Versuchsprotokoll
Wintersemester 93/94

Marcus Müller, Frank Pilhofer
Donnerstag, 03. Februar 1994
Elektronik - Praktikum / Digital - Teil

Ansteuerung eines Druckers

Übung 7

Inhaltsverzeichnis

Einleitung 2
Grundlagen 2
Datenkodes 2
Kodierung im BCD-Kode 2
Kodierung im Gray-Kode 3
ASCII-Kode 4
Die Centronics-Schnittstelle 4
Versuchsdurchführung 6
Kodeumwandlung 6
Von Gray nach Binär 6
Von Binär nach ASCII 7
Einhaltung des Zeitprotokolls 8
Die komplette Schaltung 9

Einleitung

In dieser Übung soll ein Drucker angesteuert werden. Im Gray-Kode ankommende Daten werden in den vom Drucker verwendeten ASCII-Kode gewandelt und unter Einhaltung des nötigen Protokolls über eine Centronics-Schnittstelle an den Drucker übermittelt.

Grundlagen

Datenkodes

Ein Computer oder ein ähnlich logisch arbeitendes Gerät versteht herzlich wenig von der Umgangssprache der Menschen und wird auch in absehbarer Zeit nicht sinnvoll in der Lage zu sein, Wünsche von den Lippen abzulesen. Um dennoch mit dem Gerät arbeiten zu können, muß sich der Mensch auf die Technik einlassen und lernen, ihre Sprache zu sprechen.

Um mit logischen Bauteilen zu kommunizieren, ist die binäre Sprache die sinnvollste. Alle Dateneingaben und Datenausgaben erfolgen binär und müssen andernfalls *kodiert* (oder dekodiert) werden.

Bevor wir also mit dem Gerät arbeiten können, müssen wir uns darauf einigen, *welche* Daten wir eingeben und ausgeben wollen; eventuell können auch noch Anforderungen an den Kode gestellt werden (Sicherheit gegenüber Störungen etc.). Der zweite Schritt ist es, diese Daten geeignet in das binäre Alphabet zu übersetzen und schließlich die geeigneten logischen Schaltungen aufbauen, die die gewünschten Kodierungen vornehmen.

Bei der Arbeit mit einem Drucker werden wir uns für diesen Versuch auf zwei Alphabete einigen, um den Drucker anzusprechen. Zunächst sollen nur Ziffern und drei Steuerzeichen, später das gesamte Alphabet ansprechbar sein. Damit ist der erste Schritt getan, nun müssen wir uns auf die Suche nach einer geeigneten binären Kodierung machen.

Kodierung im BCD-Kode

Der BCD-Kode (Binary Coded Decimals) ist ein einfacher Kode zur Darstellung von dezimalen Zahlen. Vom Typ her ist er ein Tetradenkode, was bedeutet, daß jede Ziffer durch eine Kombination aus vier Bits (eine Tetrade) dargestellt wird. Vier Bits erlauben insgesamt 16 Kombinationen, so daß einige Möglichkeiten von dezimalen Ziffern ungenutzt bleiben werden. Genau diese werden wir für die Darstellung der Steuerzeichen verwenden.

Im BCD-Kode entspricht jede dezimale Ziffer ihrem binären Äquivalent:

Dezimal	BCD	Dezimal	BCD	Dezimal	BCD	Dezimal	BCD
0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	LF	1010		1011
FF	1100	CR	1101		1110		1111

In die letzten Kästchen wurden bereits die später zu verwendenden Steuerzeichen eingetragen. LF steht für Line Feed (Zeilenvorschub, das Papier wird um eine Zeile nach vorne verschoben); CR für Carriage Return (Wagenrücklauf, der Druckkopf wird an den Anfang der Zeile gesetzt) und FF für Form Feed (Seitenvorschub, der Druckkopf wird auf die erste Zeile der nächsten Seite positioniert). Warum die Steuerzeichen gerade so verteilt wurden, werden wir später sehen. Es ist natürlich zu beachten, daß sie nicht zum Standardrepertoire gehören und nur laut unserer Definition des Kodes vorhanden sind.

