

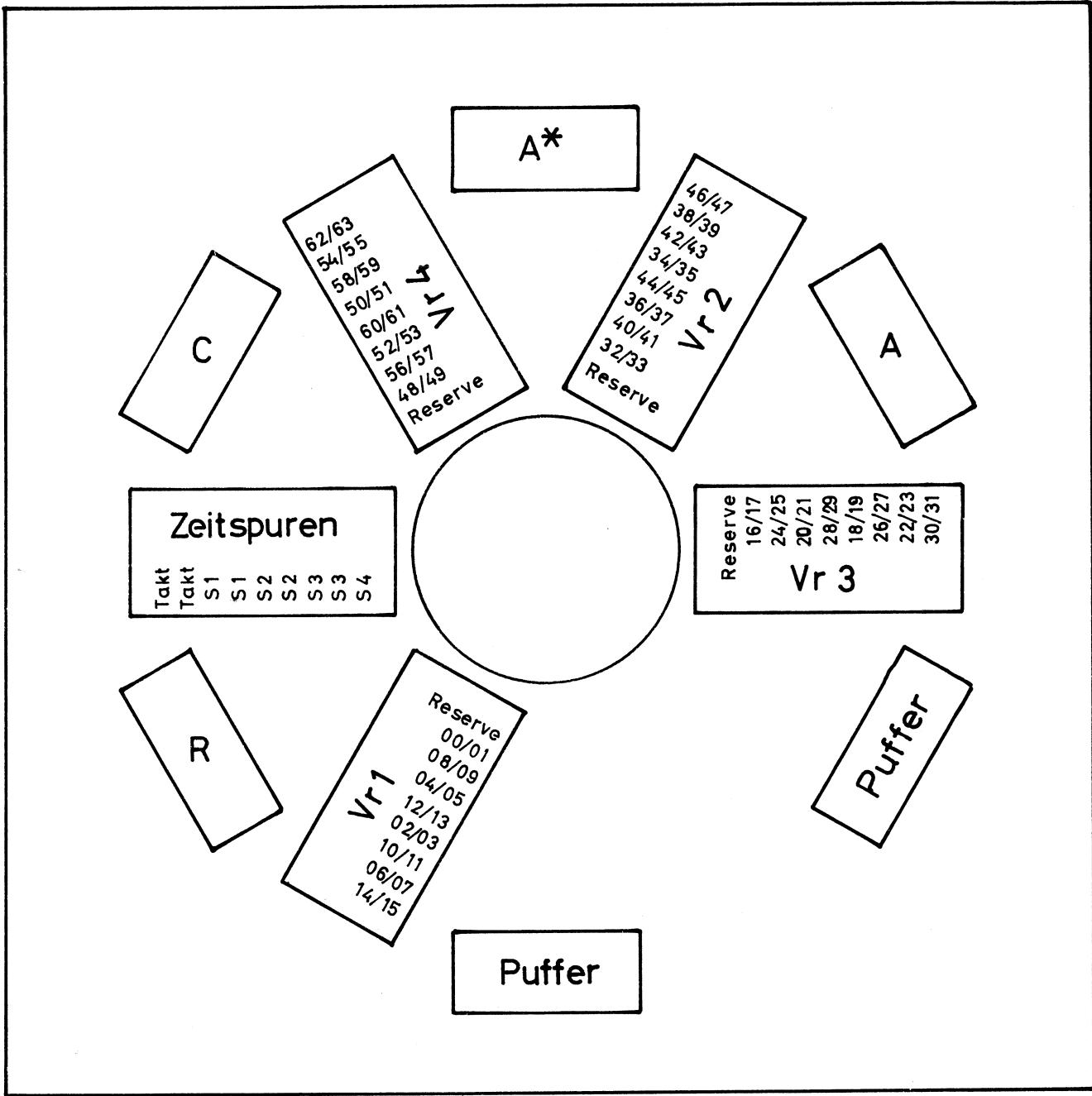
LGP-ZI

(C)

(C)

(C)

(C)

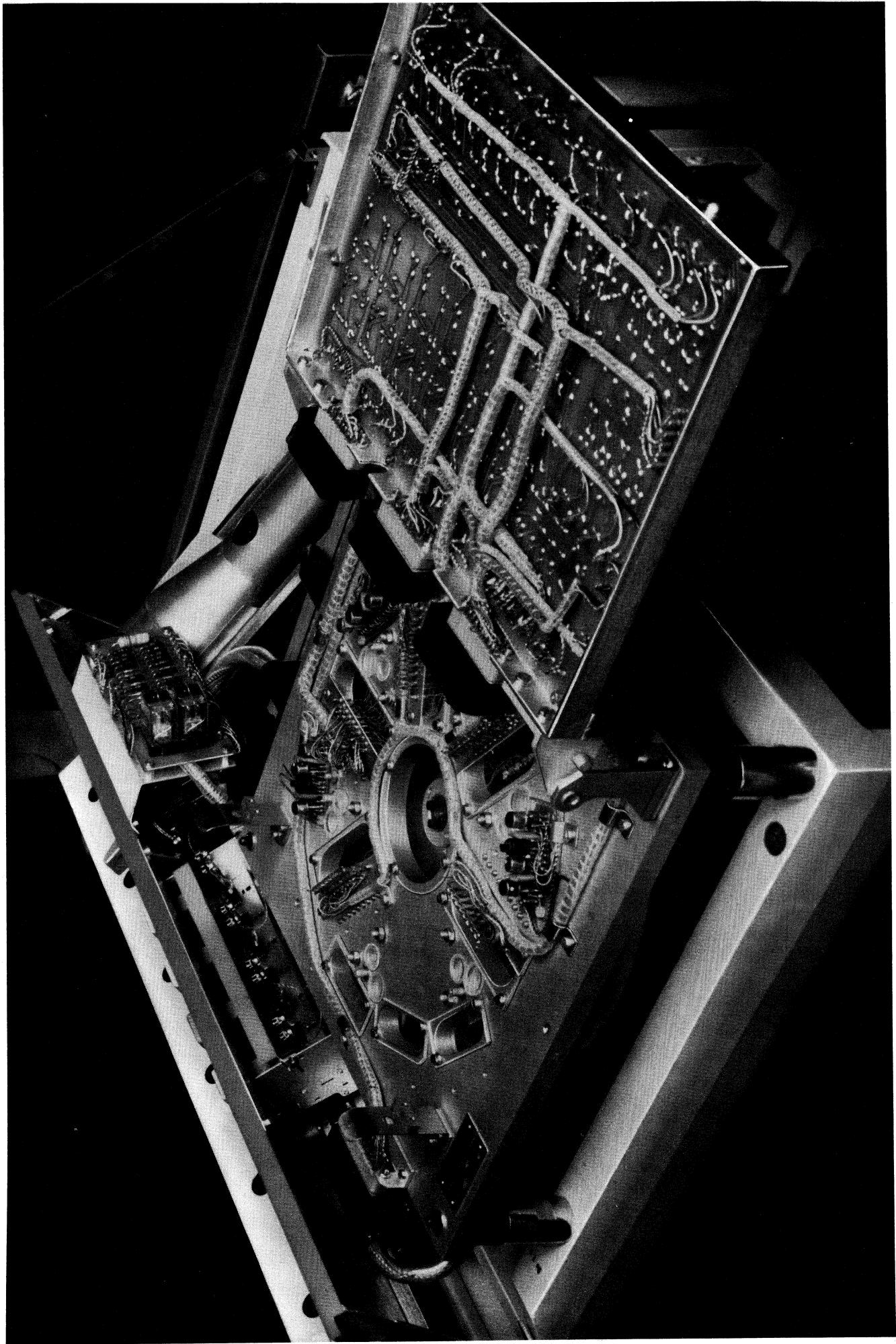


**Frontplatte Rechner**

### Plattenspeicher MSP 22

Anordnung der Köpfe







S a c h r e g i s t e r

=====

1. Allgemeines

1.1. Technische Daten der Grundausrüstung	A1 - 1
1.2. Prinzipielle Arbeitsweise	A1 - 2
1.3. Begriffe und Symbole	A1 - 4
1.4. Erweiterungen und Zusatzgeräte	A1 - 7

<b>2. Systembeschreibung</b>	A2 - 1
<b>    2.1. Stromversorgung</b>	A2 - 1
<b>        2.2. Speichereinheit und zugehörige Steuerung</b>	A2 - 1
2.2.1. Schreibverfahren	A2 - 1
2.2.2. Speicherplätze	A2 - 2
2.2.3a Druckschaltungsanordnung	A2 - 3
<b>        2.2.3. Hauptspeicher</b>	A2 - 5
2.2.3.1. Kopfanwahl	A2 - 5
2.2.3.2. Schreiben	A2 - 6
2.2.3.3. Lesen	A2 - 11
<b>        2.2.4. Umlaufregister</b>	A2 - 13
2.2.4.1. Schreibverstärker	A2 - 14
2.2.4.2. Leseverstärker	A2 - 15
<b>        2.2.5. Takt- und Zeitspuren</b>	A2 - 17
2.2.5.1. Leseverstärker für S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub>	A2 - 17
2.2.5.2. Leseverstärker für den Grundtakt cp	A2 - 17
2.2.5.3. Taktleistungsverstärker	A2 - 19
<b>    2.3. Grundschaltungen</b>	
2.3.1. Flip-Flop	A2 - 21
2.3.2. Norschaltung	A2 - 23
2.3.3. Diodengekoppelter Inverter	A2 - 24
2.3.4. Emitterfolger	A2 - 24
2.3.5. One Shot	A2 - 25
2.3.6. Astabiler Multivibrator	A2 - 27
2.3.7. Translator-Magnet-Treiber	A2 - 28
2.3.8. Translator-Kupplungs-Treiber	A2 - 29
2.3.9. Leistungs-Treiber	A2 - 30
2.3.10. Treiber	A2 - 31

## 2.4. Zeitsteuerung (Phasen)

### 2.4.1. Zeitmarkierungen

2.4.2. Phasen ( $\phi_1 \dots \phi_4$ )	A2 - 34
2.4.2.1. Phase 1 ( $\phi_1$ )	A2 - 35
2.4.2.2. Phase 2 ( $\phi_2$ )	A2 - 36
2.4.2.3. Phase 3 ( $\phi_3$ )	A2 - 37
2.4.2.4. Phase 4 ( $\phi_4$ )	A2 - 39

## 2.5. Funktion der Befehle

2.5.1. Z-Befehl	A2 - 41
2.5.1.1. Z0000 - Z0100	A2 - 41
2.5.1.2. Z-Befehl mit Spuradresse > 3	A2 - 42
2.5.1.3. -Z-Befehl	A2 - 43
2.5.2. B-Befehl	A2 - 45
2.5.3. Y-Befehl	A2 - 46
2.5.4. R-Befehl	A2 - 47
2.5.5. D-Befehl	A2 - 48
2.5.6. N-Befehl	A2 - 61
2.5.7. M-Befehl	A2 - 68
2.5.8. E-Befehl	A2 - 73
2.5.9. U-Befehl	A2 - 74
2.5.10. T-Befehl	A2 - 75
2.5.11. H-Befehl	A2 - 76
2.5.12. C-Befehl	A2 - 77
2.5.13. A-Befehl	A2 - 78
2.5.14. S-Befehl	A2 - 80

## 2.6. Bedienungsfeld

2.6.1. Ein/Aus	A2 - 81
2.6.2. Stop	A2 - 81
2.6.3. E/A	A2 - 81
2.6.4. Start	A2 - 82
2.6.5. PST, PS4, PS8, PS16, PS32	A2 - 82
2.6.6. Modus	A2 - 82
2.6.7. Füllen/Löschen	A2 - 84
2.6.8. Ausführen	A2 - 84

<b>2.7. <u>Ein-Ausgabe (E/A-Logik)</u></b>	
2.7.1. Anwahl der E/A-Einheit und Unterscheidung zwischen 4- oder 6-Bit Ein-Ausgabe	A2 - 85
2.7.2. Eingabe von Hand (Manuell)	A2 - 86
2.7.3. Eingabe über Flexowriter-Leser	A2 - 89
2.7.4. Links-Schiften	A2 - 92
2.7.5. Ausgabe über Flexowriter (Druckbefehl)	A2 - 93
2.7.6. Eingabe über Tally Leser	A2 - 95
<b>2.8. <u>Bedeutung der Buchstaben in den logischen Gleichungen</u></b>	A2 - 99
2.8.1. Flip Flops	A2 - 99
2.8.2. Logische Ausdrücke	A2 - 100
2.8.3. Schreibsignale	A2 - 101
2.8.4. Lesesignale	A2 - 101
2.8.5. Bedienungsfeld	A2 - 101
2.8.6. Ein-Ausgabe (E/A-Steuerung I)	A2 - 102
<b>2.9. <u>Logische Funktionsgleichungen</u></b>	A2 - 105
2.9.1. Phasensteuerung (80570)	A2 - 107
2.9.2. Rechenwerk (80571)	A2 - 111
2.9.3. Kommandowerk (80572)	A2 - 116
2.9.4. E/A-Steuerung I (80573)	A2 - 119
<b>2.10. <u>Blockschaltbilder</u></b>	
2.10.1. Phasensteuerung (80570)	A2 - 123
2.10.2. Rechenwerk (80571)	A2 - 125
2.10.3. Kommandowerk (80572)	A2 - 127
2.10.4. E/A-Steuerung I (80573)	A2 - 129
2.10.5. Flexowriter-Eingabe	{ A2 - 131 A2 - 133
2.10.6. Flexowriter-Ausgabe	A2 - 135
2.10.7. Tally Leser Eingabe	A2 - 137

## 1. Allgemeines

### 1.1. Technische Daten der Grundausrüstung

Das LGP-21 Standard-System setzt sich aus dem Rechner LGP-21 und einem modifizierten Flexowriter (Firma Friden) als Ein-/Ausgabe-Einheit zusammen.

Rechnertyp	Digitalrechner
Zahlendarstellung	binär
Adressensystem	Einadressensystem
Arbeitsweise	in Serie
Taktfrequenz	100 kHz
Speichertyp	magn. Plattspeicher
Speicherkapazität	4096 Worte
Wortlänge	30 Bit + Vorzeichen + Spacerbit
Speicherdrehzahl	1500 U/min
Zugriffszeit	min. 5,8msec - Mittel 21msec
Anzahl der Befehle	23

Rechengeschwindigkeit (ohne Zugriffszeit):

N-Multiplikation	ca. 21,7 msec
M-Multiplikation	ca. 22 msec
Division	ca. 22,3 msec
alle anderen Befehle	ca. 0,3 msec

Ein-/Ausgabeformat      dezimal, alphanumerisch,  
                              hexadezimal

Flexowriter                ca. 10 Zeichen/sec

Anschlußwerte:

220 V; 50 Hz, einphasig. Das Netz muß entsprechend den FTZ-Vorschriften entstört sein.

Zulässige Netzschwankungen	± 10 %
Zulässige Umgebungstemperatur	+10°C...+35°C
Leistungsaufnahme	300 Watt

Abmessungen:	Länge	Tiefe	Höhe	Gewicht
Rechner	65cm	70cm	88cm	162kg
Tisch m. Schreibmasch.	120cm	70cm	75cm	127kg

## 1.2. Prinzipielle Arbeitsweise:

Der LGP-21 arbeitet mit alphabetischen und numerischen Informationen. Die interne Zahlendarstellung ist rein dual. Mit Hilfe von gespeicherten Unterprogrammen kann jedoch eine dezimale Ein- bzw. Ausgabe erfolgen.

Auf dem Speicher befinden sich 32 Hauptspeicherspuren mit je 128 Zellen (beachte A2-3!) zur Speicherung von Befehlen und Zahlen (Rechnerprogramm).

4 weitere Spuren sind Umlaufregister: Der Akkumulator A und der doppeltlange Akkumulator A\* zur Speicherung der Operanden und Zwischenergebnisse; das Befehlsregister R, zur Speicherung des jeweils auszuführenden Befehls; und das Zählregister C, zur Speicherung der Adresse des nächsten Befehls.

Zur Zeitsteuerung des internen Rechenablaufs dienen 4 Taktspuren: Cp, S1, S2, S3.

Eine Hauptspeicherspur besteht aus 128 Zellen zu je 32 Bit. Das erste Bit gibt das Vorzeichen des Zelleninhaltes an, das letzte Bit ist eine Leerstelle. In den Bitpositionen 1 bis 30 kann der Absolutbetrag einer Zahl gespeichert werden. Die Bit-Position 0 enthält jeweils das Vorzeichen der Zahl (s. Bild 1.1).

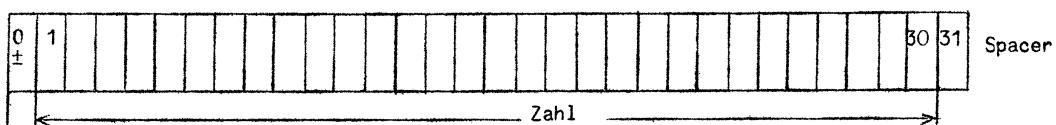


Bild 1.1 Zahlendarstellung

Eine Zelle, die zum Speichern eines Befehls benutzt wird, enthält den entsprechenden Befehl und die Adresse des Operanden. Die restlichen Bit-Positionen können noch zum Speichern von Zahlen verwendet werden (s. Bild 1.2).

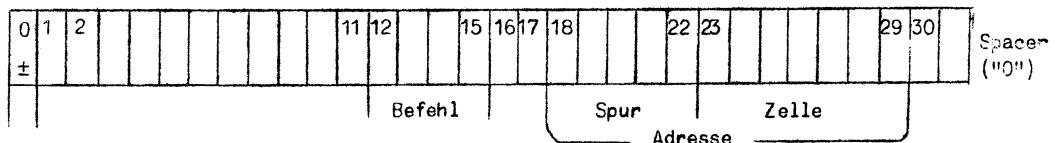


Bild 1.2. Befehlswort

Der Rechner führt einen Befehl in vier Arbeitsgängen aus:

- Phase 1 Das Zählregister bestimmt die Position im Hauptspeicher, in der sich der Befehl befindet. Die Spur wird ausgewählt, und bei Erreichen der betreffenden Zelle wird ein Koinzidenzsignal gegeben.
- Phase 2 Der Befehl wird vom Hauptspeicher in das Befehlsregister übertragen, gleichzeitig wird im Zähler eine "1" addiert.
- Phase 3 Der Operand, der durch die Adresse im Befehl bestimmt ist, wird im Hauptspeicher gesucht. Der zweite Operand befindet sich bereits im Akkumulator als das Ergebnis vorausgegangener Befehle.
- Phase 4 Der Rechenautomat führt die Operation aus, wie es im Befehl angegeben ist.

Die Zykluszeit beträgt 18 Wortzeiten, bei einer Sonderausführung des LGP-21 9 Wortzeiten. Diese Zykluszeit ist identisch mit der Zeit, die für die Ausführung eines Befehls bei optimierten Programmen notwendig ist mit Ausnahme der Befehle M, N und D, die zur Befehlsausführung eine Scheibenumdrehung plus eine Zykluszeit benötigen.

Alle Programme, die für den LGP-30 optimal sind, sind auch für die Normalausführung des LGP-21 optimal.

Programmierung siehe Programmierhandbuch.

### 1.3. Begriffe und Symbole

Der LGP 21 ist ein freiprogrammierbarer digitaler Rechenautomat. Alle arithmetischen und logischen Funktionen werden von einem gespeicherten Programm gesteuert. Der Speicher für Programm und Information wird als Hauptspeicher bezeichnet. Der Speicher für die Operationen und deren Ausführung als Register. Die logischen Verknüpfungen werden in der Logik durchgeführt. Die Zeitsteuerung (Takte) entspricht einem festverdrahteten Unterprogramm. Die E/A-Geräte dienen zur Ein- und Ausgabe.

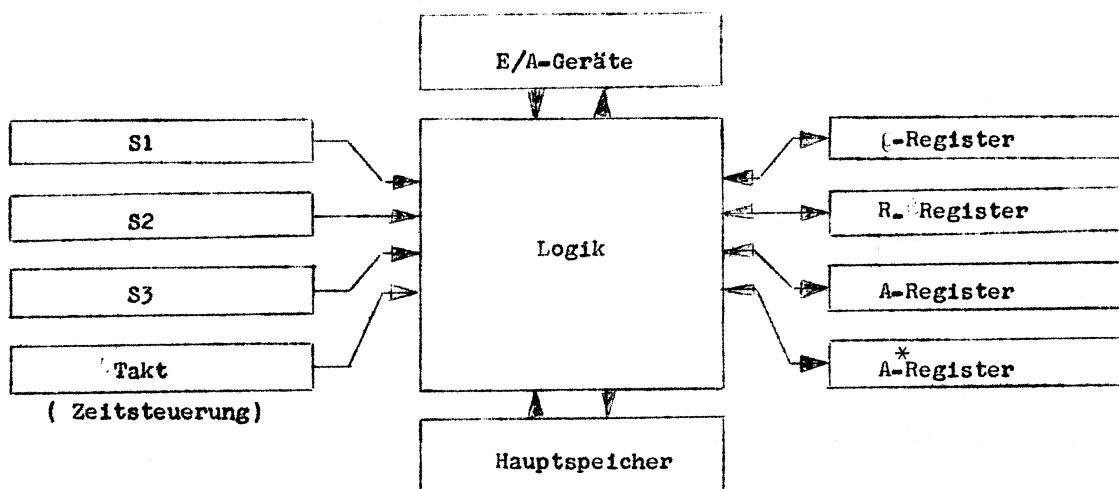


Bild 1.3.

Die Basis der Logik ist die Entscheidung zwischen zwei Möglichkeiten, und zwar zwischen logisch "nein" = "0" oder logisch "ja" = "1" (digital).

Im LGP 21 werden die beiden Zustände folgendermaßen dargestellt:

$$\begin{aligned}\text{logisch "1"} &= 0 \text{ V bis } -2 \text{ V} \\ \text{logisch "0"} &= -10 \text{ V bis } -20 \text{ V}\end{aligned}$$

Die Logik mit dieser Spannungsdarstellung bezeichnet man als "Positive Logik".

Für die einzelnen Verknüpfungspunkte lassen sich mit Hilfe der Bool'schen Algebra Gleichungen aufstellen. In diesen Gleichungen stehen alle Bedingungen die erforderlich sind, um für die Erfüllung einer logischen Funktion eine "ja"-Aussage zu erhalten.

Der invertierte Ausgang einer Verknüpfung wird als "Quer"-Ausgang bezeichnet (z.B. A invertiert = A, d.h., wenn A = 0 V hat, dann muß A = -20V haben).

Setz- bzw. Rücksetzsignale für Speicherelemente werden durch ein Apostroph ( $A'$ ) gekennzeichnet.

Beispiel einer Gleichung:

$$F' = \underline{F} \ \underline{G} \ \underline{H} \ T_3 \ P_1 + \dots$$

Das F-Flip-Flop wird gesetzt, wenn gleichzeitig F, G, H,  $T_3$  und  $P_1$  logisch "1" (0 V) sind. Das Oder (+) gibt an, daß das F-Flip-Flop noch von anderen Bedingungen gesetzt werden kann.

Der Ausgang F des Flip-Flops wird als "Ein-Seite" bezeichnet, der Ausgang F als "Aus-Seite". Das Flip-Flop wird durch ein F' zurückgesetzt (Aus-Seite = 0 V).



## 2. Systembeschreibung

Der LGP 21 besteht im wesentlichen aus folgenden Funktionsgruppen:

Stromversorgung

Speicherteil (mit Ansteuerung)

Logik

Bedienung (mit Ein- und Ausgabe)

### 2.1. Stromversorgung

Das Netzteil (Bild 3.1) des Rechners ist steckbar angeordnet. Es enthält die gesamte Stromversorgung des Rechners. Das Netzteil kann an die üblichen Netzspannungen angepaßt werden. Mit Hilfe einer Transistorverzögerungsschaltung wird der Speicherscheibenmotor nach Hochlauf (ca. 8 sec) auf 115 V umgeschaltet und gleichzeitig die verzögerte Gleichspannung -20d eingeschaltet. Außerdem enthält das Netzteil die Betriebskondensatoren für die Lüfter und den Speicherscheibenmotor. Das Netzteil ist nicht stabilisiert, da alle Schaltungen für die nach VDE zulässigen Netzspannungsschwankungen dimensioniert sind.

Die Daten des Netzteils sind im Datenblatt 80605Db1-X(4) (2 Blatt) und in der Prüfanweisung 80605 Pa1-X(4) zusammengefaßt.

### 2.2. Speichereinheit und zugehörige Steuerung

#### 2.2.1. Schreibverfahren:

Als Speichermedium dient eine mit Kobalt beschichtete rotierende Scheibe. Die Lese-Schreibköpfe schweben in einem Abstand von ca. 5 µ über der Scheibe auf einem Luftpolster, das sich durch die Drehbewegung der Scheibe bildet.

Jeder Kopf ist einer bestimmten Spur zugeordnet. Der radiale Abstand der einzelnen Spuren (ca. 0,8 mm) ist durch die Schreibkernbreite zuzüglich eines gewissen Sicherheitsabstandes gegeben. Beim Schreiben wird durch den Kopf ein so großer Strom geschickt, daß die magnetische Kobaltschicht bis in die Sättigung getrieben wird.

Durch Umkehren der Stromrichtung erfolgt eine entgegengesetzte Magnetisierung. Beim Lesen wird durch die Magnetisierungsänderung eine Spannung in den Lesekopf induziert. Jeder positive Leseimpuls zeigt einen Zustandswechsel in der einen Richtung, jeder negative einen Wechsel in der anderen Richtung an. Dieses Schreibverfahren wird als "non return to zero" (NRZ) bezeichnet, da nur die beiden Sättigungsstände vorkommen, die den beiden Binärwerten "0" und "1" zugeordnet sind.

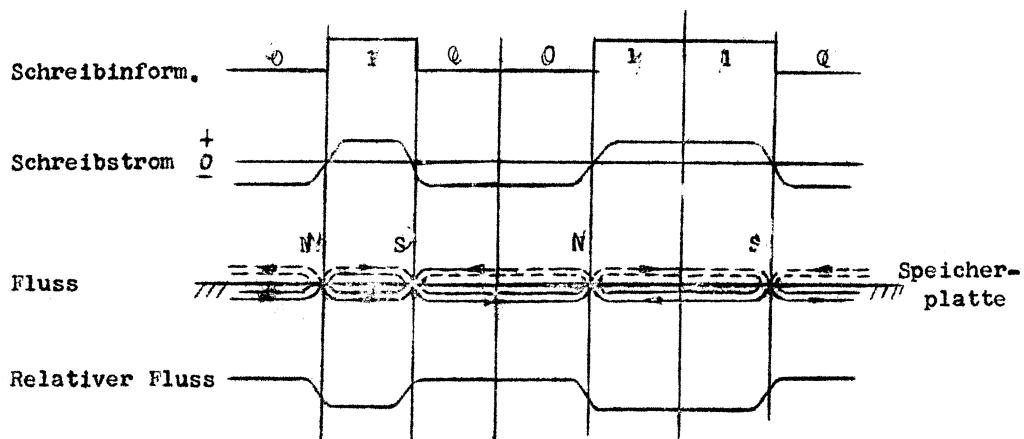


Bild 2.1.

### 2.2.2. Speicherplätze:

Die Speicherkapazität des Hauptspeichers beträgt 127 000 Informationsbits. Die Information wird in Serie (d.h. Bit nach Bit) übertragen und geschrieben. Eine Speicherspur nimmt 4096 Bit auf und ist in 128 Sektoren mit je 32 Bit unterteilt. Ein Sektor entspricht einem Wort.

Die Umdrehung der Speicherscheibe dauert ca. 41 msec, eine Bitzeit beträgt ca. 10  $\mu$ sec.

Der Hauptspeicher besteht aus 32 Spuren. Die den einzelnen Spuren zugeordneten Köpfe sind in 4 Blöcke zu je 8 (+ 1 Reservekopf) zusammengefaßt.

Die Zeitspuren  $C_p$ ,  $S_1$ ,  $S_2$ ,  $S_3$  und je eine beschriebene Reservespur befinden sich auf einem fünften Block.

Die 4 Umlaufregisterköpfe sind einzeln angeordnet.

Mechanischer Lageplan der Köpfe siehe Bild 3.2.

Beim Hauptspeicher wird zur Erleichterung der Programmierung und in Analogie zum LGP 30 Spur und Sektor durch jeweils 6 Binärstellen dargestellt. Hiermit ergibt sich eine Speichereinteilung in 64 Spuren mit je 64 Sektoren.

Beispiel:

Das gesuchte Wort sei Spur 48 Sektor 52; dann lautet die Adresse hexadezimal: 30KO (1. Code-Karte).

Die Numerierung der Hauptspeicherköpfe geht von 00 bis 31. Jede unter einem Kopf liegende Spur ist in zwei Programmspuren zu je 64 Sektoren eingeteilt.

Jede geradzahlige Spurnummer - bzw. jede ungeradzahlige Spurnummer minus eins - dividiert durch zwei ergibt die zugehörige Kopfnummer (s. Bild 3.2. und 3.3.).

#### 2.2.3. Druckschaltungsanordnung

Die einzelnen Druckschaltungen für die Speichereinheit mit zugehöriger Steuerung sind auf der Speichereinheit und in dem schwenkbaren Kartengehäuse 1 (Verdrahtungsplan 80645 VP1-X(2)) untergebracht.

Die Aufteilung der Elektronik auf die einzelnen Karten ist in den Blockschatzbildern 3.3. und 3.4. dargestellt.

Position 1 befindet sich direkt auf der Speicher-  
einheit.

Position 2 befindet sich auf der klappbaren Druck-  
schaltung über dem Speicher.

Position 3-6 befindet sich im Kartengehäuse 1.



## 2.3. Hauptspeicher (siehe auch Prüfvorschrift 80001-Pa1-X(4))

### 2.3.1. Kopfanwahl

Die Anwahl der einzelnen Köpfe erfolgt wie die des Knotenpunktes einer Matrix in Zeilen und Spalten (siehe Bild 3.3 bis 3.7).

Die einzelnen Spalten (Kanäle)  $Ch_1$  bis  $Ch_8$  werden durch  $P_3$ ,  $P_4$  und  $P_5$  angewählt, die einzelnen Zeilen (Blöcke)  $Vr_1$  bis  $Vr_4$  durch  $P_1$  und  $P_2$ .

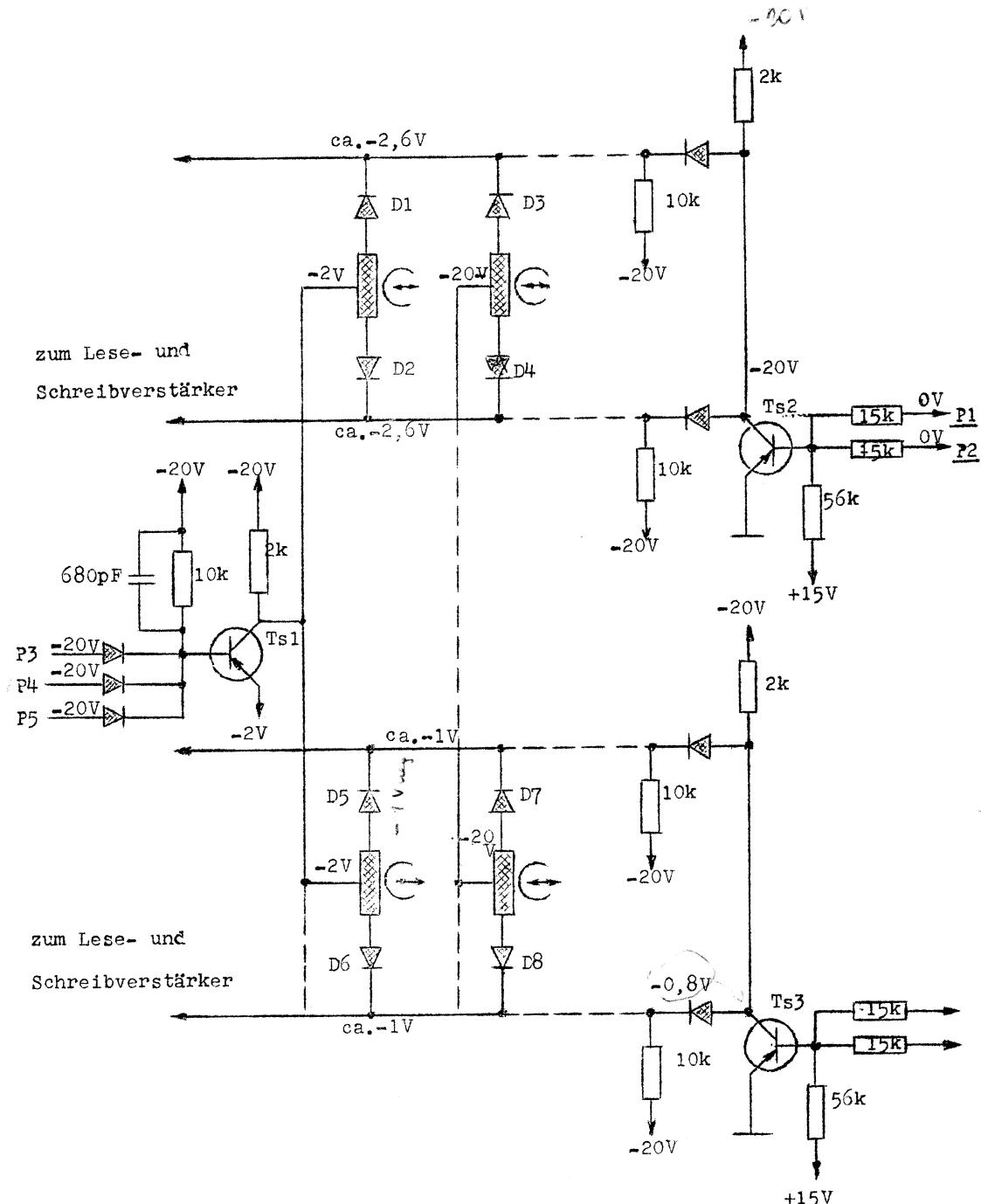


Bild 2.2.

Der Kopf 00 sei angewählt (Bild 2.2). Der entsprechende Kanal wird über den diodengekoppelten Inverter mit  $P_3$ ,  $P_4$ ,  $P_5$  ( $= -20V$ ) angewählt. Bei dem angewählten Kanal liegt an der Mittelanzapfung des Kopfes  $-2V$ , da  $T_{S1}$  durchgeschaltet ist.

Bei den nicht angewählten Kanälen liegt an der Mittelanzapfung des Kopfes  $-20V$ . Der entsprechende Block wird über ein Nor-Gatter mit  $P_1$   $P_2$  ( $= 0V$ ) angewählt.

Hiermit sind die Matrixdiode  $D_1$  und  $D_2$  leitend.

Sämtliche anderen Matrixdioden sind gesperrt. Die Sperrschwelle für das Lesesignal der nicht angewählten Köpfe beträgt einmal ca.  $2V$  (z.B.  $D_5$ ,  $D_6$ ) oder ca.  $18V$  (z.B.  $D_3$ ,  $D_4$ ).

Um beim Schreiben zu verhindern, daß mit jedem Kopf des angewählten Kanals geschrieben werden kann, ist die Schreiberlaubnis für die Schreibverstärker noch mit der Blockanwahladresse gekoppelt.

### 2.3.2. Schreiben:

Der Schreibstrom für den angewählten Kopf wird aus einer Konstantstromquelle gespeist. Dadurch erreicht man einen schnellen Stromanstieg in der Schreibspule. Der maximale Schreibstrom ist nach ca.  $2 \mu\text{sec}$  erreicht.

Arbeitsweise der Konstantstromquelle:

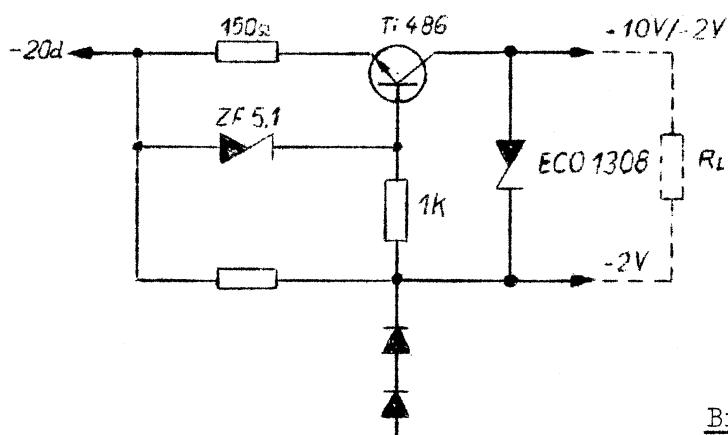


Bild 2.3.

Der Transistor wirkt als veränderbarer Widerstand, der den Strom auf  $30\text{mA}$  begrenzt. Je positiver das Potential  $U_E$  gegenüber  $U_B$  wird, je größer wird der Widerstand des Transistors. Das Basispotential  $U_B$  beträgt ca.  $14,9\text{ V}$ .

Das Emitterpotential wird durch den Spannungsabfall an  $R_1$  bestimmt

$$U_E = -20d + I_1 \cdot R_1$$

$R_1$  ist zusammen mit der Zenerdiode  $D_2$  so bemessen, daß der Strom nicht über 30 mA ansteigen kann. Der Spannungsabfall zwischen Emitter und Kollektor ist abhängig vom Lastwiderstand ( $R_L$  = klein,  $U_{CE}$  = groß;  $R_L$  = groß,  $U_{CE}$  = klein). Wird nicht geschrieben, so fließen die 30 mA über die Zenerdiode  $D_1$  und es steht am Kollektor ca. 10V. Wird geschrieben, fließen die 30 mA über den angewählten Kopf. Der Lastwiderstand  $R_L$  entspricht dann dem Innenwiderstand des Kopfes. Die Spannung am Kollektor bricht bis auf ca. -2V zusammen, da  $R_L < \frac{10V}{30mA}$  ist (siehe Bild 2.4).

