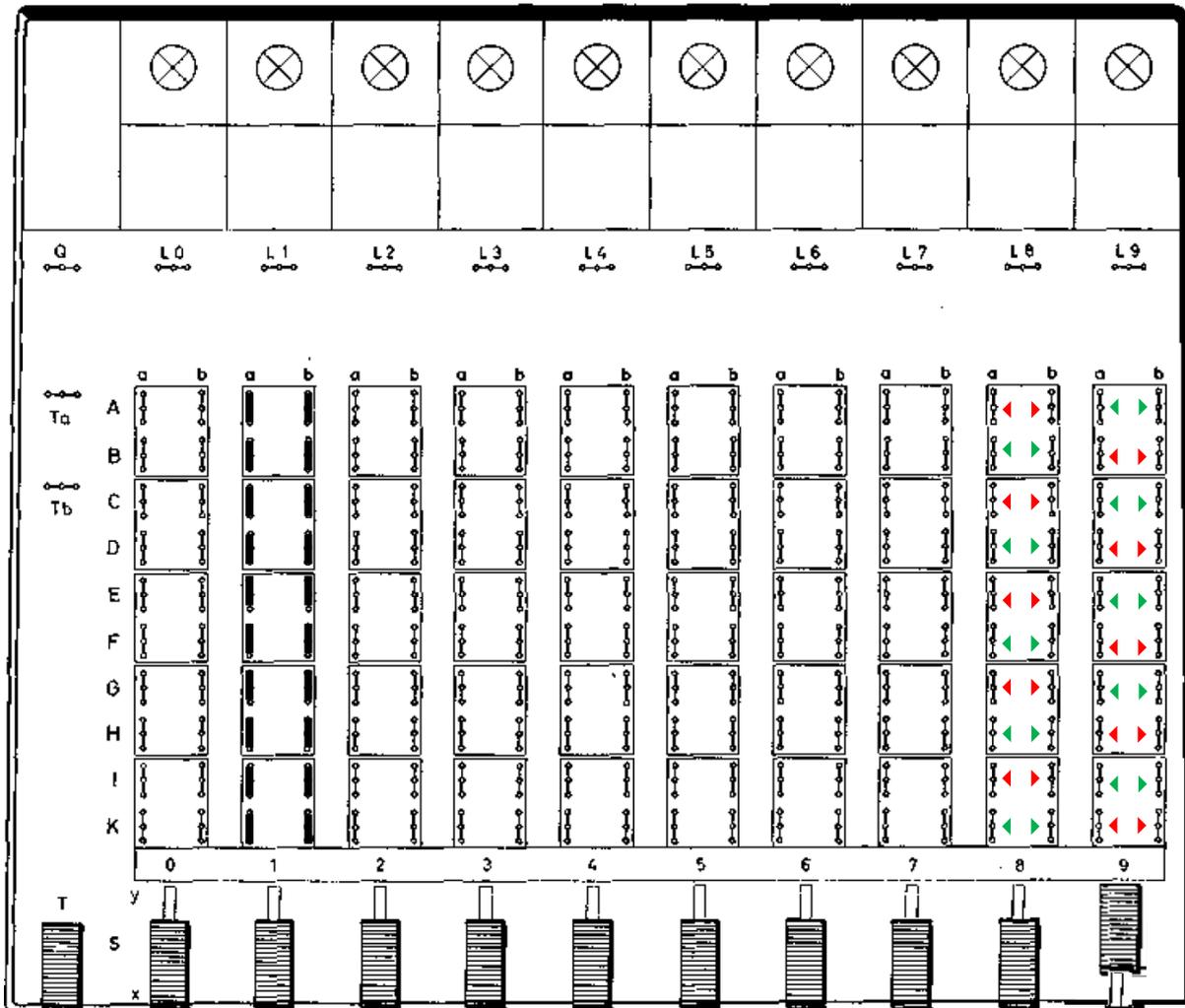
Die Schiebeschalter stellen mit Hilfe von Zungen im Inneren des Logikus den Kontakt zwischen gegenüber liegenden Federn her. Zu jeder Feder gehören drei Anschlüsse pro Seite; in welchem der drei Anschlüsse ein Kabel steckt, ist unerheblich.

nicht verbunden:
Die Zunge verbindet andere Federn; diese hier nicht.