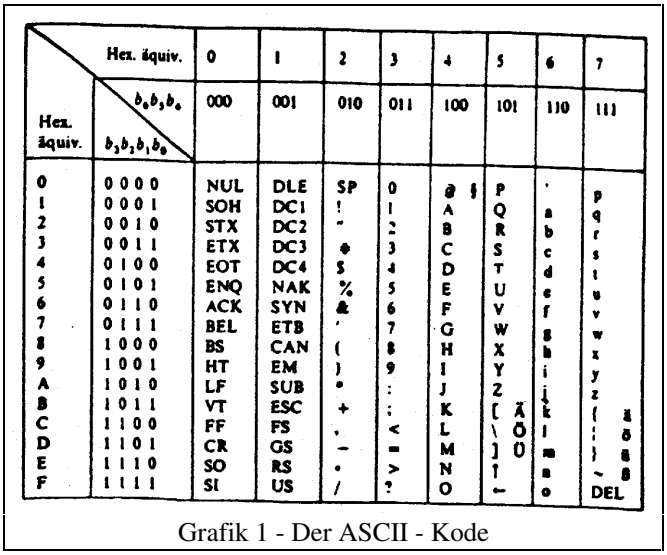
Kodierung im Gray-Kode

Auch der Gray-Kode ist ein Tetradenkode, jedoch in umgewandelter Form. Ihm liegt die Voraussetzung zugrunde, daß sich aufeinanderfolgende Ziffern in ihrer Kodierung nur in einem Bit unterscheiden. In gewissen Fällen ist so eine bessere Prüfung der Daten gewährleistet. Werden z.B. immer auf- oder absteigende Zahlen erwartet, so können unsinnige Eingaben (mehrere Bits ändern sich) besonders einfach abgefangen werden. Die Tabelle des Gray-Kodes sieht wie folgt aus:

Dezimal	Gray	Dezimal	Gray	Dezimal	Gray	Dezimal	Gray
0	0000	1	0001	2	0011	3	0010
4	0110	5	0111	6	0101	7	0100
8	1100	9	1101	LF	1111		1110
FF	1010	CR	1011		1001		1000

Auch hier definieren wir unsere Steuerzeichen auf die Positionen 10, 12 und 13.

ASCII-Kode



Hex. äquiv.	$b_6b_5b_4$ → / $b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@ §	P	·	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[Ä	k	{ ä
C	1100	FF	FS	,	<	L	\ Ö	l	| ö
D	1101	CR	GS	-	=	M	] Ü	m	} ü
E	1110	SO	RS	.	>	N	↑	n	~ ß
F	1111	SI	US	/	?	O	←	o	DEL

Grafik 1 - Der ASCII - Kode

Die beiden zuvor dargestellten Kodes hatten den entscheidenden Nachteil, nur dezimale Ziffern (und bis zu 6 Sonderzeichen) zuzulassen. Zum Schreiben wird jedoch ein Alphabet von a bis z benötigt. Wenn möglich, sollte auch noch zwischen Groß- und Kleinschreibung unterschieden werden, zuzüglich Interpunktion etc. Dies sprengt natürlich die Grenze der jeweils vier Bit der anderen Kodes. Für den ASCII-Kode werden deshalb Standardmäßig 7 Bit verwendet. Zwar ist der Kode meist als 8-Bit-Kode implementiert (der Bequemlichkeit halber, der Computer rechnet nun mal lieber mit acht Bit als mit sieben), die 128 zusätzlich darstellbaren Zeichen sind jedoch nicht international genormt; so hat jeder Hersteller oder jede Nation den zusätzlichen Platz für eigens definierte Zeichen reserviert (ä, æ, ñ, ç etc.). Die mit 7 Bit erreichbaren und meist genormten Zeichen sind in der Tabelle dargestellt.