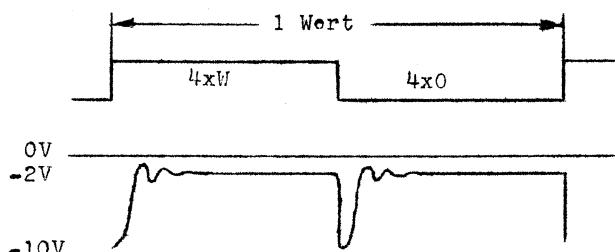


Bild 2.4.

Entsprechend dem Aufbau der Matrix werden vier getrennte Schreibverstärker benötigt. In Bild 2.5 ist der angewählte Schreibverstärker für Block  $Vr_1$  (Kopf 00) dargestellt.

Da die Schreiberlaubnis  $W$  und die Schreibinformation  $Vw'$  an allen Schreibverstärkern gleichzeitig anliegen, müssen alle nicht benötigten Schreibverstärker gesperrt werden. Dieses geschieht entsprechend der zugehörigen Blockanwahl mit  $P_1$ ,  $\underline{P}_1$ ,  $\underline{P}_2$ ,  $\underline{P}_2$  (siehe auch 2.3.1). Die Schreibinformation  $Vw'$  und  $\underline{Vw}'$  sind komplementär. Es kann nur einer der beiden Transistoren leiten. Auf dem Kopfkern befinden sich 2 Spulenhälfte mit gleichem Wicklungssinn. Entsprechend der Ansteuerung von

$Ts_1$  oder  $Ts_2$  fließt der Strom durch die eine oder andere Wicklungshälfte und erzeugt eine Magnetisierungsänderung, die einer logischen "1" oder "0" entspricht.

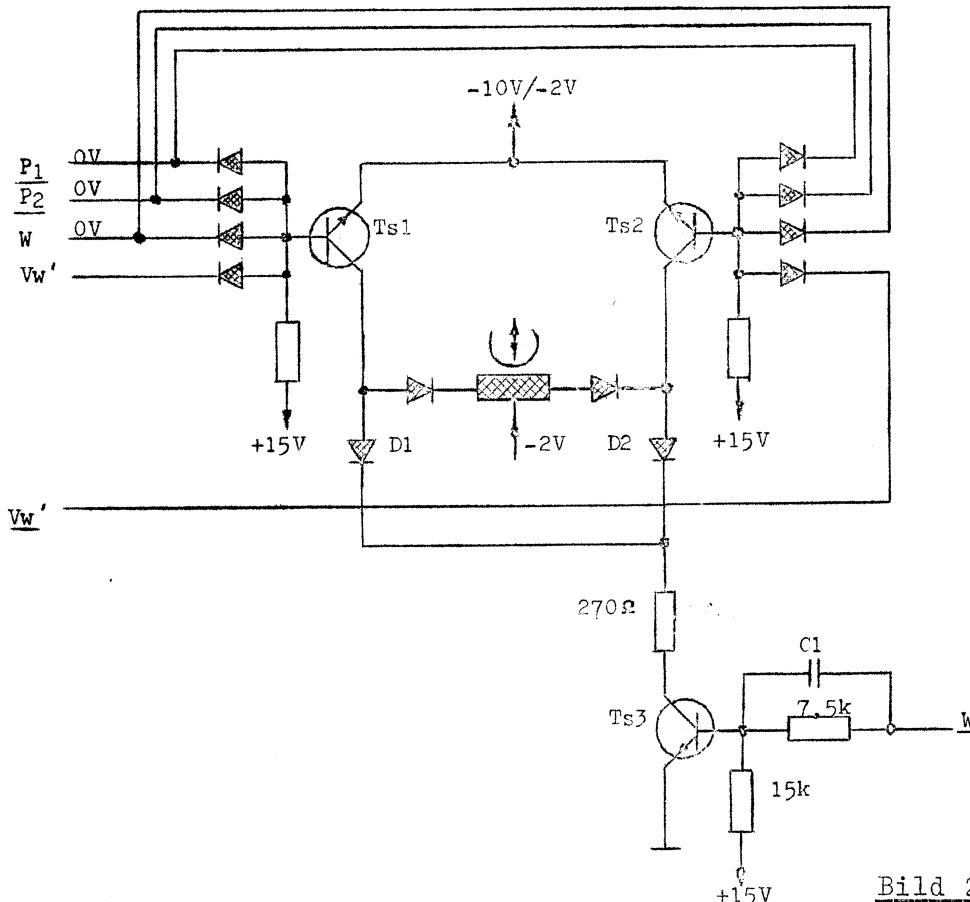


Bild 2.5

Die Schreiberlaubnis  $W$  setzt sich zusammen aus:

$$W = \underbrace{F G H}_{\phi 4} \underbrace{Q_1 Q_2 Q_3}_{H-, C-\text{Befehl}} + \underbrace{F G S_2}_{\phi 4} \underbrace{Q_1 Q_2 Q_3}_{Y-, R-\text{Befehl}}$$

Adreßzeit

Die Schreibinformation  $Vw'$  setzt sich zusammen aus:

$$Vw' = L \left[ A \underbrace{Q_1}_{R-\text{Befehl}} + A \underbrace{\frac{Q_1}{Q_4}}_{\text{Addiere "1" zum C-Register}} \underbrace{(K \underline{C} + K \bar{C})}_{\text{A-Information beim Y-Befehl}} \right]$$

$\frac{Q_1}{Q_4}$

R-Befehl      Addiere "1" zum  
C-Register

A-Information beim C-, H-Befehl

Spacerbit-Schreibinformation: "0"

Das Schreiben einer "0" zur Spacer-Bit-Zeit ist erforderlich, um beim Lesen am Anfang eines Wortes mit einem definierten Informationsbit zu beginnen (begründet durch das NRZ-Schreibverfahren).

In Bild 2.6. ist schematisch der Stromkreis beim Schreiben dargestellt.

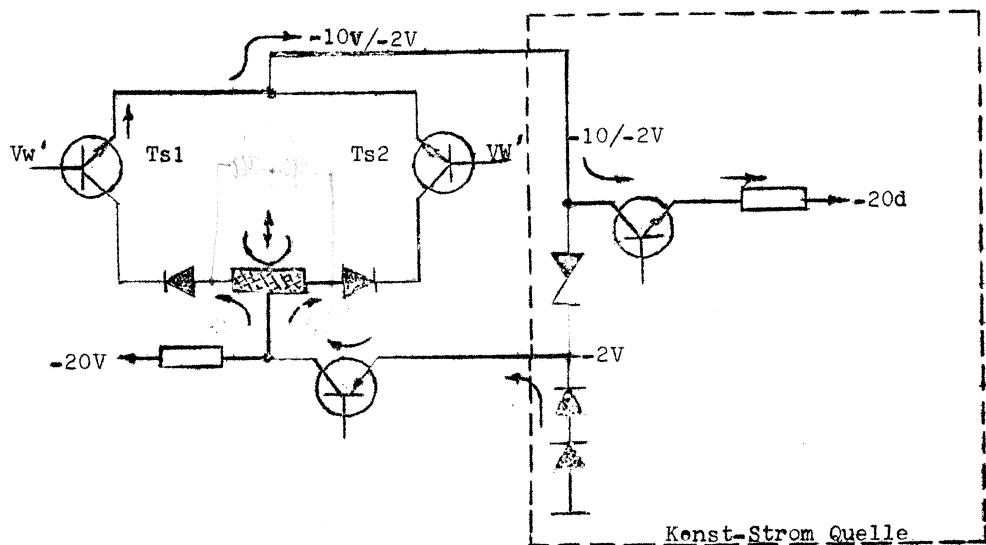


Bild 2.6.

Mit Hilfe des Transistors  $Ts_3$  (Bild 2.5) wird bei Schreibende die gespeicherte Energie im Kopf gegen OV abgeleitet. Bei Schreibende sind  $Ts_1$  und  $Ts_2$  gesperrt. Dadurch tritt eine Abschaltspannungsspitze auf, die zum Zerstören der Schreibtransistoren führen würde, wenn die Abschaltspitze nicht durch  $Ts_3$  begrenzt wäre.

Die Schaltung ist hauptsächlich beim Y- und R-Befehl von Bedeutung, da hier das Schreiben innerhalb einer Wortzeit erfolgt und das nachfolgende Bit nicht gelöscht werden darf. Das vom Kopf erzeugte Feld muß innerhalb einer Bitzeit abgebaut sein.



### 2.3.3. Lesen (siehe auch Bild 3.3, 3.5, 3.6, 3.8)

Am Ausgang des Leseverstärkers und am Ausgang des Lese-Flip-Flops stehen zu jeder Zeit Lesesignale an (auch beim Schreiben). Die Torung der Information erfolgt erst später in der Logik.

Der Leseverstärker ist ein linearer Differentialverstärker mit einer Spannungsverstärkung von ca. 290. Das gelesene Signal beträgt ca. 150 bis 350 mV. Bei größeren Signalen wird der Leseverstärker übersteuert. Um eine zu starke Übersteuerung beim Schreiben zu verhindern (lange Erholzeit des Verstärkers) wird durch die Dioden  $D_1$  und  $D_2$  (Bild 2.7) die Spannung am Eingang des Vorverstärkers auf ca. 0,6V begrenzt. Die Widerstände  $R_3$  und  $R_4$  dienen zur Entkopplung.

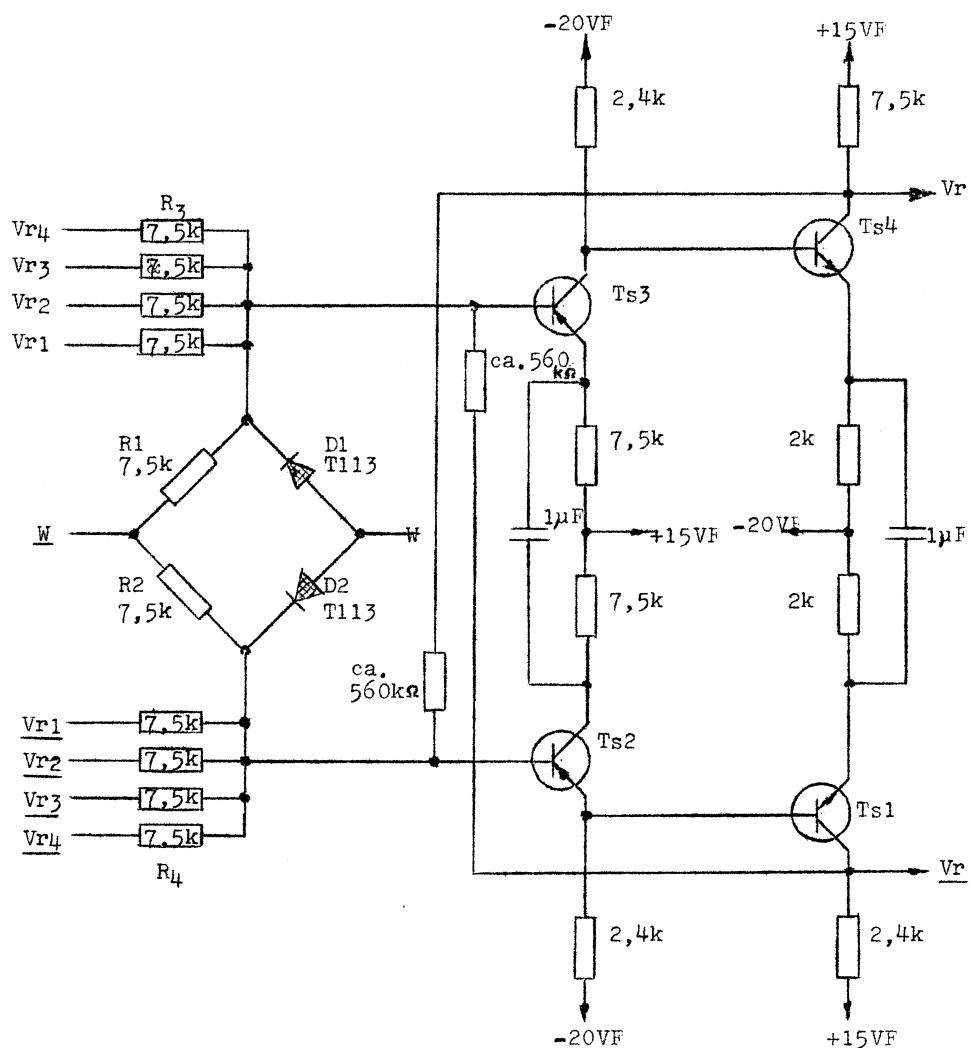


Bild 2.7.

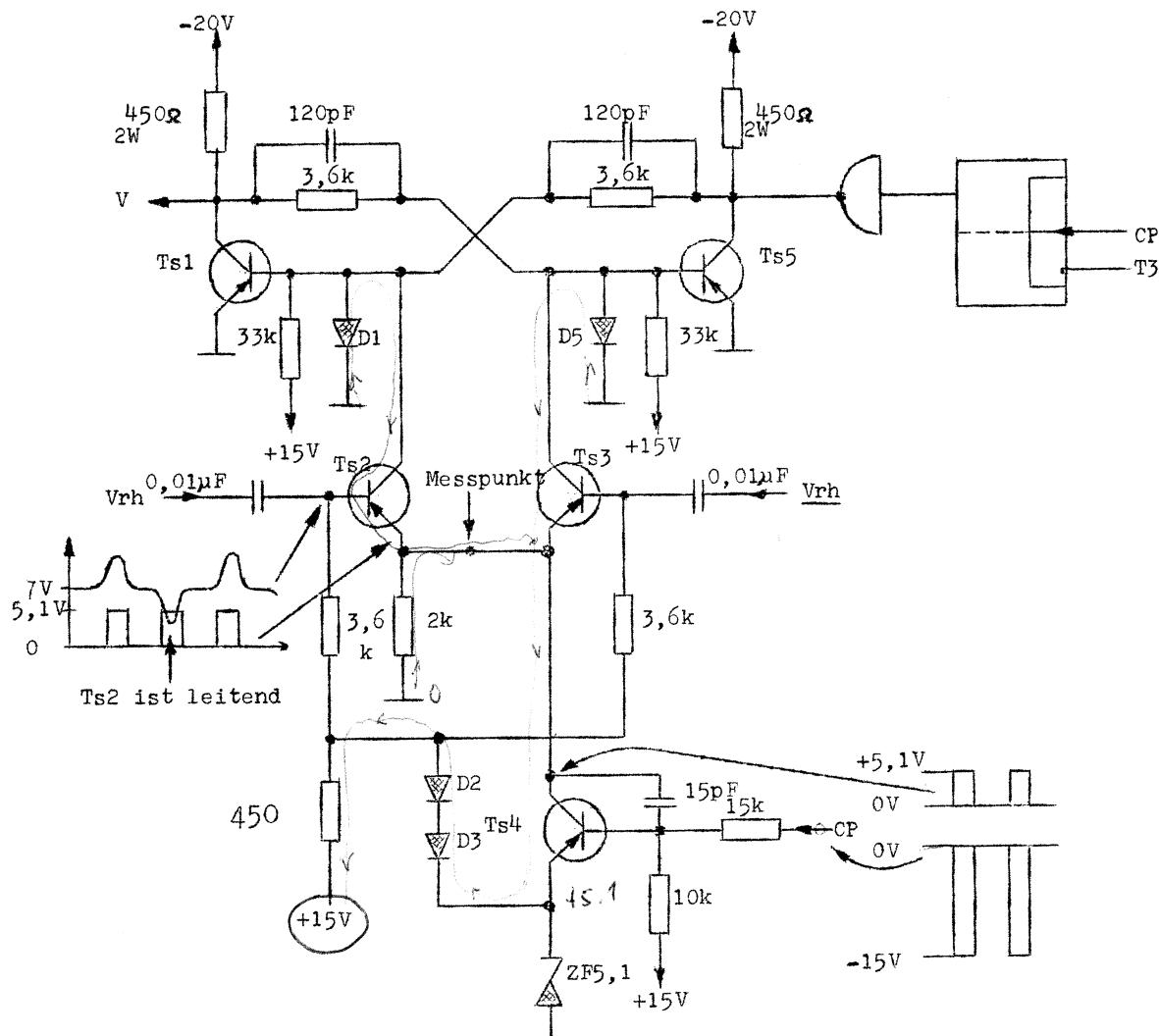


Bild 2.8.

Die gegenseitig um  $180^\circ$  phasenverschobenen Ausgangssignale des Vorverstärkers sind kapazitiv auf die Basen der Transistoren  $Ts_2$  und  $Ts_3$  gekoppelt (siehe Bild 2.8). Die Basis von  $Ts_2$  und  $Ts_3$  liegt entsprechend der Spannung  $U = U_Z + 2 \cdot U_D (= 5,1 + 2 \cdot 0,6 = 6,3V)$  gleichstrommäßig auf einem Potential von +6V bis +7V. Diesem Gleichstrompotential überlagert sich das Ausgangssignal des Vorverstärkers.

Am Emitter von  $Ts_2$  und  $Ts_3$  liegt über  $Ts_4$  ein invertierter Takt  $cp = CP_1 \dots$  von OV bzw. +5,1V. Bei  $cp=OV$  ist  $Ts_4$  durchgeschaltet. Sein Kollektor liegt auf +5,1V. Bei  $cp=-15V$  ist  $Ts_4$  gesperrt. Sein Kollektor liegt über  $R = 2k$  auf OV.

Wenn  $Ts_4$  leitend ist und am Eingang  $V_{rh}$  bzw.  $\underline{V}_{rh}$  ein negativer Impuls anliegt, wird  $Ts_2$  bzw.  $Ts_3$  durchgeschaltet und das positive Potential des Kollektors von  $Ts_4$  auf die Basis von  $Ts_1$  bzw.  $Ts_5$  gegeben. Hierdurch wird das V-Flip-Flop entsprechend der gelesenen Information gesetzt. Wenn Transistor  $Ts_4$  gesperrt ist, kann keine positive Spannung an  $Ts_1$  bzw.  $Ts_5$  gelegt werden. Dadurch wird erreicht, daß nur Signale die zur Taktzeit kommen das V-Flip-Flop setzen können.

Der Transistor  $Ts_5$  des V-Flip-Flops wird zur Spacer-Bit-Zeit durch einen Setzimpuls an seinem Kollektor auf OV gezogen.

Dadurch ist gewährleistet, daß zur Spacer-Bit-Zeit unabhängig von der gelesenen Information das V-Flip-Flop auf "0" gesetzt wird.

#### 2.2.4. Umlaufregister

Der LGP 21 besitzt 4 Umlaufregister. Das A-, C- und R-Register sind 1 Wort-Umlaufregister. Das A\*-Register ist ein 2 Wort plus 1 Bit - Umlaufregister.

Der A\*- und A-Registereingang liegen parallel. Das A\*-Register wird nur während des M-, N- und D-Befehles benutzt.

Bei den Umlaufregister-Köpfen befinden sich auf einem Schuh je ein Lese- und Schreibkopf. Beide Köpfe liegen auf der gleichen Spur, d.h. die Information, die der Schreibkopf schreibt, wird vom Lesekopf gelesen.

Der Abstand zwischen Lese- und Schreibkopfspalt beträgt je nach Art des Registers 32 oder 65 Bitzeiten.

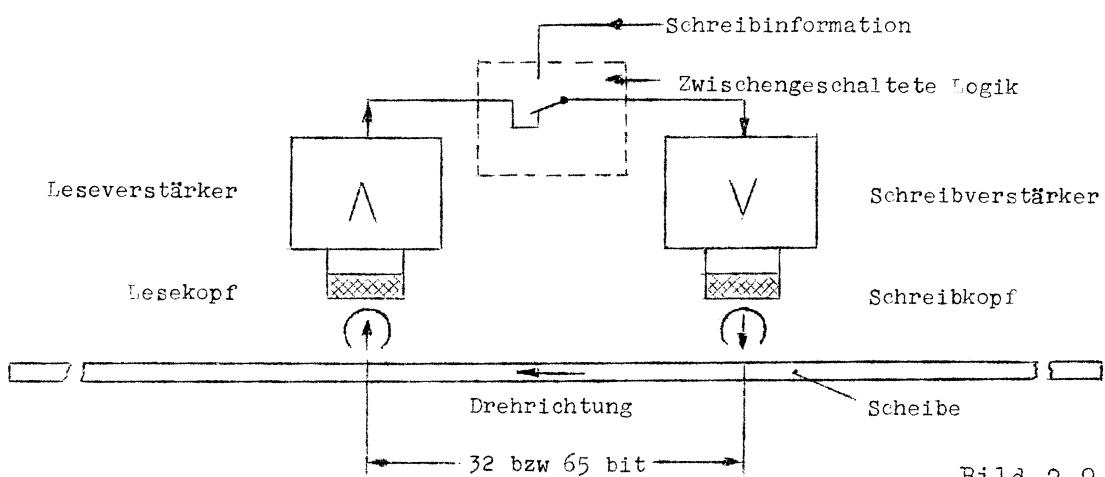


Bild 2.9.

Das geschriebene Bit wird 32 bzw. 65 Bitzeiten später wieder gelesen und gleichzeitig neu aufgeschrieben (siehe Bild 2.9). Die Information steht zu jeder Wortzeit zur Verfügung. Durch die zwischengeschaltete Logik kann der Umlauf zum Einschreiben eines neuen Wortes unterbrochen werden.

#### 2.2.4.1. Schreibverstärker (siehe auch Bild 3.10)

Der Schreibverstärker ist als Flip-Flop (Bild 2.10) ausgebildet (Erklärung des Flip-Flop s. 2.3.2.1).

Da jeweils einer der beiden Transistoren  $T_{s1}$  oder  $T_{s2}$  leitend ist, fließt ein Strom von ca. 30 mA durch den Schreibkopf. Z.B.:  $T_{s1}$  ist leitend. Der Strom fließt von OV über  $T_{s1}$ ,  $R_1$ , Schreibkopf,  $R_2$  und  $R_3$  nach -20V.

Kippt das Flip-Flop, kehrt sich die Stromrichtung um. Damit ergibt sich eine für die "0" und "1" Unterscheidung gewünschte entgegengesetzte Magnetisierungsrichtung auf der Scheibe.

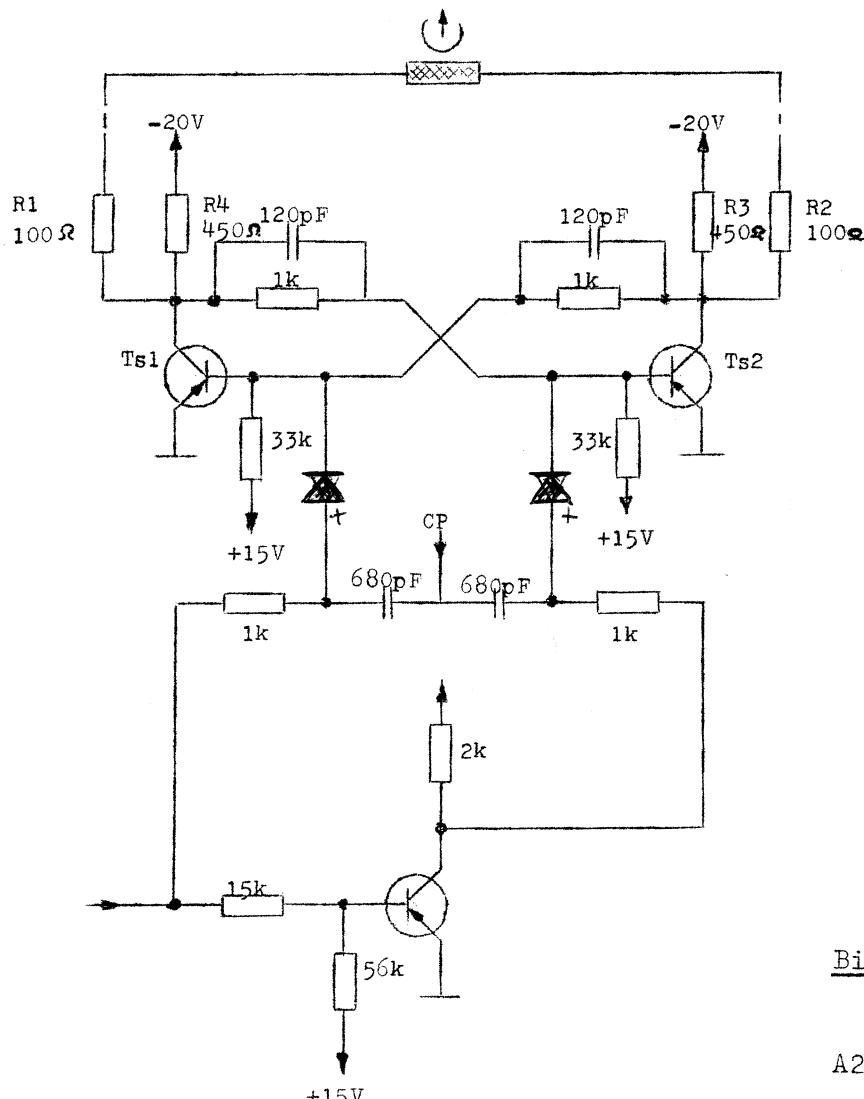
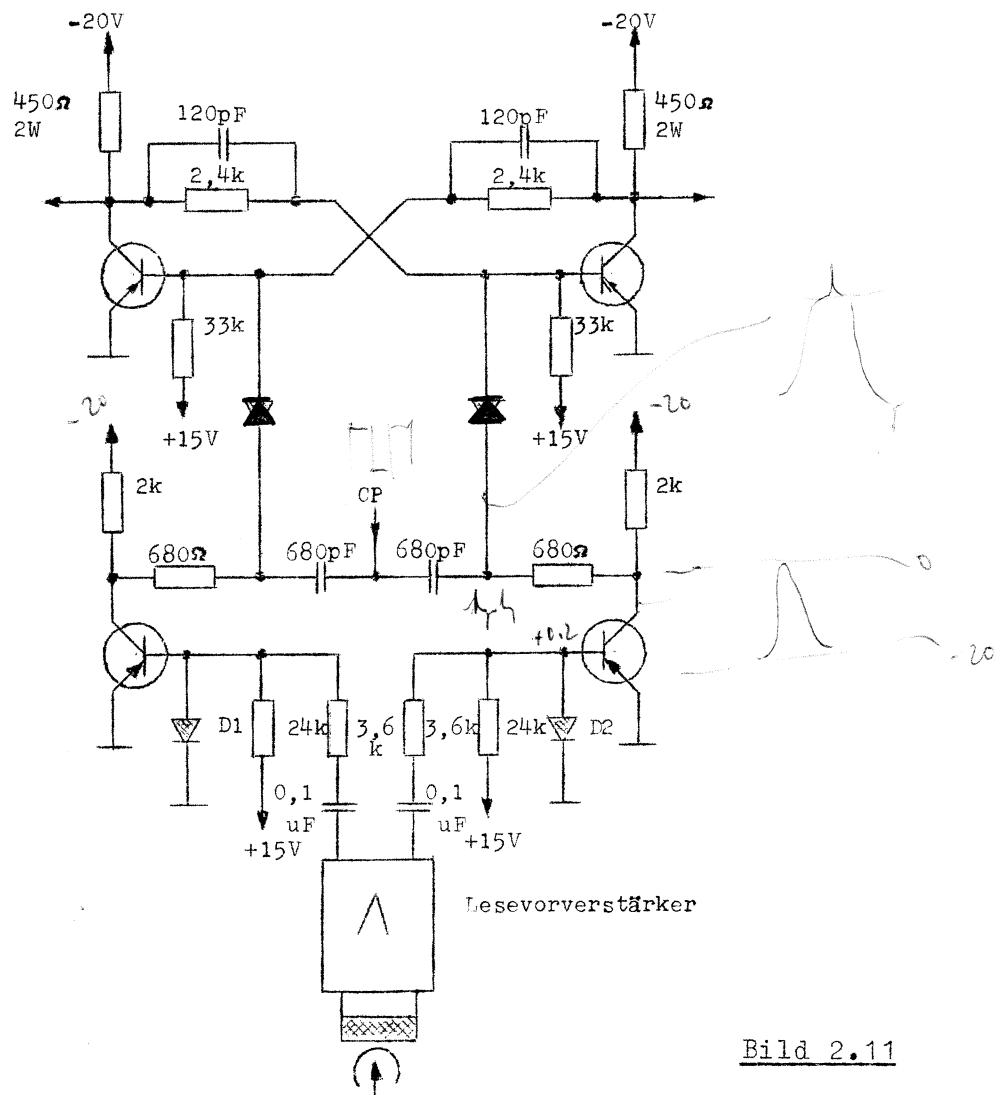


Bild 2.10

#### 2.2.4.2. Lese-Verstärker (siehe auch Bild 3.12)

Das Lese-Signal des Kopfes wird in einem linearen Differentialverstärker (ähnlicher Aufbau wie beim Hauptspeicher und bei den Taktspuren) vorverstärkt und kapazitiv auf die nächste Stufe gekoppelt (Bild 2.11). Die Dioden  $D_1$  und  $D_2$  erzeugen eine Vorspannung von ca. +0,6V und verhindern eine positive Spannung von +15V an der Basis. Sie dienen zur Störunterdrückung. Die negative Hälfte des Analogsignals steuert jeweils  $T_{S1}$  oder  $T_{S2}$  durch. Der leitende Transistor setzt dann zusammen mit der positiv gehenden Flanke von CP das nachgeschaltete Flip-Flop.





## 2.2.5. Takt- und Zeitspuren

Zur Steuerung des Rechenablaufs im LGP 21 dienen die vier Taktspuren Cp, S<sub>1</sub>, S<sub>2</sub> und S<sub>3</sub>. Diese werden bei S&F mit einem externen "Spuren-Schreibgerät" aufgeschrieben und dürfen danach nicht mehr verändert werden. Für jede Taktspur ist eine beschriebene Reservespur vorhanden (siehe Anordnung der Köpfe Bild 3.2). Die Taktspuren sind nur mit Leseverstärkern ausgerüstet. Die Köpfe sind normale Hauptspeicherköpfe.

### Achtung!

Durch Ziehen der den Spuren zugeordneten Karten unter Spannung bzw. durch Löten an den Taktspurkreisen mit einem an Netzspannung liegenden Lötkolben, können Störbits auf die Spuren geschrieben werden. Sie sind dann unbrauchbar!

### 2.2.5.1. Leseverstärker für S<sub>1</sub>, S<sub>2</sub> und S<sub>3</sub> (s. auch Bild 3.13)

Die Leseverstärker für die Spuren S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub> sind mit dem Leseverstärker für die Umlaufregister identisch (siehe Abschnitt 2.2.4.2).

### 2.2.5.2. Leseverstärker für den Grundtakt Cp (siehe auch Bild 3.13)

Das Lesesignal wird zunächst wieder in einem linearen Differentialverstärker verstärkt und dann über einen Resonanzkreis (C19, R43, L1) auf die Basis von Ts13 gegeben. Durch negative Signale wird Ts13 leitend. Sein Ausgangssignal wird mit C21 (680 pF) und R47 (470 Ω) differenziert. Die positiven Spitzen setzen das nachfolgende One Shot (monostabiler Multivibrator; Taktzeit min 3,5/μs, max 5/μs).

Der über R49, R50 normalerweise leitend gehaltene Ts14 sperrt bei einem positiven Impuls über D14. Über den Emitterfolger (Ts15) koppelt der negative Sprung am Kollektor von Ts14 auf die Basis von Ts16. Ts16 wird leitend. Über den Kondensator C23 (680pF) wird der positive Sprung am Kollektor von Ts16 kapazitiv auf die Basis von Ts14 zurückgekoppelt. Das Sperren von Ts14 wird beschleunigt. Man erhält am Kollektor von Ts14 steile negative Flanken.

Ts14 bleibt solange gesperrt, bis das Potential an seiner Basis nach der Zeitkonstanten, gebildet aus  $(R49 + R50)$ . C23 wieder negativ geworden ist. Die Zeitkonstante und damit die Sperrzeit von Ts14 kann mit dem Potentiometer R49 variiert werden. Die Zeit sollte ca. 4 bis 4,5  $\mu$ sec betragen.

Das Ausgangssignal Cp des One-Shots dient zur Ansteuerung der nachfolgenden Taktverstärker auf den einzelnen Karten.

Achtung!

Der Ausgang Cp des One-Shots ist gegen OV und positive Spannungen nicht kurzschlußfest.

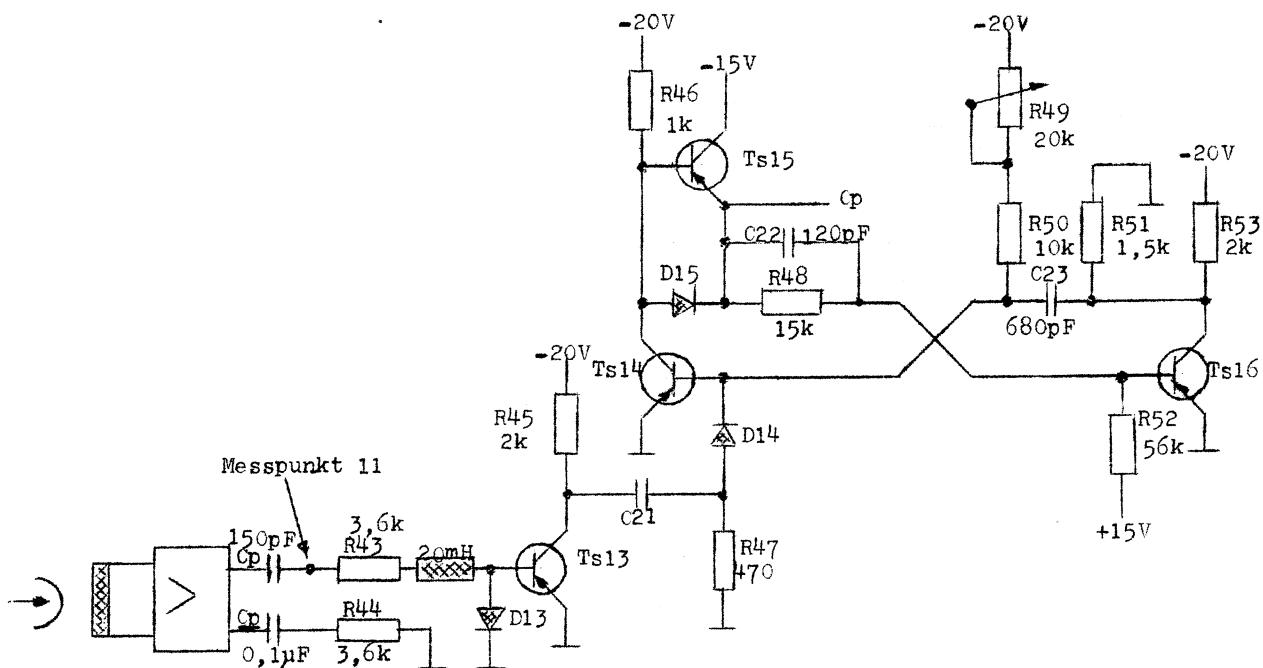


Bild 2.12

### 2.2.5.3. Takt-Leistungsverstärker

Der nachfolgende Taktverstärker besteht aus einem Emitterfolger und einem niederohmigen Inverter (Bild 2.13).

Er wird zur Ansteuerung der einzelnen Flip-Flops benutzt (max 5 Flip-Flops pro Verstärker).

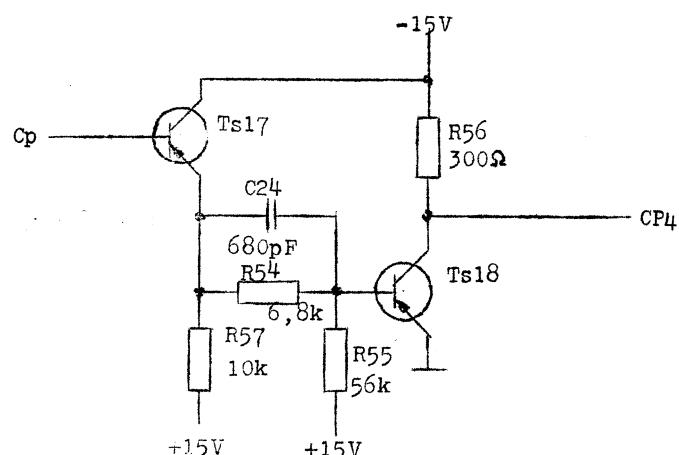


Bild 2.13



## 2.3. Grundschaltungen

### 2.3.1. Flip Flop

Das Flip Flop dient zur Informationsspeicherung. Es ist ein bistabiler Multivibrator, der durch Setzimpulse von einem in den anderen seiner zwei stabilen Zustände gekippt werden kann.

Das Flip Flop wird mit einer logischen "1" (=OV) gesetzt. Zur Setz- bzw. Rücksetzzeit müssen beide Eingänge  $A'$ ,  $\underline{A}'$  entgegengesetzte Wertigkeiten haben. Falls keine Setzbedingungen gegeben sind, liegen beide Eingänge auf logisch "0" (= -10V...-20V).

Die zwei Ausgänge A,  $\underline{A}$  des Flip Flops haben grundsätzlich entgegengesetzte Wertigkeit.

$$\begin{array}{ll} A = \text{OV} = \text{logisch "1"} & \underline{A} = -20V = \text{logisch "0"} \\ A = -20V = \text{logisch "0"} & \underline{A} = \text{OV} = \text{logisch "1"} \end{array}$$

Das Flip Flop kann nur zur Taktzeit gesetzt werden und auch nur dann, wenn das Setzsignal am Eingang  $A'\underline{A}'$  ca. 3,3 bzw. 2  $\mu$ sec vor dem Takt ansteht. Diese Eingangszeitkonstanten sollen das Flip Flop störunanfälliger machen.