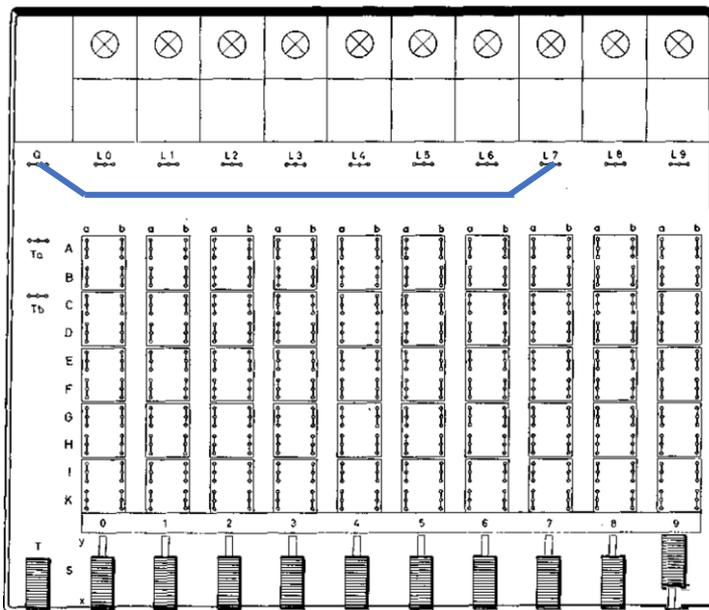


verbunden:
Die Zunge stellt den Kontakt mit der gegenüberliegenden Feder her

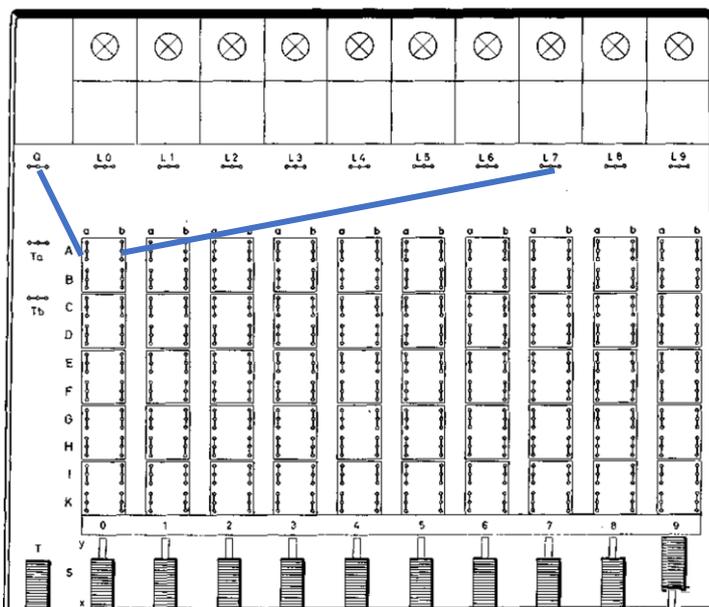
Es gibt fünf Gruppen von Federn: AB, CD, EF, GH und IK. In ihnen werden durch den Schalter immer zwei Federn miteinander verbunden, entweder die oberen (zum Beispiel Aa mit Ab) oder die unteren (Ba mit Bb) - siehe die Symbolzeichnung bei Schalter 1. Die oberen Federn sind verbunden, wenn der Schalter nach oben geschoben wird; die unteren Federn sind verbunden, wenn der Schalter nach unten geschoben wird (siehe Schalter 8 und 9) Eine Verschiebung des Schalters betrifft immer sämtliche Federngruppen.



Nun zu den Lämpchen. Man bringt ein Lämpchen zum Leuchten, indem man eine elektrische Verbindung zu ihm herstellt. Dazu muss man es lediglich mit einem Plus-Signal versorgen (das bei "Q" gleich Quelle oben links anliegt); der Rückweg zum Minuspol (Masse) verläuft im Gehäuse.

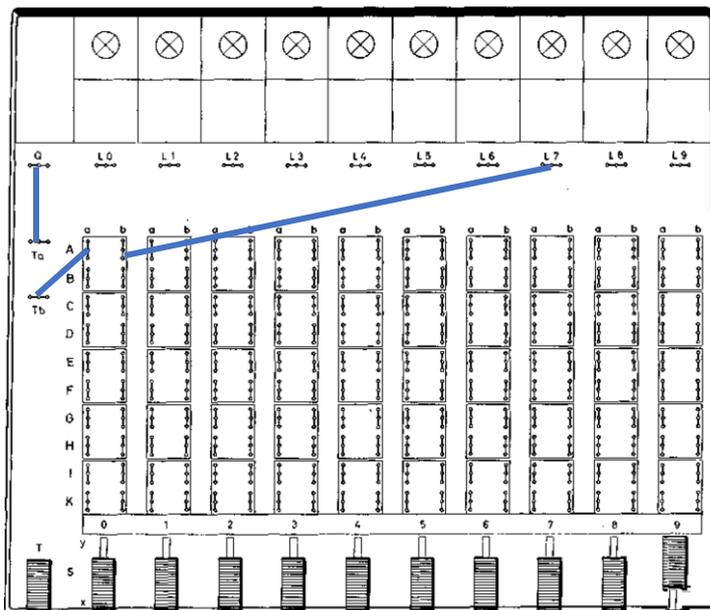


Oben sehen Sie eine einfache, aber sinnlose Schaltung: L7 wird permanent angeschlossen, leuchtet also immer. Sollen die Lämpchen nicht immer leuchten, kommen die Schalter ins Spiel. Beispiel: L7 soll leuchten, wenn Schalter S0 auf 1 steht, also nach oben geschoben wurde. Um das zu erreichen, nehmen Sie ein Stück Draht und stellen eine Verbindung zwischen Q und Aa her. Anschließend stecken Sie einen zweiten Draht in Ab und führen ihn zu L7. Schieben Sie S0 nach oben, ist das so, als würden Sie in einem Raum den Lichtschalter betätigen. Nichts Besonderes also, außer der Erkenntnis: Schaltungen sind WENN-DANN-Programmierungen, in Programmiersprache: IF - THEN.

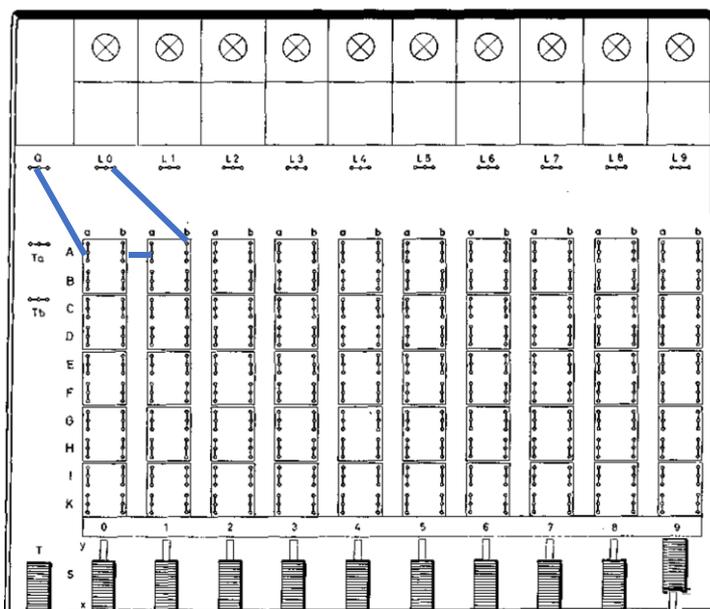


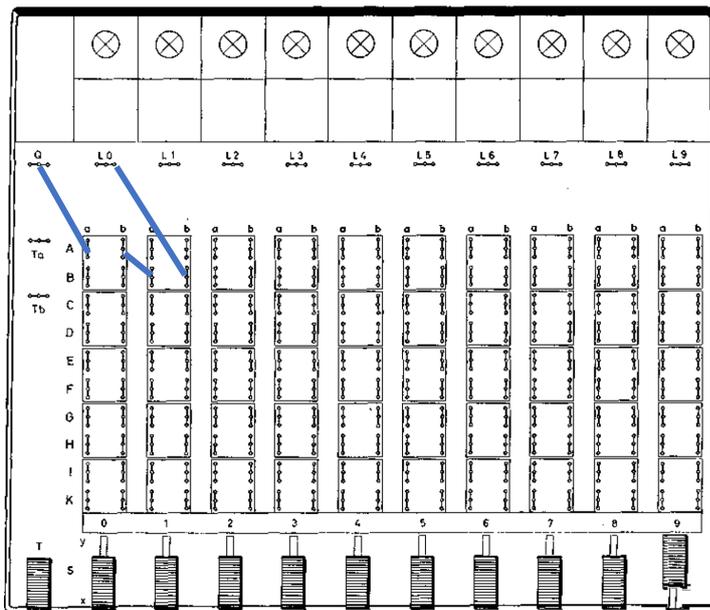
Bezieht man den Taster T in den Stromkreis ein, können die Schalter betätigt werden, ohne dass bereits ein Lämpchen leuchtet. Erst wenn man am Ende den Taster drückt, wird das Ergebnis sichtbar. Eine Parallele wäre ein Tresor, bei dem man zunächst eine Zahlenkombination einstellt. Am

Ende dreht man ein Rad, um das Schloss zu öffnen. Beim Logikus wären erst alle Schalter auf 1 oder 0 zu stellen, um am Ende durch Druck auf T zu prüfen, ob die Kombination richtig war.



