

LGP-21 BEDIENUNGSANLEITUNG

	<u>Seite</u>
1. Lochen am Flexowriter	1
2. Lesen am Flexowriter	1
3. Doppeln am Flexowriter	1
4. Doppeln mit Korrekturen am Flexowriter	1
5. Ein- und Ausschalten der Rechenanlage	2
6. Eingabe des Bootstrap und des J1-10.0	3
7. Eingabe des Bootstrap und des J1-10.1	4
8. Aufruf von J1-10.0 oder J1-10.1	5
9. Eingabe von hexadezimalen Streifen oder von dezimalen Streifen mit Startfill und Modifier	5
10. Eingabe von hexadezimalen Streifen mit Hilfe von HVI (SF-J1-2014)	6
11. Eingabe von dezimalen Streifen ohne Startfill und Modifier	6
12. Eingabe von relativen hexadezimalen Streifen nur mit J1-10.0	6
13. Speicheranzeige	7
14. Speichereingabe hexadezimaler Worte von Hand am Flexowriter	7
15. Speichereingabe dezimaler Befehle von Hand am Flexowriter	8
16. Eingabe dezimaler Befehle und hexadezimaler Worte über einen Lochstreifen	9
17. Rechnungsbeginn - Schlüsselwort auf Streifenende	9
18. Rechnungsbeginn - Manueller Aufruf der Anfangsadresse	10
19. Ausgabe von Ergebnissen	10
20. Rechnen in Einzelschritten	11
21. Unterbrechnung des Programmablaufes	11
22. Speicherausdruck	12
23. Tracen	12
24. ACT I - Compilieren	12
25. ACT I - Ausstanzen des compilierten Maschinenprogr.	13
26. ACT I - Mehrfaches Compilieren	13
27. ACT I - Rechnen mit compiliertem Programm	13
28. ACT I - Tracen	14
29. Versetztes Laden von ACT I - Unterprogrammen	14

Benutzung des Flexowriters

Nachstehende Vorschriften beziehen sich allein auf den Flexowriter.

1. LOCHEN

1. Papier einlegen
2. "Netz" ein
3. "Lochen" ein
4. Streifen in Stanzstation einlegen
5. "Bandlauf"
(Ablochen)
6. "Bandlauf", Streifen abreißen
7. "Lochen" aus
8. "Netz" aus

Bei Fehllochung während des Ablochens:

1. Streifen bis zur Fehllochung zurückdrehen
2. "Code löschen" entsprechend der Anzahl der Fehllochungen drücken
3. Ab Fehllochung neu schreiben.

2. LESEN

1. Papier einlegen
2. "Netz" ein
3. Streifen in Lesestation einlegen
4. "Bedingter Stop"
(Drücken, wenn ohne Halt gelesen werden soll.
Nicht drücken, wenn bei Stop gehalten werden soll.)
5. "Lesen Start" drücken
(Streifen wird gelesen und geschrieben)
6. "Lesen Stop" drücken
7. "Netz" aus

3. DOPPELN

1. "Netz" ein
2. "Lochen" ein
3. Zu duplizierender Streifen in Lesestation
4. Leerstreifen in Stanzstation, Papier einlegen
5. "Bandlauf" drücken
6. "Bedingter Stop" ein (Vgl. 2. 4)
7. "Lesen Start" drücken
(Streifen wird gelesen, geschrieben und gedoppelt.)
8. "Lesen Stop" drücken
9. "Netz" aus

4. DOPPELN mit KORREKTUREN

1. "Netz" ein
2. "Lochen" ein
3. Leerstreifen in Stanzstation; Papier einlegen
4. "Bandlauf" drücken
5. Zu verbessерnden Streifen in Lesestation
6. Bis kurz vor Korrekturstelle doppeln, dann "Bed. Stop" aus und in Einzelschritten bis zum Fehler (jeweils "Lesen Start" drücken).
7. Richtiges Wort eintasten (wird auf das neue Band gestanzt)
8. "Lochen ein" herausnehmen
9. "Lesen Start" drücken und bis an das Ende der Korrekturstelle lesen und schreiben ohne zu lochen.
10. "Lochen ein" drücken
11. "Bedingter Stop" ein (vgl. 2. 4) und weiterdoppeln ("Lesen Start")

In den nachstehenden Vorschriften gelten folgende Abkürzungen:

R = Rechner

F = Flexowriter

Z = Zusatzgerät

M = Dreistellungsschalter "MODUS"

5. EIN- und AUSSCHALTEN der Rechenanlage

Wichtig: Vor Einschalten von R müssen F und Z ausgeschaltet sein.
Einschalten stets in angegebener Reihenfolge.

a) Einschalten von R

1. M auf "Manuell" (R) stellen
2. "EIN/AUS" (R) drücken

Der Rechner ist betriebsbereit, wenn "E/A" (R) leuchtet. Wenn M beim Einschalten von R auf "Normal" oder "Step" steht, leuchtet "E/A" nicht, und es können gespeicherte Programmteile zerstört werden.

Normalstellung von R

M auf "Manuell" gestellt. Es leuchten an R die Tasten "E/A", "Stop" und "EIN/AUS".

Bemerkung:

Durch Drücken der Taste "E/A" wird der Akku gelöscht, unabhängig von der Schalterstellung von "MODUS".

b) Einschalten von F

1. "Netz" (F) ein

Normalstellung von F

Alle Funktionstasten in Ruhe (oben); "Netz" (F) ein

c) Einschalten von Z

1. Bei Bedarf "EIN LESER" (Z) drücken
(Taste leuchtet im eingeschalteten Zustand)
2. "NORMAL" (Z) drücken (Taste leuchtet auf)
"NORMAL" (Z) muß immer gedrückt sein, wenn eingelesen werden soll.
"VORLAUF" und "RÜCKLAUF" kann nur zum Bandumspulen benutzt werden!
Soll von "Vorlauf" auf "Rücklauf" oder umgekehrt von "Rücklauf" auf "Vorlauf" geschaltet werden, dann muß dazwischen immer erst "NORMAL" gedrückt werden.
3. Bei Bedarf "EIN STANZER" (Z) drücken
(Taste leuchtet im eingeschalteten Zustand)

Normalstellung von Z

"EIN LESER" und "NORMAL" leuchten.
"EIN STANZER" wird nur bei Bedarf gedrückt.

Vor Ein- und Ausschalten von Z stets M auf "MANUELL" (R) stellen!

d) Ausschalten der Rechenanlage

Wichtig: Vor Ausschalten von R müssen F und Z ausgeschaltet sein!

1. Ggf. "EIN STANZER" (Z) lösen
(Taste leuchtet in ausgeschaltetem Zustand nicht)
2. Ggf. "EIN LESER" (Z) lösen
(Taste leuchtet in ausgeschaltetem Zustand nicht)
3. "Netz" (F) aus
4. M auf "Manuell" (R) stellen
5. "EIN/AUS" (R) lösen.

6. EINGABE des BOOTSTRAP und des J1-10.0 (Programmeingabe 1)

Der Lochstreifen von J1-10.0 enthält 3 verschiedene Fassungen des Programms:

1. Relativ hexadezimal mit eigenen Bootstraps in Spur 00
2. Relativ hexadezimal mit eigenen Bootstraps in Spur 63
3. Dezimal in Kodierungsblattformat

Vor den Fassungen 1) und 2) stehen jeweils 2 Bootstraps. Dabei gilt der 1. Bootstrap immer für den Schnelleser und der 2. Bootstrap für den Flexowriter. Der Bootstrap ist ein Hilfsprogramm zum Einlesen von J1-10.0 und kann nur über F eingegeben werden.

