

FOLIEN 3 x 4 MATRIX TASTATUR DECODER

Realisierung mit einem GAL 16 V 8

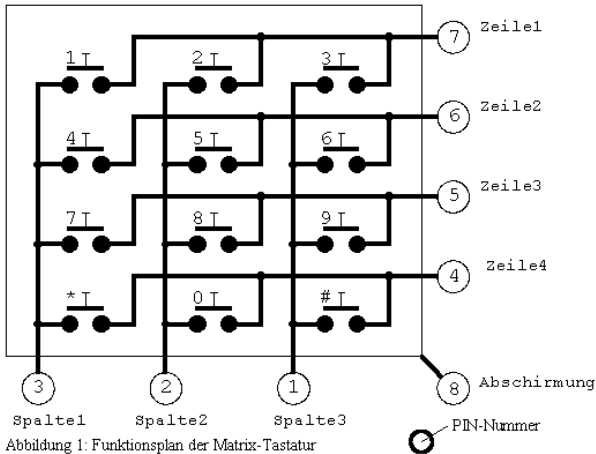
Robert SCHWAGER

Dieses Beispiel soll eine Anwendung des im Heft 44 veröffentlichte Artikel über GALs sein.

1. Die Matrixtastatur

Die Matrixtastatur ist die eleganteste Lösung von Realisierungen von großen Tastaturfeldern. Die Standardmatrixtastaturen haben $X \cdot Y$ -Anzahl von Tastaten und dafür nur $X+Y$ -Anzahl von Anschlüssen. Die Anschlüsse sind einerseits Spalten und andererseits die Zeilen. Auf einem Schnittpunkt mit einer Reihe und einer Zeile wird eine Taste gesetzt, die auf einem Tasterkontakt mit der Zeile und auf dem anderen Kontakt mit einer Spalte verbunden. Wird nun eine Folientaste an diesem Schnittpunkt gesetzt, kann das ganze Tastaturfeld mit 2 Folien und einer Maske realisiert werden.

Schematischer Aufbau einer handelsüblichen Folienmatrixtastatur
zB.: C&K 4A-01-T322NCFQ



Die Wahrheitstabelle, die sich aus dem Betätigen der Tasten ergibt, ist auf der nächsten Seite zu ersehen. Eine Verbindung zwischen zwei Pins wird durch ein X markiert.

In dieser Tabelle wird vorausgesetzt, daß nur jeweils eine Taste gedrückt wird.

		Anschlußpin						
		1	2	3	4	5	6	7
Taster	1			X				X
	2		X					X
	3	X						X
	4			X			X	
	5		X				X	
	6	X					X	
	7			X		X		
	8		X			X		
	9	X				X		
	0		X		X			
	*			X	X			
#	X			X				

Tabelle 1: Wahrheitstabelle der Matrix-Folientastatur

Aus diesen Formalismus kann man nun den Dekoder realisieren, welcher wie unter Punkt 2 aufgebaut werden muß. Zu beachten ist, daß das Steuergerät, hier am besten aus dem Anwendungsgebiet von ASIC insbesondere GALs, alle Anforderungen der Tabelle erfüllen muß. Es muß daher sichergetstellt werden, daß nur jeweils eine Taste gescannt wird.

2. Darstellung logischer Gleichungen

- / Negierung einer Größe (Variable)
- * logisch UND
- + logisch (inklusive) ODER
- = Gleichsetzung der logischen Gleichung
- := Gleichsetzung der logischen Gleichung, das Ergebnis wird einem D-Flipflop (D) zugeführt.

Darstellungsformalismus für OPAL-Junior von National-Semiconductor.

3. Der Dekoder

Der Dekoder hat nun die Aufgabe, als erstes jede Spalte abzutasten und zu schauen an welcher Zeile ein Signal "durchkommt", dieses wird nach einander mit jeder weiteren Zeile getan.

Die zweite Ausgabe ist nun von den Zeilen- und Reihenwerten ein geeignetes Ausgangssignal auszuwerten. Dieses ist zunächst ein 4 stelliges binäres Signal und ein Signal, welches angibt, daß eine Taste gedrückt wurde.