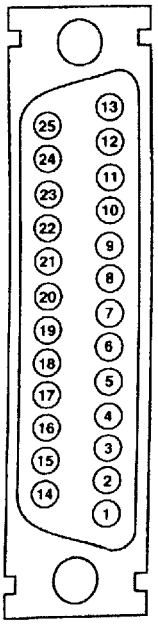
Die ersten 32 Zeichen wurden benutzt, um einige Sonderzeichen, die bei der Kommunikation mit den verschiedensten Geräten, unter anderem auch Drucker, zu definieren. So finden wir hier auch die von uns verwendeten Steuerzeichen wieder. Wir erkennen auch, warum die Anordnung der Steuerzeichen in den beiden anderen Kodes gerade so gewählt wurde: Es wurde die Norm des ASCII-Kodes übernommen.

Die Centronics-Schnittstelle

Bei allen Überlegungen, wie denn unsere Kodierung aussehen soll, darf nicht vergessen werden, daß die Daten dem Gerät auch übermittelt werden müssen, daß wir also in diesem Fall den Drucker an die Schaltung anschließen müssen. Für eine solche Grenze zwischen zwei verschiedenen Systemen (unserer Schaltung und dem vom Hersteller gefertigten Drucker) muß eine *Schnittstelle* definiert sein, eine Vereinbarung, an die sich beide Seiten halten müssen.

Für Drucker hat die wohl bekannteste Schnittstelle den Namen Centronics, benannt nach dem gleichnamigen Hersteller. Sie erlaubt eine parallele Datenübertragung zwischen den Geräten und sorgt anhand eines definierten Zeitprotokolls für einen reibungslosen und fehlerfreien Ablauf des Druckens.

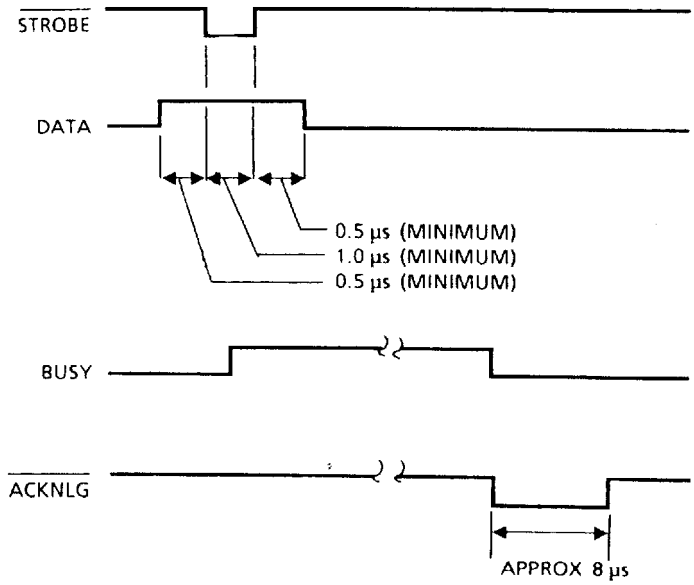
Der Centronics-Stecker hat 25 Pins (Datenleitungen) und ist in Grafik 2 dargestellt. Viele dieser Leitungen werden jedoch überhaupt nicht gebraucht und liegen deshalb auf Masse. Wichtig sind vor allem die Leitungen zur Datenübertragung (D0 bis D7, 8 Bit breit) sowie die Leitungen STROBE, ACK (Acknowledge, Akzeptiert) und BUSY (Drucker bereit). Andere Signale teilen dem Gerät den Status des Druckers mit, wie z.B. PE (Paper Empty, Papierfach leer) oder SLCT (Select, Drucker bereit). Eine weitere Leitung setzt den Drucker in seinen Einschaltzustand zurück (RESET).