Zwei Flip Flop-Typen finden im Rechner Verwendung. Sie unterscheiden sich nur in ihren Eingangszeitkonstanten von 3,3  $\mu$ sec bzw. 2  $\mu$ sec und der damit verbundenen Schaltungsänderung (unterschiedliche Eingangswiderstände, siehe Bild 2.14).

#### Arbeitsweise: (Bild 2.14)

Annahme: Das Flip Flop ist im eingeschwungenen Zustand. Der Transistor Ts2 ist leitend (Ts1 also gesperrt). Damit ist das Basispotential von Ts2 mit ca. 0,5V und das von Ts1 mit ca. +1V gegeben.

$\underline{A}'$  wird nun z.B. OV (logisch "1"). An der Anode von D2 liegt folglich ein Potential von OV. Infolge des Spannungsabfalls von ca. 0,6V an Dioden in Durchlaßrichtung, bleibt D2 noch gesperrt.

Die Anode von D1 liegt auf einem Potential von -12V... -20V, da am Eingang A' keine Setzbedingung vorliegt. D1 ist sicher gesperrt.

Über die Kondensatoren C1 und C2 wird der Spannungssprung des Taktes CP auf das Potential der Anoden von D1 und D2 differenziert. Bei einem positiven Spannungssprung wird D2 leitend und sperrt Ts2. D1 bleibt gesperrt, weil der Spannungssprung nicht größer als 12V werden kann, da die Amplitude von CP auf 12V begrenzt ist.

Über den Kopplungswiderstand R2 wird Ts1 leitend. Bei leitendem Ts1 wird Ts2 über den Kopplungswiderstand R1 gesperrt gehalten.

Die Kondensatoren C1 und C2 dienen nur zur Beschleunigung.

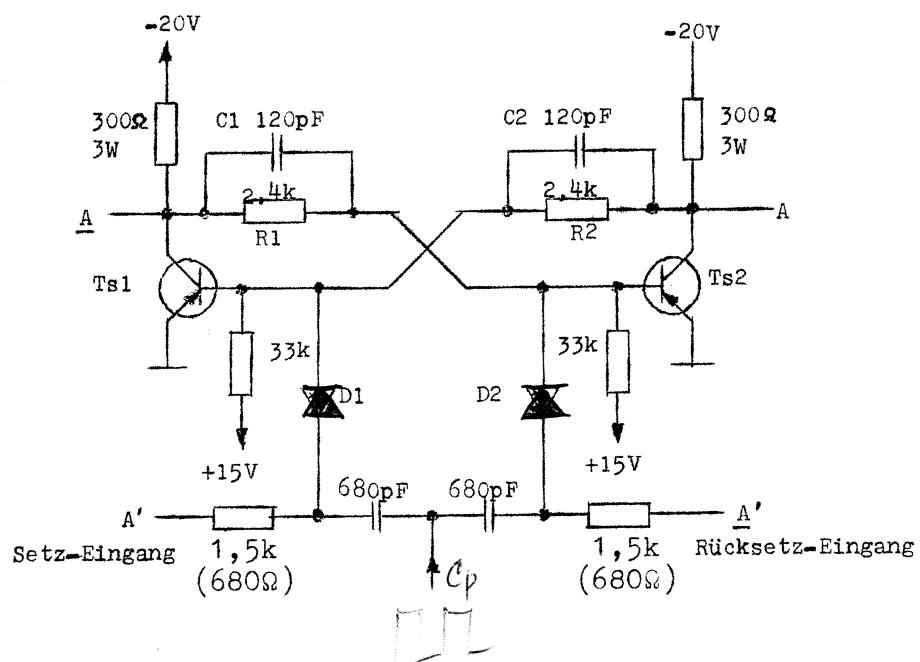


Bild 2.14

### 2.3.2. Nor-Schaltung

Die Nor-Schaltung entspricht einem ODER-Gatter mit Inverter für negative Signalrichtung, einem UND-Gatter mit Inverter für positive Signalrichtung.

Sein Ausgang ist logisch "1" (=0V), wenn mindestens ein Eingang logisch "0" (= -20V) ist.

Sein Ausgang ist logisch "0" (-20V), wenn alle Eingänge logisch "1" (=0V) sind.

Die Eingangsspegel müssen in einem Bereich von -6V bis -20V bei log. "0" und -0,1...-0,6V bei log. "1" liegen.

Die Ausgangsamplitude ist von der jeweiligen Belastung abhängig. Sie liegt zwischen -6V bis -20V.

Im LGP 21 werden zwei verschiedene Nor-Gatter-Typen eingesetzt. Sie unterscheiden sich in ihrer Ausgangsimpedanz und der damit verbundenen maximalen Eingangszahl (Bild 2.15).

Das niederohmige Nor-Gatter dient zur Einspeisung einer größeren Zahl anderer Gattereingänge (maximal 15 Stück). Die Zahl der Eingänge für dieses Gatter ist auf max 5 begrenzt.

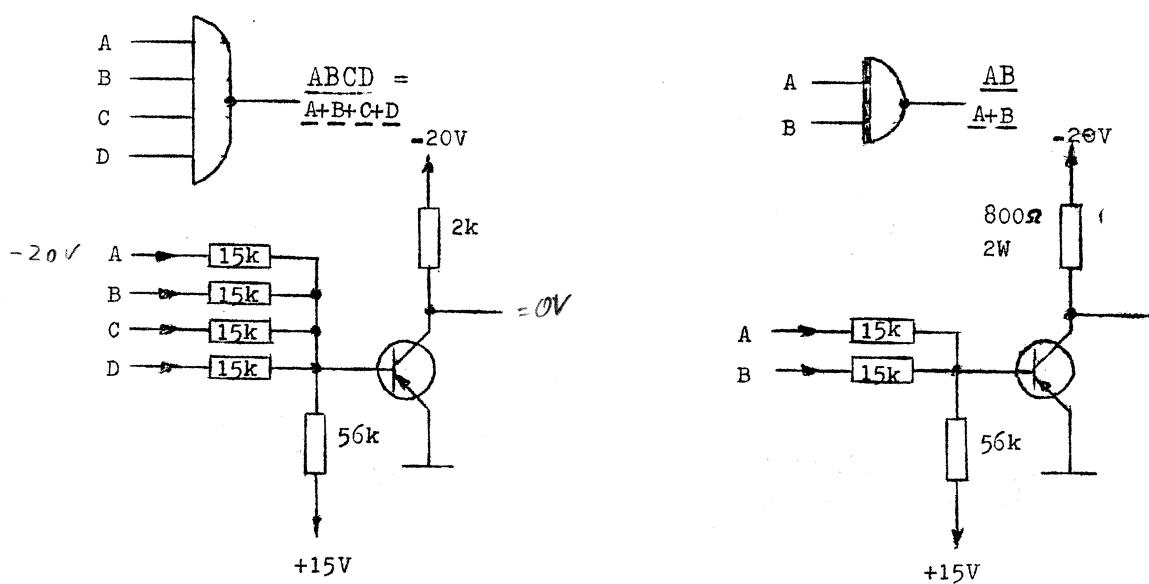


Bild 2.15

### 2.3.3. Diodengekoppelter Inverter

Der diodengekoppelte Inverter ist ein Und-Gatter mit Inverter für negative Signalrichtung, ein ODER-Gatter mit Inverter für positive Signalrichtung.

Sein Ausgang ist logisch "0" (= -20V), wenn wenigstens ein Eingang logisch "1" (=0V) ist.

Sein Ausgang ist logisch "1" (=0V), wenn alle Eingänge logisch "0" (= -20V) sind.

Der diodengekoppelte Inverter hat eine ca. 10 mal schnellere Schaltzeit als das Nor-Gatter.

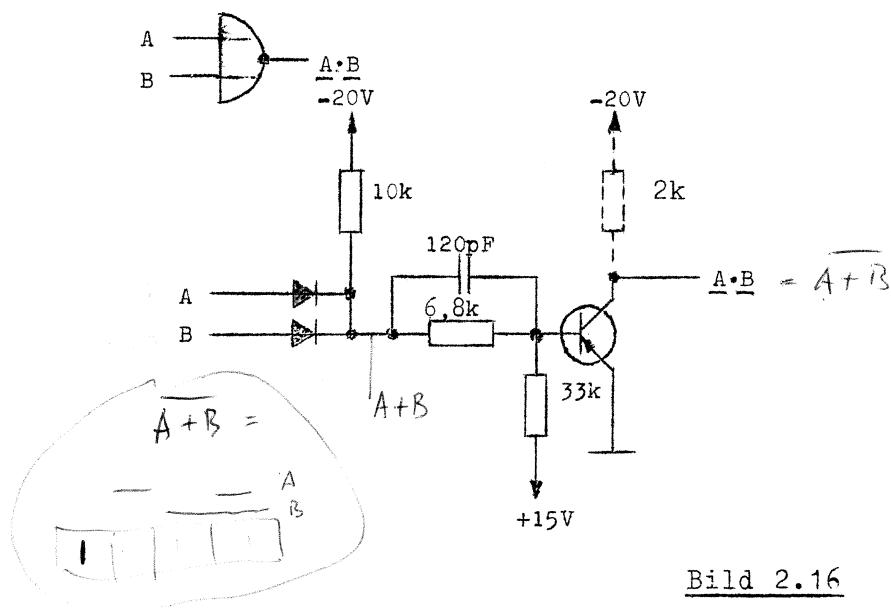


Bild 2.16

### 2.3.4. Emitterfolger

Der Emitterfolger ist ein Leistungsverstärker mit einer Spannungsverstärkung  $< 1$ .

Das Eingangssignal wird nicht invertiert!

Der Emitterfolger ist niederohmig gegen -15 Volt und wird deshalb zur Einspeisung niederohmiger Eingangsstufen bzw. einer größeren Anzahl von Folgestufen verwendet.

#### Achtung!

Der Emitterfolger ist gegen OV und positive Spannungen nicht kurzschlußfest.

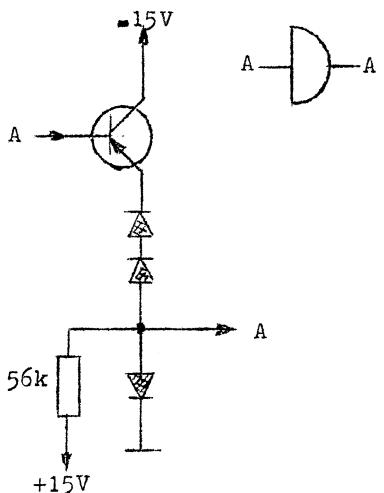


Bild 2.17

### 2.3.5. One Shot (Monostabiler Multivibrator)

Normalerweise sind die Steuersignale für die elektromechanischen Ein-/Ausgabegeräte im Verhältnis zu den internen Rechnersignalen erheblich länger. Die One-Shots werden deshalb hauptsächlich für eine Impulsverlängerung eingesetzt.

Im Ruhezustand ist der Ausgang A logisch "1" (=0V) und A logisch "0" (-20V). Mit der positiven Flanke eines Eingangssignals an A\* wird das One Shot für eine definierte Zeit gesetzt, d.h. sein Ausgang A wird logisch "1" (0V), sein Ausgang A wird "0" (-20V). Danach kippt es entsprechend seiner Zeitkonstanten in den Ruhezustand zurück.  
Mit einem Potentiometer (R1) kann die Zeitkonstante variiert werden.

Im Ruhezustand wird Ts1 über R1, R6 leitend gehalten. Sein Basispotential liegt bei ca. -0,5V. Ts2 ist entsprechend seinem Basispotential von +3V gesperrt (Spannungsteiler Ts1, R4, R9). Zur Störunterdrückung ist D1 mit ca. 7 Volt durch den Spannungsteiler R5, R3 gesperrt.

Die Setzimpulse an A\* werden über C2 R3 differenziert. Durch positive Spitzen > 8V wird D1 leitend. Die Basis von Ts1 wird dadurch positiv und Ts1 sperrt. Dadurch wird Ts2 leitend und der Kondensator C1 wird umgeladen. Die Folge ist, daß an der Basis von Ts1 entsprechend dem Spannungssprung an C1 ein positives Potential steht.

Dieses kann nur über R1 und R6 abgebaut werden, und Ts1 bleibt damit entsprechend der Zeit  $(R1+R6) \cdot C1$  gesperrt. Wird Ts1 wieder leitend, wird Ts2 gesperrt und das One Shot befindet sich wieder in Ruhelage.

Die Diode D3 liegt beim Sperren von Ts2 in Sperrrichtung. Das Kollektorpotential von Ts2 ist also unabhängig von der Umladezeitkonstanten von C1. Die negative Flanke bleibt steil.

Da sich der Kondensator C1 infolge der gesperrten Diode D3 nicht wie bei einem normalen One Shot über R11 umladen kann, mußte R8 und D2 eingefügt werden. Damit die Erholzeit des One Shots (Umladung von C1 auf ein definiertes Potential) nicht übermäßig lang wird, begrenzt der Spannungsteiler R7 R8 die negative Spannung auf ca. -8,5V.

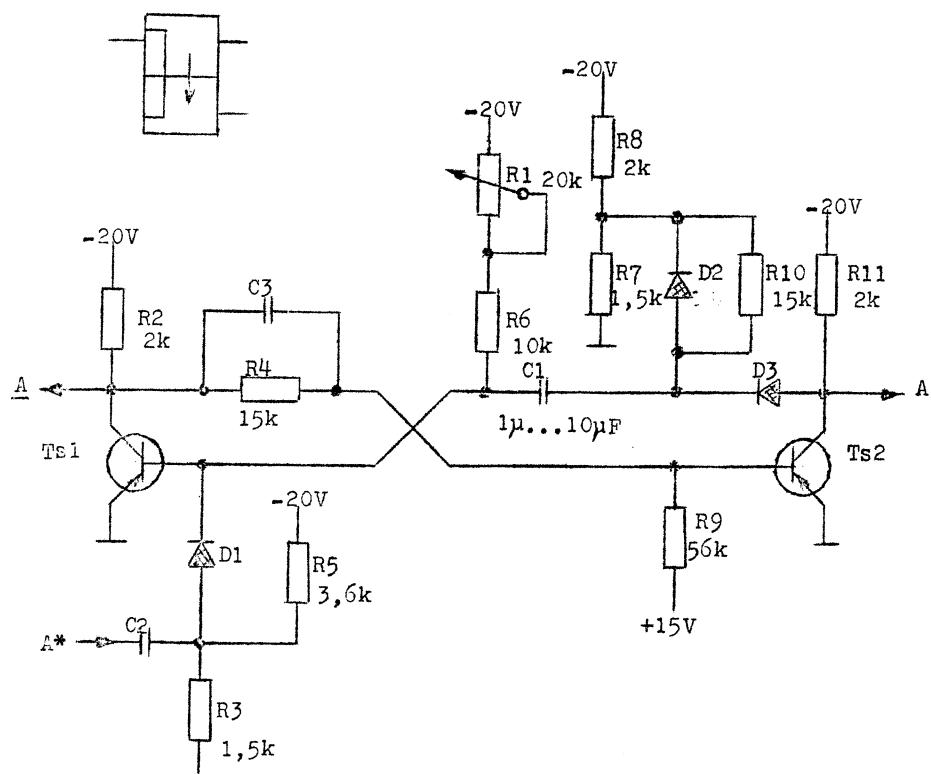


Bild 2.18

### 2.3.6. A stabiler Multivibrator

Der astabile Multivibrator (Bild 2.20) kippt selbständig in einem bestimmten Rhythmus zwischen den beiden möglichen Schaltstellungen.

Die Kippzeiten  $T_1$  und  $T_2$  sind bestimmt für:

$T_1$  durch die Zeitkonstante  $C_1 \cdot (R_1 + R_3)$

$T_2$  durch die Zeitkonstante  $C_2 \cdot (R_2 + R_{10})$

Die Spannung -20d verhindert ein Anlaufen des Multivibrators beim Einschalten des Rechners. Während dieser Zeit hat der Ausgang A ständig OV = "1".

Wird Ts2 leitend, so wird über  $C_1$  ein positiver Spannungssprung gekoppelt und Ts1 gesperrt. Diese positive Spannung an seiner Basis wird über R1 und R3 solange abgebaut, bis Ts1 seinen Basisstrom über R3 und R1 wieder ziehen kann. Dadurch wird Ts1 leitend und sperrt über D1 und C2 den Transistor Ts2. Ts2 bleibt solange gesperrt, bis das positive Potential an seiner Basis über R10 und R2 abgebaut ist. Dann wird Ts2 wieder leitend und der Vorgang läuft erneut wie oben beschrieben ab.

Die Dioden D1 und D2 liegen beim Sperren des zugehörigen Transistors in Sperrichtung. Die Kollektorpotentiale sind also unabhängig von den Umladezeitkonstanten für C1 und C2. Die negativen Flanken bleiben steil.

Damit die Erholzeit des Multivibrators (Umladung von C1 und C2 auf definierte Potentiale) nicht zu lang wird, begrenzt der Spannungsteiler R6 R8 und R5 R7 die negative Spannung auf ca. 8V.

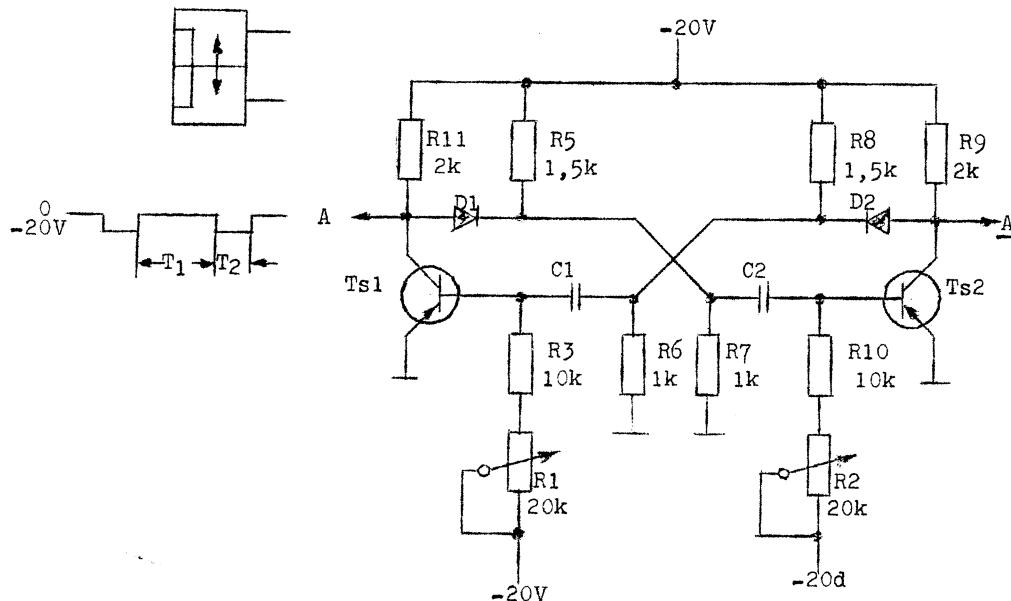


Bild 2.20

### 2.3.7. Translator-Magnet-Treiber

Der Translator-Magnet-Treiber arbeitet ähnlich wie ein Flip Flop. Er muß gesetzt und zurückgesetzt werden.

Sind die Eingänge A und B logisch "1", ist auch der Ausgang T<sub>p</sub> logisch "1".

Der Treiber kann nur durch das Eingangssignal C logisch "0" zurückgesetzt werden.

Im Ruhezustand ist Ts1 leitend, weil wenigstens ein Eingang A oder B negativ ist.

Wenn A oder B (bzw. A und B) negativ sind, ist die Diode D2 leitend, die Basis von Ts1 negativ und Ts1 leitet. Die Basis von Ts2 liegt über den Spannungsteiler Ts1, R7, R6 auf einem positiven Potential von ca. +1,6V. Damit sind Ts2 und Ts3 gesperrt. Über den Spannungsteiler R10, R5, R8 ist D1 mit ca. -4V gesperrt.

Wenn A und B logisch "1" werden, wird Ts1 gesperrt.

Ts2 kommt über den Spannungsteiler R9, R7, R6 zum Leiter. R10 ist wesentlich niederohmiger als R11. Am Emitter von Ts2 stellt sich negatives Potential ein. Damit wird Ts3 leitend. Über den Spannungsteiler R5, R8 und D1 wird D2 gesperrt. Ts1 bleibt gesperrt bis über den Eingang C negatives Potential an die Basis von Ts1 gelegt wird. Der Translator-Magnet-Treiber ist damit wieder zurückgesetzt.

D<sub>3</sub> soll verhindern, daß sich am Emitter von Ts<sub>2</sub> ein positiveres Potential als +1V einstellt (1V reicht zum sperren von Ts<sub>3</sub>), weil sonst Ts<sub>2</sub> nicht über den Spannungsteiler R<sub>7</sub>, R<sub>6</sub> gesperrt werden kann.

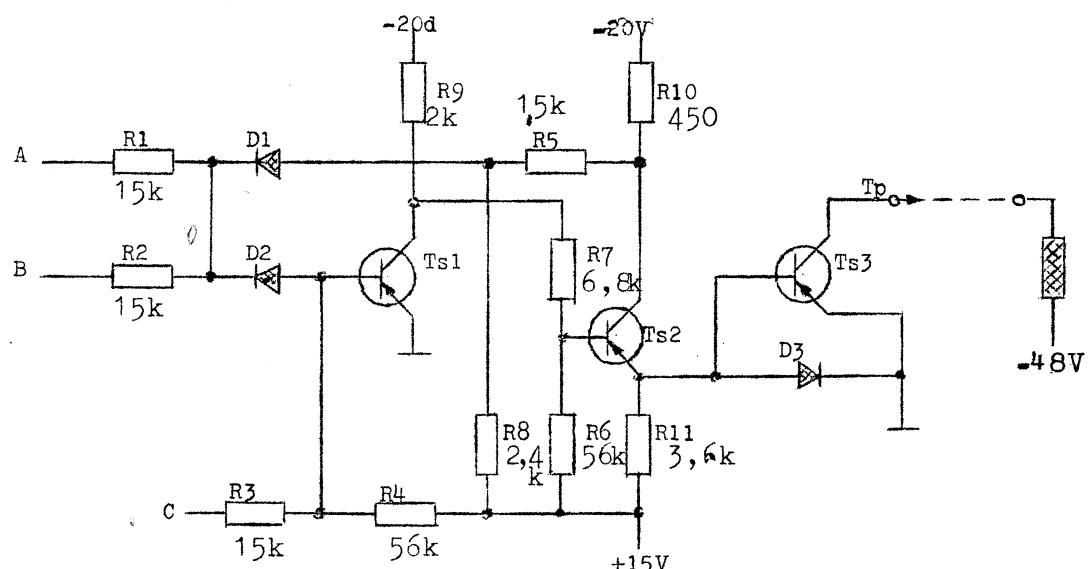


Bild 2.21

#### 2.3.8. Translator-Kupplungs-Treiber

Der Translator-Kupplungs-Treiber dient als Treiber für die Translator-Kupplung.

Die Stufe wird zur Taktzeit CP und Eingang A logisch "1" gesetzt und zur Taktzeit CP und Eingang B logisch "0" zurückgesetzt.

Im Ruhezustand ist Ts<sub>1</sub> über den Spannungsteiler R<sub>11</sub>, R<sub>3</sub>, R<sub>4</sub> leitend und damit Ts<sub>2</sub> sowie Ts<sub>3</sub> über den Spannungsteiler Ts<sub>1</sub>, R<sub>6</sub>, R<sub>9</sub> gesperrt. Wird A OV, ist D<sub>2</sub> noch gesperrt. Die folgende positive Flanke von CP öffnet D<sub>2</sub> und sperrt Ts<sub>1</sub>.

Der negative Sprung an seinem Kollektor koppelt über  $C_3$ ,  $R_6$  auf die Basis von  $T_{s2}$ .  $T_{s2}$  und  $T_{s3}$  werden leitend. Die Stufe verharrt in diesem stabilen Zustand (siehe auch Abschnitt 2.3.1.).

$T_{s2}$  und  $T_{s3}$  werden gesperrt. Der negative Sprung an seinem Kollektor koppelt über  $C_2$ ,  $R_3$  auf die Basis von  $T_{s1}$ , der somit leitend wird. Auch dieser Zustand ist stabil.

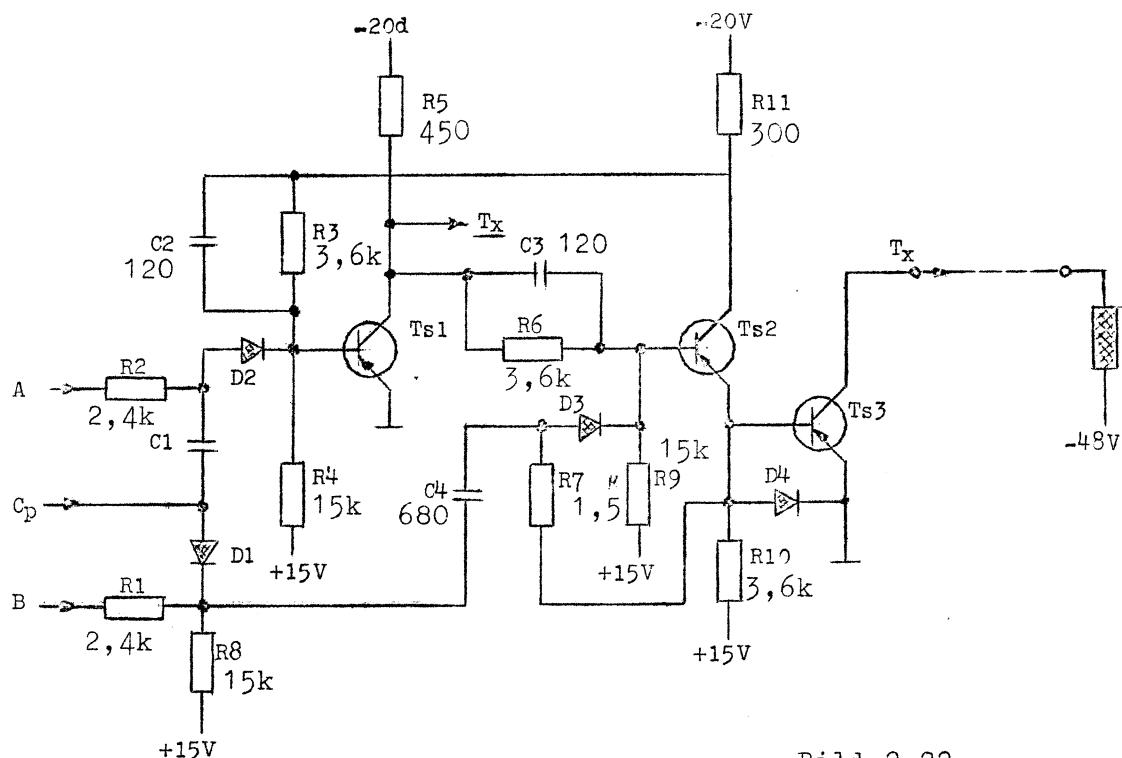


Bild 2.22

### 2.3.9. Leistungs-Treiber

Der Kupplungstreiber ist ein Leistungsverstärker.

Das Eingangssignal wird nicht invertiert. Die Stufe besteht aus Inverter, Emitterfolger, Inverter.

Die Diode D1 und D2 sollen bei leitendem  $T_{s1}$  das Sperren von  $T_{s2}$  gewährleisten. Ist  $T_{s1}$  gesperrt, so ist nur  $R_3$  für die Basisstrombegrenzung von  $T_{s2}$  maßgebend.

D3 soll verhindern, daß sich am Emitter von  $T_{s2}$  ein positiveres Potential als +1V einstellt (1V reicht zum Sperren von  $T_{s3}$ ), weil sonst  $T_{s2}$  nicht über den Spannungsteiler  $R_5$ ,  $R_6$  gesperrt werden könnte.  
(Bild 2.23)

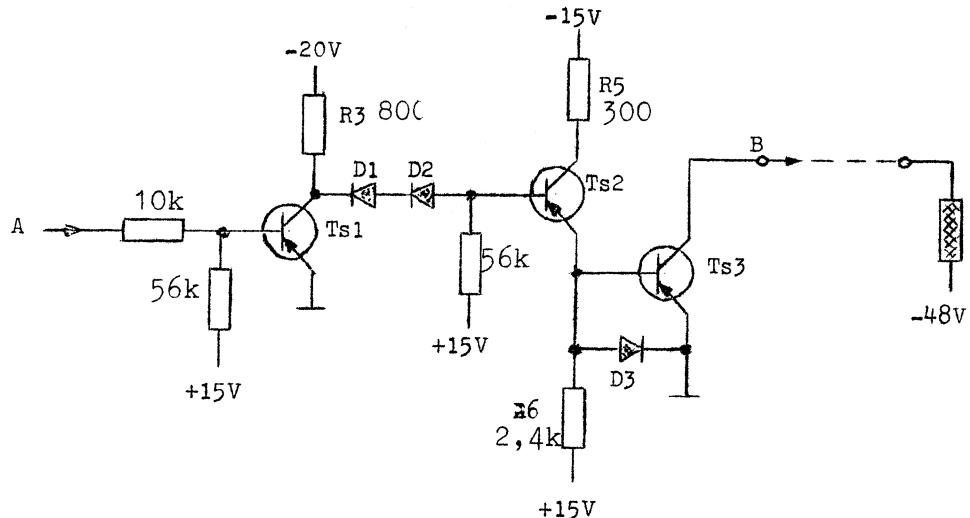


Bild 2.23

### 2.3.10. Treiber

Der Translator-Treiber ist ein Leistungsverstärker.

Der Ausgang C ist nur leitend, wenn gleichzeitig A und B logisch "1" sind.

Die Stufe besteht aus einer Nor-Schaltung und einem Inverter.

Die Diode soll bei leitendem Ts1 ein sicheres Sperrren von Ts2 gewährleisten. Ist Ts1 gesperrt, so ist nur R1 für die Basisstrombegrenzung von Ts2 maßgebend.  
(Bild 2.24)

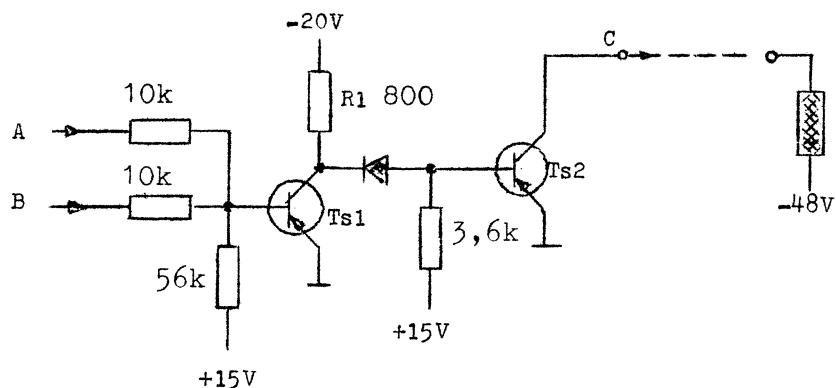


Bild 2.24



## 2.4. Zeitsteuerung

### 2.4.1. Zeitmarkierungen

Die Organisation der Worteinteilung pro Scheibennummerierung erfolgt durch 4 Taktspuren, aus denen durch Und- und Oder-Schaltungen weitere Zeiteinteilungen gewonnen werden (Bild 2.25).

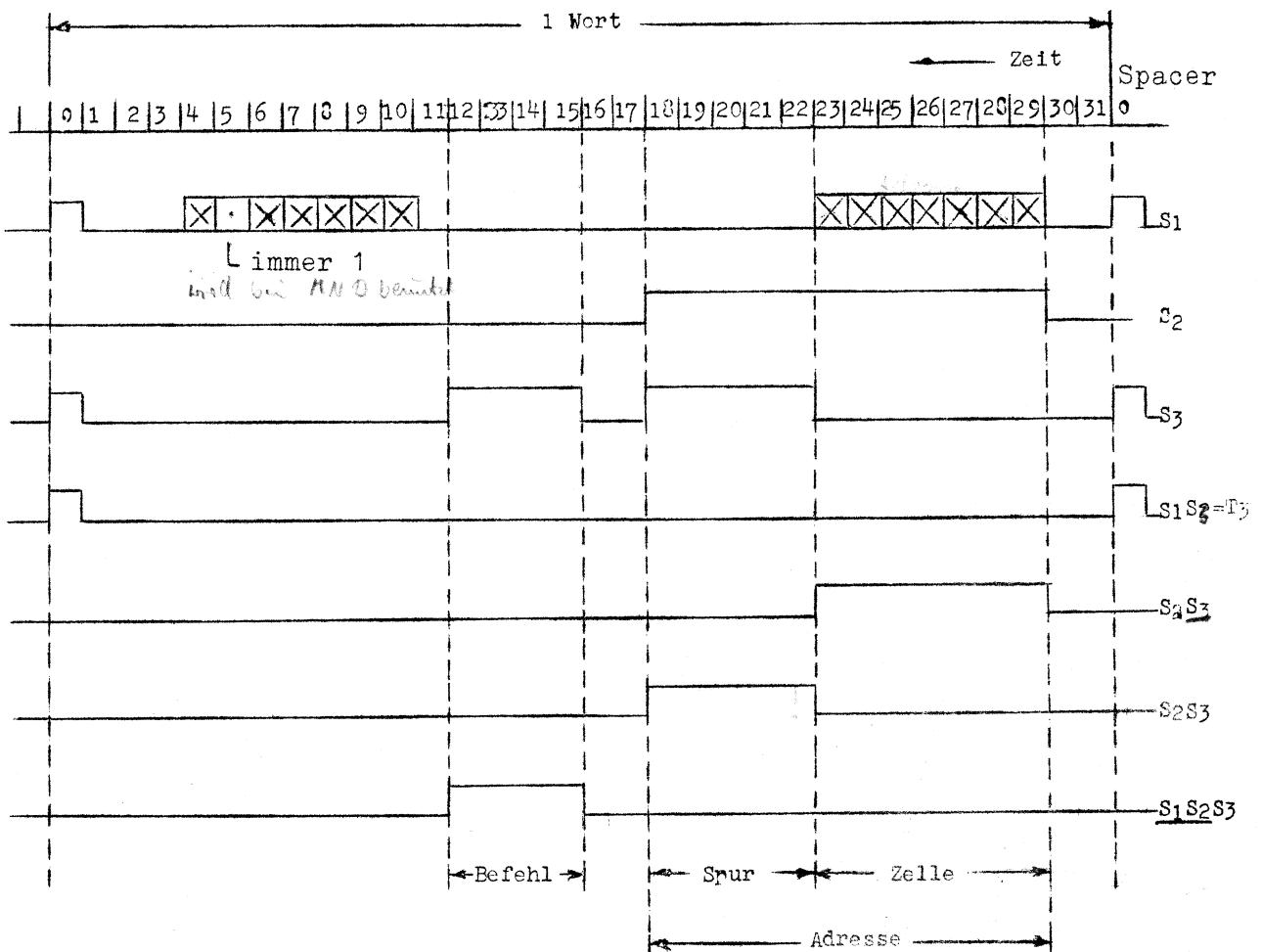


Bild 2.25

Jede Spur ist in 128 Sektoren eingeteilt, wobei die fortlaufende Numerierung der Sektoren in der S1-Spur enthalten ist.

Die optimalen Adressen siehe Optimalisierungsscheibe.

### 2.4.2. Phasen ( $\phi_1 \dots \phi_4$ )

Der Zyklus für die Ausführung eines jeden Befehls hat 4 Phasen ( $\phi_1 \dots \phi_4$ ). In Bild 2.26 sind die Aufgaben der einzelnen Phasen dargestellt.

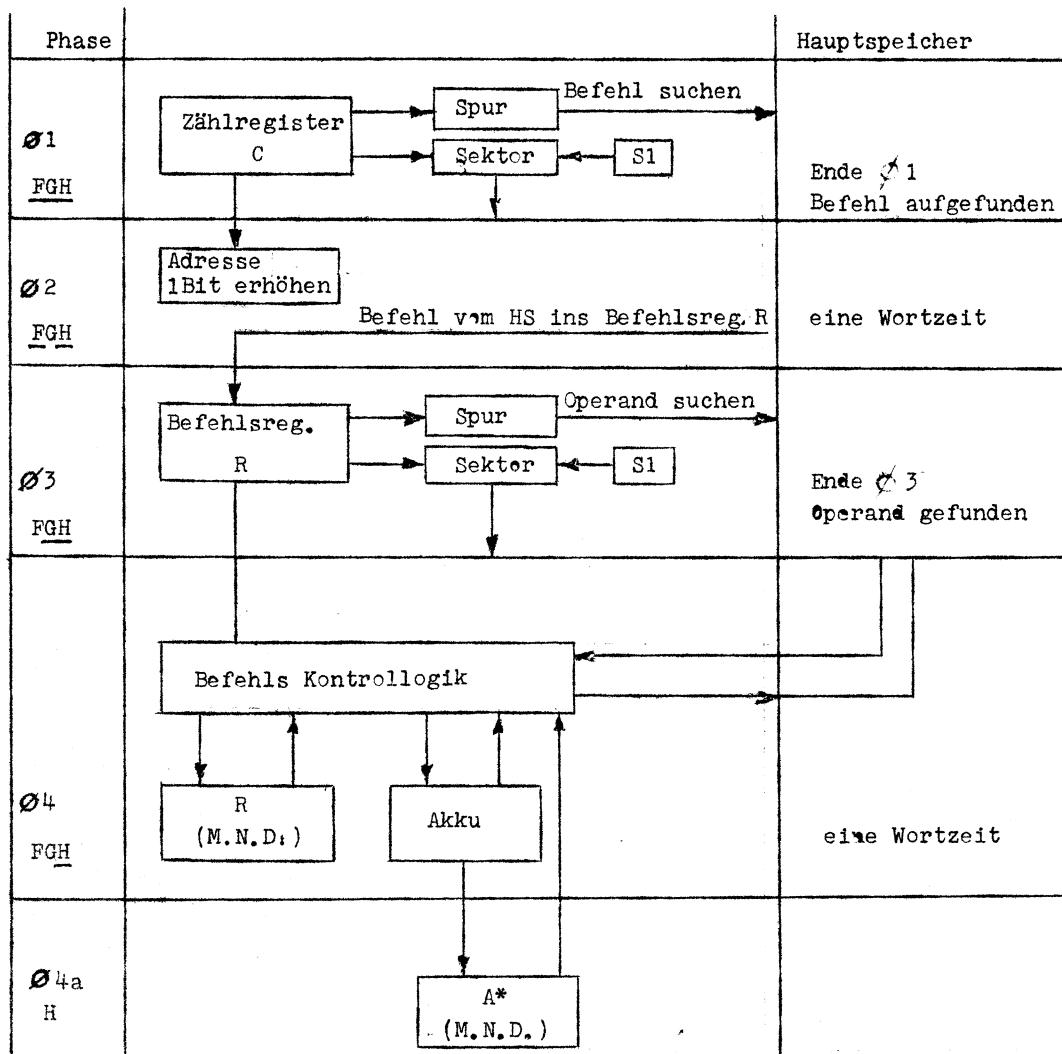


Bild 2.26

Die vier Phasen werden mit den Flip-Flops F, G und H festgelegt. Das H-Flip-Flop ist nur während der verlängerten  $\phi_4$  gesetzt.

$$\phi_1 = \underline{F} \underline{G} \underline{H}$$

$$\phi_2 = \underline{F} \underline{G} \underline{H}$$

$$\phi_3 = F \underline{G} \underline{H}$$

$$\phi_4 = F G \underline{H}$$

$$\phi_{4a} = \dots H$$

Im Normalfall erfolgt der Phasenablauf immer in dem oben angegebenen Zyklus (Besonderheiten s. einzelne Befehle).