Die Aufstellung der beinhaltet alle richtige Fällen, wenn aber nun zwei oder mehrere Tasten gedrückt werden, so ist an EN, D0 bis D4 eine Null.

	S3	S2	S1	Z4	Z3	Z2	Z1	EN	D3	D2	D1	D0
	1	2	3	4	5	6	7					
1	0	0	1	0	0	0	1	1	0	0	0	1
2	0	1	0	0	0	0	1	1	0	0	1	0
3	1	0	0	0	0	0	1	1	0	0	1	1
4	0	0	1	0	0	1	0	1	0	1	0	0
5	0	1	0	0	0	1	0	1	0	1	0	1
6	1	0	0	0	0	1	0	1	0	1	1	0
7	0	0	1	0	1	0	0	1	0	1	1	1
8	0	1	0	0	1	0	0	1	1	0	0	0
9	1	0	0	0	1	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	1	0	0	0	0
*	0	1	0	1	0	0	0	1	1	0	1	0
#	1	0	0	1	0	0	0	1	1	0	1	1

Tabelle 2: Wahrheitstabelle mit Datenbus und ENABLE-Signal der Matrix-T.

Aus dieser Tabelle werden nun 2 Tabellen erstellt.

Die dritte Tabelle beinhaltet die Abtastung der Spalten, also den Sprung von Spalte zu Spalte. Nun wenn eine Taste gedrückt wird, so wird der Sprung nicht fortgesetzt.

Die zweite Tabelle bildet den Zusammenhang von Spalten und Zeilen zu den Datenbits des vierstelligen Datenbusses und dem Steuerbit (Enable).

Spalten-Sprung

Z1	Z2	Z3	Z4	S1n	S2n	S3n	S1n+1	S2n+1	S3n+1
0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	x	x	0	1	0
0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	0	1	1	0	0
1	0	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0
0	0	1	0	1	0	0	1	0	0
0	0	0	1	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	0
1	0	0	0	0	0	1	0	0	1
0	1	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0	0	1

Tabelle 3: Wahrheitstabelle der Spalten-Sprungfunktion

Die Fußnote n gibt den momentanen Zustand der "Zähl"-Variable an, hingegen n+1 gibt die beim nächsten positiven Taktanstieg ausgegebenen Wert.

Nun kann man zu dieser Tabelle die logische Gleichung aufstellen und nach dem Sharp-Verfahren die logischen Gleichungen vereinfachen.

$$S1 = (/Z1 * /Z2 * /Z3 * /Z4 * /S1 * /S2) + (Z1 * /Z2 * /Z3 * /Z4 * S1 * /S2 * /S3) + (/Z1 * Z2 * /Z3 * /Z4 * S1 * /S2 * /S3) + (/Z1 * /Z2 * Z3 * /Z4 * S1 * /S2 * /S3) + (/Z1 * /Z2 * /Z3 * Z4 * S1 * /S2 * /S3)$$

$$S2 = (/Z1 * /Z2 * /Z3 * /Z4 * S1) + (Z1 * /Z2 * /Z3 * /Z4 * /S1 * S2 * /S3) + (/Z1 * Z2 * /Z3 * /Z4 * /S1 * S2 * /S3) + (/Z1 * /Z2 * Z3 * /Z4 * /S1 * S2 * /S3) + (/Z1 * /Z2 * /Z3 * Z4 * /S1 * S2 * /S3)$$

$$S3 = (/Z1 * /Z2 * /Z3 * /Z4 * /S1 * S2) + (Z1 * /Z2 * /Z3 * /Z4 * /S1 * /S2 * S3) + (/Z1 * Z2 * /Z3 * /Z4 * /S1 * /S2 * S3) + (/Z1 * /Z2 * Z3 * /Z4 * /S1 * /S2 * S3) + (/Z1 * /Z2 * /Z3 * Z4 * /S1 * /S2 * S3)$$

Gleichung 1 bis 3: getaktete D-Flopplaufgänge für die Spaltenausgänge S1 bis S3

Datenbus-Dekoder:

S1	S2	S3	Z1	Z2	Z3	Z4	D3	D2	D1	D0	EN
1	0	0	1	0	0	0	0	0	0	1	1
0	1	0	1	0	0	0	0	0	1	0	1
0	0	1	1	0	0	0	0	0	1	1	1
1	0	0	0	1	0	0	0	0	1	0	1
0	1	0	0	1	0	0	0	0	1	0	1
0	0	1	0	1	0	0	0	0	1	1	0
1	0	0	0	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1	0	0	1
1	0	0	0	0	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0	0	0	1	1
1	0	0	0	0	0	1	1	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	1	1	0	1	1	1

Tabelle 4: Wahrheitstabelle der Datenbus- und Enableausgänge

Die darausgebende Gleichungen, welche hier zeitunabhängig sind, im gegensatz von Gleichung 1 bis 3 im Zuge der **Tabelle 3**, wurden wieder mit einem logischen Vereinfachungsverfahren nach Sharp durchgeführt.

$$D3 = (/S1 * S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4) + (/S1 * /S2 * S3 * /Z1 * /Z2 * Z3 * /Z4) + (S1 * /S2 * /S3 * /Z1 * /Z2 * /Z3 * Z4) + (/S1 * /S2 * S3 * /Z1 * /Z2 * /Z3 * Z4)$$

$$D2 = (R1 * /R2 * /R3 * /Z1 * Z2 * /Z3 * /Z4) + (/R1 * R2 * /R3 * /Z1 * Z2 * /Z3 * /Z4) + (/R1 * /R2 * R3 * /Z1 * Z2 * /Z3 * /Z4) + (R1 * /R2 * /R3 * /Z1 * /Z2 * Z3 * /Z4)$$

$$D1 = (/R1 * R2 * /R3 * Z1 * /Z2 * /Z3 * /Z4) + (/R1 * /R2 * R3 * Z1 * /Z2 * /Z3 * /Z4) + (/R1 * R2 * R3 * /Z1 * Z2 * /Z3 * /Z4) + (R1 * /R2 * /R3 * /Z1 * /Z2 * Z3 * /Z4) + (R1 * /R2 * /R3 * /Z1 * /Z2 * /Z3 * Z4) + (/R1 * /R2 * R3 * /Z1 * /Z2 * /Z3 * Z4)$$

$$D0 = (R1 * /R2 * /R3 * Z1 * /Z2 * /Z3 * /Z4) + (/R1 * /R2 * R3 * Z1 * /Z2 * /Z3 * /Z4) + (/R1 * R2 * /R3 * /Z1 * Z2 * /Z3 * /Z4) + (R1 * /R2 * /R3 * /Z1 * /Z2 * Z3 * /Z4) + (/R1 * /R2 * R3 * /Z1 * /Z2 * Z3 * /Z4) + (/R1 * /R2 * R3 * /Z1 * /Z2 * /Z3 * Z4)$$

$$EN = (S1 * /S2 * /S3 * Z1 * /Z2 * /Z3 * /Z4) + (/S1 * S2 * /S3 * Z1 * /Z2 * /Z3 * /Z4) + (/S1 * /S2 * S3 * Z1 * /Z2 * /Z3 * /Z4) + (S1 * /S2 * /S3 * /Z1 * /Z2 * /Z3 * /Z4) + (/S1 * S2 * /S3 * /Z1 * Z2 * /Z3 * /Z4) + (/S1 * /S2 * S3 * /Z1 * Z2 * /Z3 * /Z4) + (S1 * /S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4) + (/S1 * S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4) + (S1 * /S2 * /S3 * /Z1 * /Z2 * /Z3 * Z4) + (/S1 * S2 * /S3 * /Z1 * /Z2 * /Z3 * Z4) + (/S1 * /S2 * S3 * /Z1 * /Z2 * /Z3 * Z4)$$

Gleichung 4 bis 8: logische Gleichung von Data D0 bis D3 und EN

Die Gleichung 8 mit dem Ergebnis EN kann und muß durch das einmalige Auftreten der Reihenabgabe auf vier Terme verkürzt werden, da sonst 12 terme für einen GAL 16V8 zu viel sind.

$$EN = (Z1 * /Z2 * /Z3 * /Z4) + (/Z1 * Z2 * /Z3 * /Z4) + (/Z1 * /Z2 * Z3 * /Z4) + (/Z1 * /Z2 * /Z3 * Z4)$$



Gleichung 9: gekürztes EN-Signal

Nun hat man alle notwendigen Gleichungen zusammen und diese schreibt man in eine Datei für GAL-Assembler und läßt diese in JED-Format assemblieren.