Jetzt kommen zwei Raffinessen des Logikus ins Spiel. Die erste ist, dass Sie nicht nur einen Schalter, sondern zehn Schalter haben, die Sie aufeinander aufbauend verkabeln können. Nehmen wir an, die Lampe L0 soll nur leuchten, wenn Sie den Lichtschalter S0 und den Lichtschalter S1 betätigen. Verkabeln würde man diese UND-Bedingung so:

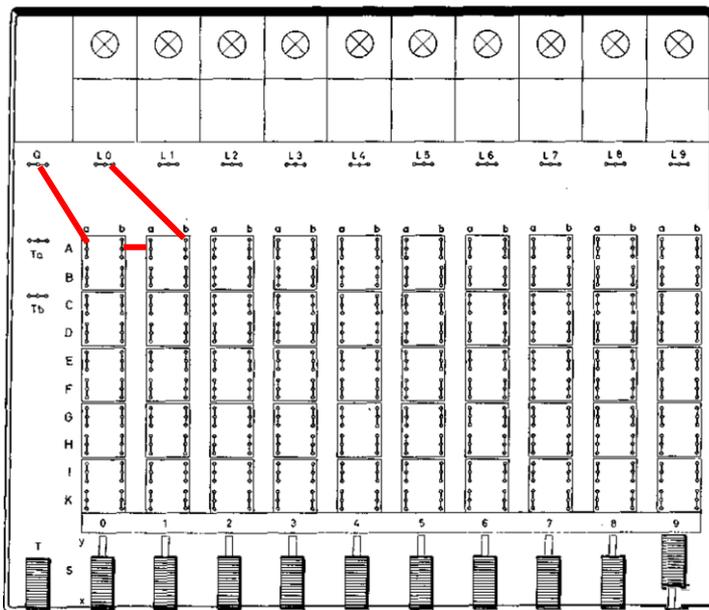




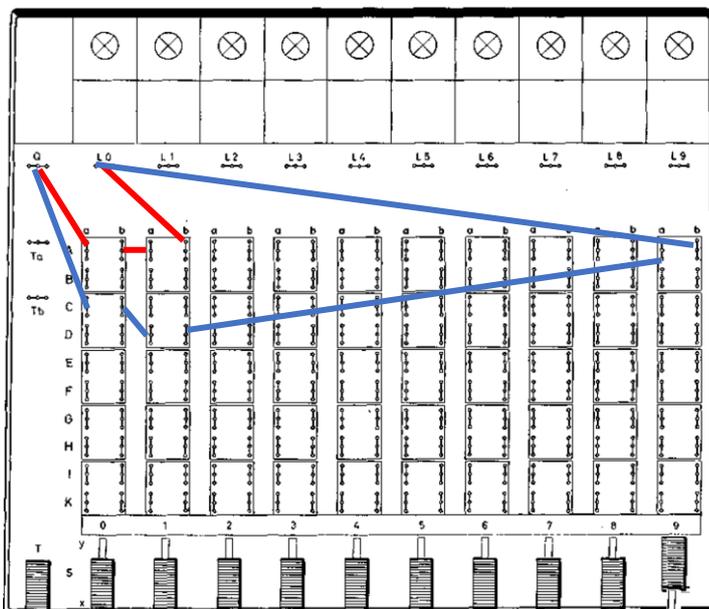
Im Beispiel oben verläuft das Kabel vom Schalter S0 in den unteren von Schalter S1. Sie können es fast ablesen: L0 soll leuchten, wenn S0 auf 1 und S1 nicht auf 1 (also auf 0) steht. Schon haben wir eine "UND NICHT"-Bedingung, die dadurch möglich wird, dass Sie eben auch mit der Stellung 0 eines Schalters etwas anfangen können.

Nun zur zweiten Raffinesse des Logikus. Jeder Schalter steuert nicht nur die Felder AB, sondern auch CD, EF, GH sowie IK. Zwischen ihnen besteht keine elektrische Verbindung. Das bedeutet, dass Sie die Gruppen unabhängig voneinander nutzen können. Erinnern Sie sich bitte: Mit der Schalterstellung "oben" (gleich 1) machen Sie die Strecken AaAb, CaCb, EaEb, GaGb und IaIb gleichzeitig leitend. Mit der Schalterstellung "unten" (gleich 0) machen Sie die Strecken BaBb, DaDb, FaFb, HaHb und KaKb gleichzeitig leitend und entziehen den zuvor genannten Verbindungen den Strom.

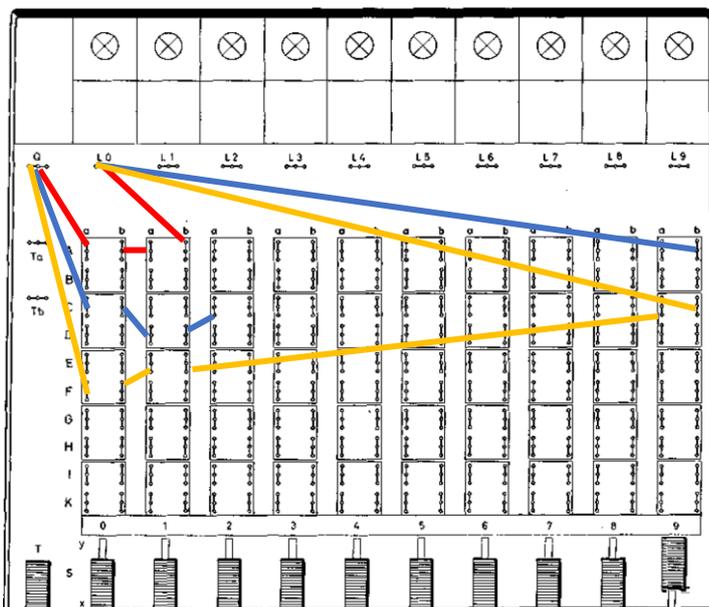
Nehmen wir an, das Lämpchen L0 soll leuchten, wenn entweder S0 und S1 auf "1" stehen oder nur S0 oder S1 eingeschaltet sind, dafür aber S9 den Wert "1" hat. Es gibt also drei Möglichkeiten der Schalterstellung. Sie verwirklichen diese Schaltung mit drei Kabelsträngen, wie auf der nächsten Seite abgebildet. Man könnte in der Schaltung übrigens Abkürzungen vornehmen, also bereits vorhandene Kabel mitbenutzen. Wir haben darauf zur besseren Verständlichkeit aber verzichtet.