Folgende Anweisungen gelten für alle Bootstraps von J1-10.0:

1. Normalstellung von R, F, Z
2. Lochstreifen des gewünschten Bootstraps in Lesevorrichtung von F einlegen.
3. "Lesen Start" (F) drücken
4. "Füllen/Löschen" (R) drücken
5. "Lesen Start" (F) drücken

Es wird mod 0000' geschrieben. Dabei bedeutet 0000 die hex. Anfangsadresse von J1-10.0. Nur wenn J1-10.0 nicht auf Spur 00 stehen soll, muß jetzt von Hand die gewünschte Anfangsadresse hex. eingegeben werden (z.B. für Spur 1800 muß "1200" über F eingegeben werden).

6. M auf "Step" (R) stellen
7. "Ausführen" (R) drücken
8. M auf "Manuell" (R) stellen
9. "Lesen Start" (F) drücken
10. "Füllen/Löschen" (R) drücken
11. "Lesen Start" (F) drücken
12. M auf "Step" (R) stellen
13. "Ausführen" (R) drücken

Die Anweisungen 8 - 13 3 mal wiederholen. Damit ist die manuelle Eingabe des Bootstrap beendet.

Einlesen von J1-10.0 1. oder 2. Fassung über F

(Es muß jeweils der 2. Bootstrap eingegeben worden sein).

14. M auf "Normal" (R) stellen
15. "Start" (R) drücken

J1-10.0 wird eingelesen. Nach dem Einlesen von 2 Spuren (nur hex. Eingabe) erfolgt ein Stop, wenn PS-32 nicht gedrückt ist. Ist PS-32 gedrückt oder wird "Start" (R) gedrückt, so wird der 2. Teil (dezimale Eingabe) eingelesen.

Einlesen von J1-10.0 1. oder 2. Fassung über Z

(Es muß jeweils der 1. Bootstrap eingegeben worden sein).

14. Lochstreifen aus Leser (F) herausnehmen, Streifenhalteklappe (F) zurückklappen, Lochstreifen in Schnelleser so einlegen, daß nur J1-10.0 eingelesen werden kann.

15. M auf "Normal" (R) stellen
16. "Start" (R) drücken

J1-10.0 wird eingelesen. Nach dem Einlesen von 2 Spuren (nur hex. Eingabe) erfolgt ein Stop. Wenn der 2. Teil (dez. Eingabe) noch eingelesen werden soll, sind folgende Operationen auszuführen:

1. "Eingabe von Hand" (F) drücken
2. "Start" (R) drücken, (Lampe an F leuchtet)
3. "S0000000!" über F eingeben
4. "Rechner Start" (F) drücken
5. "Start" (R) drücken

Dann erfolgt ebenfalls Eingabe des 2. Teils von J1-10.0 über Z.

7. EINGABE DES BOOTSTRAP UND DES J1-10.1 (Programmeingabe 2)

Der Lochstreifen des J1-10.1 enthält 3 verschiedene Fassungen des Programms:

1. Hexadezimal mit Bootstrap in Spur 63 für Flexowriter
2. Hexadezimal mit Bootstrap in Spur 63 für Schnelleser
3. Dezimal in Kodierungsblattformat

Der Bootstrap ist ein Hilfsprogramm zum Einlesen von J1-10.1 und kann nur über F eingegeben werden. Nachfolgende Anweisungen gelten für die Bootstraps der 1. und 2. Fassung:

1. Normalstellung von R, F, Z
2. Lochstreifen des Bootstrap von 1 oder 2 in Leser (F)
3. M auf "Manuell" (R) stellen
4. "Lesen Start" (F) drücken
5. "Füllen/Löschen" (R) drücken
6. "Lesen Start" (F) drücken
7. M auf "Step" (R) stellen
8. "Ausführen" (R) drücken

Anweisung 3 - 8 noch 3 mal ausführen. Damit ist die Eingabe des Bootstrap beendet.

Einlesen von J1-10.1 1. Fassung über F

9. M auf "Normal" (R) stellen
10. "Start" (R) drücken

J1-10.1 1. Fassung wird eingelesen.

Einlesen von J1-10.1 2. Fassung über Z

9. Lochstreifen aus Leser (F) herausnehmen, Streifenhalteklappe (F) zurückklappen. Lochstreifen in Schnelleser so einlegen, daß nur J1-10.1 2. Fassung eingelesen werden kann.

10. M auf "Normal" (R) stellen
11. "Start" (R) drücken

J1-10.1 2. Fassung wird eingelesen.

8. AUFRUF VON J1-10.0 ODER J1-10.1

Dabei muß J1-10.0 in Spur 0000 stehen. In den nachstehenden Vorschriften bezeichnen die Stichworte "Aufruf J1-10.0" bzw. "Aufruf J1-10.1" folgende Operationen:

1. M auf "Step" (R) stellen
2. "Füllen/Löschen" (R) drücken
3. M auf "Normal" (R) stellen
4. "Start" (R) drücken

9. EINGABE VON HEXADEZIMALEN STREIFEN ODER VON DEZIMALEN STREIFEN MIT STARTFILL UND MODIFIER MIT J1-10.0 ODER J1-10.1

Hexadezimale Streifen beginnen stets mit einem V-Schlüsselwort.

a) Eingabe über Z

1. Normalstellung von R, F, Z
2. Lochstreifen in Z einlegen
3. "Eingabe von Hand" (F) drücken
4. Aufruf J1-10.0 oder J1-10.1 (Lichtfeld am (F) leuchtet)
5. "S0000000!" eintippen
6. "Rechner Start" (F) drücken
7. "Start" (R) drücken

Anweisung 7. nur bei Eingabe mit J1-10.0 ausführen.

Lochstreifen wird eingelesen.

8. Bei Fehlen des Schlüsselwortes .000XXXX' am Ende des Streifens "E/A" (R) drücken.

b) Eingabe über F

1. Normalstellung von R und F
 2. Lochstreifen in Lesevorrichtung F einlegen
 3. Aufruf von J1-10.0 oder J1-10.1
- Lochstreifen wird eingelesen.
4. Bei Fehlen des Schlüsselwortes .000XXXX' am Ende des Streifens "E/A" (R) drücken und "Lesen Stop" (F) drücken.

10. EINGABE VON HEXADEZIMALEN STREIFEN MIT HILFE VON HVI (SF-J1-2014)

Mit dem Programm HVI können hex. Streifen schneller als mit J1-10.0 oder J1-10.1 nur über Schnelleser eingelesen werden.

Das Programm liegt in 2 Fassungen vor:

1. Hex. Lochstreifen mit eigenem Bootstrap für die Spuren 0000-0263
2. Dez. Lochstreifen beliebig verlegbar

Einzelheiten siehe Programmbeschreibung.