4. Das Programm

```
TITLE 3x4_Matrix-Folientastatur-Dekoder
PATTERN Dekoder
SEVISION A
AUTHOR Robert SCHWAGER
DATE 17. Nov. 1995
```

CHIP 3x4_Key GAL16V8

; Fuer C&K-Tastatur 4A-01-...Q geeignet.

```
; TASTATURMATRIX GAL - Ausgang
; Taste PIN 1 2 3 4 5 6 7 D3 D2 D1 D0 EN
; " 1 " _____X.....X. _____.0.0.0.1.1.1.1.
; " 2 " _____X.....X. _____.0.0.1.0.0.1.1.
; " 3 " _____X.....X. _____.0.0.1.1.1.1.1.
; " 4 " _____X.....X. _____.0.1.0.0.0.1.1.
; " 5 " _____X.....X. _____.0.1.0.1.1.1.1.
; " 6 " _____X.....X. _____.0.1.1.0.0.1.1.
; " 7 " _____X.....X. _____.0.1.1.1.1.1.1.
; " 8 " _____X.....X. _____.1.0.0.0.0.1.1.
; " 9 " _____X.....X. _____.1.0.0.1.1.1.1.
; " 0 " _____X.....X. _____.0.0.0.0.0.1.1.
; " * " _____X.....X. _____.1.0.1.1.1.1.1.
; " # " _____X.....X. _____.1.0.1.1.0.0.1.1.
```

; Pin-Belegung (DIL 20)

```
PIN 1 2 3 4 5 6 7 8 9 10
CLK NC NC NC NC Z1 Z2 Z3 Z4 GND
```

```
PIN 11 12 13 14 15 16 17 18 19 20
NC S1 S2 S3 EN D3 D2 D1 D0 VCC
```

; Gleichungen
EQUATIONS

; Zaehler fuer Spaltenschalungen

```
S1 := /Z1 * /Z2 * /Z3 * /Z4 * /S1 * /S2 +
      Z1 * /Z2 * /Z3 * /Z4 * S1 * /S2 * /S3 +
      /Z1 * Z2 * /Z3 * /Z4 * S1 * /S2 * /S3 +
      /Z1 * /Z2 * Z3 * /Z4 * S1 * /S2 * /S3 +
      /Z1 * /Z2 * /Z3 * Z4 * S1 * /S2 * /S3
```

```
S2 := /Z1 * /Z2 * /Z3 * /Z4 * S1 +
      Z1 * /Z2 * /Z3 * /Z4 * /S1 * S2 * /S3 +
      /Z1 * Z2 * /Z3 * /Z4 * /S1 * S2 * /S3 +
      /Z1 * /Z2 * Z3 * /Z4 * /S1 * S2 * /S3 +
      /Z1 * /Z2 * /Z3 * Z4 * /S1 * S2 * /S3
```

```
S3 := /Z1 * /Z2 * /Z3 * /Z4 * /S1 * S2 +
      Z1 * /Z2 * /Z3 * /Z4 * /S1 * /S2 * S3 +
      /Z1 * Z2 * /Z3 * /Z4 * /S1 * /S2 * S3 +
      /Z1 * /Z2 * Z3 * /Z4 * /S1 * /S2 * S3 +
      /Z1 * /Z2 * /Z3 * Z4 * /S1 * /S2 * S3
```