$\phi_1$  und  $\phi_3$  dauert bis zum erfolgreichen Vergleich der Sektoradresse (max 1 Scheibenumdrehung).

$\phi_2$  und  $\phi_4$  dauert eine Wortzeit. Bei der Multiplikation und Division wird die  $\phi_4$  verlängert in  $\phi_{4a}$  (Zeitdauer siehe M, N, D - Befehl).

#### 2.4.2.1. Phase 1 ( $\phi_1$ )

In  $\phi_1$  wird der Befehl gesucht, dessen Adresse im C-Register umläuft.

In jeder Wortzeit werden die Bits, die im C-Register während der Sektoradreßzeit S2, S3 erscheinen, mit der Sektornummer, die von S1 gelesen wird, Bit für Bit verglichen.

Bei Übereinstimmung aller 7 Bits ist der nächste Sektor die gewünschte Adresse. Damit wird  $\phi_1$  beendet.

Zum Vergleich wird das K-Flip-Flop benutzt.

K wird zu Beginn einer jeden Wortzeit gesetzt.

Falls auch nur ein Bit von C und S1 während der Adreßzeit S2, S3 nicht übereinstimmen, wird K zurückgesetzt.

Bei Übereinstimmung aller Bits bleibt K über die ganze Wortzeit gesetzt. Damit wird  $\phi_1$  beendet und G gesetzt.

$$K' = T3 \underline{Faf} (F + \dots)$$

└ keine Ein- oder Ausgabe

$$\underline{K'} = \underbrace{G \ H}_{\phi_1(\phi_3)} \underbrace{S2 \ S3}_{\text{Sektoradreßzeit}} \underline{Faf} (S1 \underline{r1} + \underline{S1} r1)$$

$$r1 = \underline{F} \ C + \dots$$

$$S1 \underline{r1} + \underline{S1} r1 = \dots \underline{C} \ S1 + \dots \underline{C} \ S1, \text{ d.h. dieser Aus-}$$

druck ist nur bei Nichtübereinstimmung von C und S1 logisch "1".

$$G' = \underline{G} \ \underline{H} \ T3 \ K \ Q2 \quad (\text{damit beginnt } \phi_2)$$

└ kein stop ...

Um auch die ausgewählte Spur noch vor Beginn der  $\phi_2$  zu speichern, werden die 5 Bits aus dem C-Register während der Spurzeit S2 S3 in die P-Flip-Flops gegeben. Dazu werden die P-Flip-Flops als Shift-Register geschaltet.

Das auch im Shift-Register enthaltene P6 wird nur bei der Ein/Ausgabe benötigt.

Die Bits von C setzen P1 und werden dann eine Bitzeit später nach P2 geshiftet usw.

In P5 steht das meistbedeutende Bit.

$$\begin{aligned} P1' &= i \underline{G} r1 + \dots \\ \underline{P1'} &= i \underline{G} \underline{r1} + \dots \\ &\quad \vdash_{r1} = \underline{F} C \end{aligned}$$

Übernehme C zu jeder Bitzeit

$$\begin{aligned} P2 &= i P1 \\ \underline{P2} &= i \underline{P1} \end{aligned}$$

Übernehme P1 zu jeder Bitzeit

$$\begin{aligned} P3 &= i P2 \\ \underline{P3} &= i \underline{P2} \end{aligned}$$

Übernehme P2 zu jeder Bitzeit

usw. für P4 und P5

$$\begin{aligned} i &= \underbrace{S2 \ S3}_{\text{Spur-}} \ \underbrace{H \ G}_{\text{adreß-}} + \dots \\ &\quad \vdash \phi_1 \text{ und } \phi_3 \\ &\quad \text{zeit} \end{aligned}$$

#### 2.4.2.2. Phase 2 ( $\phi_2$ )

In Phase 2 wird der Befehl, dessen Adresse in  $\phi_1$  gefunden wurde, vom Hauptspeicher ins R-Register geschrieben.

Der Umlauf in R wird unterbrochen und die neue Hauptspeicherinformation V eingegeben. Die Schreibinformation R'w lautet:

$$\begin{aligned} R'w &= \underbrace{F \ G \ H \ V}_{\text{Schreibinformation } \phi_2} + R \ \underbrace{brc}_{\substack{\text{nicht Füllen/Löschen} \\ \text{R-Umlauf in } \phi_1 \text{ und } \phi_3}} (\underline{G} + \dots) + \dots \end{aligned}$$

Der Umlauf in  $\phi_4$  ist ausgeklammert. Er ist von dem jeweiligen Befehl abhängig.

Die Adresse des gerade gefundenen Befehles steht noch im C-Register. Sie wird jetzt, vorbereitend für die nächste  $\phi_1$ , um 1 erhöht.

$$C'w = \underbrace{F \ G \ H}_{\phi_2} \ S2 \ | \ (K \underline{C} + \underline{K} \ C) + \dots$$

Adreßzeit

$$K' = T3 \ \underline{Faf} \ (\underline{F} + \dots)$$

$$\underline{K}' = \underline{F} \ G \ \underline{H} \ S2 \ \underline{Faf} \ \underline{C} + \dots$$

K wird zu Beginn der  $\phi_2$  gesetzt.

K wird zurückgesetzt, sobald eine "0" in der Adresse im C-Register erscheint. Hiermit ist eine Addition um 1 erreicht.

Die neue (Folge-)Adresse läuft in C während  $\phi_3$ ,  $\phi_4$ ,  $\phi_1$  um:

$$C'w = \underbrace{\underline{brc} \ S2 \ \underline{G} \ C}_{\text{Umlauf der Adressen}} + \underbrace{\underline{brc} \ S2 \ F \ C}_{\phi_4} \ | \ (\underbrace{Q1+Q2+Q3+Q4}_{\text{alle Befehle außer U}}) + \dots$$

$\phi_2$  ist nach einer Wortzeit beendet. F wird gesetzt und G zurückgesetzt. Damit beginnt  $\phi_3$ .

$$\underline{G}' = G \ \underline{F} \ T3$$

$$F' = G \ \underline{F} \ \underline{H} \ T3 \ Q1$$

#### 2.4.2.3. Phase 3 ( $\phi_3$ )

In Phase 3 wird die Adresse des im R-Register stehenden Befehles gesucht.

In jeder Wortzeit werden die Bits, die im R-Register während der Sektoradreßzeit S2 S3 erscheinen, mit der Sektornummer, die von S1 gelesen wird, Bit für Bit verglichen.

Bei Übereinstimmung aller 7 Bits ist der nächste Sektor die gewünschte Adresse. Damit wird  $\phi_3$  beendet.

Zum Vergleich wird das K-Flip-Flop benutzt. K wird zu Beginn jeder Wortzeit gesetzt. Falls auch nur ein Bit von R und S1 während der Adresszeit S2 S3 nicht übereinstimmen, wird K zurückgesetzt.

Bei Übereinstimmung aller Bits bleibt K über die ganze Wortzeit gesetzt.

$$K' = T_3 \underline{Faf} (\underline{F} + \dots)$$

└ keine Ein- oder Ausgabe

$$\underline{K'} = \underbrace{G \underline{H}}_{(\phi_1)\phi_3} \underbrace{S2 \underline{S3} \underline{Faf}}_{\text{Sektoradreßzeit}} (S1 \underline{r1} + S1 \underline{r1})$$

$$r1 = F \underline{H} R + \underline{\phi_3}$$

$S1\underline{r1} + S1\underline{r1} = \dots C S1 + \dots C S1$ , d.h. dieser Ausdruck ist nur bei Nichtübereinstimmung von R und S1 logisch "1"

$$G' = \underline{G} \underline{H} T_3 K Q2 \quad (\text{Damit beginnt } \phi_4)$$

└ kein Stop

Um auch die ausgewählte Spur noch vor Beginn der  $\phi_4$  zu speichern, werden die 5 Bits aus dem R-Register während der Spurzeit S2 S3 in die P-Flip-Flops gegeben. Dazu werden die P-Flip-Flops als Shift-Register geschaltet.

Das auch im Shift-Register enthaltene P6 wird nur bei der Ein-Ausgabe benötigt.

Die Bits von R werden nach P1 gegeben und eine Bitzeit später nach P2 geshiftet usw. In P5 steht das meistbedeutende Bit.

$$\begin{aligned} P1' &= i \underline{G} r_1 + \dots \\ \underline{P1'} &= i \underline{G} \underline{r_1} + \dots \quad \text{übernehme R zu jeder Bitzeit} \\ &\quad \underline{\quad} r_1 = F \underline{H} R \end{aligned}$$

$$\begin{aligned} P2' &= i P1 + \dots \\ \underline{P2'} &= i \underline{P1} + \dots \quad \text{übernehme P1 zu jeder Bitzeit} \end{aligned}$$

$$\begin{aligned} P3' &= i P2 + \dots \\ \underline{P3'} &= i \underline{P2} + \dots \quad \text{übernehme P2 zu jeder Bitzeit} \end{aligned}$$

usw. für  $P4'$  und  $P5'$

$$i = S2 \underline{S3} \underline{G} \underline{H} ia + \dots$$

Die vier Befehls-Bits, die in  $\emptyset 2$  gleichzeitig mit der Operandenadresse ins R-Register geschrieben wurden, werden in  $\emptyset 3$  zur Befehlszeit  $S1 \underline{S2} S3$  in die Q-Flip-Flops gesetzt. Dazu werden die Qs als Shift-Register geschaltet.

$$\begin{aligned} Q1' &= \alpha_{11} R + \dots \\ \underline{Q1'} &= \alpha_{11} \underline{R} + \dots \quad \text{übernehme R in die Q-Flip-Flops} \\ Q2' &= \alpha_{11} Q1 + \dots \\ \underline{Q2'} &= \alpha_{11} \underline{Q1} + \dots \quad \text{übernehme Q1 nach Q2} \end{aligned}$$

usw. für  $Q3$  und  $Q4$

$$\alpha_{11} = \underbrace{F \underline{G} \underline{H}}_{\emptyset 3} \underbrace{S1 \underline{S2} S3}_{\text{Befehls-zeit}} \underline{bq} \underline{Faf}$$

keine Ein-/Ausgabe  
keine Eingabe von Hand

#### 2.4.2.4. Phase 4 ( $\emptyset 4$ )

In der Phase 4 wird der Befehl\$ ausgeführt.

Im Folgenden werden nur die für viele Befehle gemeinsamen Ausdrücke erläutert (weitere Einzelheiten s. 2.5.).

Die Ausführung des Befehls ist primär mit dem Akkumulator A verbunden.

Der Inhalt des A läuft in  $\emptyset 1$  bis  $\emptyset 3$  für alle Befehle ohne Änderung um und in  $\emptyset 4$  für die Befehle U, T, H, Y, R, P und Z.

$$Aw' = A \underline{H} \underline{To} \left[ \underline{F} + \underline{G} + Q1 \underline{Q3} \underline{Q4} + \underline{Q2} (Q3+Q4) + \dots \right] + \dots$$

E/A-Taste nicht gedrückt

Umlauf:  $\emptyset 1 \dots \emptyset 3$  P-, H-Befehl U-, T-, Y-, R-, Z-Befehl

Für alle anderen Befehle ist die A-Logik erweitert. Sie wird bei den einzelnen Befehlen besprochen.

$\emptyset 4$  ist für alle Befehle außer M, N und D nach einer Wortzeit beendet, und der Rechner geht wieder in  $\emptyset 1$ . F und G werden zurückgesetzt:

$$\underline{F'} = G \underline{F} \underline{H} \underline{T3}$$

$$\underline{G'} = G \underline{H} \underline{T3}$$

## 2.5. Funktion der arithmetischen Befehle

Die nachfolgende Befehlslogik beschreibt die zur Ausführung der einzelnen Befehle benötigten Funktionen (s. auch Programmieranleitung LGP 21).

### 2.5.1. Z-Befehl

$$Z \triangleq \underline{Q_1} \underline{Q_2} \underline{Q_3} \underline{Q_4} \triangleq 0000$$

Beim Z-Befehl mit der Spuradresse 0000 oder 0100 erfolgt ein Stop (0100 nur bei der Normalausführung, s. auch Modell 81 und Vorrangsteuerung).

Beim Z-Befehl mit der Spuradresse 0200 oder 0300 erfolgt ein normaler Phasenablauf. Der Befehl hat sonst keine Funktion.

Beim Z-Befehl mit der Spuradresse 0300 wird der Zustand der PS-Tasten abgefragt. Bei gedrückter PS-Taste wird der nächste Befehl ausgeführt. Ist eine PS-Taste nicht gedrückt und das zugehörige P-Flip-Flop (Spuradresse) gesetzt, so wird der nächste Befehl übersprungen.

Beim Z-Befehl wird außer der normalen Z-Funktion der Überlauf getestet.

#### 2.5.1.1. Z 0000 und Z 0100

In  $\emptyset 2$  wird der Z-Befehl ins R-Register gebracht. Falls P1 gesetzt ist, erfolgt die Rücksetzung.

$$\underline{P1'} = G \ T3 \ P1 \ P1b$$

$\overbrace{\quad\quad\quad}^{\text{Faf}}$

$\emptyset 3$  ist nach einer Wortzeit beendet, da das Suchen eines Operanden entfällt. G wird gesetzt.

$$G' = \underbrace{F \ G}_{\emptyset 3} \ T3 \ \underbrace{\underline{Q_2} \ \underline{Q_3} \ \underline{Q_4}}_{\text{Z-(P)Befehl}}, \text{ be } \underline{\text{Faf}}$$

Zur Spuradresszeit (S2 S3) der  $\emptyset 3$  wird die Spuradresse des Z-Befehls aus dem R-Register in die P-Flip-Flops geschiftet.

Die Schiftzeit beträgt nur 5 Bitzeiten. Damit wird P6 immer 0 (P1 wurde in  $\phi_1$  zurückgesetzt).  
Der Z 0100 wird in den P-Flip-Flops zum Z 0000.

Nach der folgenden  $\phi_4$  (1 Wortzeit) bleibt der Rechner in  $\phi_1$  stehen. Das G-Flip-Flop wird nicht gesetzt. Das Setzen von G wird durch Q2 entschieden.

$$G' = \underline{G} \underline{H} T3 K Q2 \underline{be} \underline{bs} Ga$$

$\sqcap_{\text{Faf}}$

Zur T3-Zeit der  $\phi_4$  wird Q2 für alle Befehle außer Z 0000 und Z 0100 gesetzt:

$$Q2' = \underbrace{\underline{F} \underline{G} T3 \underline{Q2} \underline{01}}_{\phi_4} \frac{(Q1+Q2+Q3+Q4+P1+P2+P3+P4+P5+P6)}{\text{außer } Z0000, Z0100}$$

kein Step, kein Manuell

Durch Drücken der "Start"-Taste wird der Stop aufgehoben. Q2 wird wieder gesetzt.

$$Q2' = \underbrace{\underline{F} \underline{G} \underline{H}}_{\phi_1} \underline{bs} \quad \sqcap_{\text{"Start"}}$$

#### 2.5.1.2. Z-Befehl mit Spuradressen > 3

Q1 entscheidet in  $\phi_2$  über die Ausführung des Z-Befehls. Ist Q1 in  $\phi_2$  gesetzt, wird der nächste Befehl übersprungen. F kann nicht gesetzt werden. Die folgende  $\phi_3$  wird nicht ausgeführt. Da am Ende von  $\phi_2$  G jedoch wie immer zurückgesetzt wird, läuft der Rechner wieder in  $\phi_1$ . Durch die Adressenerhöhung in  $\phi_2$  ist jetzt ein Befehl übersprungen worden.

$$F' = \underline{F} \underline{G} \underline{H} \underline{Q1} T3 + \dots$$

Q1 wird am Ende von  $\phi_2$  und  $\phi_4$  zurückgesetzt. Damit beginnt wieder ein normaler Phasenablauf.

$$\underline{Q1'} = \underbrace{\underline{G} \underline{H}}_{\phi_2, \phi_4} \underline{Q1} T3 + \dots$$

Die Spuradresse des Befehls steht in  $\phi_2$  in den P-Flip-Flops. Die Setzbedingung für Q1 lautet also:

$$Q1' = \underbrace{G \ H}_{\phi_2(\phi_4)} \underbrace{Q1 Q2 Q3 Q4}_{Z\text{-Befehl}} T3 \underbrace{(P1 Tb1 + P2 Tb2 + P3 Tb3 + P4 Tb4 + \dots)}_{PS\text{-Tasten}} + \dots$$

### 2.5.1.3.-Z-Befehl

Ist bei einem -Z-Befehl durch den vorhergehenden Befehl ein Überlauf vorhanden (das C-Register ist negativ), wird das Überlauf-Bit zurückgesetzt und der nächstfolgende Befehl ausgeführt.

Ist kein Überlauf vorhanden, wird der nächste Befehl übersprungen, d.h. Q1 wird in  $\phi_2$  gesetzt (weiterer Ablauf s. 2.5.1.2.).

$$Q1' = \underbrace{G \ H}_{\phi_2(\phi_4)} T3 \underbrace{Q1 \ Q2 \ Q3 \ Q4}_{Z\text{-Befehl}} (R \ C + \dots) + \dots$$

kein Überlauf  
entschlüsselt -

Bei der Ausführung des A- oder S-Befehles werden Teile der Addierlogik zur Erkennung des Überlaufes zur Vorzeichenzeit (T3) benutzt. Das C-Register wird entsprechend gesetzt.

$$Cw' = \underbrace{F \ G}_{\phi_4} \underbrace{Q1 Q2 Q3}_{A \text{ oder } S} T3 \underbrace{S}_{A\text{-Befehl}} (I1 I2 L + I1 I2 L) + S(I1 I2 L + I1 I2 L) + \dots$$

Vorzeichenzeit

Bei der Ausführung der Division kann, wenn der Dividend größer als der Divisor ist, ein Überlauf während der zweiten Wortzeit der  $\phi_4$  stattfinden.

Da die Division ein logischer Ablauf von A- und S-Befehlen ist, werden die Vorzeichen des Dividenden und des Teilquotienten zur Erkennung des Überlaufes benutzt.

Das Vorzeichen des Teilquotienten wird während der zweiten Wortzeit der  $\emptyset 4$  in A gespeichert und das des Dividenden in P6. Damit wird das C-Register beim Überlauf durch folgenden Ausdruck gesetzt:

$$Cw' = \underbrace{F \ G \ H}_{\substack{2te \ Wortzeit \\ \emptyset 4a}} \underbrace{P1 \ T3 \ Q3}_{\substack{| \\ Divi- \\ sion}} \underbrace{(P6 \ A + P6 \ A)}_{\substack{| \\ \text{Überlauf} \\ \text{Vorzeichen}}}$$

Folgt dem Überlauf ein -Z-Befehl, wird das Überlauf-Bit zurückgesetzt.

Die Zurücksetzung erfolgt durch Unterbrechung des Umlaufes im C-Register in  $\emptyset 4$ .

$$Cw' = \underbrace{G \ S2 \ brc \ C}_{\substack{\text{Umlauf bei} \\ \text{-Befehl in} \\ \emptyset 4 \ \text{unter-} \\ \text{brochen}}} \underbrace{(R+F+T3+Q1+Q2+Q3+Q4+\dots)}_{\substack{| \\ \text{Umlauf bei je-} \\ \text{dem Befehl au-} \\ \text{ßer Z} \\ | \\ -\text{Verhindert Übernahme des Vor-} \\ \text{zeichens aus dem R-Register} \\ \text{beim U- oder T-Befehl}}}_{\substack{| \\ \text{Umlauf des} \\ \text{Überlauf-Bit} \\ \text{in } \emptyset 1 \ \text{u. } \emptyset 3}}$$

### 2.5.2. B-Befehl

$$B \stackrel{\Delta}{=} \underline{Q_1} \underline{Q_2} \underline{Q_3} Q_4 \stackrel{\Delta}{=} 0001$$

Mit dem B-Befehl wird der Inhalt einer Hauptspeicherzelle (V) in den Akkumulator (A) gebracht ( $V \rightarrow A$ ).

Der Ablauf der Phasen 1...3 siehe Abschnitt 2.4.2.

Zu Beginn der  $\phi_4$  wird der Umlauf von A unterbrochen und das vom Hauptspeicher gelesene Wort V in den Akkumulator geschrieben.

$$\begin{aligned} Aw' &= F \underline{G} \underline{H} \quad \underline{Q_1} \underline{Q_2} \underline{Q_3} Q_4 \quad V + \dots \\ &\quad \phi_4 \quad \text{B-Befehl} \end{aligned}$$

Der neue Inhalt läuft in A um.

$$\begin{aligned} Aw' &= (\underline{F} + \underline{G}) \underline{H} \underline{T_0} A + \dots \\ &\quad \phi_1, \phi_2 \quad \phi_3 \quad \begin{array}{|c|c|} \hline & \text{Umlauf} \\ \hline \end{array} \\ &\quad \quad \quad \text{-- E/A-Schalter nicht gedrückt} \end{aligned}$$

Am Ende von  $\phi_4$  werden F und G zurückgesetzt.

$$\underline{F}' = F \underline{G} \underline{H} T_3 + \dots$$

$$\underline{G}' = G \underline{H} T_3 + \dots$$

### 2.5.3. Y-Befehl

$$Y \triangleq \underline{Q_1} \underline{Q_2} \underline{Q_3} \underline{Q_4} \triangleq 0010$$

Durch den Y-Befehl wird bei einer Hauptspeicherzelle der Adreßteil mit der entsprechenden Bitkombination des Akkumulators (Adreßteil) überschrieben. Der weitere Inhalt der Hauptspeicherzelle wird nicht verändert.

Die Schreiberlaubnis W erfolgt zur Adreßzeit in  $\emptyset 4$ .

$$\begin{array}{ll} W = F G & \underline{\underline{Q_1 Q_2 Q_3}} S_2 + \dots \\ & \quad \quad \quad \text{--- Adreßzeit} \\ \emptyset 4 & Y(R)-Befehl \end{array}$$

Die Schreibinformation  $Vw'$  wird aus dem Akkumulator geholt.

$$\begin{array}{ll} Vw' = L & (\underline{Q_4} A + \dots) \\ & \quad \quad \quad \text{--- kein R-Befehl} \end{array}$$

L wird zu Beginn von  $\emptyset 4$  gesetzt.

$$\begin{array}{ll} L' = F G & \underline{Q_2} + \dots \\ & \quad \quad \quad \text{--- auch beim Y-Befehl} \end{array}$$

#### 2.5.4. R-Befehl

$$R \hat{=} Q_1 \underline{Q_2} Q_3 Q_4 \hat{=} 0011$$

Mit dem R-Befehl wird zur Adresse im C-Register eine "1" addiert. Die neue Adresse wird in den Hauptspeicher gebracht.

Die Schreiberlaubnis W erfolgt wie beim Y-Befehl (2.5.3.).

$$W = F G \underline{Q_1} \underline{Q_2} Q_3 S_2 + \dots$$

Die Schreibinformation wird aus dem C-Register geholt.

$$Vw' = L \underline{Q_1} \underline{Q_4} (\underbrace{K \underline{C} + K \underline{C} + \dots})$$

R-Befehl Addition einer "1" zur Adresse

K wird am Ende einer jeden Wortzeit gesetzt. Es wird zurückgesetzt sobald eine "0" in der Adresse in C erscheint.

Hiermit ist eine Addition von 1 zur letzten Stelle erreicht.

$$\begin{matrix} \underline{K}' & = & \underline{G} \underline{H} & S_2 \underline{C} \underline{Faf} & (Q_3 + \dots) + \dots \\ & & \emptyset 2, \emptyset 4 & & \end{matrix}$$

R-Befehl

#### 2.5.5. D-Befehl

$$D \triangleq Q_1 \ Q_2 \ Q_3 \ Q_4 \triangleq 0101$$

Beim D-Befehl wird der Inhalt des Akkumulators (Dividend) durch den Inhalt einer Hauptspeicherzelle (Divisor) dividiert. Der gerundete Quotient steht im Akkumulator.

Der LGP 21 ist eine Festkommamaschine. Das Maschinenkomma folgt der Vorzeichenstelle. Damit ergibt sich, daß in der Maschinensprache Zahlen  $\geq 1$  nicht darstellbar sind. Der Zahlenbereich in der Maschinensprache reicht von -1 bis +0,999 ( $\approx +1$ ).

Jeder Divisionsbefehl wird im Rechner als Festkommaoperation ausgeführt.

Für jede Division ist Voraussetzung, daß der absolute Inhalt des Divisors (Hauptspeicherzelle) größer als der absolute Inhalt des Dividenden (Akkumulator) ist, damit der Quotient  $< 1$  wird. Durch entsprechende Wahl des gedachten Kommas von Divisor und Dividend kann dieses erreicht werden (Differenz von Maschinenkomma zu gedachtem Komma).

Bei der Division von Zahlen mit beliebigem Vorzeichen ergeben sich die im folgenden aufgeführten Möglichkeiten. Negative Zahlen werden als Komplement zur Konstanten C = 2 dargestellt (z.B. 2-a). Die hierdurch notwendigen Korrekturen werden im Divisionsablauf berücksichtigt (s. Rint, IV. Band ab S. 665).

$$\begin{array}{ccc} a : b & \longrightarrow & a : b \\ (-a) : b & \longrightarrow & (2-a) : b \\ a : (-b) & \longrightarrow & a : (2-b) \\ (-a) : (-b) & \longrightarrow & (2-a) : (2-b) \end{array}$$

Bei der Division wechseln Subtraktion des Divisors und Schiften des laufenden Restes um 1 Bit miteinander ab. Der Divisor wird zunächst vom Dividenden, bei den folgenden Subtraktionen dann vom Rest abgezogen. Das Vorzeichen eines jeden Restwertes wird weggespeichert und

ist für den Wert einer entsprechenden Ergebnisstelle maßgebend. Die Subtraktion wird zur Addition, wenn Divisor und der vorhergehende Restwert ungleiches Vorzeichen haben. Die Ergebnisbildung geschieht am Ende der Division aus den gespeicherten Vorzeichen der Restwerte. Gegebenenfalls wird auch die Aufrundung vorgenommen.

Die Division dauert 64 Wortzeiten (32 x Subtraktion, 32 x Schiften) plus 3 weitere Wortzeiten. Ø4 des D-Befehls geht durch Setzen des H-Flip-Flops in Ø4a über. F und G dienen der Unterscheidung einzelner Wortzeitgruppen am Anfang und Ende der Division. Das abwechselnde Subtrahieren und Schiften wird mit dem P1-Flip-Flop gesteuert.

Aus dem Vorzeichen des Restes wird in der 67. Wortzeit das Vorzeichen des Quotienten ermittelt. Der Rest wird in der 3. Wortzeit aus den Bit-Positionen 1-32 in die Positionen 33-64 geschiftet. Dazu wird A\* (2 Wort plus 1 Bit) für die Dauer von 32 Takten als Umlaufregister benutzt. Durch das Vorhandensein der 65. Bitstelle erscheint der bei Pos. 33-64 stehende Restwert in der folgenden Subtraktionswortzeit gegenüber dem Divisor (R) um 1 Bit verspätet. Das 65. Bit ist eine rechtsseitige Erweiterung des Restwertes und muß darum immer Null sein. Diese Forderung wird erfüllt, wenn zwei Wortzeiten vorher an der 64 Bit weiter links befindlichen Stelle (Vorzeichen) immer eine Null geschrieben steht. Dazu wird in sämtlichen ungeradzahligen Shiftwortzeiten aus dem Schreibsignal Aw' das Vorzeichen ausgeblendet.

Bevor nach einer Shiftwortzeit die nächste Subtraktionswortzeit beginnt, nimmt das P6-Flip-Flop das Vorzeichen des vorhergehenden Restes auf.

Dieses ist zur Vorzeichenzeit T3 der Shiftwortzeit im Akku A in der Position 32 angekommen. Es kann gelesen werden.

Von der 4. Wortzeit ab findet in allen geradzahligen

Subtraktionswortzeiten eine arithmetische Verknüpfung von Divisor und geschiftetem Restwert statt. Über Subtraktion oder Addition entscheidet das ins P6-Flip-Flop übernommene Vorzeichen des zuletzt ermittelten Restwertes. Der geschiftete Restwert wird aus A\* gelesen. Während der Einschreibung eines neu berechneten Restwertes in den Akku, erreicht das Vorzeichen des vorher berechneten Restwertes zur T3-Zeit die Bitposition 65. Hier steht es jetzt als Ergebnisstelle rechts neben den schon vorhandenen Stellen.

Nach dem Einschreiben folgt mit Ausnahme der 66. Wortzeit wieder eine Schiftwortzeit. Nach der 66. Wortzeit erfolgt die Ergebnisbildung.

Die im folgenden beschriebenen Funktionen können zweckmäßigerweise parallel zum Text auch in der Divisionstabellen (s. S.A2-59) und im Blockschaltbild verfolgt werden.

Die Stellung des P1-Flip-Flops ist in  $\emptyset_4$  beliebig. Es wird jedoch am Ende  $\emptyset_4$  zurückgesetzt. In  $\emptyset_4a$  dient P1 zur Unterscheidung der geradzahligen und ungeradzahligen Wortzeiten, ausgenommen der 67. Wortzeit.

In allen geradzahligen Wortzeiten ist P1 = 0. Es wird subtrahiert. In allen ungeradzahligen ist P1 = 1. Es wird geschiftet (ausgenommen 1. und 67. Wortzeit).

$$\underline{P1'} = G \underline{T3} \underline{P1} \underline{P1b} + H \underline{P1} \underline{T3}$$

$$| \emptyset_4 \quad | \underline{Faf} \quad | \emptyset_4a$$

$$\underline{P1'} = H \underline{P1} \underline{T3}$$

$$| \emptyset_4a$$

Damit der Divisor (V) zu jeder zweiten Wortzeit für die Subtraktion verfügbar ist, wird er in der normalen Phase 4 ins R-Register geschrieben.

$$\underline{Rw'} = \underline{G} \underline{H} \underline{Q1} \underline{Q2} \underline{Q4} \quad V$$

$$| \emptyset_4 \quad | D\text{-Befehl$$

Das Vorzeichen des Divisors (V) wird für die gesamte Dauer der  $\emptyset 4a$ , wie bei der Multiplikation, im P5-Flip-Flop gespeichert.

$$P5' = \underbrace{GH}_{WZ1} \quad \underbrace{T3}_{\text{Vorzeichen}} \quad \underbrace{P5a}_{\substack{\text{Multiplikand/Divisor} \\ \underline{Faf}}} \quad V$$

$$\underline{P5'} = \underbrace{GH}_{WZ1} \quad \underbrace{T3}_{\text{Vorzeichen}} \quad \underbrace{P5a}_{\text{Multiplikand/Divisor}} \quad \underline{V}$$

In P6 wird am Ende von  $\emptyset 4$  das Vorzeichen des Dividenden (A) gespeichert und in  $\emptyset 4a$  vor Beginn einer jeden Subtraktionswortzeit das Vorzeichen des laufenden Restes.

Der Inhalt von P6 bleibt jeweils zwei Wortzeiten lang erhalten.

$$P6' = \underbrace{FGH}_{\emptyset 4} \cdot \underbrace{P6a}_{\substack{\text{Vorzeich.} \\ \text{Dividend}}} \quad \underbrace{T3 \cdot A + H}_{\substack{\emptyset 4a \\ \text{D-Bef.}}} \quad \underbrace{P1 \quad \underline{Q3} \quad T3 \quad A}_{\substack{\text{Restwert} \\ \text{Vorzeichen}}}$$

$$\underline{P6'} = \underbrace{FGH}_{\emptyset 4} \quad \underbrace{P6a}_{\emptyset 4a} \quad \underbrace{T3 \cdot A + H}_{\emptyset 4a} \quad \underbrace{P1 \quad \underline{Q3} \quad T3 \quad A}_{\text{Vorzeichen}}$$

Das Vorzeichen des Dividenden (A) wird in der  $\emptyset 4$  (1. Wortzeit) ausgeblendet.

$$Aw' = \underbrace{H \quad A \quad To}_{\substack{\emptyset 4 \\ \text{Umlauf}}} \left[ \underbrace{T3 \quad \underline{Q1} \quad Q2}_{\substack{\text{D-Befehl} \\ \text{Vorzeichen-Ausblendung}}} \quad (Q4 + \dots) + \dots \right]$$

Während der weiteren Rechnung ( $\emptyset 4a$ ) geht der Rest dem Betrag nach gegen Null, da bis zur 66. Wortzeit subtrahiert bzw. addiert wird. Die Entscheidung über Subtraktion oder Addition erfolgt durch S. Es wird immer subtrahiert, wenn P5 und P6 gleich sind.

$$S = \underbrace{H \quad \underline{Q3}}_{\emptyset 4a} \quad \underbrace{(P5 \quad \underline{P6} + P5 \quad P6)}_{\substack{\text{Gleichheit} \\ \text{Division}}}$$

In der zweiten Wortzeit wird vom Dividenden (A) entsprechend der Stellung von S der Divisor (V) subtrahiert oder addiert.

$$I_1 = \underbrace{F \underline{G} \underline{P_1} \underline{Q_3}}_{\begin{array}{l} | \\ | \\ | \\ | \\ \text{2. Wortzeit } \emptyset 4a \end{array}} (A + \underbrace{\underline{P_6} \underline{T_3}}_{\begin{array}{l} | \\ | \\ \text{Vorzeichen des Dividenden} \\ \text{Dividend} \\ \text{-D-Befehl} \end{array}}) \quad (A)$$

$$I_2 = H \underbrace{\underline{Q_3} \underline{P_1}}_{\begin{array}{l} | \\ | \\ \emptyset 4a \end{array}} R + \dots \quad \begin{array}{l} | \\ | \\ \text{Divisor} \\ \text{-geradzahlige Wortzeit} \\ \text{Division} \end{array}$$

Ein entstehender Übertrag wird im L-Flip-Flop gespeichert.

$$L' = (\underline{T_3} + \dots) (\underbrace{S \underline{I_1} \underline{I_2} \underline{L}}_{\begin{array}{l} | \\ | \\ | \\ \text{Addition} \end{array}} + \underbrace{S \underline{I_1} \underline{I_2} \underline{L}}_{\begin{array}{l} | \\ | \\ | \\ \text{Subtraktion} \end{array}}) + \dots$$

$$\underline{L'} = \underbrace{\underline{T_3} \underline{Q_3}}_{\begin{array}{l} | \\ | \\ \text{Anfang jeder D-Bef.} \\ \text{Wortzeit beim} \\ \text{D(A,S)-Befehl} \end{array}} + \underbrace{\underline{Q_2} \underline{Q_1}}_{\begin{array}{l} | \\ | \\ \text{Subtraktion} \end{array}} (\underbrace{S \underline{I_1} \underline{I_2} \underline{L}}_{\begin{array}{l} | \\ | \\ | \\ \text{Addition} \end{array}} + \underbrace{S \underline{I_1} \underline{I_2} \underline{L}}_{\begin{array}{l} | \\ | \\ | \\ \text{Addition} \end{array}}) + \dots$$

Das Ergebnis wird nach A und A\* geschrieben.

$$Aw' = A^*w' = H \underbrace{\underline{P_1}}_{\begin{array}{l} | \\ | \\ | \\ \text{2. Wortzeit} \\ \emptyset 4a \end{array}} (\underbrace{L \underline{I_1} \underline{I_2} + \underline{L} \underline{I_1} \underline{I_2} + \underline{L} \underline{I_1} \underline{I_2} + L \underline{I_1} \underline{I_2}}_{\begin{array}{l} | \\ | \\ | \\ | \\ \text{Addierlogik} \end{array}})$$

In allen ungeradzähligen Wortzeiten, von der 3. bis zur 65. Wortzeit, wird der Rest der in der vorderen Hälfte von A\* (auch in A) steht, in die zweite Hälfte von A\* geschiftet (um 32 Bit).

$$I_1 = \underbrace{G \underline{H} \underline{P_1} \underline{Q_3}}_{\begin{array}{l} | \\ | \\ | \\ | \\ \text{3.,5...63. Wortzeit} \end{array}} A^* + F \underbrace{H \underline{Q_3}}_{\begin{array}{l} | \\ | \\ \text{Division} \end{array}} A^* \quad 65.,66. \text{Wortzeit}$$

$$I_2 = \underbrace{H \underline{P_1} \underline{Q_3}}_{\begin{array}{l} | \\ | \\ \text{nicht ungeradzahlig} \end{array}} R = "0"$$

$$Aw' = H \underbrace{T_3}_{\begin{array}{c} | \\ \text{Umlauf von } A^* \\ \text{Ausblendung des} \\ \text{Vorzeichens} \end{array}} (\underbrace{L \ I_1 \ I_2}_{\text{I1}} + \underbrace{L \ I_1 \ I_2}_{\text{I2}} + \underbrace{L \ I_1 \ I_2}_{\text{I1}} + \underbrace{L \ I_1 \ I_2}_{\text{I2}})$$

Während aller weiteren geradzahligen Wortperioden wird von dem in der zweiten Hälfte von  $A^*$  stehenden Rest der Divisor ( $V \triangleq R$ ) subtrahiert bzw. addiert. Das Ergebnis wird nach A und  $A^*$  geschrieben.