Pin	Signal	Beschreibung
1	STROBE	Strobe-Signal (out, low-active)
2	D0	Datenbit 0
3	D1	Datenbit 1
4	D2	Datenbit 2
5	D3	Datenbit 3
6	D4	Datenbit 4
7	D5	Datenbit 5
8	D6	Datenbit 6
9	D7	Datenbit 7
10	ACK	Acknowledgne (in, low-active)
11	BUSY	Busy (in)
12	PE	Papierende (in)
13	SLCT	Select (in, Drucker online)
14	+5 Volt	100 mA (47Ohm pullup)
15	NC	Not connected
16	RESET	Reset-Leitung (low active)
17	GRND	Signalmasse
18	GRND	Signalmasse
19	GRND	Signalmasse
20	GRND	Signalmasse
21	GRND	Signalmasse
22	GRND	Signalmasse
23	GRND	Signalmasse
24	GRND	Signalmasse
25	GRND	Signalmasse

Grafik 2 - Anschlußbelegung der Centronics-Schnittstelle

Es ist zu beachten, daß einige Signale invertiert sind und deshalb normalerweise auf 1 liegen und mit ihrem Wechsel nach 0 eine Meldung abgeben.



Grafik 3 - Zeitprotokoll an der Centronics-Schnittstelle

Bei der Datenübertragung von der Schaltung an den Drucker ist, wie angekündigt, ein zeitliches Protokoll einzuhalten, welches in der nebenstehenden Grafik (Grafik 3)dargestellt ist. Dort sind die vier wichtigsten Leitungen (wobei DATA nur für den Sender von Interesse ist) während der Übertragung eines Datenwortes dargestellt.

Will der Sender ein Zeichen übermitteln, so öffnet er zuerst mit DATA sein eigenes Datentor und legt die Daten damit an die Datenleitungen der Centronics-Schnittstelle an. Nun läßt er eine kurze Zeitspanne (mindestens 0,5µs) verstreichen. Mit dem Setzen des STROBE-Signals auf 0 (diese Leitung ist low-active) signalisiert der Sender, daß sich die Datenleitungen stabilisiert haben und demzufolge vom Drucker abgegriffen werden können. Dieser Zustand mit "sicheren" Datenleitungen muß auch wieder eine Mindestzeit (ca. 1µs) gehalten werden.

Hat der Drucker die Daten verstanden, so antwortet er auf der Leitung BUSY, daß er nun mit der Verarbeitung des Datums beginnt. Nun kann der Sender zunächst das STROBE-Signal wieder auf 1 setzen (damit gelten die Datenleitungen als ungültig, aber das ist nicht mehr schlimm, da sie vom Drucker gespeichert wurden), das Datentor schließen und mit der Vorbereitung des nächsten Zeichens beginnen.

Solange der Drucker mit der Verarbeitung beschäftigt ist, bleibt die BUSY-Leigung auf 1. Erst wenn der Drucker alles erledigt hat, wird die Leitung geschickt. Gleichzeitig wird auch die (low-aktive) Acknowledge-Leitung auf Low gesetzt. Dies sagt dem Sender, daß der Drucker den Druckvorgang **erfolgreich** abgeschlossen hat; ein neuer Sende-Empfangs-Zyklus kann beginnen.

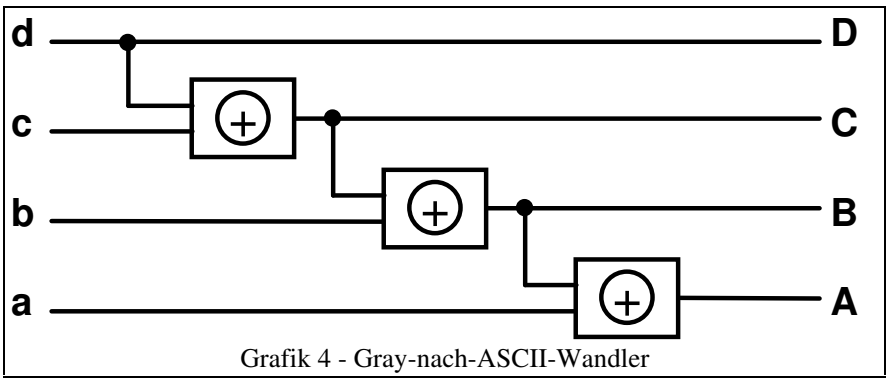
Versuchsdurchführung

Kodeumwandlung

Es wurde uns zum Ziel gesetzt, im Gray-Kode eingehende Daten auf dem Drucker darzustellen. Letzteres mußte natürlich in dem vom Drucker verwendeten ASCII-Code geschehen. Es war also eine Umwandlung von Gray nach ASCII nötig. Hierbei wurde als Zwischenschritt der Binärkode gewählt; dieser Schritt macht die Schaltung zwar nicht unbedingt einfacher, aber sehr viel übersichtlicher und Fehlersuchfreundlicher.