11. EINGABE VON DEZIMALEN STREIFEN OHNE STARTFILL UND MODIFIER

a) Eingabe über Z

1. Normalstellung von R, F, Z
2. Lochstreifen in Z einlegen
3. "Eingabe von Hand" (F) drücken
4. Aufruf J1-10.0 oder J1-10.1 (Lichtfeld am (F) leuchtet auf)
5. Schreiben der Anfangsadresse (Startfill) ;000XXXX auf (F)
6. "Rechner Start" (F) drücken (Lichtfeld am (F) leuchtet wieder auf)
7. Schreiben des Adressenmodifikators (Modifier) /000XXXX auf (F)
8. "Rechner Start" (F) drücken
9. "S0000000" über (F) eintippen
10. "Rechner Start" (F) drücken
11. "Start" (R) drücken.

Anweisung 11. gilt nur bei Eingabe mit J1-10.0
Lochstreifen wird eingelesen.

b) Eingabe über F

1. Normalstellung von R, F
2. Lochstreifen in Leser von (F) einlegen
3. "Eingabe von Hand" (F) drücken
4. Aufruf J1-10.0 oder J1-10.1 (Lichtfeld am F leuchtet auf)
5. Schreiben der Anfangsadresse (Startfill) ;000XXXX auf (F)
6. "Rechner Start" (F) drücken (Lichtfeld am F leuchtet wieder auf)
7. Schreiben des Adressenmodifikators (Modifier) /000XXXX (F)
8. "Rechner Start" (F) drücken
9. "Eingabe von Hand" (F) lösen
Lochstreifen wird eingelesen.

12. EINGABE VON RELATIVEN HEXADEZIMALEN STREIFEN nur mit J1-10.0

Relative hexadezimale Streifen beginnen stets mit einem m-Schlüsselwort.

a) Eingabe über Z

1. Normalstellung von R, F, Z
2. Lochstreifen in Z einlegen

3. "Eingabe von Hand" (F) drücken
4. Aufruf J1-10.0 (Lichtfeld am (F) leuchtet auf)
5. Schreiben des Adressenmodifikators /000XXXX auf (F)
6. "Rechner Start" (F) drücken
7. "S0000000" über (F) eintippen
8. "Rechner Start" (F) drücken
9. "Start" (R) drücken
Lochstreifen wird eingelesen.

b) Eingabe über F

1. Normalstellung von R, F, Z
2. Lochstreifen in (F) einlegen
3. "Eingabe von Hand" (F) drücken
4. Aufruf von J1-10.0 (Lichtfeld am (F) leuchtet)
5. Schreiben des Adressenmodifikators /000XXXX auf (F)
6. "Rechner Start" (F) drücken
7. "Eingabe von Hand" (F) lösen
Lochstreifen wird eingelesen.

13. SPEICHERANZEIGE nur zur Anzeige des Speicherinhalts ohne Befehlsführung

1. J1-10.0 oder J1-10.1 muß gespeichert sein
2. Normalstellung R und F
3. "Eingabe von Hand" (F) drücken
4. Aufruf von J1-10.0 oder J1-10.1 (Lichtfeld am (F) leuchtet)
5. Schreiben des Schlüsselwortes -000XXXX auf (F). XXXX = Adresse des anzugebenden Speicherplatzes
6. "Rechner Start" (F) drücken
7. Inhalt der Zelle XXXX erscheint dual im Akkumulator
8. "PST" (R) drücken
Anweisung 8. ist nur bei J1-10.0 notwendig !
9. "Start" (R) drücken
Inhalt der Zelle XXXX + 1 erscheint dual im Akkumulator.
Bei erneutem "Start" (R) erscheint jeweils der Inhalt der nächsten Zelle im Akku.

14. SPEICHEREINGABE HEXADEZIMALER WORTE VON HAND (F)

Es können Konstanten oder in hexadezimale Form umgewandelte Befehle eingegeben werden.

a) Eingabe eines einzelnen hexadezimalen Wortes

1. Normalstellung R und F
2. "Eingabe von Hand" (F) drücken

3. Aufruf J1-10.0 oder J1-10.1
4. Schreiben des Schlüsselwortes +00CXXXX oder +00HXXXX auf (F).
XXXX = Speicherplatz des hexadezimalen Wortes.
5. "Rechner Start" (F) drücken
6. Schreiben des hexadezimalen Wortes auf (F)
7. "Rechner Start" (F) drücken
Das hexadezimale Wort ist auf XXXX gespeichert.

b) Eingabe mehrerer hintereinander zu speichernder hex. Worte

1. Normalstellung R und F
2. "Eingabe von Hand" (F) drücken
3. Aufruf J1-10.0 oder J1-10.1
4. Schreiben des Schlüsselwortes ;000XXXX (Startfill) auf (F)
5. "Rechner Start" (F) drücken
6. Schreiben des Schlüsselwortes ,0000ONN auf (F)
. NN = Anzahl der hexadezimalen Worte ≤ 63
7. "Rechner Start" (F) drücken
8. Schreiben des ersten hex. Wortes auf (F)
9. "Rechner Start" (F) drücken
10. Schritte 8 und 9 wiederholen, bis alle NN-Worte eingegeben sind.
((NN - 1) mal Schritt 8 und 9 wiederholen)

15. SPEICHEREINGABE DEZIMALER BEFEHLE VON HAND (F)

Konstanten können nur in hexadezimaler Form eingegeben werden (s. 11)

1. Normalstellung R und F
2. "Eingabe von Hand" (F) drücken
3. Aufruf J1-10.0 oder J1-10.1
4. Schreiben des Schlüsselwortes ;000XXXX (Startfill) auf (F)
5. "Rechner Start" (F) drücken
6. Schreiben des Schlüsselwortes /000XXXX (Modifier) auf (F)

Es werden nur die Adressen modifiziert, vor denen kein X steht.

7. "Rechner Start" (F) drücken
8. Schreiben des Befehls auf (F)
9. "Rechner Start" (F) drücken (Befehl wird gespeichert)

Sollen mehrere Befehle in aufeinanderfolgenden Zellen gespeichert werden, so müssen die Schritte 8 und 9 für jeden einzelnen Befehl wiederholt werden.

16. EINGABE DEZIMALER BEFEHLE UND HEXADEZIMALER WORTE ÜBER EINEN LOCHSTREIFEN

1. Ablochen von dezimalen Befehlen und (oder) hexadezimalen Worten in folgender Reihenfolge:

;000XXXX (Startfill)

/000XXXX (Modifier) der Modifier ist für hex. Worte ohne Bedeutung;
Befehle und (oder) hex. Worte stehen mehrere Befehle und (oder) hex. Worte auf dem Streifen, so werden die in aufeinanderfolgenden Zellen gespeichert. Vor den aufeinanderfolgenden hex. Worten muß das Schlüsselwort ,00000NN abgelocht sein. Hierbei bedeutet NN die Anzahl der nachfolgenden hex. Worte. Bei Bedarf erneut Startfill usw.

.000XXXX (Stop und Sprung)

am Ende des Streifens

2. Eingabe des Lochstreifens nach 9

17. RECHNUNGSBEGINN - SCHLÜSSELWORT AUF STREIFENENDE

Das Schlüsselwort .000XXXX (Stop und Sprung) ist am Ende des Hauptprogrammstreifens gelocht (XXXX ist die Anfangsadresse des Rechenprogrammes).

- Speichern der benötigten Unterprogramme (nach 9, 10, 11 oder 12)
- Speichern des Hauptprogramms (nach 9, 10, 11 oder 12)

Unterprogramme und Hauptprogramme können auch auf einem Lochstreifen stehen.