Erster Kabelstrang (rot): Entweder S0 und S1 stehen auf 1 ...



Zweiter Kabelstrang (blau): ... oder S0=1 und S1=0 und S9 = 1 ...



Dritter Kabelstrang (orange): ... oder S0=0 und S1=1 und S9=1 ...

Die Syntax des Simulators

Leider ist der Logikus, selbst in gut erhaltenem Zustand, heute kaum noch bedienbar: Die Zungen und Federn haben ihre Leitfähigkeit arg eingebüßt. Deswegen gibt es diese Simulation. Der Unterschied zwischen ihr und dem realen Logikus ist nicht sehr groß. Anstatt Schiebeschalter zu bedienen, klicken Sie mit der Maus auf Schaltflächen und schalten so von 0 auf 1 und von 1 auf 0 um. Die Schalter sind die hellgrauen Feldern, auf denen S0 bis S9 steht. In den hellgrünen Feldern darüber sehen Sie den derzeitigen Status (0 oder 1). Mit einem Klick wechseln Sie von 0 auf 1 und umgekehrt.

The screenshot shows the simulator interface. At the top, there is a row of 10 light orange boxes labeled "Lämpchen L0 bis L9". Below this is a large grid labeled "Programmierfeld" with 10 columns and 10 rows. To the right of the grid are buttons for "Reset All", "Reset S", "Hide", and "Show", along with the text "Version 1.1". At the bottom, there is a row of 10 light green boxes labeled "Schiebeschalter S0 bis S9" and a "Taster" button. Below the switches, there is a row of 10 light orange boxes labeled "Schalter 1" to "Schalter 9".

0	0	0	0	0	0	0	0	0	0	N
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	Taster
Schalter 1	Schalter 2							Schalter 8	Schalter 9	

Excel-Simulation des Spielcomputers Logikus. Simulator von Werner Lauff, www.lauff.org, lauff@lauff.org. Kostenloser Download mit ausführlichem Handbuch: lauff.org/logikus.html.

Anstatt Kabel zu ziehen, programmieren Sie die Lämpchen mit Zifferangaben. Die Programmierung findet im weißen Block in der Mitte statt. Pro Lämpchen können Sie nacheinander und unabhängig voneinander zehn verschiedene Bedingungen setzen. Wollen Sie L0 so programmieren, dass es leuchten soll, wenn Schalter S0 und Schalter S1 auf 1 stehen, schreiben Sie in ein beliebiges Feld unter L0 den "Programmcode" 11xxxxxxxX. Soll L1 leuchten, wenn Schalter 8 oder Schalter 9 auf 1 stehen, schreiben Sie in zwei Felder xxxxxxx1xX und xxxxxxx1X. Gleich erfahren Sie dazu mehr.

The screenshot shows the simulator interface with the programming field filled with codes. The top row of light orange boxes is labeled "L0" and "L1". The "Programmierfeld" grid has the following codes in the first two rows: "11xxxxxxxX" and "xxxxxxx1X" in the first column, and "xxxxxxx1xX" in the second column. The rest of the grid is empty. The buttons and status indicators are the same as in the previous screenshot.

0	0	0	0	0	0	0	0	0	0	N
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	Taster
Schalter 1	Schalter 2							Schalter 8	Schalter 9	

Excel-Simulation des Spielcomputers Logikus Simulator von Werner Lauff, www.lauff.org, lauff@lauff.org. Kostenloser Download mit ausführlichem Handbuch: lauff.org/logikus.html.

Die ersten zehn Stellen dieses Ausdrucks repräsentieren die zehn Schalter S0 bis S9.

- 1 bedeutet: Der Schalter muss auf 1 stehen (Logikus: Die oberen Federn sind verbunden).
- 0 bedeutet: Der Schalter muss auf 0 stehen (Logikus: Die unteren Federn sind verbunden).
- x bedeutet: Es ist egal, wie der Schalter steht (Logikus: Der Schalter ist gar nicht verkabelt).

Eine Programmieranweisung besteht allerdings aus elf Zeichen. Das elfte Zeichen steht für den Taster. Für den Taster gelten besondere Zeichen:

- J bedeutet: Der Taster muss gedrückt sein (Ja).
- N bedeutet: Der Taster darf nicht gedrückt sein (Nein).
- X bedeutet: Es ist egal, wie der Taster steht.

Sie können hier übrigens auch Kleinbuchstaben verwenden.

Findige Logikus-Experten werden erkennen, dass wir eine Funktion hinzugemogelt haben. Beim realen Logikus kann man den Taster einsetzen (J) oder bei der Verkabelung aussparen (X); die Variante N (der Taster darf nicht gedrückt sein) ist nicht vorgesehen, weil es an entsprechenden Kontakten fehlt. Da das schade ist, haben wir die Funktion hier hinzugefügt. Wenn Sie damit nicht einverstanden sind, verzichten Sie einfach auf den Einsatz von N. Nicht verändert ist, dass der Taster für jedes Lämpchen individuell einprogrammiert werden kann, also nicht notwendig für alle einheitlich. Das ist auch beim Logikus der Fall: Ob der Taster für ein spezifisches Lämpchen eine Rolle spielt, hängt davon ab, woher das "erste" Kabel kommt, das zu diesem Lämpchen führt, von T oder von Q.

Manche von Ihnen mögen sich auch fragen, ob nicht fünf Programmierfelder pro Lämpchen gereicht hätten, denn es gibt ja auch nur fünf Gruppen AB, CD, EF, GH und IK. Ja, meist, aber nicht immer. Ein Beispiel dafür ist die Zählaufgabe im Logikus-Arbeitsbuch Band 1, Seite 65 und 66. Die Anforderung, von fünf möglichen Schaltern die "aktiven" (auf 1 gestellten) zusammenzuzählen, ist aufgrund der Varianz bei der Schalterstellung nur mit zehn Wenn-Dann-Bedingungen zu bewältigen.

Ein Schalter	Zwei Schalter	Drei Schalter	Vier Schalter	Fünf Schalter
00001	00011	00111	01111	11111
00010	00101	01011	10111	
00100	00110	01101	11011	
01000	01001	01110	11101	
10000	01010	10011	11110	
	01100	10101		
	10001	10110		
	10010	11001		
	10100	11010		
	11000	11100		

Verkürzte Syntax

Zur Wiederholung:

111xxxxxxN = Die Schalter S1, S2 und S3 müssen auf 1 stehen und der Taster darf nicht gedrückt sein. Wie die Schalter S4, S5, S6, S7, S8, S9 und S10 stehen, ist egal.

xxxxx11111J = Die Schalter S6 bis S10 müssen auf 1 stehen; zusätzlich muss der Taster gedrückt sein. Wie die Schalter S1, S2, S3, S4 und S5 stehen, ist egal.