Von Gray nach Binär

Dieser Schritt ist recht simpel: Jede eingehende Gray-Kodierung wird in die Binärdarstellung übertragen, wobei keine Sonderregeln zu beachten sind. Nach etlichem Überlegen erkennt man, daß die Umwandlung wie folgt bewerkstelligen kann:



Grafik 4 - Gray-nach-ASCII-Wandler

Links werden die Eingabedaten eingegeben, rechts liegen die binären Ausgangsdaten an. Die Schaltung ist nur sehr schwer nachzuvollziehen, am besten ist es, man vergleicht anhand der Logiktabellen beider Kodes.

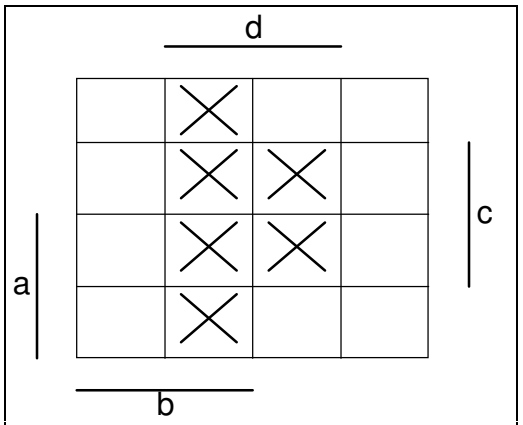
Die Funktion ist mit Hilfe eines Chips des Typs SN74LS86, der vier Exklusiv-Oder Verknüpfungen enthält, aufbaubar.

Von Binär nach ASCII

Wir haben die Daten jetzt in binärem Format und müssen sie noch immer in den ASCII-Kode wandeln. Hier hilft ein Blick in die Tabelle weiter: Die Ziffern, die wir darstellen wollen, entdecken wir ab Position 48 beginnend in der richtigen Reihenfolge. Die 48 entspricht im Zweiersystem dem vierten und fünften Bit $2^4 + 2^5 = 48$. Bei jeder Ziffer brauchen wir also nur diese Bits zu setzen; die unteren vier Bits können einfach von der binären Dekodierschaltung übernommen werden.

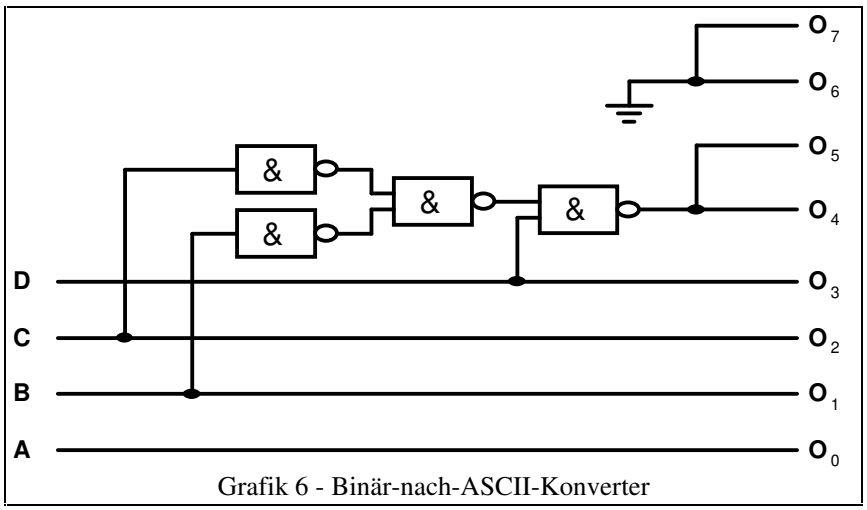
Genau das darf bei den gewünschten Steuerzeichen nicht geschehen. Diese können ohne weitere zugesetzte Bits einfach übernommen werden. Die Schaltung braucht also bloß die binäre Ausgabe des Gray-nach-ASCII-Wandlers zu testen, ob dort eine Zahl kleiner oder gleich 9 anliegt. Wenn ja, so werden die Bits 4 und 5 gesetzt, ansonsten gelöscht. Das 6. und das 7. Bit werden zu keinem Zeitpunkt gebraucht und deshalb statisch auf die logische Null (Masse) gelegt.