- Nach dem Einlesen des Programms bewirkt das Schlüsselwort .000XXXX einen Stop.

Die Auswahl der Einheit für die Dateneingabe ist im Programm jeweils durch den Spurteil des I-Befehls festgelegt. Dabei bedeutet Spuradresse 02 Eingabe über Flexowriter, Spuradresse 00 Eingabe über Schnelleser.

- Dateneingabe von Hand über F (800I0200 oder I0200)

- "Eingabe von Hand" (F) drücken
- "Start" (R) drücken

Beginn der Rechnung; wenn vom Programm Daten verlangt werden, leuchtet Lichtfeld (F) auf.

- Schreiben des vom Programm verlangten Wertes auf (F)
- "Rechner Start" (F) drücken

Datenwort muß entsprechend dem Eingabeprogramm vorbereitet sein. Werden in einem Programm mehrere Daten benötigt, müssen die Schritte 3 und 4 genügend oft wiederholt werden. Das nächste Datenwort immer erst dann schreiben, wenn das Lichtfeld (F) leuchtet.

e) Dateneingabe auf Lochstreifen über F (800I0200 oder I0200)

1. Lochstreifen mit Daten in Lesevorrichtung (F) legen
2. "Eingabe von Hand (F) lösen, falls gedrückt
3. "Start" (R) drücken

Die Daten werden programmgemäß eingelesen und verarbeitet.

f) Dateneingabe auf Lochstreifen über Z (800I0000 oder I0000)

1. Lochstreifen mit den Daten in Schnelleser einlegen
2. "Start" (R) drücken

Die Daten werden programmgemäß eingelesen und verarbeitet.

g) Fehler bei der Dateneingabe

Ist der Datenstreifen in die falsche Eingabeeinheit gelegt worden, so bleibt der Rechner auf dem jeweiligen I-Befehl stehen und der angewählte Leser läuft leer.

Es leuchten die Tasten "Start" und "E/A" am Rechner. Den angewählten Leser durch Drücken von "E/A" stoppen. Im Befehlsregister der Oszilloskopanzeige die Spuradresse des I-Befehls lesen, danach die richtige Eingabeeinheit auswählen oder, falls die angewählte Einheit nicht vorhanden ist, die Spuradresse des I-Befehls ändern.

Anschließend die Rechnung neu beginnen, wie es im folgenden Abschnitt beschrieben wird.

18. RECHNUNGSBEGINN - MANUELLER AUFRUF DER ANFANGSADRESSE

Diese Anweisung gilt für den Fall, daß das Schlüsselwort .000XXXX nicht am Ende des Programmstreifens abgelocht worden ist oder für den Fall, daß irgendwelche Bedienungsfehler während der Rechnung gemacht worden sind und mit der Rechnung neu begonnen werden soll.

Manueller Aufruf der Anfangsadresse

1. Normalstellung R, F, Z
2. "Eingabe von Hand" (F) drücken
3. Aufruf J1-10.0 oder J1-10.1
4. Schreiben des Schlüsselwortes .000XXXX (Stop und Sprung, XXXX = Anfangsadresse des Programms)
5. "Rechner Start" (F) drücken
6. "Start" (R) drücken

Die Eingabe der Daten erfolgt dann wie in 17d, 17e oder 17f.

19. AUSGABE VON ERGEBNISSEN

Die Auswahl der Einheit für die Datenausgabe ist im Programm jeweils durch den Spurteil des P-Befehls festgelegt. Spuradresse 02 bedeutet Ausgabe über Flexowriter, Spuradresse 06 Ausgabe über Schnellstanzer.

1. Ausschreiben der Ergebnisse (F) gemäß Programm

Hierzu ausreichend Papier (F) einlegen.

2. Ausschreiben der Ergebnisse (F) und Stanzen (F)

- a) Ausreichend Papier (F) einlegen
- b) Leerstreifen (F) einlegen
- c) "Lochen ein" (F) drücken ; "Bandlauf" (F)

3. Stanzen auf Leerstreifen (Z)

- a) Leerstreifen (Z) einlegen
- b) "Bandlauf" (Z) drücken

20. RECHNEN IN EINZELSCHRITTEN

1. Rechnen in Einzelschritten vom Programmbeginn an:

Vor dem Drücken der Taste "Start" (R) zum Beginn des eigentlichen Rechenablaufs darf M nicht auf "Normal", sondern muß auf "Step" stehen. Die Befehle werden einzeln nach jeweiligem Drücken von "Start" (R) ausgeführt.

2. Rechnen in Einzelschritten von beliebiger Programmstelle an:

M auf "Step" (R) stellen
"Start" (R) drücken

Die Befehle werden einzeln nach jeweiligem Drücken von "Start" (R) ausgeführt.

Soll die Rechnung fortlaufend, d.h. nicht mehr in Einzelschritten, weitergeführt werden, so ist M wieder von "Step" (R) auf "Normal" zu stellen und "Start" (R) zu drücken.

21. UNTERBRECHEN DES PROGRAMMABLAUFS

a) kurzzeitige Unterbrechung

- 1. M auf "Step" (R) stellen (Programmablauf wird gestoppt)
- 2. M auf "Normal" (R) stellen
- 3. "Start" (R) drücken (Normaler Programmfortgang)

b) Unterbrechung bei Ausschalten des Rechners

- 1. M auf "Step" (R) stellen
- 2. "Start" (R) so oft drücken, (vgl. 18) bis ein B-Befehl im Befehlsregister des Oszilloskopes erscheint.
- 3. Ablesen und Notieren der im Zählregisterfeld des Oszilloskopes erscheinenden Adresse
- 4. Notieren der Schalterstellungen F, R, Z
- 5. Notieren der Lochstreifenlage (F, Z)
- 6. Ausschalten der Rechenanlage nach 5.
- 7. Einschalten der Rechenanlage nach 5.
- 8. Normalstellung R
- 9. "Eingabe von Hand" (F) drücken
- 10. Aufruf J1-10.0 oder J1-10.1
- 11. Schreiben des Schlüsselwortes .000XXXX (F). XXXX ist die um 1 verminderte am Zählregisterfeld des Oszilloskopes abgelesene Adresse .

12. "Rechner Start" (F) drücken
 13. Stellung der Schalter an F und Z, evtl. auch R, sowie Lochstreifenlage (F, Z) wie zum Zeitpunkt des Ausschaltens wieder herstellen.
 14. "Start" (R) drücken (Programm wird normal weitergerechnet)
22. SPEICHERAUSDRUCK durch Verwendung von Unterprogrammen
Für den Betrieb sind die entsprechenden Programmbeschreibungen maßgebend.
1. Hexadezimaler Speicherausdruck
Unterprogramme J4-10.0, J4-10.1, J4-10.2, J4-10.3 Die Unterprogramme eignen sich zur Zusammenstellung von Haupt- und Unterprogrammen zu vollständigen Programmstreifen (Bildung von Prüfsummen). Bedienungsvorschriften und genauere Angaben siehe Unterprogrammbeschreibung.
 2. Dezimaler Speicherausdruck
Unterprogramm K2-10.0 oder K2-10.1 ohne Prüfsummenbildung oder dgl., d.h. nur zum Ausdruck der Programmfolge. Die Reihenfolge entspricht der laufenden Nr. der Speicherplätze.
23. TRACEN zum Testen von Programmen durch Verwendung von Unterprogrammen
Für den Betrieb sind die entsprechenden Programmbeschreibungen maßgebend. Es sind zu benutzen:
Für FK-Programme: Unterprogramm K1-10.0