1010xxxxxX = Die Schalter 1 und 3 müssen auf 1 stehen, die Schalter 2 und 4 dürfen nicht auf 1 stehen (müssen 0 sein), wie der Taster steht, ist egal. Ebenso egal ist, wie die Schalter 5 bis 10 stehen.

Neben dieser vollständigen Syntax gibt es noch eine verkürzte Syntax, die hilfreich ist, um die Logik bei kleineren Schaltungen auf den ersten Blick zu erkennen: Wenn nach 1 und 0 nur noch x (Schalter) oder X (Taster) folgen, können x und X dort weggelassen werden. Anders ausgedrückt: Wenn es nicht auf den Taster ankommt und wenn es rechts von den relevanten Schaltern nur noch irrelevante Schalter gibt, dann kann man die Symbole dafür weglassen. Also:

111xxxxxxN bleibt 111xxxxxxN, weil es auf den Taster ankommt.

xxxxx11111J bleibt xxxxx11111J, weil es auf den Taster ankommt und rechts sowieso kein x mehr ist.

1010xxxxxX kann aber zu 1010 werden, weil nur noch x folgt und es auf den Taster nicht ankommt.

Hier ist das "Flug ist ausgebucht" Beispiel aus dem Logikus-Arbeitsbuch Band 1, Seite 29. Ein Flug nach New York hat 100 Plätze. Je 30 können in München, Stuttgart, Frankfurt und Hamburg vergeben werden. Der Flug soll als ausgebucht gelten, wenn drei der vier Flughäfen ihr Kontingent verkauft haben.

München	Stuttgart	Frankfurt	Hamburg	ausgebucht
1	x1	xx1	xxx1	x111
				1x11
				11x1
				111x
0	0	0	0	0
S0	S1	S2	S3	S4
München	Stuttgart	Frankfurt	Hamburg	

In diesem Beispiel sind notwendige "x" enthalten und überflüssige weggelassen; auch spielt der Taster keine Rolle. Schauen Sie in die Spalte "ausgebucht":

Zeile 1: Wenn Stuttgart und Frankfurt und Hamburg
 Zeile 2: oder wenn München, Frankfurt und Hamburg
 Zeile 3: oder wenn München, Stuttgart und Hamburg
 Zeile 4: oder wenn München, Stuttgart und Frankfurt

ihre Kontingente verkauft haben, leuchtet "ausgebucht" auf. Die einzelnen Meldungen ("30 Tickets verkauft") werden durch L0 bis L3 widerspiegelt; das könnte man auch weglassen.

Noch einmal zu den "x": Führende x sind zwingend erforderlich, nachrangige x (unbenutzte Schalter, unbenutzter Taster) können weggelassen werden.

Übrigens: Die rosa Felder sind Beschriftungsfelder. Sie entsprechen den Pergamentfolien beim Logikus und können von Ihnen beschriftet werden. Bitte ändern Sie nichts in Reihe 4 unter der rosafarbenen Reihe ; dann würde der Simulator nicht mehr funktionieren. Ebenso sind alle Zeilen unterhalb des grauen Rahmens tabu. Sollten Sie versehentlich etwas verändert haben, laden Sie sich einfach eine neue Datei von der Website herunter.

1	x1	xx1	xxx1	xxxx1	xxxxx1	xxx1	xx1	x1	1
0	0	0	0	0					
S0	S1	S2	S3	S4					

Führende x sind zwingend erforderlich, nachrangige x (unbenutzte Schalter, unbenutzter Taster) können weggelassen werden. In diesem Beispiel ist ein "Lauflicht" programmiert: L0 und L9 leuchten, wenn S0 auf 1 steht. L1 und L8 leuchten, wenn S1 auf 1 steht. L2 und L7 leuchten, wenn S2 auf 1 steht. L3 und L6 leuchten, wenn S3 auf 1 steht. L4 und L5 leuchten, wenn S4 auf 1 steht. Die Lichter laufen also aufeinander zu.

Nochmal zu den Flügen: Im vorigen Beispiel konnte ein Flug "ausgebucht" sein, der in Wirklichkeit sogar "überbucht" war. In diesem Beispiel wird der überbuchte Flug gesondert angezeigt:

München	Stuttgart	Frankfurt	Hamburg	ausgebucht	überbucht
1	x1	xx1	xxx1	x111	11111
				1x11	
				11x1	
				111x	
0	0	0	0	0	0
S0	S1	S2	S3	S4	S5
München	Stuttgart	Frankfurt	Hamburg		

Jetzt erkennen Sie vielleicht bereits die Vorteile der Simulation gegenüber der Arbeit mit dem realen Spielcomputer Logikus:

- Komplexe Verkabelungen sind mit wenigen Zeilen Code zu realisieren.
- Die Codierung erfolgt viel schneller als per Draht.
- Mit etwas Übung erkennen Sie die Logik auf den ersten Blick.
- Die Schaltungen sind nicht mehr flüchtig; Sie können sie speichern.

Schäfer, Wolf, Ziege, Kohlkopf

Programmierern wir eine der bekanntesten Aufgaben für den Logikus, das Spiel mit dem Schäfer, dem Wolf, der Ziege und dem Kohlkopf. Alle vier sind am linken Ufer eines Flusses. Ihr Ziel ist das rechte Ufer. Aber das Boot ist eng. Der Schäfer kann es allein oder nur zusammen mit dem Wolf oder der Ziege oder dem Kohlkopf nutzen. Die Tücke besteht darin, dass ohne Aufsicht ein Desaster droht: Die Ziege frisst den Kohl und / oder der Wolf die Ziege. Also ist eine gute Strategie gefragt.

Wir steuern die vier Objekte mit dem Schaltern 0 bis 3. Die Schalterstellung "0" bedeutet: Das Objekt ist am linken Ufer. Die Schalterstellung "1" bedeutet: Das Objekt ist am rechten Ufer. Das "Übersetzen ans andere Ufer" findet jeweils durch den Wechsel von 0 auf 1 oder 1 auf 0 statt. Die Lämpchen L4 und L5 zeigen an, an welchem Ufer Gefahr droht.

Schäfer	Wolf	Ziege	Kohlkopf	GEFAHR LINKS	GEFAHR RECHTS
				100	011
				1x00	0x11
0	0	0	0	0	0
S0	S1	S2	S3	S4	S5
Schäfer	Wolf	Ziege	Kohlkopf		

Wenn man zunächst darauf verzichtet, auch den Standort der vier Objekte anzuzeigen (das sieht man ja eigentlich bereits an der Schalterstellung), erfordert das Ganze nur vier Zeilen Programmcode.