Dadurch, daß  $A^*$  65 Bit lang ist, erscheint der Rest in der Subtraktionswortzeit gegenüber dem Divisor ( $R$ ) um 1 Bit verspätet.

$$I_1 = \underbrace{F \ G \ H}_{4.6 \dots 64. \text{Wortzeit}} A^* + \dots$$

$$I_2 = H \underbrace{P_1 \ Q_3}_{\text{geradzahlige Wortzeit}} R + \dots$$

S, L, A-Logik wie in der zweiten Wortzeit.

Mit dem Ende der 66. Wortzeit steht der tatsächliche oder invertierte ganze Quotient im Akkumulator ( $Aw'$ -Logik wie bei der 3., 5....63. Wortzeit).

Das Vorzeichen des Divisors ( $V \triangleq R$ ) gibt an, ob der Quotient noch invertiert werden muß. Ist der Divisor positiv ( $P_5 = 1$ ), wird invertiert.

Der Quotient wird gerundet, wenn das Vorzeichen von Rest ( $P_6$ ) und Divisor ( $P_5$ ) übereinstimmen. Die Rundung erfolgt durch Subtraktion einer 1 von jeder Bitstelle; dieses entspricht der Addition einer 1 zur wenigstbedeutenden Stelle.

Invertierung und Rundung erfolgen in der 67. Wortzeit.

$$I_1 = \underbrace{F \ G \ H}_{67. \text{Wort-zeit}} \underbrace{A^* \ P_5}_{\text{Divisor +}} + \underbrace{G \ H \ Q_3}_{65..67. \text{Wort-zeit}} \underbrace{P_5 \ A^*}_{\text{Divisor -}} + \dots$$

$$I_2 = \underbrace{F \quad G \quad H}_{\text{67. Wortzeit}} (\underbrace{P_5 \quad P_6 + P_5 \quad P_6}_{\text{Subtraktion einer 1}}) + \dots$$

67. Wortzeit Subtraktion einer 1

$$S = \underbrace{F \quad G \quad H}_{\text{67. Wortzeit}} + \dots$$

67. Wortzeit

$$Aw' = \underbrace{F \quad G \quad H}_{\text{67. Wortzeit}} (\underbrace{L \quad I_1 \quad I_2 + L \quad I_1 \quad I_2 + L \quad I_1 \quad I_2 + L \quad I_1 \quad I_2}_{\text{Addierlogik}})$$

Addierlogik

Fall (+a) : (+b)

Sind Divisor und Dividend positiv ( $P_5 = 0$ ) und teilbar ohne bleibenden Rest ( $P_6 = 1$ ), so wird

$$I_1 = (P_5 \quad A^* + \underbrace{P_5 \quad A^*}_{\begin{matrix} 0 \\ 1 \end{matrix}}) \stackrel{\Delta}{=} \underline{A^*}$$

$$I_2 = (P_5 \quad P_6 + \underbrace{P_5 \quad P_6}_{\begin{matrix} 0 \\ 1 \end{matrix}}) \stackrel{\Delta}{=} 0$$

$$S = \underbrace{F \quad G \quad H}_{\text{und damit}} + \stackrel{\Delta}{=} 1$$

$$\text{und damit } Aw' \stackrel{\Delta}{=} \underline{A^*}$$

Ist eine Aufrundung aufgrund eines bleibenden Restes erforderlich ( $P_6 = 0$ ), so wird

$$I_2 = (P_5 \quad P_6 + \underbrace{P_5 \quad P_6}_{\begin{matrix} 0 \\ 0 \end{matrix}}) \stackrel{\Delta}{=} 1$$

Subtraktion von 1,111...1 ist gleichbedeutend mit Addition des entsprechenden Komplements (0,000...1).

Fall (-a) : b

Das negative Vorzeichen des Dividienden geht am Ende der normalen Phase 4 in das  $P_6$ -Flip Flop. In der 2. Wortzeit wird dadurch das Signal  $S = 0$ , so daß die Division mit einer Addition anstelle einer Subtraktion beginnt. Wegen der vorausgesetzten Größenverhältnisse muß der Rest positiv werden.

Aus seiner Null in der Vorzeichenstelle wird in der 67. Wortzeit das Vorzeichen des Quotienten "1" ( $Aw' \hat{=} A^*$ ). Außer in der 2. Wortzeit hat das negative Vorzeichen des Dividenden keinen direkten Einfluß auf den Divisionsablauf. Nur indirekt pflanzt sich seine Wirkung über folgende Kette fort:

Signal S (Wortzeit 2) - neues Restwertvorzeichen - P6-Flip-Flop - Signal S (Wortzeit 4) - neues Restwertvorzeichen usw.

Die Ergebnisbildung in der 67. Wortzeit geschieht wie bei positivem Dividenden, der Quotient ist negativ und steht als Komplement im Akkumulator.

#### Fall a : (-b)

Das negative Vorzeichen des Divisors wird am Ende der normalen Phase 4 vom P5-Flip Flop übernommen und darin bis zum Divisionsende gehalten. Dadurch wird in der 2. Wortzeit das Signal S = 0. Die Division beginnt also mit einer Addition ( $a+(-b)$ ), wobei das Vorzeichen des Restwertes wegen der vorausgesetzten Größenverhältnisse negativ werden muß. In den weiteren Wortzeiten wird jetzt subtrahiert, wo unter den gleichen Bedingungen ( $P6=1$ ) bei positivem Divisor addiert werden wäre.

Sind Dividend und Divisor ohne bleibenden Rest teilbar, so wird in der 67. Wortzeit

$$I1 = \left( \begin{array}{c} P_5 \\ 1 \end{array} \right) A^* + \left( \begin{array}{c} P_5 \\ 0 \end{array} \right) \hat{=} A^* \quad \text{und mit } P6=1 \text{ auch}$$

$$I2 = \left( \begin{array}{cc} P_5 & P_6 \\ 1 & 1 \end{array} \right) + \left( \begin{array}{cc} P_5 & P_6 \\ 0 & 0 \end{array} \right) \hat{=} 1$$

und damit  $Aw' = A^* - 1,111\dots 1$

Das in diesem Beispiel erforderliche negative Ergebnis erhält man aus dem Vorzeichen des Restwertes der 2. Wortzeit, der Bedingung  $I1 = A^*$  in der 67. Wortzeit und der Addition einer "1" zur letzten Stelle. Durch die Aufrundung des Ergebnisses erhöht sich die letzte Stelle des vorhandenen Komplements um 1, jedoch wird dadurch der

Betrag der negativen Zahl in der letzten Stelle um 1 vermindert.

Ist bei der angenommenen Division eine Aufrundung aufgrund eines bleibenden Restes erforderlich, so wird die Subtraktion einer 1 von der letzten Stelle des Betrages der negativen Zahl unterlassen. Hierfür wird mit

$$P_6 = 0 \text{ auch } I_2 = \left( \begin{array}{c} P_5 \\ \cdot \\ 1 \end{array} \begin{array}{c} P_6 \\ \cdot \\ 0 \end{array} + \begin{array}{c} P_5 \\ \cdot \\ 0 \end{array} \begin{array}{c} P_6 \\ \cdot \\ 1 \end{array} \right) \hat{=} 0$$

Fall (-a) : (-b)

Die negativen Vorzeichen werden wieder in der 1. Wortzeit in die Flip Flops  $P_6$  und  $P_5$  gespeichert. In der 2. Wortzeit wird das Signal  $S = 1$ . Die Division beginnt wieder normal mit einer Subtraktion  $(-a)-(-b)$ . Wegen der vorausgesetzten Größenverhältnisse muß der Restwert positiv werden. Aus seiner Null in der Vorzeichenstelle wird in der 67. Wortzeit das Vorzeichen des Quotienten zu Null ermittelt ( $Aw' \hat{=} A^*$ ).

Das negative Vorzeichen des Dividenden übt einen direkten Einfluß auf den Divisionsablauf nur in der 2. Wortzeit aus. In den weiteren Wortzeiten wird wegen des negativen Divisors ( $P_5=1$ ) jetzt subtrahiert, wo unter den gleichen Bedingungen ( $P_6=1$ ) bei positivem Divisor addiert worden wäre. Die Ergebnisbildung läuft wie im "Fall a : (-b)" beschrieben ab.

### Wortzeitgruppen in Phase $\phi_4$ und $\phi_{4a}$

F und G werden in der  $\phi_{4a}$  zur Unterscheidung der einzelnen Wortzeitgruppen benutzt.

Am Ende von  $\phi_4$  werden F und G für zwei Wortzeiten zurückgesetzt.

$$\begin{array}{lcl} \underline{F}' = F \underline{G} \underline{H} T3 & \longrightarrow & \underline{F} \underline{G} H = 2. \text{ und } 3. \text{ Wort-} \\ \underline{G}' = G \underline{H} T3 & & \text{zeit} \end{array}$$

Am Ende der 3. Wortzeit wird F wieder gesetzt.

$$F' = \underline{F} \underline{G} H P1 T3 \longrightarrow \underline{F} \underline{G} H = 4. \text{ bis } 64. \text{ Wort-} \\ \text{zeit}$$

Am Ende der 64. Wortzeit wird G gesetzt.

$$G' = \underline{G} H Q4 Ga \underline{be} T3 K \longrightarrow \underline{F} \underline{G} H = 65., 66. \text{ Wortzeit} \\ | \\ \text{Sektorkoinzidenz nach 64 Wortzeiten}$$

Das K-Flip-Flop dient zum Sektorvergleich. K wird zu Beginn einer jeden Wortzeit gesetzt. Bei Nichtübereinstimmung der Sektoradreßspur S1 und der Bits 4-10 aus dem C-Register wird K zurückgesetzt. Bei Übereinstimmung bleibt K über die ganze Wortzeit gesetzt.

$$K' = T3 \underline{Faf} (F + G + R + Q3 + Q4 + \dots)$$

$$\underline{K}' = \underline{H} \underline{S2} (\underline{C} \underline{S1} + \underline{C} \underline{S1} \underline{S3}) \\ \phi_{4a} \quad \underline{\text{Antikoinzidenz}} \quad \text{Bit } 1 \dots 11, 16, 17, 30, 31$$

Aus der S1-Spur wird zu jeder Wortzeit der  $\phi_3$  während S2 S3 eine (neue) Sektoradresse in das C-Register geschrieben:

$$Cw' = \underline{G} \underline{H} \underline{S2} \underline{S3} S1 + \dots \\ \phi_2$$

Ende  $\phi_3$  steht die Operandenadresse des D (N, M)-Befehles im C-Register und läuft in  $\phi_4$  um.

$$Cw' = G \underset{\emptyset 3, \emptyset 4}{|} \underline{S2} \underset{\text{Umlauf}}{\downarrow} C \underline{brc} (Q1+Q2+Q3+Q4+\dots) + H \underset{\substack{\text{Nicht F\ddot{u}llen/} \\ \text{L\ddot{o}schen}}}{|} \underline{C} \underline{brc} + \dots$$

Umlauf  $\emptyset 4a$

Nach der Sektorkoinzidenz kann G am Ende der 64. Wortzeit gesetzt werden.

Am Ende der 66. Wortzeit wird F zurückgesetzt:

$$\underline{F'} = F G \underline{P1} T3 \quad \underline{F} G H \overset{\wedge}{=} 67. \text{ Wortzeit}$$

L gerade Wortzeit

Wortzeit LGP 21	Wortzeit Beispiel	Akkuinhalt am Wortzeitende						Bemerkungen A*				
		F	G	H	P1	P5	P6	I1	I2	S	Aw	
1	1	1	1	0	X	X	X	(A)	(V)	(1)	A.T3	0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0
2	2	0	0	1	0	0	0	(A+T3P6)	R	$\Sigma = (A+T3P6) - R$	$\Sigma = A^* \cdot T3$	1 1 1 0 1 1 0 0 0 0 1 1 1 1 0 0 0
3	3	0	0	1	1	0	0	A*	0	$\Sigma = 1$	$\Sigma = A^* \cdot T3$	0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0
4	4	1	0	1	0	0	1	A*	R	= 0	$\Sigma = A^* \cdot T3$	0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1
5	5	1	0	1	1	0	1	A*	0	= 0	$\Sigma = A^* \cdot T3$	0 1 1 1 0 0 0 1 0 0 1 0 0 0 0 1
6	6	1	0	1	0	0	0	A*	R	= 1	$\Sigma = A^* - R$	0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0
7	7	1	0	1	1	0	0	A*	0	= 1	$\Sigma = A^* \cdot T3$	0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
8	8	1	0	1	0	0	0	A*	R	= 1	$\Sigma = A^* - R$	1 0 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0
9	9	1	0	1	1	0	0	A*	0	= 1	$\Sigma = A^* \cdot T3$	0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0
10	10	1	0	1	0	0	1	A*	R	= 0	$\Sigma = A^* \cdot T3$	1 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1
11	11	1	0	1	1	0	1	A*	0	= 0	$\Sigma = A^* \cdot T3$	0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0
12	12	1	0	1	0	0	1	A*	R	= 0	$\Sigma = A^* \cdot T3$	1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1
13	13	1	0	1	1	0	1	A*	0	= 0	$\Sigma = A^* \cdot T3$	0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0
14, 16...62 15, 17...63	14 15 16 17 18 19	1	0	1	0	0	1	A*	R	= 0	$\Sigma = A^* \cdot T3$	1 0 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1
64	64	1	0	1	0	0	1	A*	R	= 0	$\Sigma = A^* \cdot T3$	0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0
65	65	1	1	1	1	0	1	A*	0	= 0	$\Sigma = A^* \cdot T3$	1 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 1
66	66	1	1	1	0	0	1	A*	R	$(P5P6+P5P6) = 0$	$\Sigma = A^* \cdot T3$	0 1 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1
67	67	0	1	1	1	0	1	(P5A*+P5A*) (P5P6+P5P6)	1	$(P5P6+P5P6) = 0$	$\Sigma = A^* - 0$	0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0

b / > / a /      a : 0.0 1 1 1 1 0 0 ;      b : 0.1 0 1 0 0 0 0 ;      = 0.1 1 0 0 0 0 0  
 Dezimal: 15      5 : 5 (@ 3 = @ 2 : 3      Divisor

Divisionsstabelle  
LGP 21



### 2.5.6. N-Befehl

$$N \triangleq Q_1 \ Q_2 \ Q_3 \ Q_4 \triangleq 0110$$

Beim N-Befehl wird der Inhalt des Akkumulators (Multiplikator) mit dem Inhalt einer Hauptspeicherzelle (Multiplikand) multipliziert.

Die wenigstbedeutende Hälfte des Produktes steht im Akkumulator.

Jeder Multiplikationsbefehl wird im LGP 21 intern als Festkommaoperation ausgeführt. Das Maschinenkomma steht zwischen Vorzeichen und meistbedeutendem Bit.

Die Multiplikation besteht aus wiederholten Additionen des Multiplikanden zum laufenden Teilprodukt und Schiften des Teilproduktes um 1 Bit nach jeder Addition.

Weil jeder Faktor 32 Bit hat, wird 32 mal addiert und 32 mal geschiftet. Damit umfaßt das Produkt einschließlich Vorzeichen und Spacerbit 64 Bit. Das gesamte Ergebnis steht in A\*. Für eine Weiterverarbeitung kann bei der N-Multiplikation aber nur die wenigstbedeutende Hälfte des A\*-Inhaltes verwendet werden. In der Vorzeichenstelle steht der meistbedeutende Wert (kein Vorzeichen!).

Bei der Multiplikation von Zahlen mit beliebigem Vorzeichen, ergeben sich die im folgenden aufgeführten Möglichkeiten. Negative Zahlen werden als Komplement zur Konstanten 2 dargestellt (z.B. 2-a) (siehe auch Division):

	Ist-Wert	Soll-Wert
$a \times b \triangleq a \times b \triangleq a \cdot b$		$+ab \triangleq ab$
$-a \times b \triangleq (2-a) \times b \triangleq 2b - ab$		$-ab \triangleq (2-ab)$
$-a \times (-b) \triangleq (2-a) \times (2-b) \triangleq 4-2b-2a+ab$		$+ab \triangleq ab$
$a \times (-b) \triangleq a \times (2-b) \triangleq 2a-ab$		$-ab \triangleq (2-ab)$

Aus dem Ist-Wert läßt sich der Sollwert durch Addition einer Korrektur berechnen:

$$\text{Ist} + \text{Korr.} = \text{Soll} \longrightarrow \text{Korr.} = \text{Soll} - \text{Ist}$$

	Soll	Ist	Korrektur, kompl.	Korrektur, normal
a x b	ab	ab	0	0
-a x b	(2-ab)	(2b-ab)	(2-2b)	-2b
-a x (-b)	ab	(4-2b-2a+ab)	-(2-2b+2-2a)	+2a+2b
a + (-b)	(2-ab)	(2a-ab)	(2-2a)	-2a

Für den Sonderfall  $(-a) \times (-b) = (-1) \cdot (-a)$  erhält man das falsche Ergebnis:  $(-1)$ , da Zahlen  $\geq 1$  nicht darstellbar sind. Es wird aber nichts gemeldet.

Die im folgenden beschriebenen Funktionen können zweckmäßigerweise auch parallel zum Text in der Multiplikationstabelle (Seite und ) und im Blockschaltbild des Rechenwerkes verfolgt werden.

Während der ersten Wortzeit  $\phi_4$  wird der Multiplikand V in das R-Register geschrieben, um ihn während der Ausführung des Befehls bei jeder Addition sofort zur Verfügung zu haben.

$$Rw' = \underbrace{G \underline{H} V \underline{Q1 Q2 Q3}}_{\phi_4} + \underbrace{R \underline{brc H}}_{\substack{\text{N,M-Multiplikation} \\ \text{Multiplikand}}} \quad \text{Umlauf in } \phi_4a$$

Gleichzeitig wird das Vorzeichen des Multiplikanden (V) im P5-Flip-Flop gespeichert.

$$P5' = \underbrace{G \underline{H} T3 V}_{\phi_4} P5a + \dots \quad \begin{array}{l} \text{E/A-Verriegelung} \\ \text{- Multiplikand} \end{array}$$

$$\underline{P5'} = \underbrace{G \underline{H} T3 V}_{\phi_4} P5a + \dots \quad \begin{array}{l} \text{+ Multiplikand} \end{array}$$

Der Multiplikator wurde durch den vorhergehenden Befehl in den Akkumulator gebracht.

$$Aw' = \underline{H} \underline{To} A (\underline{F} + \underline{G} + \dots)$$

↓      ↓      ↓  
 φ1, φ2 φ3  
 - Umlauf A

In der φ4 wird das Vorzeichen des Multiplikators (A) in P6 gespeichert und aus dem Akkumulatorumlauf ausgebendet.

$$Aw' = \underline{H} \underline{To} A \underline{T3} \underbrace{\underline{Q1} \underline{Q2} \underline{Q3}}_{\substack{| \\ N-Multiplikation}} + \dots$$

↓  
 Ausblendung des Vorzeichens

$$P6' = \underbrace{F G \underline{H}}_{\substack{| \\ φ4}} \underbrace{T3}_{|} \underbrace{A}_{\substack{| \\ -A(Multiplikator)}} P6a + \dots$$

↓  
 E/A-Verriegelung

$$\underline{P6'} = \underbrace{F G \underline{H}}_{\substack{| \\ φ4}} \underbrace{T3}_{|} \underbrace{A}_{+A(Multiplikator)} P6a$$

Ab der zweiten Wortzeit dient P1 zur Unterscheidung der geradzahligen und ungeradzahligen Wortzeiten. In den geradzahligen Wortzeiten (Additionszeit) ist P1 zurückgesetzt, in den ungeradzahligen Wortzeiten (Shiftzeit) ist P1 gesetzt.

$$\underline{P1'} = \underbrace{G \underline{T3} \underline{P1}}_{\substack{| \\ φ4}} \underbrace{P1b}_{\substack{| \\ Faf}} + \underbrace{H \underline{T3} \underline{P1}}_{\substack{| \\ φ4a}}$$

$$P1' = \underbrace{H \underline{T3} \underline{P1}}_{\substack{| \\ φ4a}}$$

Um das Vorzeichen des Multiplikators (in P6) zu berücksichtigen, erfolgt in der zweiten Wortzeit eine Subtraktion des Multiplikanden (R) von Null in Abhängigkeit von P6 (Berechnung des Korrekturfaktors (-b)).

$$I_1 = 0$$

$$I_2 = \underline{G} \ H \ \underline{P_1} \ P_6 \ R$$

└ Multiplikand  
└ Multiplikator

$$S = \underline{\underline{F}} \ H \ Q_3$$

2 WZ. └ M, N-Befehl  
∅4a

$$L' = (\underline{T_3} + \underline{H} \ \underline{P_1} \ Q_3) \ (S \ \underline{I_1} \ I_2 \ \underline{L}, + \dots)$$

WZ 2 - MN-Befehl └  
Übertragslogik

$$L' = T_3 \ H + \underline{Q_2} \ (Q_3 + \underline{Q_1}) \ (S \ \underline{I_1} \ \underline{I_2} \ L, + \dots)$$

└ auch MN-Befehl      Übertragslogik  
└ Rücksetzer am Ende aller ∅

$$Aw' = H \ \underline{P_1} \ (L \ \underline{I_1} \ I_2 + \cancel{L \ I_1 \ I_2} + \cancel{L \ I_1 \ I_2} + \cancel{L \ I_1 \ I_2})$$

Komplementbildung

In der 3. Wortzeit (siehe auch Multiplikationstabelle) wird die Berechnung des Korrekturfaktors (-2b) abgeschlossen. Die Multiplikation von (-b) mit 2 erhält durch Schiften des Ergebnisses um 1 Bit.

Damit wird auch der Multiplikand um 1 Bit verschoben, und es verschwindet die zuletzt ausmultiplizierte Stelle. Die neu zu multiplizierende Stelle wird ausgeblendet und in P6 (alle ungeradzahligen Wortzeiten) gespeichert.

$$P_6' = \underline{H} \ \underline{T_3} \ \underline{P_1} \ Q_3 \ A^* + \dots$$

ungeradzahlige WZ

$$\underline{P_6}' = H \ T_3 \ P_1 \ Q_3 \ A^*$$

Um das Vorzeichen des Multiplikanden zu berücksichtigen, wird bei negativem Multiplikanden ( $P_5 = "1"$ ) eine "1" von jedem Bit des Multiplikators subtrahiert. Bei positivem Multiplikanden ( $P_5 = "0"$ ) wird von jedem Bit des Multiplikators eine "0" subtrahiert (Korrekturfaktor).

$$I_1 = \underbrace{H \ P_1 \ Q_3}_{\substack{| \\ 3.WZ}} \ A^*$$

└ M, N-Befehl  
└ ungerade Wortzeiten ♂4a

$$I_2 = \underbrace{F \ H \ P_1}_{\substack{| \\ 3.WZ}} \ Q_3 \ P_5$$

└ Vorzeichen Multiplikand (R)  
└ M,N-Befehl

$$L' = T_3 \ S \ \underbrace{I_1 \ I_2 \ L}_{\substack{| \\ \text{Übertragslogik}}} + \dots$$

$$\underline{L'} = T_3 \ \underbrace{P_1}_{\substack{| \\ \text{ungerade WZ}}} + \underbrace{Q_2 \ Q_3}_{\substack{| \\ M,N-Befehl}} \ \underbrace{S \ I_1 \ I_2 \ L}_{\substack{| \\ \text{Übertragslogik}}} + \dots$$

$$A_{w'} = A_{w''} = H \ T_3 \ (L \ \underbrace{I_1 \ I_2}_{\substack{| \\ \text{kein Vorzeichen}}} + \underbrace{L \ I_1 \ I_2}_{\substack{| \\ \text{Addierlogik}}} + \underbrace{L \ I_1 \ I_2}_{\substack{| \\ \text{Addierlogik}}} + L \ I_1 \ I_2)$$

In der 4. bis 64. Wortzeit (61 Wortzeiten) der Multiplikation wird in den geraden Wortzeiten in Abhängigkeit von P6 der Multiplikand oder Null zum geschifften Teilprodukt addiert.

Eine weitere Addition erfolgt in den ungeraden Wortzeiten, wenn der Multiplikand negativ ist ( $P_5 = "1"$ ) und das Multiplikatorbit eine 1 ist ( $P_6 = "1"$ ).

In diesem Falle wird zu jeder Stelle des Teilproduktes eine 1 addiert, sonst wird der Inhalt von  $A^*$  um 1 Bit geschiftet.

$S = 0$  ab 4. Wortzeit

$$I_1 = \underbrace{F \ G \ H}_{\substack{| \\ 4. \text{ bis } 64. \text{ WZ}}} \ A^*$$

$$I_2 = \underbrace{G \ H \ P_1}_{\substack{| \\ \text{ungerade WZ}}} \ P_6 \ R + H \ P_1 \ Q_3 \ P_5 \ P_6$$

└ M,N-Befehl  
└ gerade WZ

$$L' = (T_3 + H \ Q_3 \ P_1) \ S \ I_1 \ I_2 \ L + \dots$$

$$\underline{L}' = T3 \underline{P1} + Q2 \underline{Q3} \underline{S} \underline{I1} \underline{I2} L + \dots$$

$$A_w' = A_w^{**} = (H \underline{T3+HP1+\dots})(\underline{LI1I2} + \underline{LI1I2} + \underline{LI1I2} + \underline{LI1I2})$$

Am Ende der 64. Wortzeit stehen die wenigstens bedeutenden Bits in A und das volle Produkt in A\*.  
Die N-Multiplikation ist beendet.

#### Wortzeitgruppen in $\emptyset_4$ und $\emptyset_{4a}$

F und G werden in der  $\emptyset_{4a}$  zur Unterscheidung der einzelnen Wortzeitgruppen benutzt.

Am Ende von  $\emptyset_4$  werden F und G für zwei Wortzeiten zurückgesetzt.

$$\underline{F}' = F \underline{G} \underline{H} T3 \quad \underline{F} \underline{G} H = 2. \text{ und } 3. \text{ Wortzeit}$$

$$\underline{G}' = G \underline{H} T3$$

Am Ende der 3. Wortzeit wird F wieder gesetzt.

$$F' = \underline{F} \underline{G} H P1 T3 \quad \underline{F} \underline{G} H = 4. \text{ bis } 64. \text{ Wortzeit}$$

Am Ende der 64. Wortzeit werden F und H zurückgesetzt

$$\begin{aligned} \underline{F}' &= F H T3 \underline{Q4} K \\ &\quad | \\ &\quad \text{Sektorkoinzidenz} \\ &\quad \text{nach 64 Wortzeiten} \end{aligned}$$

$$\begin{aligned} \underline{H}' &= H T3 \underline{Q4} K \\ &\quad | \\ &\quad \text{N-Befehl} \end{aligned}$$

Das K-Flip Flop dient zum Sektorvergleich. K wird zu Beginn einer jeden Wortzeit gesetzt. Bei Nichtübereinstimmung der Sektoradreßspur S1 und der Bits 4 bis 10 aus dem C-Register wird K zurückgesetzt. Bei Übereinstimmung bleibt K über die ganze Wortzeit gesetzt.

$$K' = T3 \underline{Faf} (\underline{F} + G + R + Q3 + Q4 + \dots)$$

$$K' = H \underline{S2} (C \underline{S1} + C \underline{S1} \underline{S3})$$

Ø4a	Anti- koinzidenz	Bit 1...1, 16, 17, 30, 31
-----	---------------------	---------------------------

Aus der S1-Spur wird zu jeder Wortzeit der Ø3 während S2 S3 eine (neue) Sektoradresse in das C-Register geschrieben:

$$Cw' = \underbrace{G \underline{H}}_{Ø3} \underline{S2} \underline{S3} S1 + \dots$$

Ende Ø3 steht die Operandenadresse des M,N(D)-Befehls im C-Register und läuft in Ø4 um.

$$Cw' = G \underline{S2} C \underline{brc} (Q1+Q2+Q3+Q4+\dots) + H C \underline{brc}$$

Ø3, Ø4	Umlauf	nicht "Füllen/ Löschen"
--------	--------	----------------------------

Umlauf Ø4a

Nach der Sektorkoinzidenz werden F und H zurückgesetzt, und der Rechner steht wieder in Ø1.

## 2.5.7. M-Befehl

$$M \stackrel{?}{=} Q_1 \ Q_2 \ Q_3 \ Q_4 \stackrel{?}{=} 0111$$

Beim M-Befehl wird der Inhalt des Akkumulators (Multiplikator) mit dem Inhalt einer Hauptspeicherzelle (Multiplikand) multipliziert.

Die meistbedeutende Hälfte des Produktes steht im Akkumulator.

Die M-Multiplikation entspricht bis einschließlich der 64. Wortzeit der N-Multiplikation.

Nach der 64. Wortzeit stehen die wenigstbedeutenden Bit in A und die meistbedeutenden in der 2. Hälfte von A\*.

In der 65. Wortzeit werden die meistbedeutenden Bits von A\* nach A übertragen. Am Ende dieser Wortzeit stehen die Bits jedoch infolge des Schiftens in A\* um 1 Bit, auch in A um 1 Bit verschoben.

In der 66. Wortzeit wird das Produkt durch Addition von A + A (entspricht Multiplikation von A mit 2) um 1 Bit nach links geschiftet. Danach stehen die meistbedeutenden Bits des Produktes in A.

Die P- und Addierlogik in der 65. Wortzeit entspricht der in den ungeraden Wortzeiten der N-Multiplikation.

In der 66. Wortzeit erfolgt die Addition A + A:

$$I1 = F \ G \ Q_3 \underline{P1} \ A + \dots$$

$$I2 = G \ H \ Q_3 \underline{P1} \ A + \dots$$

$$Aw' = H \underline{P1} (L \ I1 \ I2 + L \ \underline{I1} \ \underline{I2} + L \ I1 \ \underline{I2} + L \ \underline{I1} \ I2)$$

## Wortzeitgruppen

Am Ende der 64. Wortzeit wird nach dem Sektorvergleich G gesetzt (s. Wortzeitgruppen N-Befehl S. A2-66).

$G' = \underline{G} \ H \ Q4$     G a be T3 K  
            |                |  
            M(D)-Befehl      Sektorkoinzidenz nach 64 WZ

Die 65. Wortzeit ist gegeben mit

G F H P1.

Die 66. Wortzeit ist gegeben mit

F G H P1

Am Ende der 66. Wortzeit werden F,G,H zurückgesetzt,  
und der Rechner steht wieder in  $\emptyset 1$ .

F' = F G T3 P1 + ...

G' = G H T3 Q3 P1

H' = G H T3 Q3 P1



Wortzeit LGP 21	Wortzeit Beispiel	Akkuinhalt am Wortzeitende								Bemerkungen			
		F	G	H	P1	P5	P6	I1	I2	S	A <sub>W</sub>	A <sub>W</sub>	A*
1	1	1	1	1	0	X	X	(A)	(V)	X	A. <u>T3</u>	0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0	Multiplikator durch B-Befehl in Akku
2	2	0	0	1	0	0	0	P6R	1	$\Sigma = 0$	0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0		
3	3	0	0	1	1	0	0	A*	P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0: Null geworden durch T3, P6 geht an für WZ 4 und 5	
4	4	1	0	1	0	0	1	A*	P6R	$\Sigma = A^* p1R$	0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0	P6 geht aus für WZ 6 und 7	
5	5	1	0	1	1	0	1	A*	26P5	$\Sigma \cdot T3 = A \cdot T3$	0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0		
6	6	1	0	1	0	0	0	A*	P6R	$\Sigma = A^*$	1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0		
7	7	1	0	1	1	0	0	A*	P6P5	$\Sigma \cdot T3 = A \cdot T3$	0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0		
8	8	1	0	1	0	0	1	A*	P6R	$\Sigma = A^* p1R$	1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1		
9	9	1	0	1	1	0	1	A*	P6P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1		
10	10	1	0	1	0	0	0	A*	P6R	$\Sigma = A^*$	1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1		
11	11	1	0	1	1	0	0	A*	P6P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1		
12	12	1	0	1	0	0	0	A*	P6R	$\Sigma = A^*$	1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1		
13	13	1	0	1	1	0	0	A*	P6P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0		
14,16...62	14	1	0	1	0	0	0	A*	P6R	$\Sigma = A^*$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1		
15,17...63	15	1	0	1	1	0	0	A*	P6P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0		
64	16	1	0	1	0	0	0	A*	P6R	$\Sigma = A^*$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	N endet hier	
65	17	1	1	1	1	0	0	A*	P6P5	$\Sigma \cdot T3 = A^* \cdot T3$	0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0		
66	18	1	1	1	0	0	0	A	0	$\Sigma$	0 0 1 1 1 0 0 0 0 0 1 1 1 0 0	M endet hier	

Multiplicationstabelle  
LGP 21

Dezimal  
bei q  
 $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$  =  $\begin{pmatrix} 2 \\ 2 \end{pmatrix}$

$$0.1010000 \times 0.100000 = 0.0111100$$

Multiplicationstabelle  
LGP 21



#### 2.5.8. E-Befehl

$$E \triangleq Q_1 \underline{Q_2} \underline{Q_3} Q_4 \triangleq 1001$$

Mit dem E-Befehl können beliebige Stellen eines Wortes ausgeblendet werden. Das auszublendende Wort steht in A, die Maske (V) im Hauptspeicher und das Ergebnis wieder in A.

A bleibt "1", wenn in A und V an derselben Position eine "1" steht. Alle anderen Positionen werden "0".

Der E-Befehl entspricht einer Multiplikation ohne Berücksichtigung der einzelnen Überträge.

$$Aw' = \underbrace{F \ G \ H}_{\emptyset 4} \cdot \underbrace{\underline{Q_2} \underline{Q_3} Q_4}_E \cdot \underbrace{V \ A}_{d.h., \text{nur wenn beide } "1"} + \dots$$

wird eine "1" geschrieben.

## 2.5.9. U-Befehl

$$U \triangleq Q_1 \underline{Q_2} Q_3 Q_4 \triangleq 1010$$

Mit dem U-Befehl wird die Adresse im Zähler verändert, d.h. der Adreßteil des U-Befehls gibt die Hauptspeicherzelle an, aus der der nächste Befehl geholt werden soll.

Der Umlauf des C-Registers wird in  $\emptyset 4$  unterbrochen und der Inhalt des R-Registers nach C geschrieben ( $Cw'$ ). Die neue Information läuft danach in C um (siehe Gleichung für  $Cw'$ ).

In  $\emptyset 2$  wurde der U-Befehl in das R-Register gebracht. In der folgenden  $\emptyset 3$  braucht kein Operand gesucht zu werden.

$\emptyset 3$  wird nach einer Wortzeit beendet. G wird gesetzt.

$$G' = \underbrace{F \underline{G} T_3}_{\emptyset 3} \underbrace{Q_1 \underline{Q_2} Q_3}_{\text{be}} + \dots$$

$\emptyset 3$                            | Taste Ausführen nicht gedrückt  
U(T)-Befehl

$$Cw' = \underbrace{F \underline{G} \underline{H} T_3}_{\emptyset 4} \underbrace{Q_1 \underline{Q_2} Q_3 \underline{Q_4}}_{\text{U-Befehl}} R$$

                                 | Inhalt des R-Registers

### 2.5.10. T-Befehl

$$T \hat{=} Q_1 \underline{Q_2} Q_3 Q_4 \hat{=} 1011$$

Der T-Befehl ist ähnlich dem U-Befehl. Es wird ein Sprung nach der im Adreßteil des T-Befehles angegebenen Zelle ausgeführt.

Ein Sprung erfolgt immer bei negativem Akkumulatorinhalt oder wenn bei einem -T-Befehl die PST-Taste gedrückt ist.

Der U- und T-Befehl unterscheiden sich nur im Setzen von Q4. Sind obige Bedingungen erfüllt, wird Q4 zurückgesetzt. Es folgt ein normaler Ablauf des U-Befehles (siehe 2.5.9.).

Sind die Bedingungen nicht erfüllt, bleibt Q4 gesetzt. Der Umlauf von C bleibt unverändert. Der Sprungbefehl ist unwirksam.

Die Entscheidung über einen Sprung erfolgt in  $\phi_3$ .

$$\underline{Q4'} = \underline{\underbrace{Q_1 \underline{Q_2} Q_3}_{\substack{\text{T-Befehl} \\ \text{A negativ?}}} \ T_3 \ A + \underbrace{Q_1 \underline{Q_2} Q_3 \ T_3 \ R}_{\substack{\text{-T-Befehl}}} \ T_c}$$

Signal der  
PST-Taste

### 2.5.11. H-Befehl

$H \triangleq Q1 \ Q2 \ \underline{Q3} \ \underline{Q4} \triangleq 1100$

Mit dem H-Befehl wird der Inhalt des Akkumulators (A) in eine Hauptspeicherzelle gebracht. Der Akkumulator bleibt unverändert.