24. ACT I - COMPILEIREN
Der Compiler ist ein hexadezimaler Streifen, der die Speicher 4000 bis 5963 belegt, von 0000 - 0263 muß J1-10.1 gespeichert sein.
1. Laden des Compilers nach Vorschrift Nr. 9 mit Hilfe J1-10.1 oder HVI
 2. Normalstellung R, F, Z
 3. Streifen des Quellenprogrammes in Lesevorrichtung (F) einlegen
 4. "Eingabe von Hand" (F) drücken
 5. Aufruf J1-10.1
 6. Schreiben des Schlüsselwortes .0004000 auf (F)
 7. "Rechner Start" (F) drücken
 8. "Eingabe von Hand" (F) lösen
 9. "Start" (R) drücken

Quellenprogramm wird in Maschinencode übersetzt. Nach Beendigung des Compilierens wird folgendes ausgedruckt:

i xxxx Anfangsadresse des Maschinenprogramms
f xxxx Endadresse des Maschinenprogramms
sol xxxx Absolute Adresse der Anweisungskennzeichen
soc xxxx usw.

Danach stoppt (R); das compilerte Programm steht auf den Speicherplätzen i xxxx bis f xxxx.
Anschließend Programm austanzen (Vorschrift Nr. 25), Compilieren eines weiteren Quellenprogrammes (Vorschrift Nr. 26) oder Beginn der Rechnung (Vorschrift Nr. 27).

25. ACT I - AUSSTANZEN des kompilierten Maschinenprogrammes

Ausstanzen und Ausdrucken des kompilierten Maschinenprogrammes erfolgt mit Hilfe des im Compiler enthaltenen Druckprogrammes.

a) Ausstanzen unmittelbar nach Compilieren über Flexowriter

1. Papier und Leerstreifen (F) einlegen
2. Schalterstellung (R) nach Compilieren unverändert lassen
3. Normalstellung F und Z
4. "Lochen ein" (F) drücken
5. "Bandlauf" (F) drücken
6. "Start" (R) drücken

Maschinenprogramm wird hexadezimal geschrieben und ausgestanzt.

7. "Bandlauf" (F) drücken
8. Normalstellung F und R

b) Ausstanzen mit besonderem Aufruf über Flexowriter

Besonderer Aufruf des Druckprogrammes ist notwendig, wenn zwischen Compilieren und Ausstanzen der Inhalt des Befehlszählregisters verändert wurde (z.B. bei Rechnung im Anschluß an Compilier-Vorgang vor Ausstanzen des Maschinenprogramms oder bei Bedienungsfehlern).

1. Papier und Leerstreifen (F) einlegen
2. Normalstellung R, F, Z
3. "Eingabe von Hand" (F) drücken
4. Aufruf J1-10.1

5. Schreiben des Schlüsselwertes .0005501 auf (F)

6. "Rechner Start" (F) drücken
7. "Lochen ein" (F) drücken
8. "Bandlauf" (F) drücken
9. "Start" (R) drücken

Maschinenprogramm wird hexadezimal geschrieben und ausgestanzt.

10. "Bandlauf" (F) drücken
11. Normalstellung R, F, Z

26. ACT I - MEHRFACHES COMPILEIEREN

Der Compiler erlaubt es, mehrere Quellenprogramme nacheinander zu compilieren. Nach dem Compilieren jedes einzelnen Quellenprogrammes nach Vorschrift 24 ist jeweils das compilierte Maschinenprogramm nach Vorschrift 25 auszustanzen. Schalterbedienung wie bei den Vorschriften 24 und 25.

27. ACT I - RECHNEN MIT COMPILEIRETEM PROGRAMM

1. Speichern des hexadezimalen Maschinenprogrammes nach Vorschrift Nr. 9 mit J1-10.1 oder HVI entfällt, wenn das Maschinenprogramm nach dem Compilieren noch auf der Trommel gespeichert ist.
2. Speichern der benötigten Unterprogramme UPA, UFB-GK oder UPB-FK nach Vorschrift Nr. 9 mit J1-10.1 oder HVI.

Speicherbelegung der ACT I-Unterprogramme:

<u>ACT I - UPA</u>	<u>Plätze</u>
Trace, Float und Unfloat	5000 - 5163
Gleitkommaoperationen	5200 - 5763
Festkomma Ein- und Ausgabe	5800 - 5963

<u>ACT I - UPA</u>	<u>Gleitkomma</u>	<u>Festkomma</u>
Arc sin	3300 - 3631	4432 - 4563
Arc cos		
Exponentialfunktion (e, 10)	3632 - 3931	4100 - 4163
Logarithmus (ln, log)	3932 - 4163	4200 - 4431
arctg	4200 - 4463	4600 - 4663
sin, cos	4500 - 4731	4700 - 4763
Quadratwurzel	4732 - 4863	4800 - 4863
linkages	4900 - 4963	4900 - 4963

UPA, UPB-GK und UPB-FK sind jeweils als hexadezimaler Streifen vorhanden und werden nach Vorschrift 9 gespeichert.

Darüber hinaus sind die unter UPB zusammengefaßten Unterprogramme als Einzelunterprogramme auf dezimalen Lochstreifen mit Startfill und Modifier vorhanden. Sie werden nach Vorschrift 9 auf die o.a. Plätze gespeichert.

3. Rechnungsbeginn

Rechnungsbeginn verläuft wie bei Vorschrift 18. Das einzugebende Schlüsselwort ist .OOXXXX, wobei XXXX die als i xxxx im Anschluß an das Compilieren ausgedruckte Anfangsadresse des Maschinenprogramms bedeutet.

28. ACT I - TRACEN

Das Trace-Unterprogramm ist im Compiler enthalten.

1. Tracen einzelner Programmteile

Durch Schreiben von "trace" hinter die zu tracende Anweisung im Quellenprogramm. Das Tracen erfolgt bei der Ausführung des Maschinenprogrammes nur, wenn PST (R) gedrückt ist.

2. Tracen einzelner Programmteile oder Tracen des gesamten Programmes

1. Drücken der Sprungtaste (R) vor Compilieren

2. Drücken der Sprungtaste (R) vor der Ausführung des gesamten Maschinenprogrammes oder der zu tracenden Teile des Maschinenprogrammes.

Nach Lösen der Sprungtaste (R) wird Normal weitergerechnet.

Hinsichtlich des Ausdruckes beim Tracen vgl. ACT I-Beschreibung Seite 11.