Gefahr links droht bei 100 und 1x00. Gefahr rechts droht bei 011 und 0x11. Im Klartext:

100 Der Schäfer ist am rechten Ufer, der Wolf und die Ziege sind links. Gefahr links.

1x00 Der Schäfer ist am rechten Ufer, die Ziege und der Kohlkopf aber links. Gefahr links.

011 Der Schäfer ist am linken Ufer, rechts sind Wolf und Ziege. Gefahr rechts.

0x11 Der Schäfer ist am linken Ufer, rechts sind Ziege und Kohlkopf. Gefahr rechts.

Wenn Sie ganz genau hinschauen, erkennen Sie, dass 100 das Gegenstück zu 011 und 1x00 das Gegenstück zu 0x11 ist. Invertieren Sie jede 1 und jede 0, dann wird es klar.

Schäfer	Wolf	Ziege	Kohlkopf	LINKS!	RECHTS!	Schäfer	Wolf	Ziege	Kohlkopf
0	x0	xx0	xxx0	100	011	1	x1	xx1	xxx1
				1x00	0x11				
0	0	0	0	0					
S0	S1	S2	S3	S4					
Schäfer	Wolf	Ziege	Kohlkopf						

Hier ist eine erweiterte Form. Zusätzlich zur Gefahrenermittlung wird angezeigt, wo sich Schäfer, Wolf, Ziege und Kohlkopf gerade befinden.

Sie können bei diesem Spiel auch gut den Taster einsetzen. So kann man erst einmal die Objekte transferieren und dann durch Tastendruck ermitteln, ob am linken Ufer oder am rechten Ufer jeweils eine Gefahr entsteht. Das erfordert dann die komplette Programmierung; Sie sollten sie ohnehin nicht aus dem Auge verlieren:

LINKS!	RECHTS!
100xxxxxxxxJ	011xxxxxxxxJ
1x00xxxxxxxxJ	0x11xxxxxxxxJ

Programm "Binäre Zahlen"

Jetzt programmieren wir eine Schaltung, die Dezimalzahlen in Binärzahlen umwandelt. Schalten Sie nacheinander S0, S1, S2 und so weiter von 0 auf 1, werden die Binärzahlen von 1 bis 10, also von 0001 bis 1010 angezeigt.

	8	4	2	1
	111111100X	1111000000X	1100000000X	1000000000X
	111111110X	1111100000X	1110000000X	1110000000X
	111111111X	1111110000X	1111110000X	1111100000X
		1111111000X	1111111000X	1111111000X
			1111111111X	1111111110X
0	0	0	0	0
S5	S6	S7	S8	S9

Zur Kontrolle: 1 = 0001, 2 = 0010, 3 = 0011, 4 = 0100, 5 = 0101, 6 = 0110, 7 = 0111, 8 = 1000, 9 = 1001, 10 = 1010.

Programm "Binäre Zahlen II"

1	2	3	4	5	6	7	8	9	10
0000000001J	0000000010J	0000000011J	0000000100J	0000000101J	0000000110J	0000000111J	0000001000J	0000001001J	0000001010J
						0	0	0	0
						S6	S7	S8	S9

Das ist genau die umgekehrte Übung: Geben Sie mit den Schaltern S6 bis S9 (also den vier rechten Schaltern) binär die Zahlen 1, 2, 3, 4, 5, 6, 7, 8, 9 und 10 ein. Also: 0001 = 1, 0010 = 2 und so weiter. Haben Sie die Zahl gebildet, drücken Sie zur Kontrolle den Taster. Da man die Eingaben schwer lesen kann, sind sie hier nochmal zweireihig für L0 bis L5 (oben) und L6 bis L9 (unten):

0000000001J	0000000010J	0000000011J	0000000100J	0000000101J
0000000110J	0000000111J	0000001000J	0000001001J	0000001010J

An dieser Stelle eine Zwischenbemerkung: Die Beispiele sollen Ihnen verdeutlichen, wie Sie den Simulator einsetzen. Das zugrunde liegende Thema, zum Beispiel das binäre Zahlensystem, können und wollen wir bei dieser Gelegenheit natürlich nicht erläutern. Glücklicherweise gibt es die beiden "methodischen Arbeitsbücher" zum Logikus eingescannt unter

- <https://www.logikus.info/downloads/handbuch-1.zip> und
- <https://www.logikus.info/downloads/handbuch-2.zip>.

Dort finden Sie ausführliche und didaktisch gut aufbereitete Erläuterungen zu vielen Schaltungen.

Der Zehnercode

Sie sollen eine zehnstellige Kombination von Einsen und Nullen erraten. L10 leuchtet, wenn alle Stellen korrekt gewählt wurden. Denken Sie zunächst kurz darüber nach, wie viele Kombinationen Sie ausprobieren und notieren müssten, um den richtigen Code zu ermitteln. Bei einem Schalter wären es bis zu zwei Versuche. Bei zwei Schaltern bis zu vier. Bei drei Schaltern bis zu neun. Bei vier Schaltern bis zu 16. Erkennen Sie die Systematik? Sie quadrieren die Zahl der Schalter, um auf die Zahl der Versuche zu kommen. Bei 10 Schaltern brauchen Sie also bis zu 100 Versuche: $10^2=100$.

Könnten Sie drei Ziffern einstellen (0,1,2), wären es $10^4=1000$. Bei zehn Ziffern (0 bis 9) sind es 10^{10} gleich 10 Milliarden Kombinationen!

1	0	1	1	1	0	1	1	0	1	N
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	Taster

Excel-Simulation des Spielcomputers Logikus Simulator von Werner Lauff, www.lauff.org, lauff@lauff.org. Kostenloser Download mit ausführlichem Handbuch: lauff.org/logikus.html

Zur besseren Lesbarkeit: Die Werte sind 101, 101110 und 1011101101. Sie können aber auch beliebige eigene Kombinationen bilden.

Um die Sache zu erleichtern, bekommen Sie nach den ersten drei und nach den ersten sechs Stellen ein Signal, falls Sie bis dahin richtig lagen. Das heißt: Sie brauchen maximal neun Versuche bis zur ersten Etappe, dann nochmal bis zu neun Versuche bis zur zweiten Etappe und zum Schluss bis zu 16 Versuche, um Ihr endgültiges Ziel zu erreichen. Erraten Sie also zunächst die erste Dreier-Kombination mit den drei linken Schaltern (L2 leuchtet bei Erfolg), danach die zweite Dreier-Kombination mit den Schaltern 3 bis 5 (L5 leuchtet bei Erfolg) und schließlich mit den weiteren vier Schaltern den Rest der Kombi. Sie können selbstverständlich Ihren eigenen Code programmieren und auch die Schritte verändern, zum Beispiel auf 4-4-2.