Eine Herleitung der benötigten Schaltung folgt leicht über ein KV-Diagramm, in das wir der Einfachheit halber die Kombinationen eintragen, die keine Ziffer darstellen; wir beschreiben also den invertierten Ausgang. Die Kombinationen $K = \{dcba \mid 1010,1011,1100,1101,1110,1111\}$ stellen keine Ziffern dar. Ins Diagramm eingetragen, erkennt man die beiden Primimplikanden cd und bd. Für den Ausgang gilt also $\overline{x} = cd + bd = d(c+b) = d(\overline{c}\,\overline{d})$, d.h. $x = \overline{d(\overline{c}\,\overline{d})}$.



Grafik 5 - KV-Diagramm der Nicht-Ziffern

Die Funktion wurde bereits auf die Schaltfunktion des NAND reduziert und kann mit einem Chip 74LS00 (4xNAND) leicht aufgebaut werden:



Grafik 6 - Binär-nach-ASCII-Konverter

Beide Dekodierschaltungen wurden nun zusammengeschaltet (d.h. die Ausgänge der Gray-Binär-Wandlers an die Eingänge des Binär-ASCII-Wandlers gelegt. Die Eingänge wurden über Schalter gesteuert, die Ausgänge an Leuchtdioden gelegt. Es zeigte sich, daß die Schaltungen problemlos zusammenarbeiteten und das richtige Ergebnis lieferten.

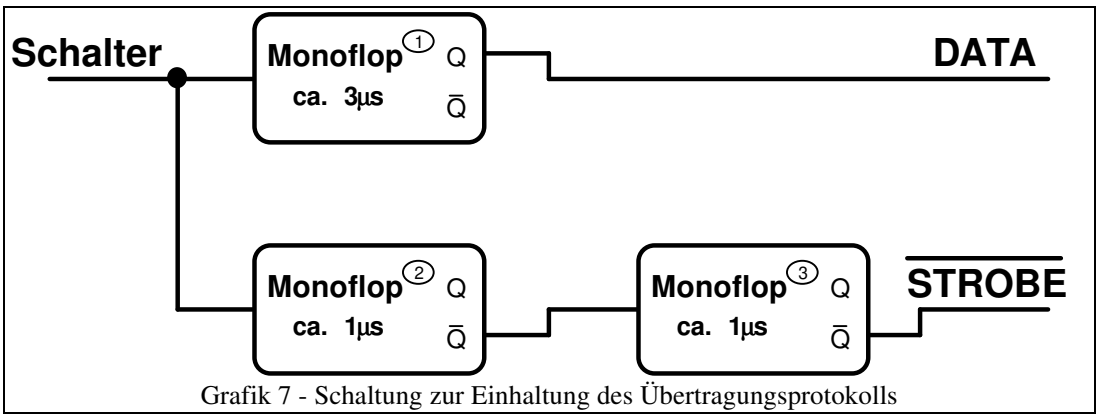
Einhaltung des Zeitprotokolls

Bislang funktionierten die Schaltungen asynchron. Bei der Ansteuerung des Druckers ist jedoch darauf zu achten, daß das definierte zeitliche Protokoll eingehalten wird.