29. VERSETZTES LADEN VON ACT I - UNTERPROGRAMMEN

Werden nicht alle im UPB-GK enthaltenen Unterprogramme benötigt, so können die benötigten Unterprogramme zur Einsparung von Speicherplätzen abweichend von ihrer Normallage gespeichert werden. Da der Übergang vom Rechenprogramm in die jeweiligen UPB-Unterprogramme durch die in den "linkages" enthaltenen Sprungbefehle vermittelt wird, müssen diese Sprungbefehle bei versetztem Laden der Unterprogramme abgeändert werden. Es handelt sich hierbei um folgende Sprungbefehle:

<u>UPB-GK-Unterprogramm</u>	<u>Speicherplatz</u>	<u>Sprungbefehl</u>
arc sin	4945	xU(L _o + 0004)
arc cos	4959	xU(L _o + 0261)
Exponentialf. (e)	4926	xU(L _o + 0214)
Exponentialf. (10)	4933	xU(L _o + 0000)
Logarithmus (ln)	4954	xU(L _o + 0200)
Logarithmus (log)	4958	xU(L _o + 0000)
arctg	4939	xU(L _o + 0000)
sin	4908	xU(L _o + 0000)
cos	4915	xU(L _o + 0210)
Quadratwurzel	4920	xU(L _o + 0000)

Der angegebene Speicherplatz gilt für die Normallage nach 27.2; L_o ist die Anfangsadresse des jeweiligen Unterprogrammes. Zum versetzten Laden der Unterprogramme sind zunächst die linkages einzulesen; sodann sind die benötigten Sprungbefehle den veränderten Anfangsadressen der Unterprogramme entsprechend nach Nr. 15 einzugeben.

Das versetzte Speichern der UPB-GK-Unterprogramme (dezimale Lochstreifen) geschieht mit Eingabe des geänderten Startfills und Modifiers nach Vorschrift Nr. 9, wobei zu beachten ist, daß die auf den einzelnen dezimalen Lochstreifen enthaltenen Startfills und Modifiers nicht mit eingelesen werden, da sie sich nur auf die Normallage beziehen.

LGP-ZI

()

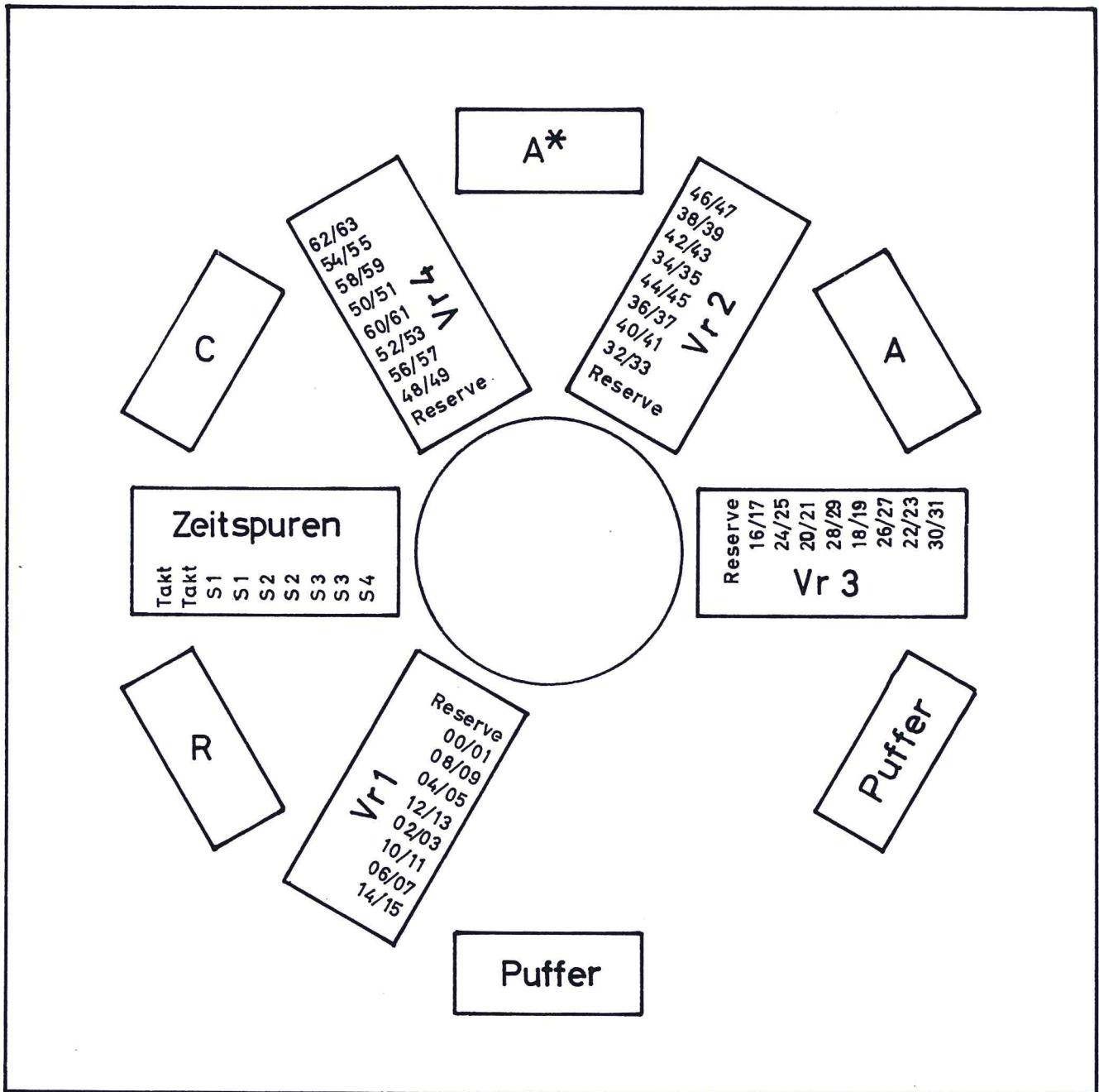
C

()

C

()

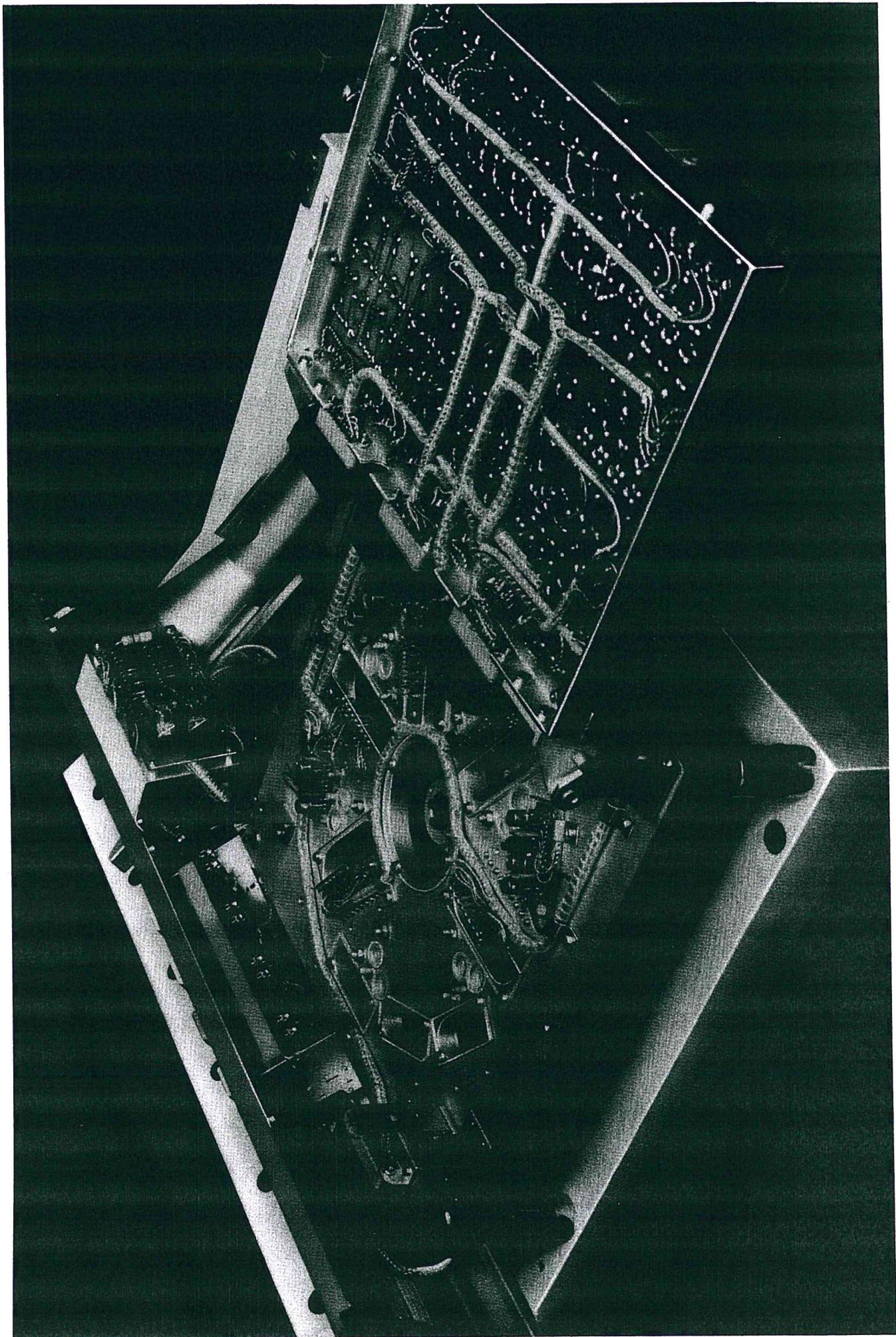
()