Die Schreiberlaubnis W erfolgt in  $\emptyset 4$

$$W = \underbrace{F \ G \ H}_{\emptyset 4} \ \underbrace{Q1 \ Q2 \ \underline{Q3}}_{(C)H-\text{Befehl}} + \dots$$

Die Schreibinformation  $Vw'$  lautet:

$$Vw' = L (Q1 A + \dots)$$

$$L' = W + \dots$$

$$\underline{L}' = T3 (\underline{H} + \dots) +$$

Der Umlauf in A erfolgt über:

$$Aw' = \underbrace{\underline{H} \ To \ A}_{\begin{array}{l} \text{Umlauf in } \emptyset 1-\emptyset 3 \\ \text{Umlauf in } \emptyset 4 \text{ für alle} \\ \text{Befehle außer C} \end{array}} \left[ \underbrace{F+G+Q1 \underline{Q3} \underline{Q4}}_{\text{Umlauf in } \emptyset 1-\emptyset 3} + \underbrace{T3 \underline{Q1} \underline{Q2} (Q3+Q4)}_{\text{Umlauf in } \emptyset 4} + \underbrace{Q2 (Q3+Q4) Faf}_{\text{Umlauf in } \emptyset 4} \right]$$

### 2.5.12. C-Befehl

$$C \stackrel{\wedge}{=} Q_1 \ Q_2 \ \underline{Q_3} \ Q_4 \stackrel{\wedge}{=} 1101$$

Mit dem C-Befehl wird der Inhalt des Akkumulators (A) in eine Hauptspeicherzelle gebracht.

Der Akkumulator wird gelöscht.

Die Ausführung des Speicherns erfolgt wie unter 2.5.11.

Die Schreiberlaubnis W erfolgt in  $\emptyset 4$ :

$$W = \underbrace{F \ G \ H}_{\emptyset 4}, \underbrace{Q_1 \ Q_2 \ \underline{Q_3}}_{C(H)-\text{Befehl}} + \dots$$

Die Schreibinformation in Vw' lautet:

$$Vw' = L (Q_1 \ A + \dots)$$

$$L' = W + \dots$$

$$\underline{L'} = T_3 \ \underline{H} + \dots$$

Der Umlauf in A wird beim Löschen unterbrochen.

$$Aw' = \underbrace{H \ To \ A}_{\begin{array}{l} \text{Umlauf in } Q_1, \emptyset 2, \emptyset 3 \\ \text{Umlauf in } \emptyset 4 \end{array}} \left[ \underbrace{F+G+Q_1 \underline{Q_3} Q_4 + T_3 Q_1 Q_2 (Q_3+Q_4)}_{\text{für alle Befehle außer C}} + \underline{Q_2 (Q_3+Q_4) Faf} \right]$$

## 2.5.13. A-Befehl

$$A \hat{=} Q_1 \ Q_2 \ Q_3 \ \underline{Q_4} \hat{=} 1110$$

Mit dem A-Befehl wird der Inhalt des Akkumulators (A) mit dem Inhalt einer Hauptspeicherzelle addiert. Das Ergebnis läuft in A um.

Das L-Flip Flop speichert bei der Addition in  $\phi_4$  die Überträge. L wird am Ende einer jeden Wortzeit (ausgenommen  $\phi_4$ ) zurückgesetzt ( $L' = T_3 \underline{H}$ ). Bei gleichzeitiger "1" in A und V wird L gesetzt, bei gleichzeitiger "0" in A und V wird L zurückgesetzt. Bei unterschiedlichem A und V bleibt die Stellung von L unverändert.

Die Eingänge der Addier-Logik werden mit I1 und I2 bezeichnet.

$$I_1 = \underline{H} \ A + \dots$$

$$I_2 = \underline{H} \ V + \dots$$

$$L' = \underline{T_3} \ \underline{S} \ I_1 \ I_2 \ \underline{L} + \dots$$

$$\underline{L'} = T_3 \ \underline{H} + \underbrace{L \ S \ I_1 \ I_2}_Y \ \underbrace{Q_2 \ (Q_3 + \dots)}_{\text{unwichtig für A-Befehl. Verhindert das Rücksetzen bei einem Y-, R-, H- und C-Befehl}} + \dots$$

Rücksetzer für A und V beide "0"

Das Signal S gibt die Addition an

$$\underline{S} = Q_4 \ \underline{H}$$

T3 in der Setzbedingung für L verhindert, daß L zur  $T_3$ -Zeit gesetzt wird, da dann Setzer und Rücksetzer ( $L' = T_3 \underline{H}$ ) gleichzeitig auf L einwirken würden.

Die Schreibinformation  $Aw'$  ergibt sich dann wie folgt:

$$Aw' = (\underbrace{FG \ Q_1 \ Q_2 \ Q_3}_{\phi_4} + \dots) (\underbrace{LI_1 \ I_2 + LI_1 \ I_2 + LI_1 \ I_2 + LI_1 \ I_2}_{A(S)-\text{Befehl}}) + \dots$$

$\phi_4 \quad A(S)-\text{Befehl} \qquad \text{Addierlogik}$

Die Addition negativer Zahlen erfolgt im Rechner  
in gleicher Weise wie die der positiven Zahlen.

Mathematisch wird sie durch Addition des Komple-  
mentes der negativen Zahlen gebildet.

Der LGP 21 bildet bei der Addition kein Komplement.

Die negativen Zahlen müssen deshalb schon als  
Komplementwerte gespeichert sein.

Beispiel:    -14                : 0,1110  
                                    Kehrwert : 1,0001  
                                    +"1"        :       1  
                                    Komplement: 1,0010     $\hat{=}$    -14

#### 2.5.14. S-Befehl

$$S \triangleq Q_1 \ Q_2 \ Q_3 \ Q_4 \triangleq 1111$$

Mit dem Subtraktionsbefehl wird der Inhalt ( $V$ ) einer Hauptspeicherzelle vom Inhalt des Akkumulators ( $A$ ) subtrahiert. Die Differenz läuft in  $A$  um.

Die Subtraktion erfolgt mathematisch, indem man das Komplement der Information vom Hauptspeicher ( $V$ ) zum Akkumulator ( $A$ ) addiert.

$$\begin{array}{ll} \text{Beispiel 1: } A = 30 \triangleq 11110 & +27 \triangleq 11011 \\ V = \underline{-27} \triangleq \underline{00101} & -27 \triangleq 00100 \quad \text{invers} \\ 3 & 00101 \quad \text{Komplement} \\ 3 & 00011 \triangleq 3 \end{array}$$

Das gleiche Ergebnis erhält man, indem man den Inhalt des Akkumulators ( $A$ ) invertiert, zu der Information vom Hauptspeicher ( $V$ ) addiert und das Ergebnis wieder invertiert.

$$\begin{array}{ll} \text{Beispiel 2: } A = 30 & 30 \triangleq 11110 \quad \text{invertiert: } 00001 \\ V = \underline{-27} & 27 \triangleq 11011 \quad \begin{array}{r} +11011 \\ \hline 11100 \end{array} \\ 3 & \text{invertiert: } 00011 = 3 \end{array}$$

Die Subtraktion wird im LGP 21 praktisch wie die Addition verwirklicht. Nur bei der Übertragsbildung ( $L$ -Flip-Flop) wird statt  $A$   $\underline{A}$  benutzt (s. auch 2.5.13).

$S = Q_4 \underline{H} =$  Angabe des Subtraktionsbefehles

$I_1 = A \underline{H}$

$I_2 = V \underline{H}$

$L' = T_3 \underline{H} \ S \ \underline{L} \ \underline{I_1} \ I_2$   
 ↓ Invertierung von  $A$

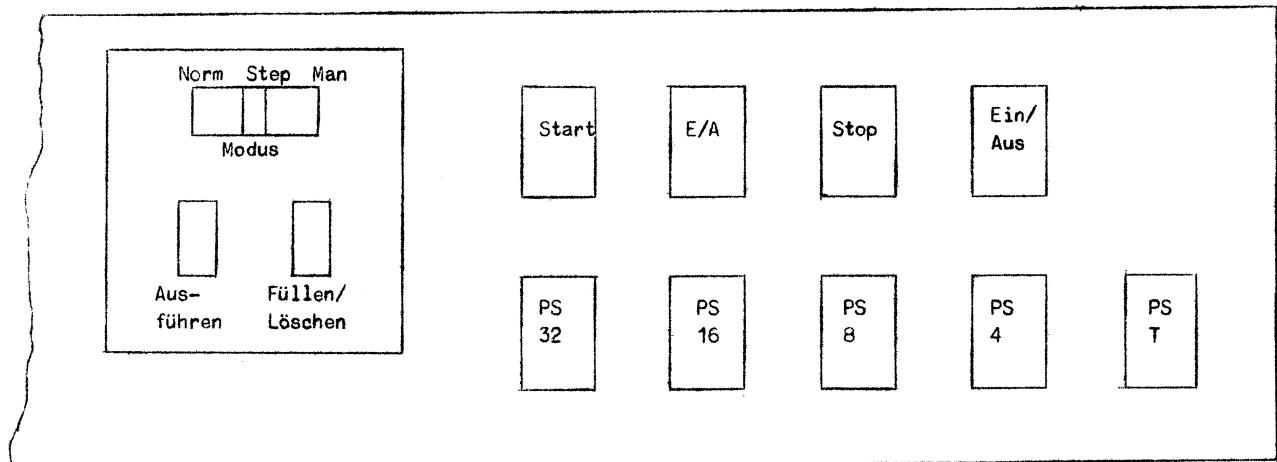
$L' = T_3 \underline{H} + L \ S \ \underline{I_2} \ I_1 \ \underline{Q_2(Q_3+\dots)}$   
 ist für den S-Befehl unwichtig

Die Schreibinformation  $Aw'$  ergibt sich dann wie folgt:

$$\begin{array}{ll} Aw' = (\underbrace{FG \ Q_1 Q_2 Q_3 + \dots}_{\emptyset 4 \quad (A)\text{-Befehl}})(\underbrace{LI_1 I_2 + LI_1 \underline{I_2} + LI_1 I_2 + LI_1 \underline{I_2}}_{\text{Addierlogik}}) \end{array}$$

## 2.6. Bedienungsfeld

Die Anordnung der Bedienungstasten ist in Bild 2.27 schematisch dargestellt. (Siehe auch Bild 3.37.)



### 1. Ein/Aus

Bild 2.24

Dieses ist der Hauptschalter, mit dem der Rechner in Betrieb gesetzt wird (Netzschalter).

Die Taste leuchtet auf, wenn alle Versorgungsspannungen des Rechners vorhanden sind.

### 2. Stop

Das rote Feld "Stop" dient nur als Anzeige. Es leuchtet, wenn der Rechner in "Stop" steht (kein Befehlsablauf,  $Kc = "1"$ )

$Kc = \frac{F \cdot G \cdot Q2}{\emptyset 1} \quad \text{wird Relais B (Bedienungsfeld)}$   
"Stop"  $\quad \quad \quad \text{nicht angesteuert.}$

### 3. E/A

Durch Drücken der E/A-Taste wird der Akkumulatorinhalt durch Unterbrechung des Umlaufes gelöscht und das jeweilige E/A-Gerät durch Zurücksetzung des entsprechenden Anwahl-Flip Flops abgewählt.

$$Aw' = \frac{To \cdot H \cdot A}{\text{Umlauf}} [ F + G + \dots ]$$

E/A-Signal

F(n)' = To + ...      Löschen Anwahl  
!\_ Anwahl-Flip Flop

Die E/A-Taste leuchtet, solange ein Anwahl-Flip Flop gesetzt ist ( $F_{af} \leq "1"$ , auch bei Manuell).

#### 4. Start

Die Start-Taste gibt das Signal (bs) zum Beginn einer Operation. Der Stop wird hierdurch aufgehoben.  
(Erklärung siehe Z-Befehl, Abschnitt 2.5.1.1.)  
Die Start-Lampe leuchtet in Abwechslung mit der Stop-Lampe. Sie leuchtet, wenn  $K_c$  nicht erfüllt ist  $\leq 0$  Volt

$$K_c = \frac{\underline{F} \quad \underline{G}}{\underline{\phi}1}, \frac{Q2}{\text{Stop}}$$

#### 5. PST, PS4, PS8, PS16, PS32

Durch diese "Programm-Sprungtasten" hat der Bediener die Möglichkeit, manuell in ein Programm einzugreifen und verschiedene Programmteile anzurufen.  
Eine gedrückte Taste leuchtet.  
Erklärung siehe Z-Befehl, Abschnitt 2.5.1.1. und T-Befehl, Abschnitt 2.5.

#### 6. Modus

Der Schalter Modus ist ein Dreistellungsschalter für "Normal-Step-Manuell". Mit ihm können drei Betriebszustände ausgewählt werden:

In Stellung "Normal" durchläuft der Rechner fortlaufend automatisch das gesamte Programm entsprechend der Befehlsfolge; in Mittelstellung "Step" stoppt der Rechner nach  $\phi 4$ , führt also jeweils nur einen Befehl aus; bei "Manuell" kann der Akkumulator manuell vom Tastenfeld des Flexowriters gefüllt werden (Simulation eines I-Befehles).

### 6.1. Normal

Diese Schalterstellung enthält keine speziellen logischen Ausdrücke oder Operationen. Die Verriegelungen über Step und Manuell sind aufgehoben.

### 6.2. Step

In Stellung "Step"-Signal 01- erfolgt nach jeder Befehlausführung ein Stop. Der Rechner bleibt in  $\emptyset 1$  stehen, da das G-Flip Flop nicht gesetzt wird. Das Setzen von G wird durch Q2 entschieden (siehe auch Abschnitt 2.5.1.1.).

$$G' = \frac{G \ H}{\emptyset 1} T3 \ K \ Q2 \ \underline{\underline{be}} \ \underline{\underline{bs}} \ Ga + \dots$$

$\emptyset 1$  Entscheidung | Faf  
nicht Start

Q2 wird nach jeder Befehlausführung zurückgesetzt.

$$\underline{Q2}' = \frac{G \ T3 \ 01}{\emptyset 4} \ \underline{Q1} + \dots \ \underline{01} \ \underline{H'} + \dots$$

Step | für M,N,D-Befehl nach  
Ablauf von  $\emptyset 4a$   
alle Befehle außer M,N,D

Durch Drücken der Start-Taste erfolgt der nächste Befehlsablauf (siehe auch Abschnitt 2.5.1.1. Setzen Q2).

Nach jeder Eingabe eines Zeichens erfolgt ebenfalls ein Stop:

$$\underline{Q2} = G \ T3 \ \underline{Q1} \ \underline{Q2} \ \underline{Q3} \ \underline{Q4} \ \underline{01} + \dots$$

I-Befehl Step

### 6.3. Manuell

Das Signal 01 für Step und das Signal bq für Manuell sind logisch verknüpft. Signal 01 wird "1" vor dem Signal bq, und 01 bleibt "1" während der Stellung Manuell, d.h. in Manuell erfolgt ebenfalls ein Stop wie unter 6.2. Außerdem wird das Flexowriteranwahl-Flip Flop gesetzt.

## 7. Füllen/Löschen

Durch Drücken der Taste Füllen/Löschen-Signal brc-, wird einmal der Akkumulatorinhalt in das R-Register geschrieben, zum anderen wird das C-Register zu bestimmten Zeiten und Befehlen gelöscht.

$$Rw' = \underbrace{A \ brc}_{A \rightarrow R} + R \ \underline{brc} \left[ \underline{G} + \dots \right]$$

R-Umlauf

$$Cw' = \underbrace{\underline{brc}}_{\text{Löschen}} \left[ F \ S2 \ C \ (\underline{Q1} + \dots) + \underline{G} \ S2 \ C + G \ \underline{S2} \ C \ (\underline{F} + \dots) + \dots \right]$$

## 8. Ausführen

Durch Drücken der Taste "Ausführen" -Signal bespringt der Rechner von  $\phi_1$  nach  $\phi_3$  und führt den im R-Register stehenden Befehl aus.

$$F' = \underline{be} \ T3 + \dots \quad (\text{Springen nach } \phi_3)$$

Der Rechner verharrt in  $\phi_3$ , bis die Taste "Ausführen" wieder losgelassen ist, da be in jeder Setzbedingung für das G-Flip Flop eingeschlossen ist.

Außerdem wird das Setzen aller E/A-Anwahl-Flip Flops zur Zeit be = "1" verhindert.

## 2.7. Ein- Ausgabe-Logik (E/A-Logik)

Bei der Normalausführung des LGP 21 ist ein Flexowriter als Standard-E/A-Einheit vorgesehen. Die Ansteuerung des ersten Flexowriters erfolgt über die E/A-Steuerung I (Zchngs.-Nr. 80573). Auf dieser Karte ist noch zusätzlich die Eingabelogik für einen Tally-Leser-Anschluß.

### 2.7.1. Anwahl der E/A-Einheit und Unterscheidung zwischen 4- oder 6 Bit Ein-Ausgabe

Die einzelnen E/A-Einheiten werden über Flip Flops angewählt.

Das Flexowriteranwahl-Flip Flop wird zur T3-Zeit der Ø3 durch einen (800) IO200- oder (800) P0200-Befehl bzw. durch "Manuell" gesetzt, das Tally-Leser-Anwahl-Flip Flop durch einen (800) I0000-Befehl.

$$Ff' = \underbrace{F \ G}_{\emptyset 3} \ T3 \ \underbrace{P1 \ P2 \ P3 \ P4 \ P5}_{0200} \ be \ Faf \ (Q1 \ Q2 \ Q3 \ Q4) +$$

P-Befehl

keine andere Einheit angewählt (E/A-Verriegelung)

nicht "Ausführen"

$$+ \underbrace{Q1 \ Q2 \ Q3 \ Q4}_{I-Befehl}) + bq$$

Manuell

$$Ft' = \underbrace{F \ G}_{\emptyset 3} \ T3 \ \underbrace{P1 \ P2 \ P3 \ P4 \ P5}_{0000} \ be \ Faf \ \underbrace{Q1 \ Q2 \ Q3 \ Q4}_{I-Befehl},$$

Es kann jeweils nur eine E/A-Einheit angewählt sein.  
Das Signal Faf verhindert weitere Anwahlen.

$$Faf = Ff + Ft + \dots F(n)$$

Tally-Leser      weitere E/A-Einheiten  
Flexowriter

Die Ein- Ausgabe kann ein 4- oder 6-Bit erfolgen. Der negative E/A-Befehl kennzeichnet 4-Bit. Die Unterscheidung erfolgt durch das K-Flip Flop. K wird zur T3-Zeit der Ø3 bei 4-Bit Ein- Ausgabe gesetzt und bei 6-Bit zurückgesetzt.

Die Eingabe in "Manuell" erfolgt nur in 4 Bit.

$$K' = T_3 \underbrace{Faf}_{\begin{array}{l} \text{negativer I-} \\ \text{oder P-Befehl} \end{array}} (R + \dots) + bq \quad (\cong 4 \text{ Bit Mode})$$

Manuell

$$\underline{K'} = F \underbrace{G \ Faf}_{\begin{array}{l} \text{kein negativer Befehl} \end{array}} \underbrace{T_3 \ R}_{\begin{array}{l} \text{P-Befehl} \end{array}} (\underbrace{Q_1 \ Q_2 \ Q_3 \ Q_4}_{\text{I-Befehl}} + \underbrace{Q_1 \ Q_2 \ Q_3 \ Q_4}_{\text{I-Befehl}}) \quad (= 6 \text{ Bit Mode})$$

#### 2.7.2. Eingabe von Hand (Manuell) (s. auch 2.10.5.)

In Stellung "Manuell" erfolgt eine 4 Bit-Eingabe vom Tastenfeld des Flexowriters in den Akkumulator.

Der Rechner steht in  $\phi 1$ . In Stellung "Manuell" wird  $bq = -20V$ .

$Q_1$  wird zur Kennzeichnung des Eingabebefehls zurückgesetzt (Simulation eines I-Befehles).

$$\underline{Q_1}' = bq$$

Das Flexowriteranwahl-Flip Flop wird gesetzt:

$$Ff' = bq$$

Das K-Flip Flop wird zur Auswahl der 4 Bit-Eingabe gesetzt:

$$K' = bq$$

Beim Anschlagen einer Flexowriter-Type werden entsprechend dem jeweiligen Code die Umschaltkontakte SC6, SC8 bis SC 12 betätigt. Danach schaltet SC 7 verzögert um und legt  $bq = -20V$  an die Umschaltkontakte und an JL 12. Entsprechend der Stellung von SC 6, SC 8 ... SC 12 werden die P-Flip Flops gesetzt. JL 12 leitet zur folgenden  $T_3$ -Zeit durch Setzen von F die  $\phi 3$  ein.

$F' = F_c = JL\ 12 \quad \underline{Sc} \ \underline{Sk} \ T3$

keine unlesbaren Zeichen

kein Stop Code

$P1 = P1c$

$P2 = P2c$

usw. bis  $P6$

$\underline{P1} = P1d$

$\underline{P2} = P2d$

Der Rechner steht in  $\emptyset 3$  bis SC7 wieder umschaltet und -20V an JL11 legt. Hierdurch wird G gesetzt und  $\emptyset 4$  eingeleitet.

$G' = G_c = JL11 \underbrace{F \ G}_{\emptyset 3} T3$

Die P-Flip Flops werden für die Dauer der  $\emptyset 4$  durch das i-Signal als Schiftregister geschaltet.

$i = ic + \dots = Faf \underbrace{G \ F}_{\emptyset 4}$

Das P-Schiftregister von P1 bis P4 wird in den Umlauf des Akkumulators eingeschlossen. Der A-Umlauf verlängert sich somit um 4 Bit (Bild)

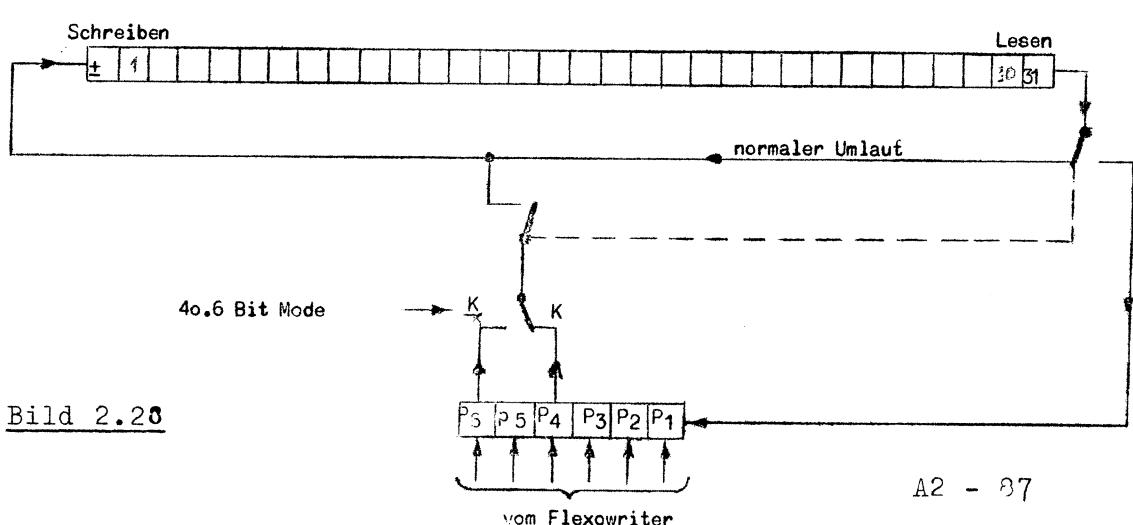
$P1' = G_i A \quad P2' = G_i P1$

usw. bis  $P6$

$\underline{P1}' = G_i \underline{A} \quad \underline{P2}' = G_i \underline{P1}$

$Aw' = Ac1 = \underbrace{F \ G}_{\emptyset 4} Q1 \ Faf \ K \ P4$

Eingabe      Übernahme P4 (4 Bit)



Die  $\phi_4$  wird nach einer Wortzeit beendet, und der Rechner wartet in  $\phi_1$  auf das nächste Zeichen vom Flexowriter.

Bei den Zeichen für die Maschinenfunktionen (unten, oben, Farbe usw.) wird der SC7-Kontakt nicht betätigt. Die P-Flip Flops werden nicht gesetzt. Der Rechner bleibt in  $\phi_1$ , da  $\phi_3$  und damit auch  $\phi_4$  nicht gesetzt werden können.

Die 4 Bit-Information wird von P4 in die meistbedeutende Stelle von A eingeschrieben und wandert im Laufe der  $\phi_4$  bis zur wenigstbedeutenden Stelle (Schiften um 32 Bit des erweiterten Akkumulators), d.h. die Information von P4 steht nach der  $\phi_4$  -eine Wortzeit- in der Position 31 des Akkumulators, P3 in 30, P2 in 29 und P1 in 28. Die alte Information aus Position 31 steht nun in 27 usw.

In der  $\phi_4$  läuft der Inhalt von A wieder normal um. Es folgt die  $\phi_1$  und ein weiteres Zeichen kann eingegeben werden. Dann verschiebt sich in  $\phi_4$  das vorhergehende Zeichen in A um 4 Positionen auf die Stellen 24, 25, 26, 27, da A um 4 Bit verlängert ist.

Bei jeder weiteren Eingabe verschieben sich also die bereits eingegebenen Zeichen zur meistbedeutenden Stelle hin.

Die Eingabe sei beendet. Der Rechner steht in  $\phi_1$ .

Die Taste "Manuell" wird zurückgesetzt, d.h. bq wird -20V. Der Flexowriter wird abgewählt.

Ff' = F G T3 Q3 Xs Sr' bQ  
 $\phi_1$        $\overset{\uparrow}{\text{Eingabe}}$

### 2.7.3. Eingabe über Flexowriter-Leser Q1 Q2 Q3 Q4 = 0100

(siehe auch 2.10.5.)

Mit einem (800) I0200-Befehl erfolgt eine (4 Bit)  
6 Bit-Eingabe vom Flexowriterleser in den Akkumulator.

In  $\emptyset_3$  wird der I-Befehl in das R-Register und in die  
Q-Flip Flops gegeben.

Das Flexowriteranwahl-Flip Flop wird gesetzt. Gleich-  
zeitig werden die P-Flip Flops zurückgesetzt.

$Ff' = F \underline{G} T3 \underline{P1} \underline{P2} \underline{P3} \underline{P4} P5 \underline{be} \underline{Faf} \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4}$

$P1d = P2d = P3d = P4d = P5d = P6d =$   
Rücksetzer  $P1 \dots P6$

$= \underline{F} \underline{\cancel{G}} \underline{T3} \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4} \underline{be} \underline{Faf}$   
 $\emptyset_3 \quad \text{I-Befehl}$

$\emptyset_3$  wird nach einer Wortzeit beendet, da keine Operanden-  
adresse gesucht zu werden braucht.

G wird gesetzt und der Rechner steht in  $\emptyset_4$ .

$G' = F \underline{G} T3 \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4} \underline{be} \underline{Faf}$

Die P-Flip Flops (alle "0") werden für die Dauer von  
 $\emptyset_4$  durch das i-Signal als Schiftregister geschaltet

$i = ic1 + \dots = Faf F G + \dots$

Das P-Schiftregister wird in den Umlauf des Akkumula-  
tors eingeschlossen.

$Aw' = Ac1 = F G \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4} (K P4 + \underline{K} P6)$

$P1' = G i A \quad P2' = G i P1 \quad \text{usw. bis}$   
 $\underline{P1}' = G i \underline{A} \quad \underline{P2}' = G i \underline{P1} \quad P4 \text{ bzw. } P6$

Das Flexowriter-One Shot Sr wird für ca. 50msec gesetzt und bringt das kcri-Relais.

Sr\* = F Q1 Ff Xs bq  
! - Signal vom Flexowriter  
(keine Eingabe)

$\phi_4$  dauert eine Wortzeit, danach beginnt wieder  $\phi_1$ .

G' = T3 H

F' = F G H T3

Über kcri 3-4 kann das Relais KRC anziehen. KRC hält sich mit seiner zweiten Wicklung über den Kontaktweg  
/+/ "Start-Rechnen" / "Stop-Lesen" / krc 3-4 /kpe 21-22 /  
KRC /-/

Nach 50msec wird das Sr-Signal wieder negativ und KCR I fällt ab. Dadurch kann KOC anziehen.

/+/ "Start-Rechnen" / "Stop-Lesen" / SR / krc 3-4 /  
kcri 21-22 / kfb 25-26 / STC-2 / KOC /-/

KOC hält sich über:

/+/ "Start-Rechnen" / "Stop-Lesen" / SR / krc 3-4/  
koc 1-2 / KOC /-/

Die Leser-Kupplung LR zieht an:

/+/ SCRT 1-2 / SBS / LKL 1-2 / "Start-Lesen" / SCD 1-2 /  
kdc 3-4 / SF-4 / SF-1 / koc 25-26 / krc 5-6 / kmi 1-2 /  
LR / - /

Wenn ein Zeichen vom Lochstreifen gelesen wird, schließen die entsprechenden Leserkontakte SR(n) und die entsprechenden Translator-Magnete LT sowie die Translator-Kupplung LTC ziehen an. Das gewünschte Zeichen wird über die Typenhebel ausgedruckt.

Durch das Anschlagen der Type werden entsprechend dem jeweiligen Code die Umschaltkontakte SC6, SC8 ... SC12 betätigt. Danach schaltet SC7 verzögert um und legt -20V an die Umschaltkontakte und an JL12.

Ff = -20V/JL32/koc28-29/krc25-26/SC7/

Entsprechend der Stellung von SC6, SC8 ... SC12 werden die P-Flip Flops gesetzt. JL12 leitet zur folgenden T3-Zeit durch Setzen von F die  $\phi_3$  ein.

$F' = F_c = JL12 \quad \underline{SC} \quad \underline{Sk} \quad T3$

keine unlesbaren Zeichen  
kein Stop Code

$$\begin{array}{ll} \underline{P1^*} = \underline{P1c} & \underline{P2^*} = \underline{P2c} \\ P1^* = \underline{P1d} & P2^* = \underline{P2d} \end{array} \quad \text{usw. bis } P6$$

Der Rechner steht in  $\phi_3$  bis SC7 wieder umschaltet und -20V an JL11 legt. Hierdurch wird G gesetzt und  $\phi_4$  eingeleitet.

$$G' = G_c = JL11 \quad F \quad \underline{G} \quad T3$$

Die P-Flip Flops werden für die Dauer der  $\phi_4$  durch das i-Signal als Schiftregister geschaltet und das P-Schiftregister in den A-Umlauf eingeschlossen (s. Bild 2.28)

$$\begin{array}{ll} P1' = G_i A & P2' = G_i P1 \\ \underline{P1'} = G_i \underline{A} & \underline{P2'} = G_i \underline{P1} \end{array} \quad \text{usw. bis } P6$$

$$Aw' = Ac1 = F \quad G \quad \underline{Q1} \quad Q2 \quad \underline{Q3} \quad \underline{Q4} \quad (K \quad P4 + \underline{K} \quad P6)$$

Übernahme von P4 bzw. P6

Die  $\phi_4$  wird nach einer Wortzeit beendet, und der Rechner wartet in  $\phi_1$  auf das nächste Zeichen vom Flexowriter. Beim Stop-Code öffnet der Leserkontakt SR1. Die Relais KOC und KRC und damit auch die Leserkupplung LR fallen ab.

Das Signal Xs wird OV über koc 21-22, krc 21-22 und Ff wird zurückgesetzt.

$$\underline{Ff'} = \underline{F} \quad \underline{G} \quad \underline{Q3} \quad T3 \quad \underline{Sr'} \quad Xs \quad \underline{bq}$$

Faf wird wieder OV.

Bei den Zeichen für die Maschinenfunktionen (unten, oben, Farbe usw.) wird der SC7-Kontakt nicht betätigt. Die P-Flip Flops werden nicht gesetzt. Der Rechner bleibt in  $\phi_1$ , da  $\phi_3$  und damit auch  $\phi_4$  nicht gesetzt werden können.

#### 2.7.4. Links-Schiften Q1 Q2 Q3 Q4 $\hat{=}$ 0100

Der I-Befehl wirkt gleichzeitig als Schiftbefehl. Der Akkumulator wird um 4 bzw. 6 Bit zur meistbedeutenden Stelle hin geschiftet, und in die wenigstbedeutenden Stellen werden Nullen geschiftet.

Falls nur geschiftet werden soll, darf die Adresse des I-Befehls keine Ein- Ausgabeeinheit anwählen.

Der Funktionsablauf ist wie unter 2.7.3.

Die P-Flip Flops werden zur T3-Zeit der  $\phi_3$  gelöscht. Hierdurch werden Nullen in die wenigstbedeutenden Stellen geschrieben.

P1d = P2d = ... P6d = F G T3 Q1 Q2 Q3 Q4 be Faf

Das aus den P-Flip Flops gebildete Schiftregister wird in den Umlauf von A mit eingeschlossen:

i = ic = F G Q1 Q2 Q3 Q4  
 $\phi_4$  Schift-Befehl (I)

Aw' = Ac1 = F G Q1 Q2 Q3 Q4 (K P4 + K P6)

Da Faf nicht OV wird, gilt dieser "Eingabe Zustand" nur für einen Phasenablaufzyklus und wird wie ein gewöhnlicher Befehlsablauf interpretiert.

## 2.7.5. Ausgabe über Flexowriter (Druckbefehl)

$P \hat{=} Q_1 \underline{Q_2} \underline{Q_3} \underline{Q_4} \hat{=} 1000$

Siehe auch Blockschaltbild 2.10.6.

Mit einem (800) P0200-Befehl wird das Zeichen, das durch die an den meistbedeutenden Stellen im Akkumulator stehenden (4 Bit) 6 Bit gekennzeichnet ist, ausgedruckt. Der Akkumulatorinhalt bleibt unverändert.

$\phi_1$  und  $\phi_2$  verlaufen normal.  $\phi_3$  wird auf eine Wortzeit begrenzt, da keine Operandenadresse gesucht zu werden braucht.

$G' = \underbrace{F \ G}_{\phi_3} T_3 \underline{Q_2} \underline{Q_3} \underline{Q_4} \text{ be Faf} + \dots$   
P-Befehl

Gleichzeitig wird das Flexowriteranwahl-Flip Flop Ff gesetzt und die 4- oder 6 Bit Ausgabeunterscheidung vorgenommen (s. 2.7.1.).

$\phi_4$  dauert eine Wortzeit.

Die P-Flip Flops werden in  $\phi_4$  als Schiftregister geschaltet. Während des Umlaufs von A wird sein Inhalt gleichzeitig in die P-Flip Flops geschiftet. Nur die 6 meistbedeutenden Bit werden in den P-Flip Flops gehalten.

$i = \underbrace{F \ G}_{\phi_4} \text{ Faf}$

$P1' = G \ i \ A \quad P2' = i \ P1 \quad P3' = i \ P2$   
 $\underline{P1}' = G \ i \ \underline{A} \quad \underline{P2}' = i \ \underline{P1} \quad \underline{P3}' = i \ \underline{P2} \quad \text{usw. bis } P_6$

$AW' = \underline{H} \ A \ To \ \underline{Q_1} \underline{Q_3} \underline{Q_4}$   
|  
Umlauf  
P(H)-Befehl

Am Ende  $\phi_4$  wird zur Kennzeichnung des P-Befehls für den weiteren Ablauf  $Q_3$  gesetzt.

$Q3' = G \ T_3 \ \underline{Q_1} \underline{Q_2} \underline{Q_3} \underline{Q_4}$   
P-Befehl

Der eigentliche Druckvorgang beginnt in der folgenden  $\phi_1$ .

$$\underline{F}' = F \text{ } G \text{ } \underline{H} \text{ } T3$$

$$\underline{G}' = G \text{ } \underline{H} \text{ } T3$$

Bei der 4 Bit Ausgabe werden nur die hexadezimalen Zeichen ausgedruckt. Dazu wird P5 gesetzt und P6 zurückgesetzt.

$$P1, P2, P3, P4, P5, P6 \triangleq \underline{\underline{XXX10}} \\ \text{beliebig}$$

$$P5' = \underline{F} \text{ } \underline{G} \text{ } K \text{ } Q3 \text{ } Faf$$

$$\underline{P6}' = \underline{\underline{F}} \text{ } \underline{\underline{G}} \text{ } K \text{ } Q3 \text{ } Faf \\ \begin{array}{c} \phi_1 \\ | \\ 4 \text{ Bit} \end{array} \quad \begin{array}{l} \text{P-Befehl} \end{array}$$

Das X-Flip Flop löst den Druckvorgang aus und hält den Rechner in  $\phi_1$  bis der Druckvorgang ausgeführt ist, wenn ein neuer Druckvorgang folgt.

$$X' = \underline{\underline{F}} \text{ } \underline{\underline{G}} \text{ } \underline{Tx} \text{ } Q3 \text{ } T3 \text{ } Ff \text{ } (\text{JL-33})(\text{krc 23-24})(\text{kfb 5-6}) \cdot \\ \begin{array}{c} \phi_1 \\ | \\ (\text{P-Befehl}) \end{array} \\ \begin{array}{l} \text{Translator Kupplungstreibersignal} \\ \cdot (\text{STC-4})(\text{koc 3-4}) \end{array}$$

Die Translatortreiber TP1...TP6 und die Translatorkuppung LTC werden entsprechend der jeweiligen Code-Kombination der P-Flip Flops gesetzt:

$$TP1 = X \text{ } P1$$

$$TP2 = X \text{ } P2$$

$$TP6 = X \text{ } P6$$

$$Tx = X \text{ } CP \text{ (Translator-Kupplungstreibersignal)}$$

Die Druckoperation wird ausgeführt, damit kann die Ausgabeanwahl gleichzeitig zurückgenommen werden.

Ff = X

Hierdurch wird auch Faf wieder "1", und der Rechner befindet sich wieder in einer normalen  $\emptyset 1$ .

X bleibt nur eine Taktzeit lang gesetzt (10  $\mu$ sec).

X' = F G Tx Q3 T3 Ff (JL-33)(krc 23-24)(kfb 5-6).  
(STC-4)(koc 3-4)

Falls mehrere Druckbefehle einander folgen, muß der Rechner solange in  $\emptyset 1$  warten, bis der jeweilige Druckbefehl ausgeführt ist.