Bei diesem Spiel gibt es einen kleinen Nachteil: Sie und Ihre Spielpartner sehen im Programmierfeld die richtige Lösung. Beim echten Logikus wird das durch die komplizierte Verkabelung praktisch unmöglich. Außerdem könnten Sie das Programmierfeld einfach abdecken. In der Simulation haben wir die Button-Kombination "Hide" und "Show" hinzugefügt. Mit "Hide" verstecken Sie die Programmierung, mit "Show" holen Sie sie wieder zurück. Da wir gerade bei den Buttons sind: Mit "Reset S" setzen Sie alle Schalter auf 0. Mit "Reset All" löschen Sie zusätzlich die Programmierung und der Simulator steht wieder auf seiner Ausgangsposition.

Die Wettervorhersage

KOSMOS meint im Arbeitsbuch (Band 1, Seite 34), dass man aus der Entwicklung von Luftdruck, Luftfeuchte und Temperatur das Wetter vorhersagen kann. Allerdings sind die Werte im Winter und im Sommer anders zu interpretieren.

wird besser	wird schlechter			
1101	1010			
0100	0011			
0	0	0	0	
S0	S1	S2	S3	
Som = 1, Win = 0	Luftdruck	Luftfeuchte	Temperatur	

Je nachdem, ob es Sommer ($S0=1$) oder Winter ($S0=0$) und der Luftdruck steigt ($S1=1$) oder fällt ($S1=0$), die Luftfeuchtigkeit zu- ($S2=1$) oder abnimmt ($S2=0$) und die Temperatur steigt ($S3=1$) oder fällt ($S3=0$) sagt der Spielcomputer das Wetter voraus (Logikus, Band 1, Seite 34).

Farben mischen

orange	grün	violett	braun	
110	101	011	111	
0	0	0		
S0	S1	S2		
gelb	rot	blau		

Diese Schaltung können Sie nun bereits völlig ohne Erläuterung verstehen (Logikus, Band 1, Seite 42).

Eine Übersetzungsmaschine

Hier sehen Sie eine interessante Programmierung, bei der es sich der Programmierer auf den ersten Blick schwerer gemacht hat als nötig - braucht man wirklich den gesamten Satzbau, kann man nicht Wort für Wort übersetzen? Aber die "Übersetzungsmaschine" spiegelt die reale Arbeit eines Dolmetschers gut wieder, denn Dolmetscher und Übersetzer können einen Satz eigentlich erst dann in der Zielsprache wiedergeben, wenn er komplett ausgesprochen (gleich eingegeben) ist.

DER	DAS	JUNGE	MÄDCHEN	HAT	VIELE	RINGE	LÄUTET	DIE	GLOCKE
1101x1		1101x1		1101x1	110111	1101x1			
	1011x1		1011x1	1011x1	101111	1011x1			
11000111		11000111					11000111	11000111	11000111
	10100111		10100111				10100111	10100111	10100111
0	0	0	0	0	0	0	0	0	0
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
THE	BOY	GIRL	HAS	MANY	RINGS	THE	BELL		

Logikus, Band 1, Seite 46.

Ein Taschenrechner

2	3	plus	mal	3		2	5	6	9
1	x1	xx10	xx01	xxxx1		11101	10101	11011	11011
								01101	01011
								10011	
								11101	
0	0	1	1	0	0	0	0	0	0
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
2	3	plus	mal	3					

Logikus, Band 1, Seite 59.

Abgewandeltes NIM-Spiel

Im Logikus-Arbeitsbuch Band 1 gibt es auf Seite 68 ein sehr vereinfachtes Nim-Spiel. Die Erläuterung lautet etwa so:

Abwechselnd dürfen Sie und die Maschine die Schaltschieber, die zu Anfang alle auf 0 stehen, nacheinander auf 1 schieben - also bei S0 anfangend bis zu S9. Wer von Ihnen beiden an der Reihe ist, wenn S9 von 0 auf 1 geschoben wird, der hat gewonnen. Sowohl Sie wie die Maschine dürfen einen, zwei oder auch drei Schalter nacheinander verschieben. Da die Maschine selbst ja nicht schieben kann, lässt sie bei Tasterdruck die Zahl 1, 2 oder 3 aufleuchten und bittet Sie dadurch, einen, zwei oder drei Schalter für sie zu stellen. Es ist egal, wer anfängt.

Wichtig ist, dass Sie den Taster drücken, um die vom Logikus erbetene Zahl abzulesen, ihn dann aber sofort loslassen, also zurückstellen. Wenn "die Maschine" anfangen soll, drücken Sie den Taster gleich zu Beginn.

Dies ist die Verkabelung:

1	2	3
1111100000J	1111000000J	1110000000J
1000000000J	1111111100J	1111111000J
1100000000J	0000000000J	
1111111100J		

Schon aus dem Code können Sie die Lösung erkennen. Es ist kaum glaubhaft, dass, wie der SPIEGEL 1968 schrieb, der damalige Informatikprofessor Karl Steinbuch erst beim zehnten Versuch eine Gewinnstrategie gefunden hat. Stellen Sie sich einfach ein Feld aus zehn Elementen (0 bis 9) vor, wie unten abgebildet. Sie dürfen ein, zwei oder drei Schritte machen. Noch stehen Sie links außerhalb des Spielfelds. Ein Schritt führt Sie also auf Feld 0, drei Schritte führen Sie auf Feld 2.

Damit Ihr Gegner Feld 9 nicht erreicht, müssen Sie auf Feld 5 abzielen. Denn dann kommt Ihr Spielpartner bei seinem Zug allenfalls auf die Felder 6, 7 oder 8, aber nicht auf 9.

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Damit Sie auf Feld 5 kommen, darf - wenn Sie anfangen - Ihr erster Zug nicht 1 sein (Sie springen damit auf Feld 0), denn dann antwortet Ihr Gegner ebenfalls mit 1, ist nun auf Feld 1 und Sie kommen maximal auf Feld 4. Ihr erster Zug kann aber auch nicht 3 sein, womit Sie auf Feld 2 kommen, denn dann springt der Gegner direkt auf das begehrte Feld 5. Es bleibt nur übrig, dass Ihr erster Zug 2 ist (Sie sind dann auf Feld 1). Ihr Gegner kann nun auf die Felder 2, 3 oder 4 kommen; das alles nutzt ihm aber nichts, denn Sie kommen im nächsten Schritt auf Feld 5. Also: Gewinner ist, wer anfängt und auf Feld 2 kommt.