Hierfür werden die zu übertragenden Daten aus dem ASCII-Konverter zunächst an ein Datentor gelegt, welches die Daten nur zu bestimmten Zeitpunkten durchläßt. Dieses wird einfach dadurch erreicht, daß jedes einzelne Signal über ein UND mit der DATA-Leitung verknüpft wird. Solange DATA auf 0 liegt, ist das Tor geschlossen, nur wenn es auf 1 liegt, können die Daten der Dekodierung an den Bus gelangen.

Das Verhalten der Steuerleitungen DATA und STROBE muß noch gesteuert werden. Wir möchten, daß die Übertragung eines Schalters durch das Umlegen eines Schalters gesteuert werden soll. Nach Grafik 3 soll nach diesem Umlegen zunächst die DATA-Leitung auf 1 gelegt werden, was unser Datentor öffnet. Mindestens fünf Mikrosekunden später wird STROBE ebenfalls auf 1 gelegt, was die Gültigkeit der Ausgabedaten am Datentor bestätigt. Weitere 5 Mikrosekunden später fällt STROBE zurück, und schließlich wird auch das Datentor geschlossen.

Die zeitliche Steuerung kann mit Monoflops eingestellt werden. Die prinzipielle Schaltung hierfür sieht wie folgt aus:



Grafik 7 - Schaltung zur Einhaltung des Übertragungsprotokolls

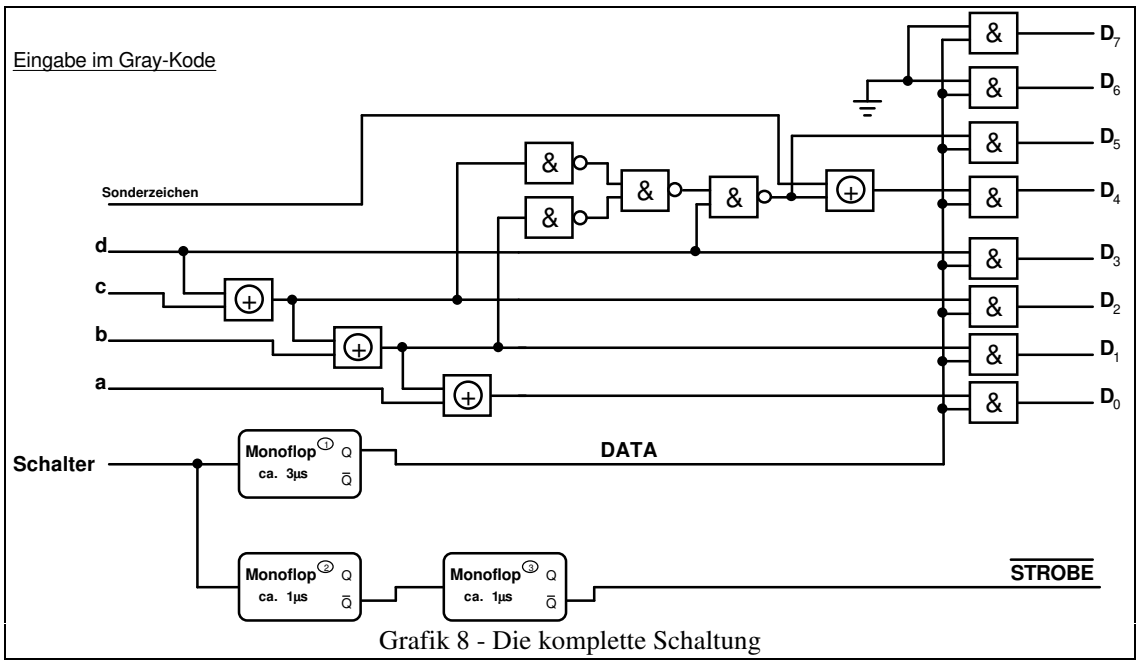
Der Schaltung liegt folgende Idee zugrunde: Die Monoflops schalten immer, wenn der Eingang von 0 auf 1 wechselt; im Ruhezustand liegt Q auf 0. Wird nun der Schalter betätigt, so schalten die ersten beiden Monoflops, die DATA-Leitung wird auf 1 gelegt, womit das Datentor geöffnet wird. Am Ausgang des zweiten Monoflops schaltet $\overline{Q}$ von 1 (Ruhezustand) auf 0, der dritte Monoflop wird also nicht verändert, $\overline{\text{STROBE}}$ bleibt also zunächst auf 1. Eine Mikrosekunde später haben sich die Ausgangswerte des Datentors stabilisiert; zu diesem