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 Grundausrustung	A1 - 1
1.2. Prinzipielle Arbeitsweise	A1 - 2
1.3. Begriffe und Symbole	A1 - 4
1.4. Erweiterungen und Zusatzgeräte	A1 - 7

<u>2. Systembeschreibung</u>	A2 - 1
<u>2.1. Stromversorgung</u>	A2 - 1
<u>2.2. Speichereinheit und zugehörige Steuerung</u>	A2 - 1
2.2.1. Schreibverfahren	A2 - 1
2.2.2. Speicherplätze	A2 - 2
2.2.3a Druckschaltungsanordnung	A2 - 3
<u>2.2.3. Hauptspeicher</u>	A2 - 5
2.2.3.1. Kopfanwahl	A2 - 5
2.2.3.2. Schreiben	A2 - 6
2.2.3.3. Lesen	A2 - 11
<u>2.2.4. Umlaufregister</u>	A2 - 13
2.2.4.1. Schreibverstärker	A2 - 14
2.2.4.2. Leseverstärker	A2 - 15
<u>2.2.5. Takt- und Zeitspuren</u>	A2 - 17
2.2.5.1. Leseverstärker für S ₁ , S ₂ , S ₃	A2 - 17
2.2.5.2. Leseverstärker für den Grundtakt cp	A2 - 17
2.2.5.3. Taktleistungsverstärker	A2 - 19
<u>2.3. Grundschaltungen</u>	
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 (ϕ_1)	A2 - 35
2.4.2.2. Phase 2 (ϕ_2)	A2 - 36
2.4.2.3. Phase 3 (ϕ_3)	A2 - 37
2.4.2.4. Phase 4 (ϕ_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

2.7. <u>Ein-Ausgabe (E/A-Logik)</u>	
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
2.8. <u>Bedeutung der Buchstaben in den logischen Gleichungen</u>	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
2.9. <u>Logische Funktionsgleichungen</u>	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
2.10. <u>Blockschaltbilder</u>	
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			
Anschriftwerte:				
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:	<u>Länge</u> <u>Tiefe</u> <u>Höhe</u> <u>Gewicht</u>			
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).

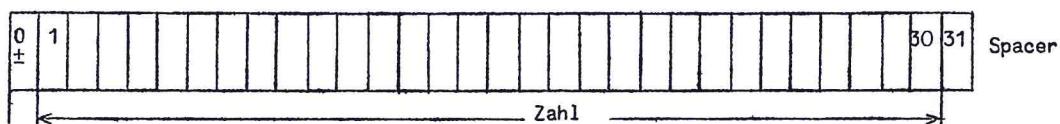


Bild 1.1 Zahlendarstellung

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).

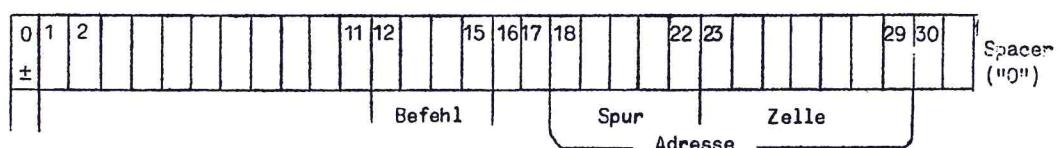


Bild 1.2. Befehlswort

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 Befehlausführung 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.

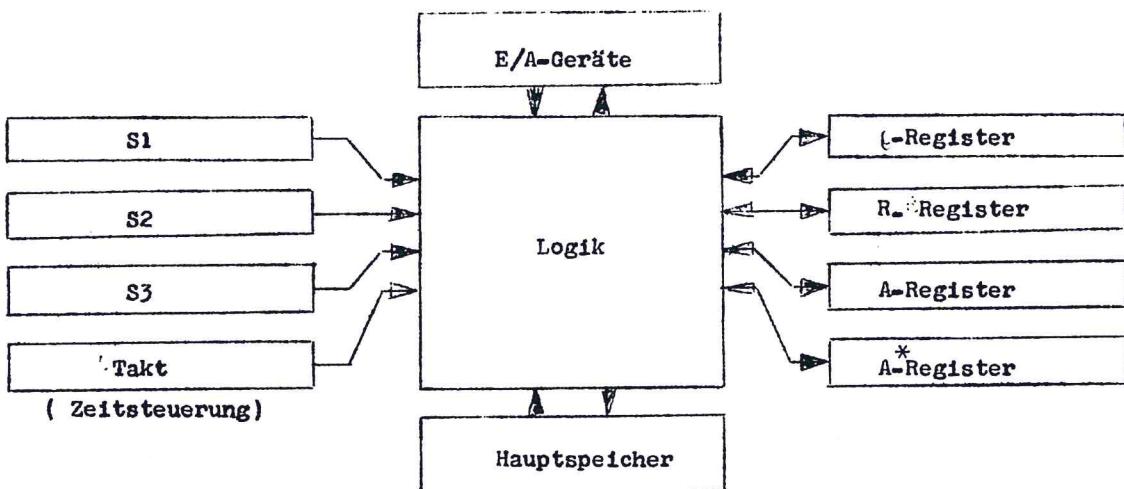


Bild 1.3.

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:

logisch "1" = 0 V bis - 2 V

logisch "0" = -10V bis -20 V

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. 5μ 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.