Im Originalbuch gibt es noch eine Lämpchen-Anzeige "Sie haben verloren". Aber: Fangen Sie an, "weiß" der Logikus schon nach dem ersten Zug, ob Sie gewinnen oder nicht. Fängt der Logikus an, "weiß" er sofort, dass er gewinnt, denn er gewinnt immer, wenn er beginnt (was der SPIEGEL übrigens zutreffend berichtete). Also leuchtet die Lampe bereits nach dem ersten Zug auf und entzaubert das Ganze ein wenig; wir haben auf diese Programmierung daher verzichtet.

Exkurs: Wie funktioniert der Simulator?

Der Simulator funktioniert allein durch Excel-eigene Formeln; es gibt keinen VBA-Code. Die Berechnung der Formeln erfolgt in den Zeilen 66 ff. Für jedes Lämpchen L0 bis L9 gibt es einen eigenen Programmierblock. Wir konzentrieren uns im Folgenden auf L0.

Zeile 67 (grün) repliziert den Status der Schalter S1 bis S10. Das ist hilfreich, weil die Schalterstellung beim Scrollen des Arbeitsblatts ja nicht mehr sichtbar ist.

In den Zeilen 68 bis 77 werden die zehn 11 Stellen langen Programmierzeilen "auseinandergenommen". Aus 1010101011J wird dadurch 1-0-1-0-1-0-1-0-1-1-J. Aus 111xxx1111N wird 1-1-1-x-x-x-1-1-1-1-N. Aus 101 wird 1-0-1- - - - - - - - .

In den Zeilen 78 bis 87 wird festgestellt und aufsummiert, inwieweit die Soll-Schalterstellung (Zeile 17) mit der tatsächlichen Schalterstellung (Zeilen 68 bis 77) übereinstimmt.

Gibt es genau elf Übereinstimmungen (alle 10 Schalter stehen so wie gewünscht, außerdem steht der Taster richtig), wird die im Feld rechts außen die Summe 11 ausgegeben.

Zeile 4 fragt ab, ob bei den zehn Bedingungen mindestens einmal die Summe 11 erreicht ist. Dann springt der Wert von 0 auf 1. Ist der Wert 1, wird das Feld orange hinterlegt - die Lampe "leuchtet".

Maier, Müller, Schulze

Das "Rätsel um Herrn Müller und Herrn Maier" in Band 1, Seite 48, zeigt, dass bereits die Programmierung ein Problem lösen kann; anders ausgedrückt: Der Weg ist das Ziel. Es gibt die drei Herren Maier, Müller und Schulze. Einer von ihnen ist Bürgermeister, einer Arzt, einer Apotheker. Gleich werden wir von einigen Inkompatibilitäten hören, aber zunächst können wir den Logikus so programmieren, dass sämtliche Berufe bei sämtlichen Namen aufleuchten. Danach müssen wir nur noch die Bedingungen löschen, die nicht passen.

Bürgermeister	Arzt	Apotheker		
100	100	100		
010	010	010		
001	001	001		
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Nun wissen wir nach den Angaben in Band 1, Seite 49: Müller ist nicht der Arzt.

Bürgermeister	Arzt	Apotheker		
100	100	100		
010		010		
001	001	001		
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Maier ist nicht der Apotheker.

Bürgermeister	Arzt	Apotheker		
100	100			
010		010		
001	001	001		
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Und Maier ist auch nicht der Arzt.

Bürgermeister	Arzt	Apotheker		
100				
010		010		
001	001	001		
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Schalten wir nun der Reihe nach S0 bis S2 ein (und wieder aus!), verrät uns der Logikus: Maier ist der Bürgermeister. Das ist eindeutig. Da das Amt des Bürgermeisters nun vergeben ist, löschen wir die restliche Programmierung zum Thema Bürgermeister.

Bürgermeister	Arzt	Apotheker		
100				
		010		
	001	001		
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Eine erneute Probe zeigt: Müller ist der Apotheker. Da das Amt des Apothekers nun ebenfalls vergeben ist, löschen wir die restliche Programmierung zum Thema Apotheker.

Bürgermeister	Arzt	Apotheker		
100				
		010		
	001			
0	0	0		
S0	S1	S2		
Maier	Müller	Schulze		

Damit ist nun auch durch Schalterschieben nachweisbar: Schulze ist der Arzt.

Addition zweistelliger Dualzahlen

Die Addition zweistelliger Dualzahlen (Band 1, Seite 77 ff., Schaltbild Seite 82) bekommen wir mit dem Logikus-Simulator noch mühelos hin. Wir haben dazu den kleinen Trick angewendet, die Schalter S0 und S1 für die linke Zahl (vor dem Plus-Zeichen) und die Schalter S2 und S3 für die rechte Zahl (nach dem Plus-Zeichen) einzusetzen. So sparen wir uns jede Menge x in der Schaltung.

0/1	0/1	0/1	0/1
	1010	0101	0001
	1011	0110	0100
	0111	1000	0110
	1101	1001	1001
	1110	1100	1011
	1111	1111	1100
			1110
1	0	1	0
S0	S1	S2	S3
links	links	rechts	rechts

$$10 + 10 = 0100$$

Das Logikus-Handbuch enthält auf Seite 83 auch eine Schaltung zur Addition zweier fünfstelliger Dualzahlen, also beispielsweise 11111 plus 11111 gleich 111111.

Akkorde

Der zweite Band des Logikus Handbuchs enthält eine Schaltung, mit der man Akkorde bilden und identifizieren kann. Die Schaltung ist natürlich einfach im Simulator nachprogrammierbar.

B Dur	g moll	F Dur	d Moll	C Dur	a Moll	G Dur	e Moll	D Dur	h Moll
0101000100J	0100010100J	1001001000J	0101001000J	1010010000J	1010001000J	0100010010J	0010010010J	0100101000J	0100100010J
0	1	0	0	0	1	0	1	0	0
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
c	d	e	f	fis	g	a	b	h	

Richtig interessant wäre aber, alle gebräuchlichen Dur- und Moll-Akkorde samt ihrer Variationen (C7, Cm7, Cm6, C7-5 usw.) darstellen zu können. Das ist mit dem Logikus nicht möglich. Dafür haben wir eine spezielle Datei erstellt: den "Akkordikus". Sie finden die xlsx-Datei und die Anleitung dazu ebenfalls über die Seite www.lauff.org/excel.html. Die Programmierung ist ganz ähnlich wie beim

Logikus und folgt ebenfalls dem Zuordnungsprinzip, frei nach dem Motto: "Was nicht als Daten eingegeben ist, kann auch nicht herauskommen."

Zum Schluss

Danke fürs Lesen und viel Spaß mit dem Akkordikus! Ergänzungen sind willkommen. Und schauen Sie auf der Seite www.lauff.org/excel.html ab und zu nach Aktualisierungen.

Sollten Sie Interesse an einer raffinierten Excel-Anwendung für Ihre Firma haben, schreiben Sie bitte eine E-Mail an lauff@lauff.org.