X kann erst gesetzt werden, wenn Tx und JL-33 beide "1" sind. Der Rücksetzer für die Translator-Treiber und für die Translatorykupplung wird durch STC-3 bestimmt.

Tx wird "1" über:

-20V, JL-28, koc 5-6, STC-3, JL-29.

JL-33 wird "1" über:

Null, JL-31, koc 3-4, STC-4, kfb 5-6, krc 23-24, JL-33.

#### 2.7.6. Eingabe über Tally-Leser

I  $\hat{=}$  Q1 Q2 Q3 Q4  $\hat{=}$  0100

(Siehe auch Blockschaltbild 2.10.7.)

Mit einem (800) I0000-Befehl erfolgt eine (4-Bit) 6-Bit Eingabe vom Tally-Leser in den Akkumulator.

In  $\emptyset 3$  wird der I-Befehl in das R-Register und in die Q-Flip Flops gegeben (s. 2.4.2.3.).

Das Tally-Anwahl-Flip Flop wird gesetzt. Gleichzeitig werden die P-Flip Flops zurückgesetzt.

Ft' = F G T3 P1 P2 P3 P4 P5 be Faf Q1 Q2 Q3 Q4

P1d = P2d = P3d = P4d = P5d = P6d =  
Rücksetzer für P1...P6

= F G T3 Q1 Q2 Q3 Q4 be Faf  
Ø3 I-Befehl

$\phi_3$  wird nach einer Wortzeit beendet, da keine Operandenadresse gesucht zu werden braucht.

G wird gesetzt und der Rechner steht in  $\phi_4$ .

$G' = F \underline{G} T3 \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4} \underline{be} \underline{Faf}$

Die P-Flip Flops (alle "0") werden für die Dauer von  $\phi_4$  durch das i-Signal als Schiftregister geschaltet.

i = ic1 + ... = Faf F G + ...

Das P-Schiftregister wird in den Umlauf des Akkumulators eingeschlossen.

$Aw' = Ac1 = F G \underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4} (K P4 + K P6)$

$P1' = G i A$        $P2' = G i P1$   
 $P1'$  = G i A       $P2'$  = G i P1      usw. bis P6

$\phi_4$  dauert eine Wortzeit, danach beginnt wieder  $\phi_1$ .

Zur Synchronisierung des Tally-Lesers mit dem Rechnergatkt wird das Flip Flop Di gesetzt.

$Di' = \frac{F G}{\phi_1} Ft \underline{Xp}$   
astabiler Multivibrator  
Tally-Anwahl-Flip Flop

Die Ansteuerung des Tally-Leser erfolgt über

$R_p = D_t = D_i X_p$   
|  
Tally Multivibrator  
Tally-Leser Kupplungstreiber Signal

$\underline{R_p} = \underline{T_r}$   
|  
Signal für die Tally-Leser Kontakte

Solange  $D_i$  gesetzt ist, verläuft das Signal  $R_p$ ,  $\underline{R_p}$ ,  $T_r$  bzw.  $D_t$  im Takte des Tally-Multivibrators  $X_p$  (4,5msec).

Der 4,5 msec Impuls gibt das Treibersignal für die Kupplung.

Sobald  $R_p$  "1" wird, geht der Rechner zur  $T_3$ -Zeit durch Setzen von  $F$  in  $\emptyset 3$ :

$F_c = \underline{F} \underline{G} \underline{T_3} \underline{S_c} \underline{C_d} \underline{S_k} \underline{F_t} \underline{R_p}$   
|  
kein Maschinenzeichen  
kein Code delete  
kein Stop Code

$F' = \underline{F_c} + \dots$

Die Tally-Leser Kupplung zieht an (4,5msec Impuls). Bevor jedoch das nächste Zeichen erreicht wird (nach ca. 4msec), werden entsprechend der Lage der Tally-Leser Kontakte über das Signal  $T_r = \underline{R_p} = -20V$  die entsprechenden P-Flip Flops gesetzt.

$P_1 = P_{1c}$   
 $\underline{P_1} = P_{1d}$  usw. bis  $P_6$

Sobald  $R_p = -20V$  wird (nach 4,5msec), geht der Rechner zur  $T_3$ -Zeit durch Setzen von  $G$  in  $\emptyset 4$

$G_c = \underline{F} \underline{G} \underline{T_3} \underline{F_t} \underline{R_p}$   
 $\emptyset 3$

$G' = \underline{G_c} + \dots$

Die P-Flip Flops werden für die Dauer der  $\phi_4$  durch das i-Signal als Schiftregister geschaltet und das P-Schiftregister in den A-Umlauf eingeschlossen (s. Bild 2.28).

$$\begin{array}{ll} P1' = G \text{ i } A & P2' = G \text{ i } P1 \\ \underline{P1}' = G \text{ i } \underline{A} & \underline{P2}' = G \text{ i } \underline{P1} \end{array} \quad \text{usw. bis } P6$$

$$Aw' = Ac1 = \underbrace{F \text{ G}}_{\phi_4} \underbrace{Q1 \text{ Q2 \cancel{Q3} \cancel{Q4}}}_{\text{I-Befehl}} \underbrace{(K P4 + K P6)}_{\text{Übernahme von P4 bzw. P6}}$$

Die Phase 4 wird nach einer Wortzeit beendet und der Rechner wartet in  $\phi_1$  auf das nächste Zeichen vom Leser.

Wird ein Stop Code, ein Code delete oder ein Maschinenzeichen gelesen, wird F zur T3-Zeit der  $\phi_3$  zurückgesetzt und der Rechner steht in  $\phi_1$ .

$$\begin{array}{l} \underline{F}' = Fd = \underbrace{F \text{ G}}_{\phi_3} T3 (Sc + Cd + Sk) Ft \\ \text{Stop Code} \quad \downarrow \quad \text{Maschinenzeichen} \\ \text{Code delete} \end{array}$$

Durch einen Stop Code oder durch Drücken der E/A-Taste wird die Tally-Leser-Anwahl zurückgenommen.

$$\begin{array}{l} \underline{Ft}' = \underbrace{F \text{ G}}_{\phi_1} T3 Sc \underbrace{Xp}_{\downarrow \text{E/A-Taste}} + To \end{array}$$

Hierdurch wird auch Faf wieder "1" und damit Di zurückgesetzt:

$$\underline{D1}' = \underline{Faf}$$

## 2.8. Bedeutung der Buchstaben in den logischen Gleichungen

### 2.8.1. Flip Flops

F, G, H Phasen-Flip Flops. Sie stellen  $\phi_1 \dots \phi_4$  dar.

$$(\phi_1 = \underline{F} \underline{G} \underline{H}; \phi_2 = \underline{F} G \underline{H}; \phi_3 = F \underline{G} \underline{H}; \\ \phi_4 = F G \underline{H}; \phi_4a = \dots H)$$

Q1 ... Q4 Befehls-Flip Flops. Sie werden zur Speicherung des Befehles für die Befehlausführung benutzt (Schiftregister).

Mehrfach ausgenutzt wird:

Q1 als Sprungindikator

Q2 als Stopindikator

Q3 als Ausgabebefehlindikator.

P1 ... P6 Spuradreß- und E/A-Flip Flops. Sie werden zur Speicherung der Spuradresse und bei einem Ein-/Ausgabe-Befehl zur Speicherung der Information benutzt (Schiftregister).

Mehrfach ausgenutzt wird:

P1 zur Unterscheidung geradzahliger und ungeradzahliger Wortzeiten in  $\phi_4a$  beim M,N,D-Befehl.

P5 zur Speicherung des Vorzeichens des Operanden (Divisor, Multiplikand) vom Hauptspeicher beim M,N,D-Befehl.

P6 zur Speicherung des Vorzeichens des Dividenden bzw. des laufenden Restes und des Multiplikators.

K Entscheidungs-Flip Flop für Sektorübereinstimmung in  $\phi_1$  und  $\phi_3$ .

Übertrags-Flip Flop in  $\phi_2$  bei der Addition von 1 zum C-Register.

Übertrags-Flip Flop in  $\phi_4$  für die Addition von 1 zum C-Register bei einem R-Befehl.

Entscheidungs-Flip Flop bei M,N,D-Befehl für die Sektorübereinstimmung in der 64. Wortzeit.

Als Indikator für 4 oder 6 Bit Mode bei der Ein-/Ausgabe oder Links-Schift-Befehl.

L Übertrags-Flip Flop beim A,S,M,N,D-Befehl.  
Es zwingt beim Speicher-Befehl eine "0" in  
die Spacer-Bit-Stelle.

$F_f$ ,  $F_t$  Anwahl-Flip Flop für E/A-Geräte.

### 2.8.2. Logische Ausdrücke

i Schaltet die P-Flip Flops als Schift-Register  
zur Übernahme der Spuradresse oder eines Zei-  
chens bei der Ein-/Ausgabe.

$\alpha 11$  Schaltet die Q-Flip Flops als Schift-Register  
zur Übernahme des Befehls-Code.

J1 Addierer-Eingang (vom A-Register usw.).

J2 Addierer-Eingang (vom Hauptspeicher usw.).

B3 Arithmetischer Übertragsausdruck.

$\alpha 5 - \alpha 8$  Überlauf-Begriffe

S Entscheidung über Addition - Subtraktion.

W Schreiberlaubnis bei einem Schreibbefehl.

r1 Eingangssignal zu dem P-Flip-Flop-Schift-  
Register.

Kc Gibt den blockierten Zustand an und steuert  
die Stop- und Start-Anzeigelampe.

Faf Zeigt den Ein/Ausgabe-Zustand an.

T3 Vorzeichenzeit ( $T3 = S1 S3$ ).

Index a Verriegelung für die Setzseite eines Flip Flops  
Index b Verriegelung für Rücksetzseite eines Flip Flops

Index c Setzer für ein Flip Flop  
Index d Rücksetzer für ein Flip Flop

#### 2.8.3. Schreibsignale:

Aw' Schreibsignal für das A-Register  
(Akkuulator)

Cw' Schreibsignal für das C-Register  
(Zählregister)

Rw' Schreibsignal für das R-Register  
(Befehlsregister)

Vw' Schreibsignal für den Hauptspeicher

#### 2.8.4. Lesesignale

A Lesesignal vom A-Register

A\* Lesesignal vom A\*-Register

C Lesesignal vom C-Register

R Lesesignal vom R-Register

V Lesesignal vom Hauptspeicher

S1 Lesesignal Sektoradresse

S2 Lesesignal Sektor-Spuradreßzeit

S3 Lesesignal Spur-Befehls-T3-Zeit

#### 2.8.5. Bedienungsfeld

be Taste "ausführen" gedrückt

bq Modus-Taste in "Manuell"

brc Taste "Füllen/Löschen"

bs	Taste "Start"
01	Modus-Taste in "Step"
Tb1-Tb4	Signal der PS-Tasten
Tc	Signal der PST-Taste
To	Signal der E/A-Taste

#### 2.8.6. Ein-/Ausgabe (E/A-Steuerung 1, 80573)

Cd	Code delete
Di	Synchronisierungs-Flip Flop für Tally-Leser
Dt	Tally-Leser Kupplungstreiber signal
Ff	Flexowriteranwahl-Flip Flop
Ft	Tally-Leser-Anwahl-Flip Flop
JL 11	Ende Eingabezyklus Flexowriter (Setzsignal für Ø4)
JL 12	Anfang Eingabezyklus Flexowriter (Setzsignal für Ø3)
JL 29	Rücksetzer für die Translator treiber
P1*...P6*	Zeicheninformation vom Flexowriter
Rs1...Rs6	Zeicheninformation vom 1. Tally-Leser
Rp	One Shot, legt Dauer des Kupplungstreiberimpulses beim 1. Tally-Leser fest
Sc	Stop Code
Sk	Maschinenfunktion (Farbe, oben, unten usw.)
Sr	One Shot, gibt den Flexowriter zur Ein- gabe frei
Sr*	Setzbedingung für das Sr-One Shot
Tp1...Tp6	Treibersignale für die Translator-Magnete des Flexowriters

Tr1	Impuls zu den Abfühlkontakte des Tally-Lesers
Tx	Flexowriter Kupplungstreibersignal
X	Steuer-Flip Flop für die Ausgabe zum Flexowriter
Xp	Astabil Multivibrator, gibt die Dauer des Tally-Leser-Zyklus an
Xs	Signal vom Flexowriter, gibt Ende des Eingabezyklus an
85	Definiert Drucken in $\phi 1$ bei 4 Bit-Mode.



## 2.9. Logische Funktionsgleichungen



Ende WZ 3 in Ø4a + gehe in Ø3 beim Z-Befehl  
 $F' = \underline{F} \underline{G} \underline{H} T3 P1 + \underline{F} \underline{G} \underline{H} T3 Q1 +$   
 Ø4a Ø2 Sprungindikator

+ "ausführen" + gehe in Ø3 (v. E/A-Gerät)  
 + be T3 + Fc  
 "ausführen"

Ende Ø4 + WZ 66, D, M-Befehl +  
 $\underline{F}' = \underline{F} \underline{G} \underline{H} T3 + \underline{F} \underline{G} T3 \underline{P1} +$   
 Ø4 Ø4a

+ WZ 64 (Ende) N-Befehl + Stop Code, Code Delete  
 + F H T3 Q4 K + Fd  
 Ø4a Sektorvergleich  
 - N-Befehl

gehe in Ø4 bei Eingabe +  
 $G' = \underline{F} \underline{G} T3 Q1 Q2 Q3 Q4, be \underline{Faf} +$   
 Ø3 I-Befehl | E/A-Verriegelung  
 nicht "ausführen"

+ gehe in Ø4 nach Sektorvergleich +  
 + F G H T3 K Ga be  
 Ø3 | Faf Sektorvergleich

+ gehe in Ø2 nach Sektorvergleich, falls kein Stop +  
 + G H T3 K Ga Q2 be bs +  
 Ø1 Sektorvergl. | Faf | | nicht Start  
 kein Stop | nicht "ausführen"

+ Ende WZ 64, D,M-Befehl +  
 + G H T3 K Ga Q4 be +  
 Ø4a | Faf | | nicht "ausführen"  
 Sektorvergleich D,M-Befehl

+ gehe in Ø4 nach 1 WZ, U,T-Befehl +  
 + F G T3 Q1 Q2 Q3 be +  
 Ø3 U,T-Befehl | | nicht "ausführen"

+ gehe in Ø4 nach 1 WZ bei Z,P-Befehl Ende Eingabezyklus  
 + F G T3 Q2 Q3 Q4 be Faf + Gc (gehe in Ø4)  
 Ø3 Z,P-Befehl | | E/A-Verriegelung  
 nicht "ausführen"

Ende Ø4 nach 1 WZ + Ende Ø2 nach 1 WZ oder Ende D-Befehl +  
G' = G H T3 + F G T3 +  
 Ø4 Ø2 oder WZ 67 beim D-Befehl

+ Ende WZ 66 beim M-Befehl +  
 + G H P1 T3 Q3 + Gd  
 Ø4a M-Befehl

gehe in Ø4a nach 1 WZ Ø4 beim M,N,D-Befehl  
H' = F G H T3 Q1 Q2 α9  
 Ø4 (3 + Q4)  
 M,N,D-Befehl

Ende WZ 67 beim D-Befehl + Ende WZ 64 beim N-Befehl +  
H' = F G H T3 + H T3 K Q4 +  
 Ø4a

Ende WZ 66 beim M-Befehl  
 + β6  
 = G H T3 Q3 P1

Ende WZ 66 beim M-Befehl  
β6 = G H T3 Q3 P1  
 Ø4a M-Befehl

Stop (Anzeige)  
Kc = F G Q2  
 Ø1 Stop-Indikator

Übernehme V in Ø2 + Übernehme V in Ø4 beim M,N,D-Befehl +  
Rw' = F G H V + G H V Q1 Q2 α9 +  
 Ø2 Hauptspeicher- Ø4 (Q3+Q4)  
 information M,N,D-Befehl

Übernehme A wenn "Füllen/Löschen" gedrückt +  
 + A brc +  
 Füllen/Löschen  
 Akkumulatorinhalt

Umlauf von R in Ø1, Ø1, Ø4a und in Ø4 bei allen Befehlen außer M,N,D-Befehl

+ R brc [ G + H + F ( Q1 + Q2 + Q3 Q4 ) ]  
 Ø1, Ø3 Ø4a Ø4 I-Befehl  
 nicht Füllen/Löschen Z,B,Y,R-Befehl  
 R-Umlauf P,E,U,T,H,C,A,S-Befehl

addiere 1 zum Zähler im C beim R-Befehl +  
 $Vw' = \underline{L} (\underline{Q1} \underline{Q4} \underline{K} \underline{C} + \underline{Q1} \underline{Q4} \underline{K} \underline{C} +$   
 R-Befehl R-Befehl  
 schreibe 0 zur Spacerzeit Addition von 1

übernehme A beim C,H-Befehl + übernehme A beim Y-Befehl  
 $+ Q1 \underline{A} + \underline{Q4} \underline{A} )$   
 A-Inhalt A-Inhalt  
 C,H-Befehl Y-Befehl

Schreiberlaubnis in Ø4 beim H,C-Befehl +  
 $W = \underline{F} \underline{G} \underline{H} \underline{Q1} \underline{Q2} \underline{Q3}$  +  
 Ø4 H,C-Befehl

Schreiberlaubnis zur Adresszeit in Ø4 beim Y,R-Befehl  
 $+ \underline{F} \underline{G} \underline{S2} \underline{Q1} \underline{Q2} \underline{Q3}$   
 Ø4 Y,R-Befehl  
 Adresszeit

übernehme C in Ø1 + übernehme C in Ø4a + übernehme R in Ø3  
 $r1 = \underline{F} \underline{C} + \underline{H} \underline{C} + \underline{F} \underline{H} \underline{R}$   
 Ø1 Ø4 Ø3

Flexowriter-Anwahl + Tally-Anwahl + usw.  
 $Faf = Ff + Ft + F(n)$

Eingabe von Hand + Setze K zu jeder T3-Zeit in Ø1, Ø2, Ø4 und  
 $K' = bQ + T3 \underline{Faf} (\underline{F} + \underline{G} + \underline{R} + \underline{Q1+Q2+Q3+Q4})$   
 Manuell Ø1, Ø2 Ø4 jeder Befehl außer +I  
 -Befehl

bei einem -Befehl, nicht beim +E/A-Befehl  
 $\cdot (\underline{F} + \underline{G} + \underline{R} + \underline{Q1} + \underline{Q2} + \underline{Q3} + \underline{Q4})$   
 Ø1, Ø2 Ø4 jeder Befehl außer +P  
 -Befehl

$\underline{K}' =$  (siehe Rechenwerk 80571, Seite A2-113)

Schalte die P-Flip Flops in Ø4 als Schiftregister für

Ac1 =  $\frac{F \quad G}{Ø4} \quad \underline{Q1} \quad BL \quad (Faf + \underline{Q1 \quad Q2 \quad Q3 \quad Q4})$  •  
I-Befehl n.U. E/A-Mode I- und Schiftbefehl

4- oder 6-Bit Mode

• (  $K \quad P4$  +  $K \quad P6$  )  
[ Übernehme P4 ] [ übernehme P6 (6 Bit) ]  
-(I)Befehl +(I)Befehl

Schalte die P-Flip Flops in Ø4 als Schiftregister bei einem I-Befehl

ic =  $\frac{F \quad G}{Ø4} \quad BL \quad (Faf + \underline{Q1 \quad Q2 \quad Q3 \quad Q4})$   
n.U. E/A-Mode I- und Schiftbefehl

P1d = P2d = P3d = P4d = P5d = P6d =

Rücksetzer für P's beim I-Schiftbefehl zur T3-Zeit Ø3

=  $\frac{F \quad G}{Ø3} \quad T3 \quad \underline{Q1 \quad Q2 \quad Q3 \quad Q4}$ , be Faf  
I- und Schiftbefehl E/A-Verriegelung  
nicht "ausführen"

E/A-Verriegelung

P1b = P5a = P6a = ia = Ga = Faf  
E/A-Verriegelung

Vorzeichenzeit

T3 = S1 S3

übernehme V beim B-Befehl, und V,A beim E-Befehl				+		
$Aw' =$	F	G	H	<u>Q2 Q3 Q4</u> V      ( <u>Q1 + A</u> )		
	<u>Ø4</u>		B,E-Befehl			
+ A-Umlauf in Ø1...Ø3, in Ø1...Ø4 beim P,H,U,T,Y,R,Z-Befehl				+		
+ H	A	To	[ F + G + Q1 <u>Q3 Q4</u> ]	+		
	nicht Ø4a		E/A nicht Ø1, Ø2      Ø3			
+ und beim D,N,M-Befehl außer Vorzeichen				+		
+ T3	Q1	Q2	a9	+ <u>Q2</u> (Q3 + Q4)      Faf ]		
	D,N,M-Befehl		(Q3+Q4)      U,T,Y,R,Z-Befehl			
+ E/A-Verriegelung						
+ Schiftregister in Ø4 bei Eingabe				+		
+ Ac1				+		
+ übernehme die Addierlogik						
+ $\beta_1$	[ (L <u>I1 I2</u> + L <u>I1 I2</u> + L <u>I1 I2</u> + L <u>I1 I2</u> ) ]					
	s. nächste Gleichung		Addierlogik			
Ø4a keine Vorzeichenzeit				+		
$\beta_1 =$	H	T3		Ø4a gerade WZ		
			H P1			
+ WP 67, D-Befehl				+		
+ F	G	H		Ø4 A,S-Befehl		
			F G Q1 Q2 Q3			
	Ø4		A,S-Befehl			
+ Adressenumlauf für alle Befehle außer U				+		
$Cw' =$	F	S2	C	<u>brc</u> ( Q1+Q2+Q3+Q4 )		
	Ø3, Ø4		nicht Fü/Lö      kein U-Befehl			
	Adreßzeit					
+ Adressenumlauf in Ø1, Ø3				+ C-Umlauf in Ø4a		
+ G	S2	C	<u>brc</u>	+		
	Ø1, Ø3		nicht Fü/Lö			
	Adreßzeit					
+ H C <u>brc</u> nicht Fü/Lö				+		
+ G S2 C <u>brc</u> ( F + T3 + R +				+		
	Ø2, Ø4		nicht Fü/Lö			
	kein -Befehl					
	nicht Adreßzeit					
+ C-Umlauf außer Adresse, außer Vorzeichenzeit, außer -Z-Befehl						
+ G S2 C <u>brc</u> ( F + T3 + R +						
	Ø2, Ø4		nicht Fü/Lö			
	kein -Befehl					
	nicht Adreßzeit					

+ Q1 + Q2 + Q3 + Q4, + Faf + To ) +  
 alle Befehle außer Z E/A-Verriegelung E/A-Taste

+ Addiere 1 zur Adresse in C in Ø2 +  
 + F G H S2 C K + F G H S2 C K +  
 Ø2 Adreßzeit Ø2 Adreßzeit  
 Addiere 1

+ Vorzeichenumlauf Ø1, Ø3 + übernehme die 2te Sektoradresse  
 + G T3 C brc + G H S1 S2 S3 +  
 Ø1, Ø3 nicht Fü/Lö Ø1, Ø3 2te Sektoradresse

+ Setze das Überlaufbit zur Vorzeichenzeit Ø4 A,S-Befehl +  
 + F G T3 Q1 Q2 Q3 ( $\alpha_5 + \alpha_6 + \alpha_7 + \alpha_8$ ) +  
 Ø4 A,S-Befehl normale Addierlogik s. A2-115

+ Übernehme R in Ø4 beim U-Befehl +  
 + F G H T3 Q1 Q2 Q3 Q4 R +  
 Ø4 U-Befehl

+ Setze das Überlaufbit zur Vorzeichenzeit Ø4 beim D-Befehl in WP 2  
 + F G H T3 Q3 P1 P6 A +  
 Ø4a, 2.WP Division gibt Überlauf an

siehe vorhergehende Zeile  
 + F G H T3 Q3 P1 P6 A  
 Ø4a, 2.WP Division gibt Überlauf an

I1 = H A + F G P1 Q3 A +  
 nicht Ø4a WZ 2 D-Befehl  
 übernehme A + D-Befehl: übernehme in WP 2 den A (Dividend)

und Vorzeichen (P6)  
 + F G P1 S1 S3 Q3 P6 +  
 WZ 2 = T3 Vorzeichen  
 D-Befehl

+ übernehme A\* in WP 3,5...63 + übernehme A\* in WP 65, 66 +  
 + G H P1 Q3 A\* + F H Q3 A\* +  
 WZ 3,5...63 D-Befehl WZ 65,66 D-Befehl

+ übernehme A\* in WP 65...67 bei -Divisor + übernehme A\* in WP 67 bei +Divisor  
 + G H Q3 P5 A\* + F G H A\* P5 +  
 WZ 65-67 Divisor - WZ 67 Divisor +  
 D-Befehl

+ D,M,N-Befehl: WP 4,6...64 + M,N-Befehl: übernehme A\* in WP 3,5...65 +  
 + F G H A\* + H P1 Q3 A\* +  
 WZ 4,6...64 WZ 3,5...65 M,N-Befehl

+ übernehme A in WP 66  
 + F G P1 Q3 A  
 WZ 66 M-Befehl

I2 = H V + D-Befehl: übernehme in WP 2,4...64 R (Divisor) +  
 nicht Ø4a + H P1 Q3 R +  
 WZ 2,4...64 D-Befehl

+ D,M,N-Befehl: Aufrundung (Subtrahiere 1) in WZ 67 +  
 + F G H P5 P6 + F G H P5 P6 +  
 WZ 67 Subtraktion von 1

+ M,N-Befehl: übernehme R in WZ 2,4...64 + übernehme Vorzeichen des Multiplikanden  
 + G H P1 P6 R + F H P1 Q3 P5 + in WP 3  
 WZ 2,4...64 Multiplizierbit WZ 3 Vorzeichen  
 M,N-Befehl

+ übernehme 1 bei gleichen Vorzeichen + übernehme A in WZ 66  
 + H P1 Q3 P5 P6 + G H P1 Q3 A  
 WZ 5,6-65 Vorzeichen WZ 66 M(N)-Befehl  
 M,N-Befehl

K' = gibt die erste 0 in C an, für Addition von 1 / R-Befehl +  
 G H S2 C Faf ( F + Q3 ) +  
 Ø2,Ø4 E/A-Verriegelung Ø2 R-Befehl  
 gelung

+ keine Sektorkoordinzidenz in Ø4a, M,N,D-Befehl +  
 + H S2 S1 r1 + H S3 S2 S1 r1 +  
 Ø4a =C Ø4a 2.Sektoradr. =C  
 2.Sektoradresse Sektorvergleich

keine Sektorübereinstimmung in Ø1 und Ø3 +  
 + G H S3 S2 S1 r1 Faf + G H S3 S2 S1 r1 Faf +  
 Ø1, Ø3 Sektorzeit E/A-Verriegel. Ø1, Ø3 Sektorzeit E/A-Verriegel.  
 ≡C in Ø1; ≡R in Ø3 ≡C in Ø1; ≡R in Ø3  
 Sektorvergleich

+ 6 Bit Mode bei +P-Befehl +  
 + F G T3 R Q1 Q2 Q3 Q4 Faf +  
 Ø3 Druckbefehl  
 kein negativer Befehl

+ 6 Bit Mode bei + I-Befehl  
 + F G T3 R Q1 Q2 Q3 Q4 Faf  
 Ø3 I-Befehl E/A-Verriegelung  
 kein -Befehl

Setzer für K' = siehe A2-109

für Spacerbit beim H,C-Befehl + erlaubt Schreiben des 1. Bits +  
 L' = W + F G Q2 +  
 Schreiberlaubnis Ø4 Y, R-Befehl

+ nicht Vorzeichen und gerade WZ bei M,N,D  
 $\beta_3$  (α5 + α7)  
 = T3 + H Q3 P1      L = S 11 12 L      = S 11 12 L  
 Addition Subtraktion

T3-Zeit Ø1...Ø4, D-Befehl, ungerade WZ M,N-Befehl +  
 L' =  $\beta_3$  +  
 T3 (H + Q3 + P1)

+ Übertragslogik  
 Q2 (Q1 + Q3) (α6 + α8)  
 D,M,N,A,S - Befehl      L = S 11 12 L      = S 11 12 L  
 Subtraktion Addition

Subtrahiere + bei D-Befehl bei Vorzeichengleichheit  
 S = H Q4 + H Q3 P5 P6 + H Q3 P5 P6 +  
 Ø4a D-Befehl Ø4a D-Befehl Vorzeichengleichheit

M,N-Befehl WZ 2,3 + D-Befehl WZ 67  
F H Q3 + F G H  
 WZ 2,3 M,N-Befehl WZ 67

$\alpha_5 = \underline{S} \quad \underline{I1} \quad \underline{I2} \quad \underline{L}$

A-Befehl Addierer-Eingänge

$\alpha_6 = \underline{S} \quad \underline{I1} \quad \underline{\underline{I2}} \quad \underline{L}$

S-Befehl Addierer-Eingänge

$\alpha_7 = \underline{S} \quad \underline{\underline{I1}} \quad \underline{I2} \quad \underline{L}$

S-Befehl

$\alpha_8 = \underline{S} \quad \underline{\underline{I1}} \quad \underline{\underline{I2}} \quad \underline{L}$

A-Befehl

$\beta_3 = \underline{T3} + H \quad Q3 \quad \underline{P1}$

Schalte die P-Flip Flops als Schiftregister

$$i = ia \underline{G} \underline{H} S2 S3 + ic$$

Faf    Ø1, Ø3    Spurzeit

s. Phasensteuerung A2-110

$$P1' = \underline{G} i r1 + \underline{G} i A +$$

Ø1, Ø3    Ø1=C, Ø3=R

+ übernehme C oder R + übernehme A beim I,P-Befehl +

+ H P1 T3 + P1c

Ø4a

$$P1' = \underline{G} i \underline{r1} + \underline{G} i \underline{A} +$$

Ø1, Ø3    Ø1=C, Ø3=R

+ gerade WZ, M,N,D-Befehl + Rücksetzer von Eingabegerät +

+ H P1 T3 + P1d

Ø4a

+ 1. WZ, M,N,D-Befehl

+ G P1 T3 P1b

Ø4                  Faf

$$P2' = i P1 + \text{Setzer von Eingabegerät}$$

Schifftsignal

$$P2' = i \underline{P1} + \text{Rücksetzer v. Eingabegerät}$$

Schifftsignal

$$P3' = i P2 + \text{Setzer v. Eingabegerät}$$

Schifftsignal

$$P3' = i \underline{P2} + \text{Rücksetzer v. Eingabegerät}$$

Schifftsignal

übernehme P3 + Setzer v. Eingabegerät  
 $P4' = i \underline{P3} + P4c$   
 Schifftsignal

übernehme P3 + Rücksetzer v. Eingabegerät  
 $\underline{P4}' = i \underline{P3} + P4d$   
 Schifftsignal

übernehme P4 + Setzer v. Eingabegerät  
 $P5' = i \underline{P4} + P5c$   
 Schifftsignal

Vorzeichen von V bei M,N,D-Befehl + Setze P5 bei 4 Bit Ausgabe  
 $+ G \underline{H} T3 V P5a + \beta 5$   
 $\underline{\theta 4} \quad \text{Hauptsp.} \quad \underline{Faf} \quad \text{siehe weiter unten}$   
 (E/A-Verriegelung)

übernehme P4 + Rücksetzer v. Eingabegerät +  
 $\underline{P5}' = i \underline{P4} + P5d$   
 Schifftsignal

Vorzeichen von V bei M,N,D-Befehl  
 $+ G \underline{H} T3 V P5a$   
 $\underline{\theta 4} \quad \text{Hauptsp.} \quad \underline{Faf} \quad (\text{E/A-Verriegelung})$

übernehme P5 + Setzer v. Eingabegerät +  
 $P6' = i \underline{P5} + P6c$   
 Schifftsignal

Vorzeichen von A in  $\theta 4$  bei M,N,D-Befehl +  
 $+ F G \underline{H} T3 A P6a$   
 $\underline{\theta 4} \quad \text{Akku} \quad \underline{Faf} \quad (\text{E/A-Verriegelung})$

Vorzeichen v. A (laufender Rest) D-Befehl + Multiplizierbit beim N,M-Befehl  
 $+ H T3 P1 \underline{Q3} A + H T3 P1 \underline{Q3} A^*$   
 $\underline{\theta 4a} \quad \text{D-Befehl} \quad \underline{\theta 4a} \quad \text{M,N-Befehl}$

$P6' = i \underline{P5}$  + Rücksetzer v. Eingabegerät +  
 Schiftsignal

+ Vorzeichen v. A in Ø4 bei M,N,D-Befehl +  
 + F G H T3 A P6d +  
 Ø4 Akku | Faf (E/A-Verriegelung)

+ Vorzeichen v. A (laufender Rest) D-Befehl +  
 + H T3 P1 Q3 A +  
 Ø4a | D-Befehl

+ Multiplizierbit beim M,N-Befehl +  
 + H T3 P1 Q3 A\* +  
 Ø4a | M,N-Befehl

+ Rücksetzer bei 4 Bit Ausgabe  
 + β 5  
 | s. weiter unten

$Q1' = \alpha_{11} R + G H T3 Q1 Q2 Q3 Q4 Faf To$   
 | s. weiter | Ø4 | Z-Befehl | E/A nicht gedrückt  
 unter | | | | E/A-Verriegelung

- Z-Befehl, Überlaufbit  
 $(R : C + P1 Tb1 + P2 Tb2 + P3 Tb3 + P4 Tb4)$   
 Überlauf PS-Tasten Koinzidenz  
 - Z-Befehl

$Q1' = \alpha_{11} R + bq + G H T3 Q1$   
 | s. weiter unten | Ø2, Ø4

$Q2' = \alpha_{11} Q1 + F G H bs$   
 | s. weiter unten | Ø1 | "Start"

+ Setzer für Q2 für alle Befehle außer Z0000, Z0100  
 + F G T3 Ø1 Q2 (Q1+Q3+Q4+P1+P2+P3+P4+P5+P6)  
 Ø4 nicht "Step" alle Befehle außer Z0000, Z0100

Übernehme Q1 + Stop, nach jeder Befehlsausführung in Step +
   
 $\underline{Q2}' = \alpha_{11} \underline{Q1} + \underline{H}' 01 + G T3 Q1 01 +$   
 ↓ s. weiter unten ↓ Step ↓ Step  
 Ende D,M,N-Befehl P,E,U,T,H,C,A,S-Befehl

+ Stop, nach Eingabe in Step oder Manuell + E/A-Rücksetzer  
 + G T3 Q1 Q2 Q3 Q4 01 + To  
 ↓ 04 Eingabe

Übernehme Q2 + Druckbefehl  
 $\underline{Q3}' = \alpha_{11} Q2 + G T3 Q1 \underline{Q2} \underline{Q3} \underline{Q4}$   
 ↓ s. weiter unten ↓ 04 Druckbefehl

Übernehme Q2 + Manuell Eingabe  
 $\underline{Q3}' = \alpha_{11} Q2 + bq$   
 ↓ s. weiter unten

Übernehme Q3  
 $\underline{Q4}' = \alpha_{11} Q3$   
 ↓ s. weiter unten

Übernehme Q3 + Manuell Eingabe +  
 $\underline{Q4}' = \alpha_{11} \underline{Q3} + bq +$   
 ↓ s. weiter unten

+ Testbefehl, A negativ + - Testbefehl  
 + T3 Q1 Q2 Q3 A + T3 Q1 Q2 Q3 R Tc  
 ↓ T-Befehl ↓ T-Befehl PST-Taste  
 ↓ A negativ ↓ - T-Befehl

$\alpha_9 = Q3 + Q4$

Schalte Q-Flip Flops als Schiftregister  
 $\alpha_{11} = F \underline{G} \underline{H} \underline{S1 S2 S3}, bq Faf$   
 ↓ 03 Befehlszeit ↓ E/A-Verriegelung  
 nicht "Manuell"

$\beta_5 = \underline{F} \underline{G} K Q3 Faf$   
 ↓ 01 Ausgabe ↓ E/A-Verriegelung  
 4 Bit Mode

Kommandowerk (80572)

gehe in Ø3 (Flexowriter) +  
 Fc = JL12 T3 Sc Sk +  
 vom Flexowriter | Maschinenfunktion  
 kein Stop-Code

+ gehe in Ø3 (Tally Leser)  
 + F G T3 Rp Sc Cd Ft Sk  
 Ø1 | kein Code | kein Maschinenzeichen  
 delete | Tally-Anwahl Flip Flop  
 kein Stop Code

gehe nicht in Ø3 (Flexowriter und Tally Leser)  
 Fd = F G T3 (Sc + Cd + Sk) (Ff + Ft)  
 Ø3 Stop Code Code delete | Flexowriteranwahl- Tally Anwahl  
 Maschinenfunktionen Flip Flop Flip Flop

gehe in Ø4 (Flexowriter) + gehe in Ø4 (Tally Leser)  
 Gc = JL11 F G T3 + F G T3 Rp Ft  
 vom Flexo- Ø3 | Tally Anwahl Flip Flop  
 writer Ø3 | Tally Eingabe-Ende

Flexowriteranwahl beim (800) P0200 - Befehl +  
 Ff' = F G T3 P1 P2 P3 P4 P5 Q1 Q2 Q3 Q4 be Faf +  
 Ø3 0200 | P-Befehl

+ Flexowriteranwahl beim (800) I0200 - Befehl +  
 + F G T3 P1 P2 P3 P4 P5 Q1 Q2 Q3 Q4 be Faf +  
 Ø3 0200 | I-Befehl | E/A-Verriegelung  
 nicht "ausführen"

Manuell  
 + b Q

Rücksetzer bei Eingabe-Ende +  
 Ff' = F G T3 Q3 Xs Sr' bq +  
 Ø1 Eingabe Eingabe- | nicht "Manuell"  
 Ende | Flexowriter Eingabe frei