Zeitpunkt fällt Monoflop Nr. 2 in den Ruhezustand zurück, dessen $\overline{Q}$ schaltet also von 0 auf 1 und kippt damit den dritten Monoflop und setzt $\overline{\text{STROBE}}$ auf 0. Eine weitere Mikrosekunde später schaltet Nr. 3 zurück und legt den $\overline{\text{STROBE}}$ wieder auf 1. Zu diesem Zeitpunkt muß die BUSY-Leitung des Druckers auf 1 gewechselt sein, was uns anzeigen würde, daß das Zeichen auch angekommen ist. Da hier auf die Rückmeldungen des Druckers verzichtet wurde (es wurde nicht mit kritisch kleinen Zeitspannen gearbeitet), wird die Leitung einfach nicht abgefragt.

Nach 3 Mikrosekunden schaltet auch der 1. Monoflop zurück und schließt das Datentor. Nach Abwartung des Acknowledge vom Drucker kann ein neuer Übertragungszyklus beginnen.

In der Praxis sind die Monoflops in einem Baustein untergebracht, dessen Schaltzeit über den Anschluß eines CR-Gliedes bestimmt werden kann. Ein Differenzieren des Eingangssignals war nicht nötig.

Die komplette Schaltung



Grafik 8 - Die komplette Schaltung

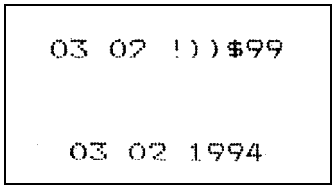
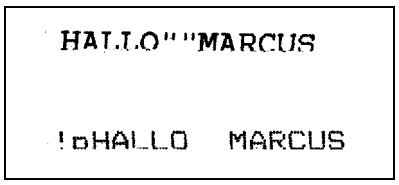
In die Leitung D_4 wurde noch ein zusätzliches Exklusiv-Oder geschaltet. Über einen Schalter konnte die Leitung invertiert werden. Diese Möglichkeit wurde von uns genutzt, um auch noch die Schriftzeichen, die sich auf der Tastatur über den Ziffern befinden, zu drucken. Unter anderem war auch ein Leerzeichen möglich (alle Datenleitungen 0, "Sonderzeichen" auf 1), welches zur schöneren Formatierung verwendet wurde.

Nachdem die entsprechenden Leitungen mit dem Drucker verbunden waren, konnte der Aufbau getestet werden. Die gestellte Aufgabe, das Datum des Versuchstages zu drucken, konnte ohne Probleme bewerkstelligt werden.

Um auch die Rückmeldung des Druckers zu testen, wurde dessen Acknowledge-Signal auf den Schaltereingang zurückgeführt. Dies bedeutete, daß die am Schalter anliegenden Daten so schnell wie möglich mehrmals an den Drucker geschickt wurden, was in etlichen Zeilen Ausdruck des gleichen Datums endete.

Interessehalber wurde zum Schluß auch die Gray-Binär-ASCII-Dekodierung von der Schaltung umgangen, um direkt mit den Schaltern ASCII-Werte an den Drucker übergeben zu können. Das Umschalten des Druckers in einen anderen Schriftmodus funktionierte auf Anhieb, beim Druck von alphanumerischen Zeichen lag der einzige Fehler beim Herr der Schalter, der in der Eile nicht alle Schalter auf Anhieb getroffen hat.

Der Ausdruck des Datums - ein erfolgloser und ein gelungener Versuch



Mit der Rechtschreibung hapert es noch etwas - aber der Ansatz ist vielversprechend