; Auswertung der Ein/Ausgaenge (Zeilen zu Spalten)

```
D3 = /S1 * S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4 +
      /S1 * /S2 * S3 * /Z1 * /Z2 * Z3 * /Z4 +
      S1 * /S2 * /S3 * /Z1 * /Z2 * /Z3 * Z4 +
      /S1 * /S2 * S3 * /Z1 * /Z2 * /Z3 * Z4
```

```
D2 = S1 * /S2 * /S3 * /Z1 * Z2 * /Z3 * /Z4 +
      /S1 * S2 * /S3 * /Z1 * Z2 * /Z3 * /Z4 +
      /S1 * /S2 * S3 * /Z1 * Z2 * /Z3 * /Z4 +
      S1 * /S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4
```

```
D1 = /S1 * S2 * /S3 * Z1 * /Z2 * /Z3 * /Z4 +
      /S1 * /S2 * S3 * Z1 * /Z2 * /Z3 * /Z4 +
      /S1 * /S2 * S3 * /Z1 * Z2 * /Z3 * /Z4 +
      S1 * /S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4
```

```
S1 * /S2 * /S3 * /Z1 * /Z2 * /Z3 * Z4 +
/S1 * /S2 * S3 * /Z1 * /Z2 * /Z3 * Z4

D0 = S1 * /S2 * /S3 * Z1 * /Z2 * /Z3 * /Z4 +
      /S1 * /S2 * S3 * Z1 * /Z2 * /Z3 * /Z4 +
      /S1 * S2 * /S3 * /Z1 * Z2 * /Z3 * /Z4 +
      S1 * /S2 * /S3 * /Z1 * /Z2 * Z3 * /Z4 +
      /S1 * /S2 * S3 * /Z1 * /Z2 * Z3 * /Z4 +
      /S1 * /S2 * S3 * /Z1 * /Z2 * /Z3 * Z4

EN = Z1 * /Z2 * /Z3 * /Z4 +
      /Z1 * Z2 * /Z3 * /Z4 +
      /Z1 * /Z2 * Z3 * /Z4 +
      /Z1 * /Z2 * /Z3 * Z4
```

; Programmende

Programmlisting 1: GAL-Equations-File 3x4_key.eqn

5. Pinbelegung

Pin	Belegung
1	Takteingang (1MHz bis 14MHz)
2	nicht belegt, daher auf GND
3	nicht belegt, daher auf GND
4	nicht belegt, daher auf GND
5	nicht belegt, daher auf GND
6	Zeile 1, Pin 7
7	Zeile 2, Pin 6
8	Zeile 3, Pin 5
9	Zeile 4, Pin 4
10	GND (Ground, 0V)
11	nicht belegt, daher auf GND
12	Spalte 1, Pin 3
13	Spalte 2, Pin 2
14	Spalte 3, Pin 1
15	ENABLE-Signal (Ausgang)
16	DATA 3 (Ausgang)
17	DATA 2 (Ausgang)
18	DATA 1 (Ausgang)
19	DATA 0 (Ausgang)
20	VCC (+5V)

Abbildung 5: Pin Belegung des programmierten GALs 16V8 mit 3x4_key.jed

6. Demoboard

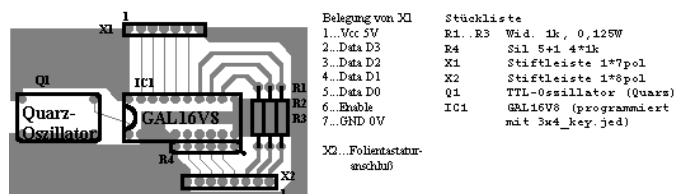


Abbildung 2: Bestückung des Demoboards

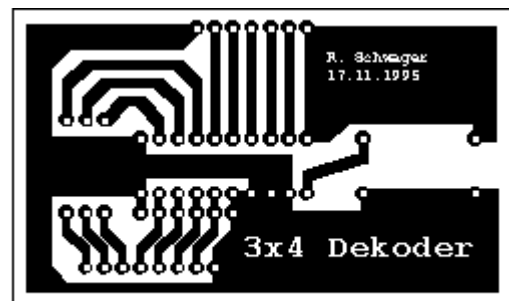


Abbildung 3: Layoutplan des Demoboards

□