+ Druckende + E/A-Taste  
 + X + To

Tally Leser (1)-Anwahl beim 10000-Befehl

Ft' = F G T3 P1 P2 P3 P4 P5 Q1 Q2 Q3 Q4 be Faf  
03 0000 I-Befehl E/A-Verriegelung  
nicht "ausführen"

Rücksetzer für Tally beim Stop Code + E/A-Taste

Ft' = F G T3 Sc Xp + To  
01 Stop Code Multivibrator  
(Zyklus-Zeit)

Rp = Dt = Xp Di  
Start Eingabe Tally Synchron.  
Multivibrator (Zykluszeit)

Stop - Code

Sc = P1 P2 P3 P4 P5 P6

Code - delete

Od = P1 P2 P3 P4 P5 P6

Synchronisierung Tally

Di' = F G Xp Ft  
01 Multivibrator (Zykluszeit)

Di' = Faf

E/A-Verriegelung

Vorbereitung Flexowriter Eingabe

Sr\* = F Q1 Xs bQ Ff  
04 I-Befehl nicht "Manuell"  
nicht in Eingabe-Zustand

Ausgabebefehl zum Flexowriter

X' = F G T3 Tx JL33 Q3 Ff  
01 Kupplungstreiber Ausgabe-Indikator  
nicht gesetzt Translator Cam

X' = F + G + T3 + Tx + JL33 + Q3 + Ff  
03,04 02 Translator- Translator Ausgabe- Flexowriter  
Kupplung Cam Indikator nicht ange- wählt

Drucke P1-Bit

Tp1 = X P1

Drucke P2-Bit

Tp2 = X P2

Drucke P3-Bit

Tp3 = X P3

Drucke P4-Bit

Tp4 = X P4

Drucke P5-Bit

Tp5 = X P5

Drucke P6-Bit

Tp6 = X P6

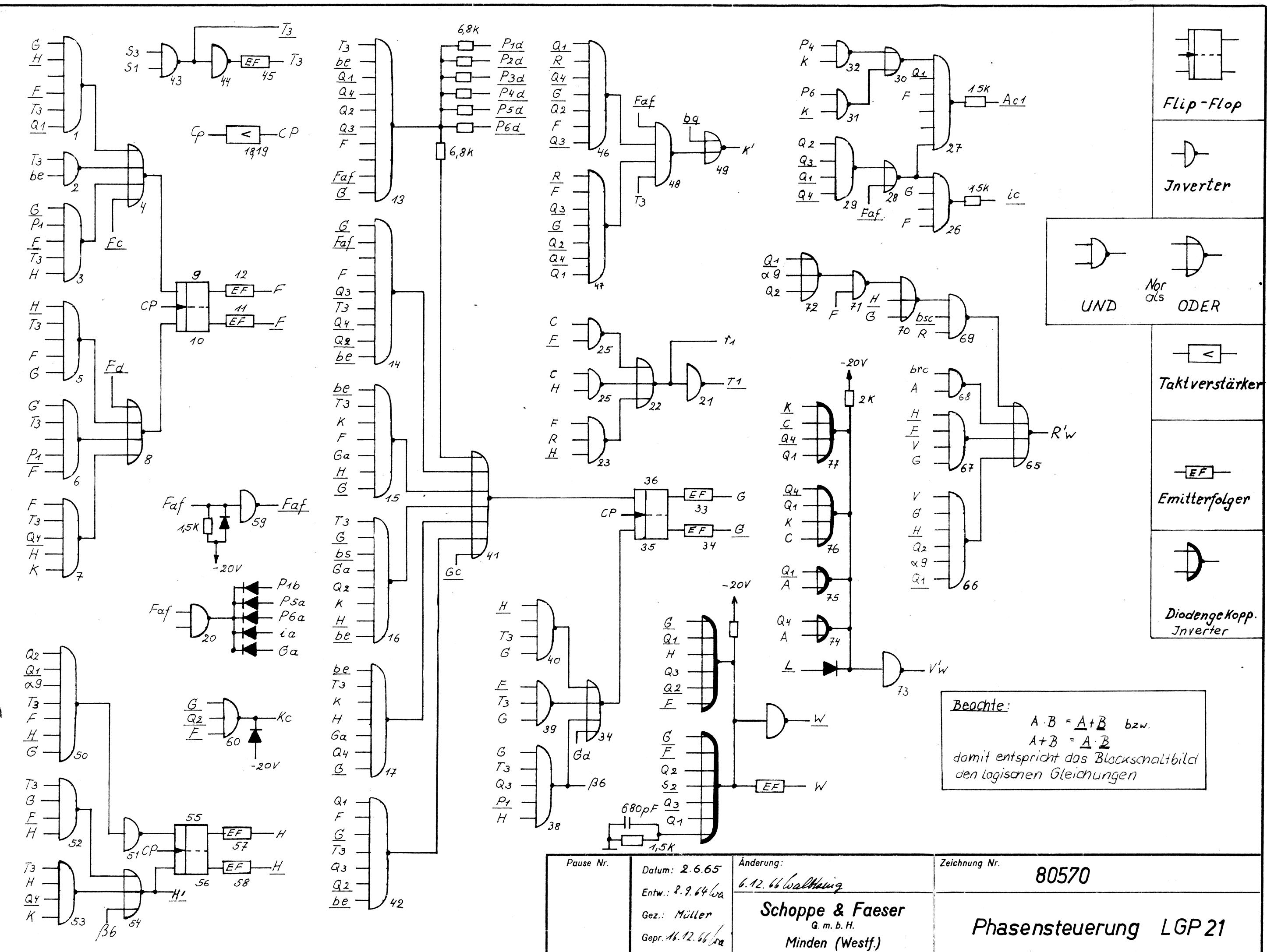
Drucke

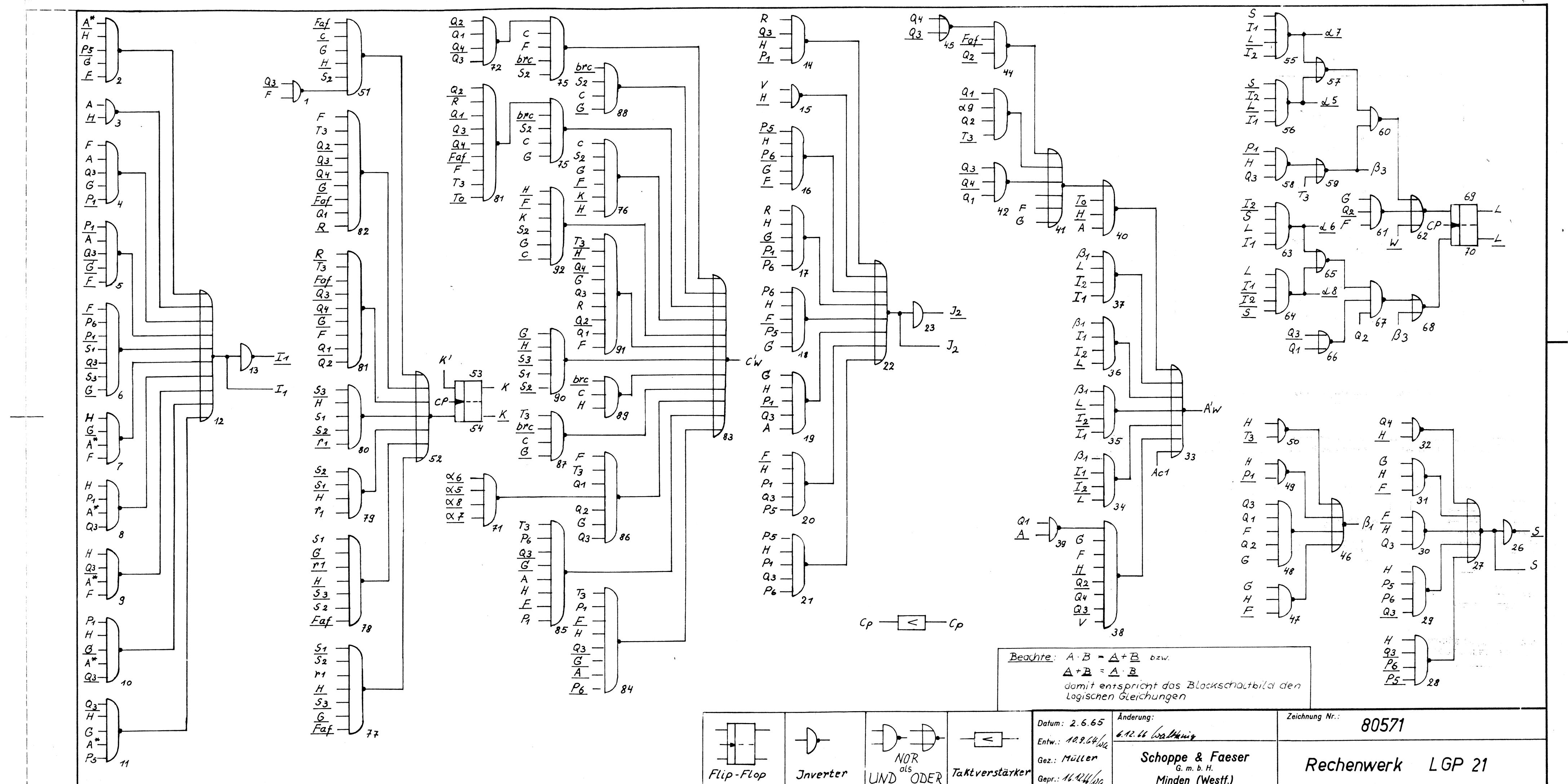
Tx = X = JL7

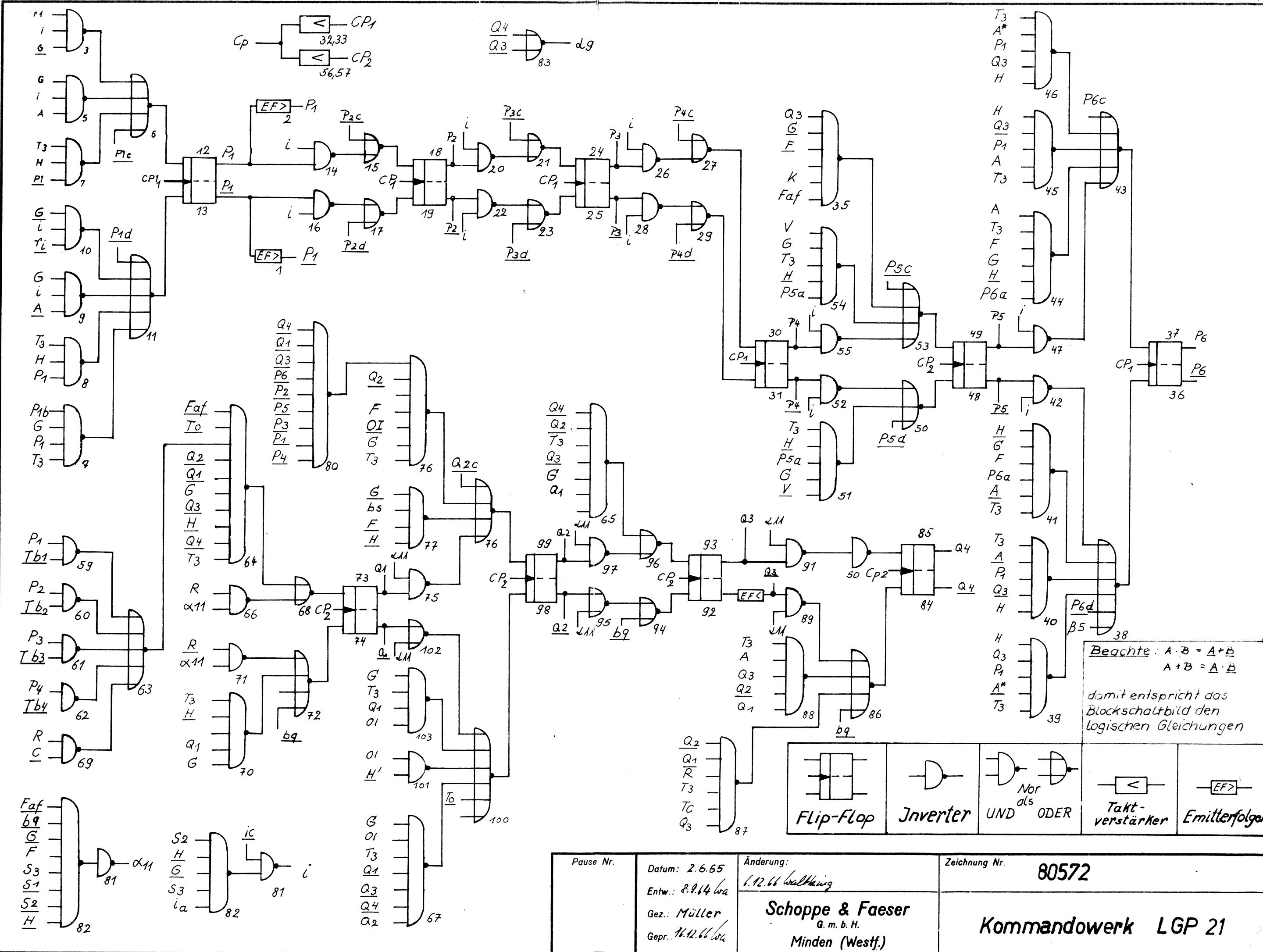
Maschinenezeichen

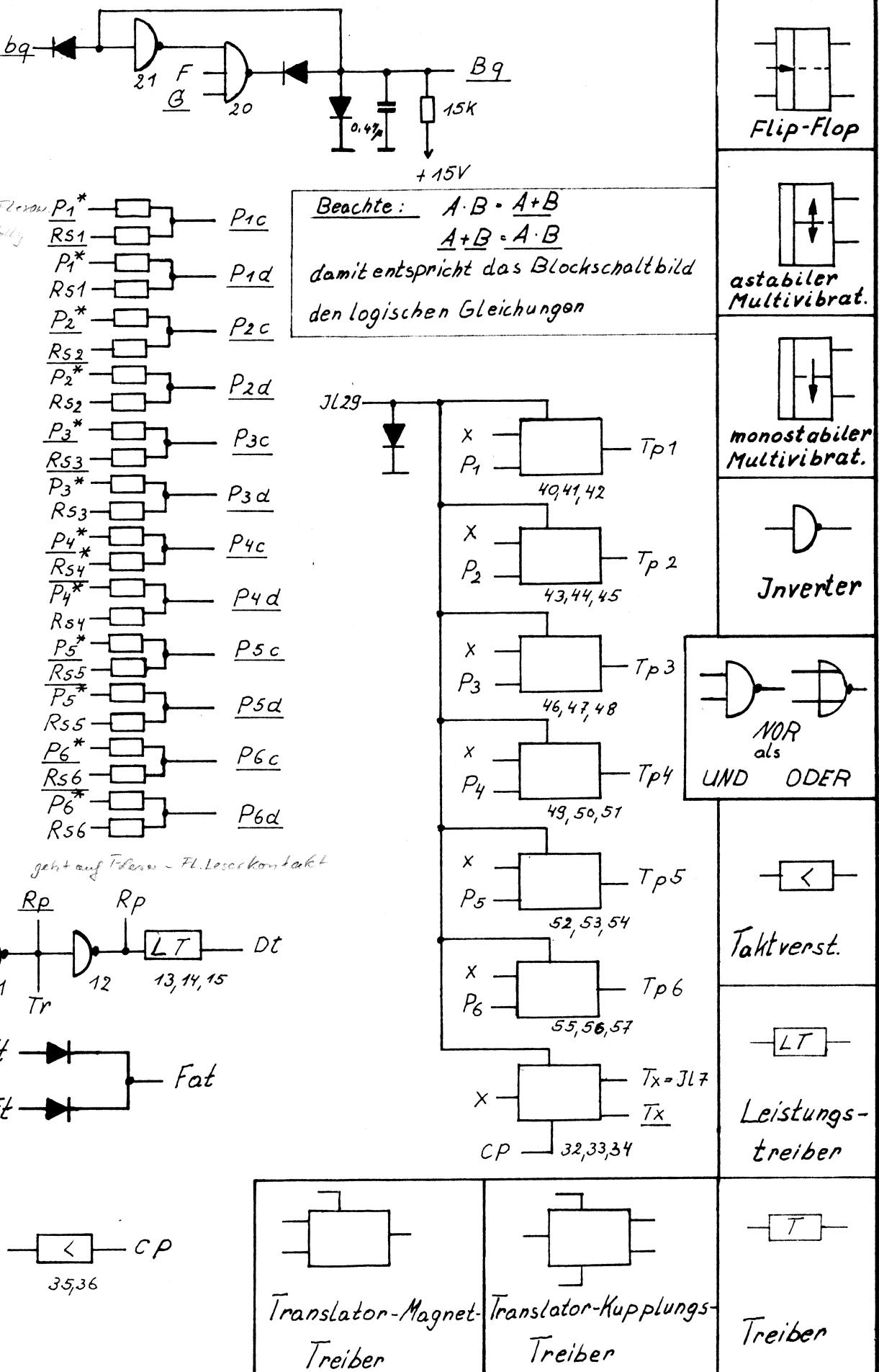
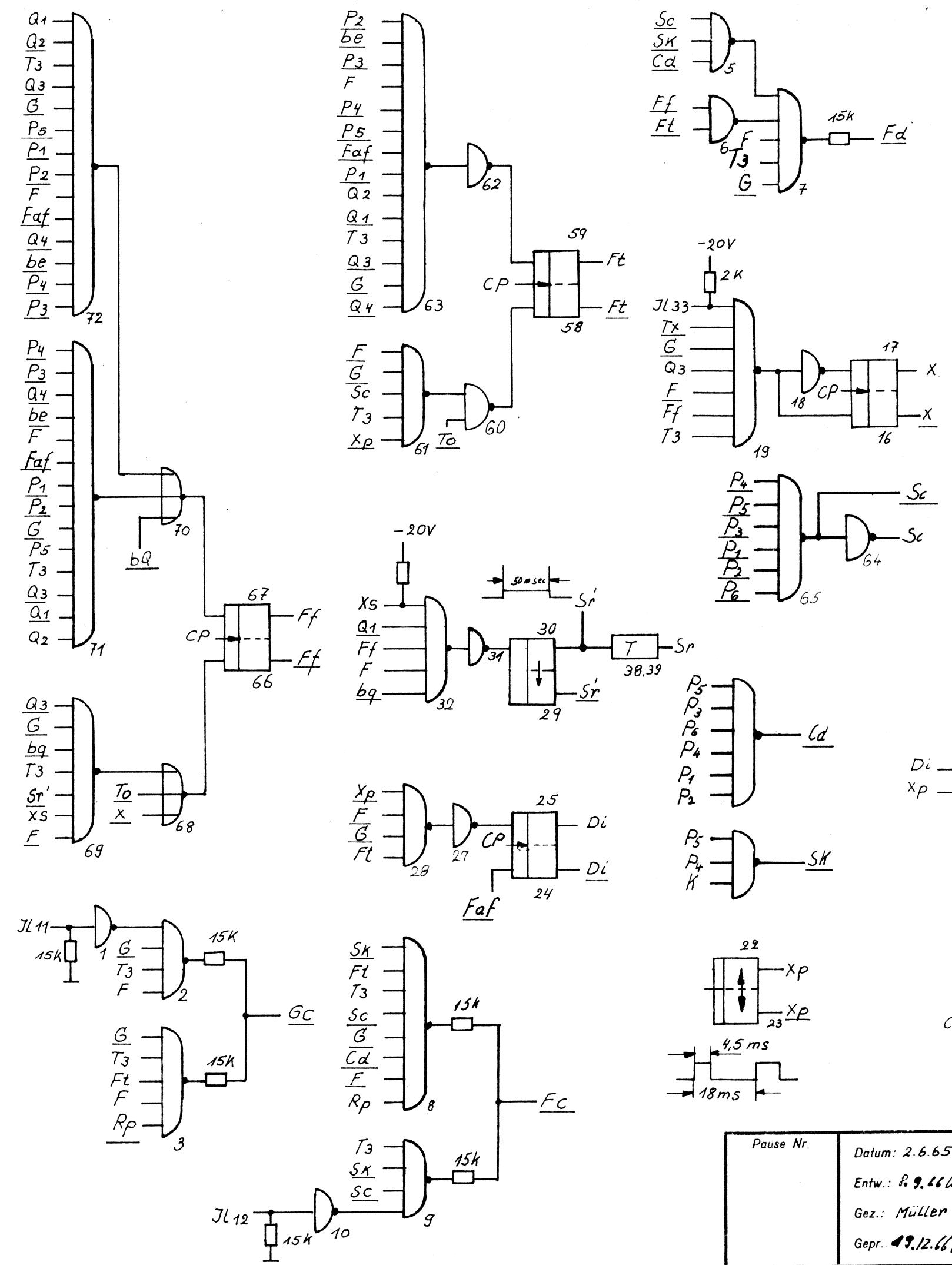
Sk = P5 P6 K

4 Bit Modus

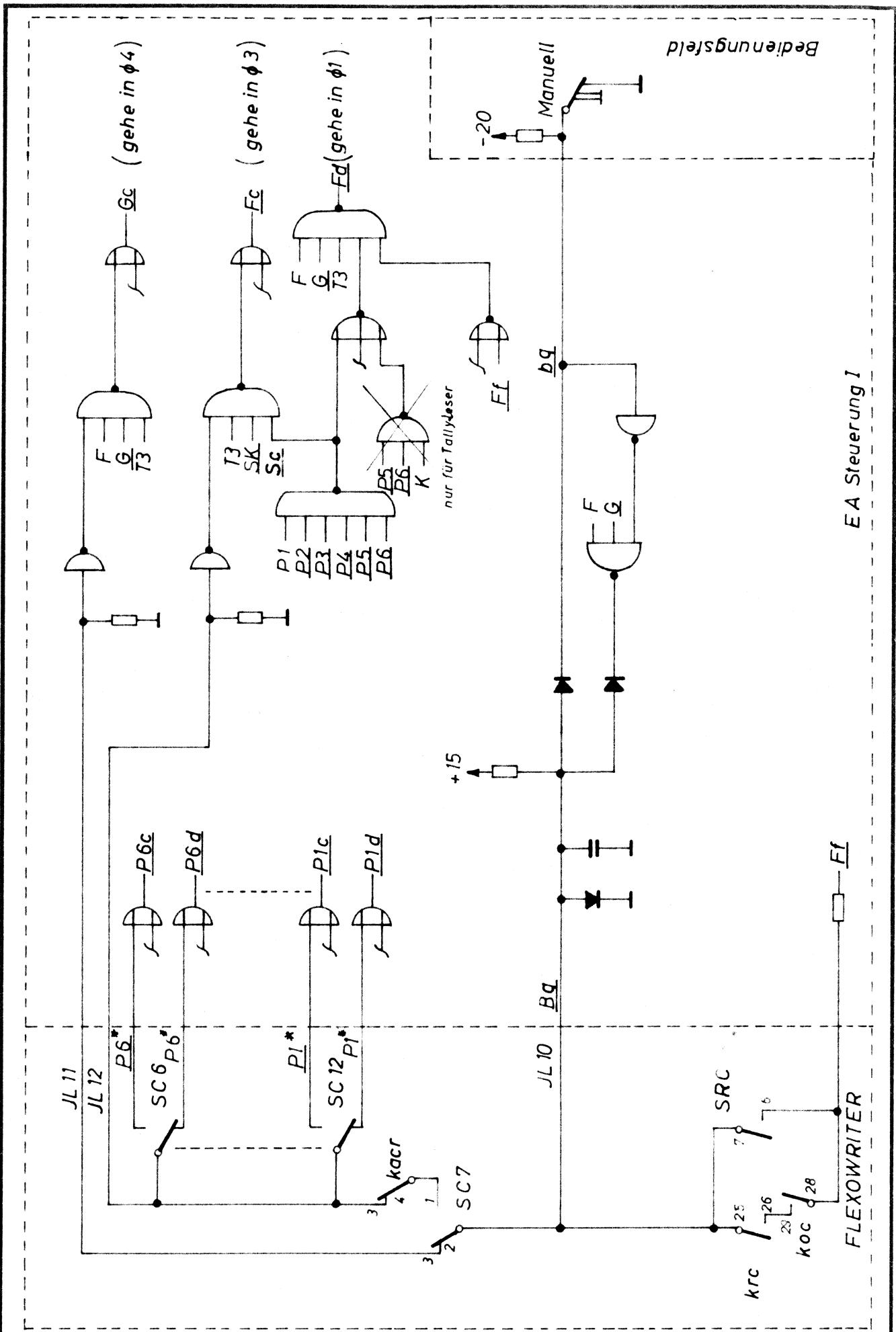




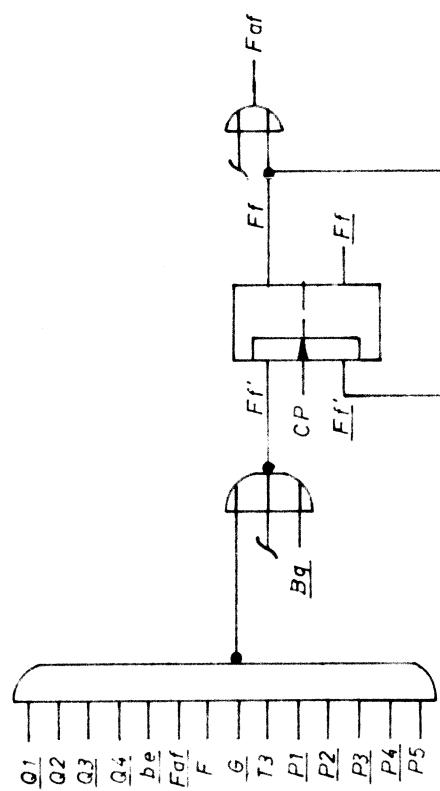




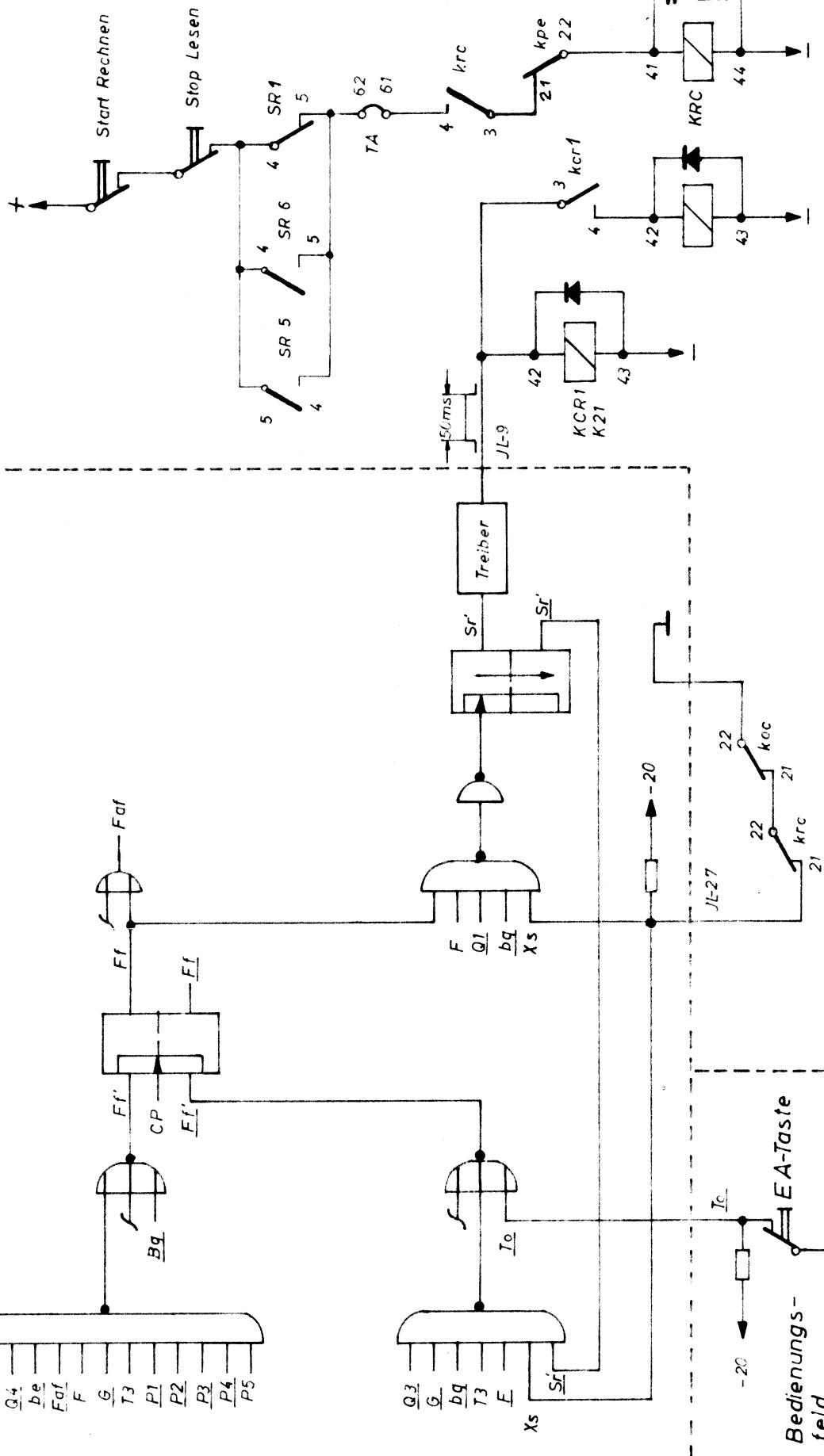
Pause Nr.	Datum: 2.6.65 Entw.: 8.9.66 6a Gez.: Müller Gepr.: 19.12.66 6a	Änderung: 6.12.66 Wartung	Zeichnung Nr. 80573
	Schoppe & Faeser G. m. b. H. Minden (Westf.)		EA - Steuerung 1 LGP 21



### EA Steuerung I



### Flexowriter



8.12.1966

wa

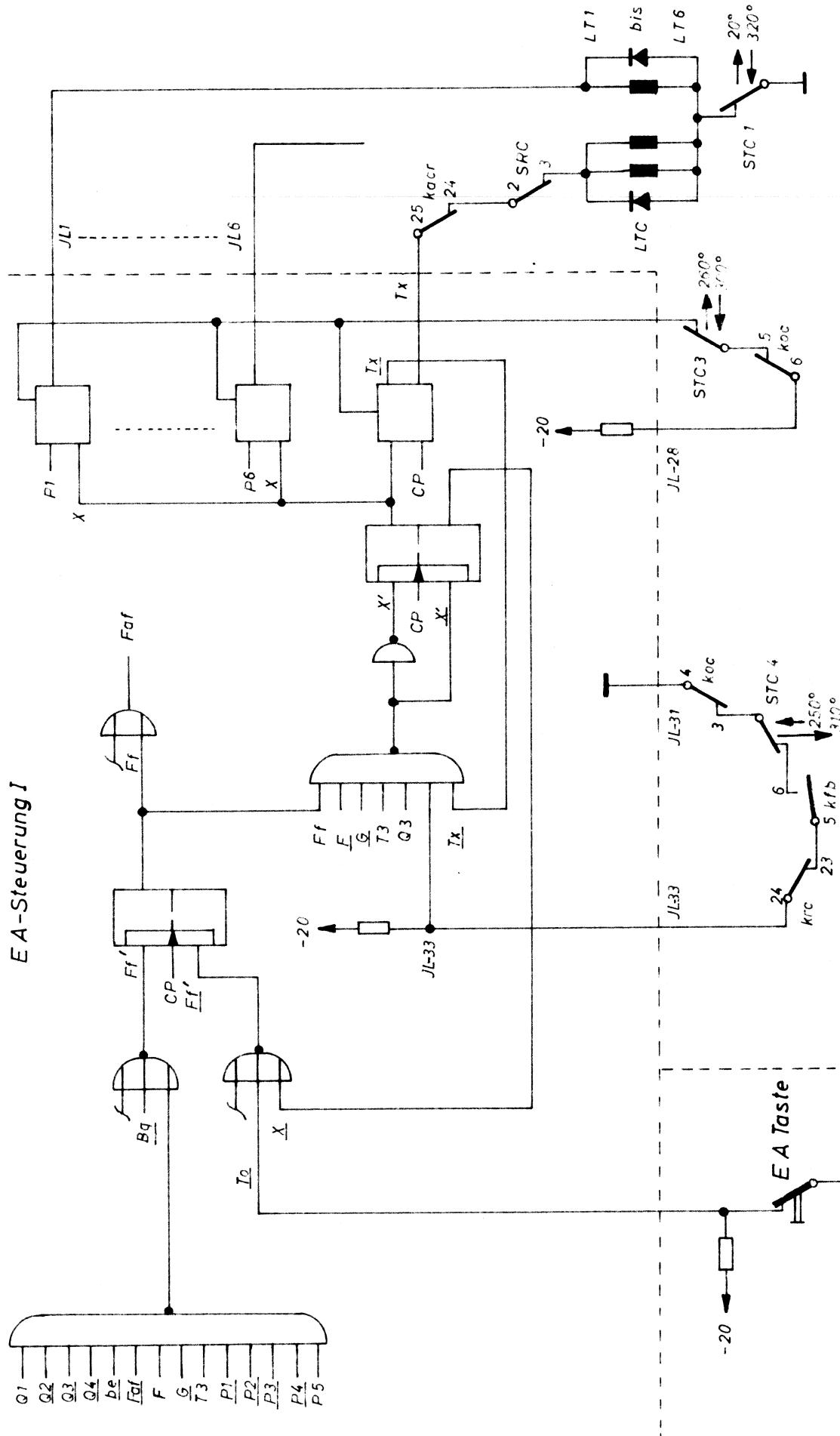
Schoppe & Faeser  
G. m. b. H.  
Minden (Westf.)

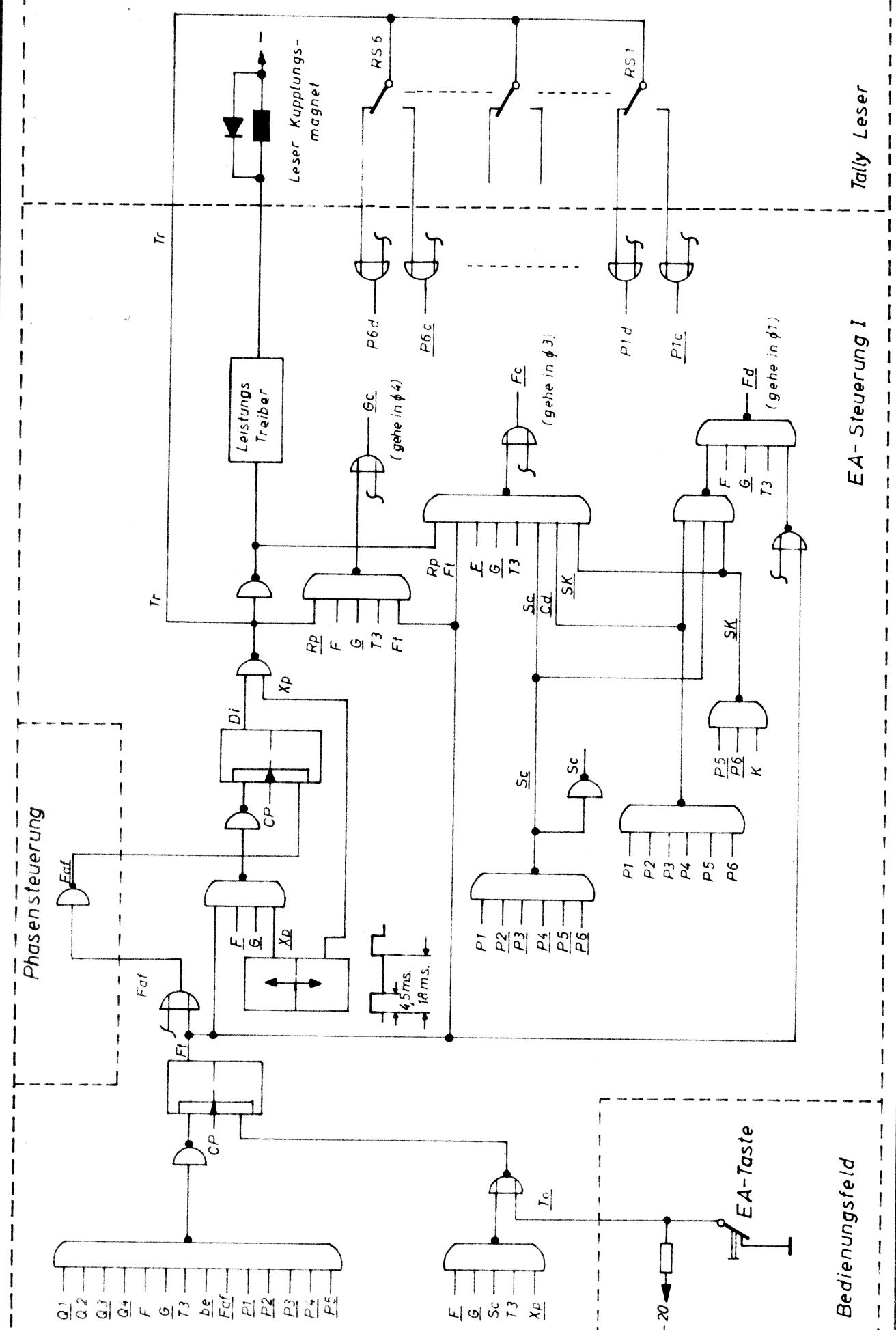
Eingabe über Flexowriter  
Blockschatzbild

Blattzahl: 2  
Blatt: 2

A2 - 133

### EA-Steuerung I





8.12.1966  
*Wa*  
**Schoppe & Faeser**  
G. m. b. H.  
Minden (Westf.)

**Eingabe über Tally Leser**  
Blockschaltbild

A2 - 137

**Bedienungsfeld**

**EA-Steuerung I**  
Tally Leser