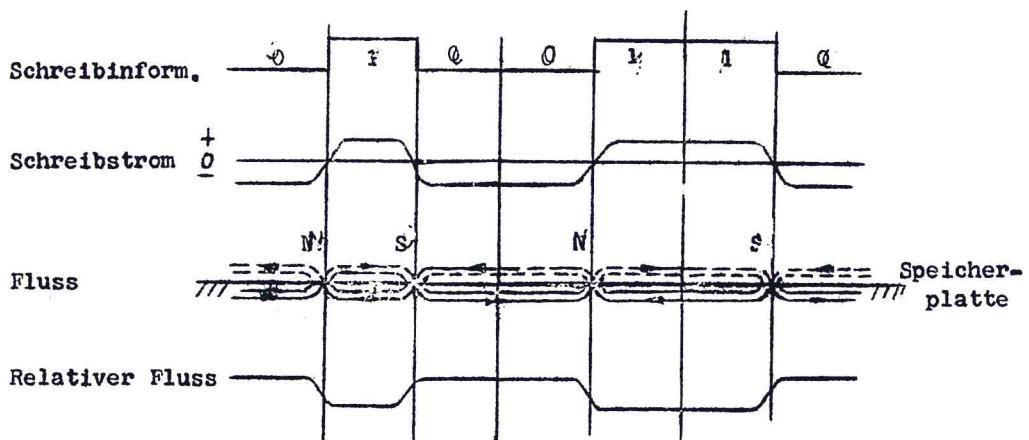


Bild 2.1.

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 μ 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: 3OKO (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 Blockschaltbildern 3.3. und 3.4. dargestellt.

Position 1 befindet sich direkt auf der Speicheleinheit.

Position 2 befindet sich auf der klappbaren Druckschaltung 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) Ch₁ bis Ch₈ werden durch P₃, P₄ und P₅ angewählt, die einzelnen Zeilen (Blöcke) Vr₁ bis Vr₄ durch P₁ und P₂.

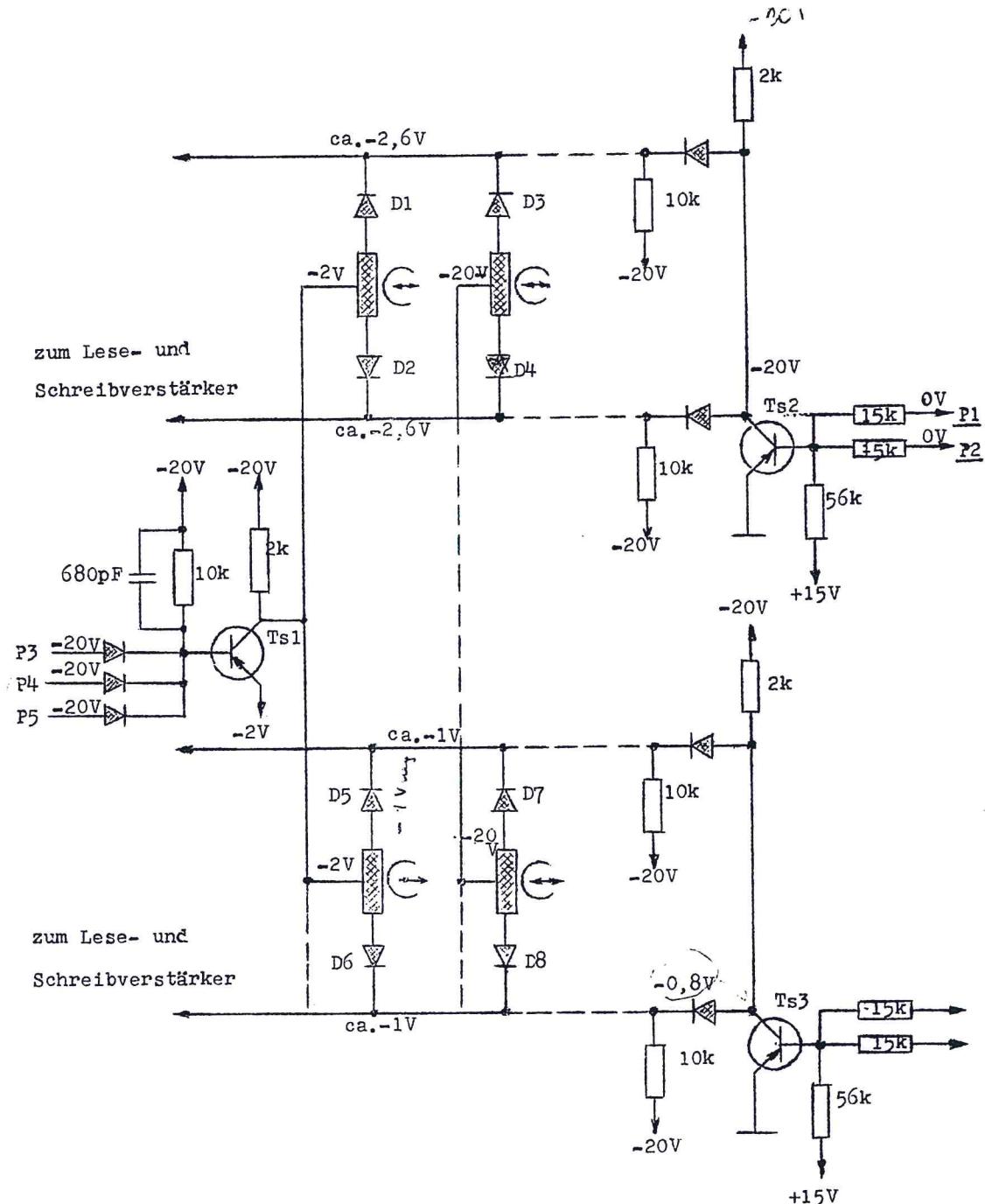


Bild 2.2.

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:

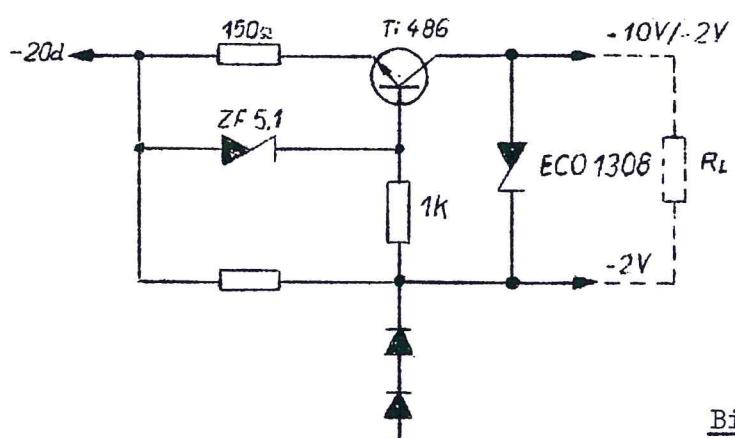


Bild 2.3.

Der Transistor wirkt als veränderbarer Widerstand, der den Strom auf 30mA 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).

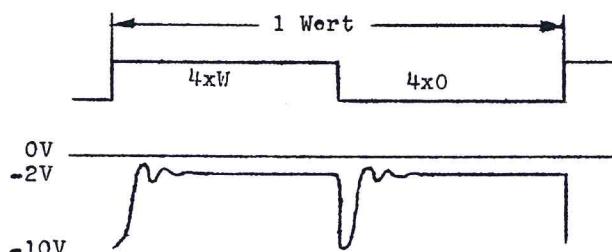


Bild 2.4.

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 , 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.

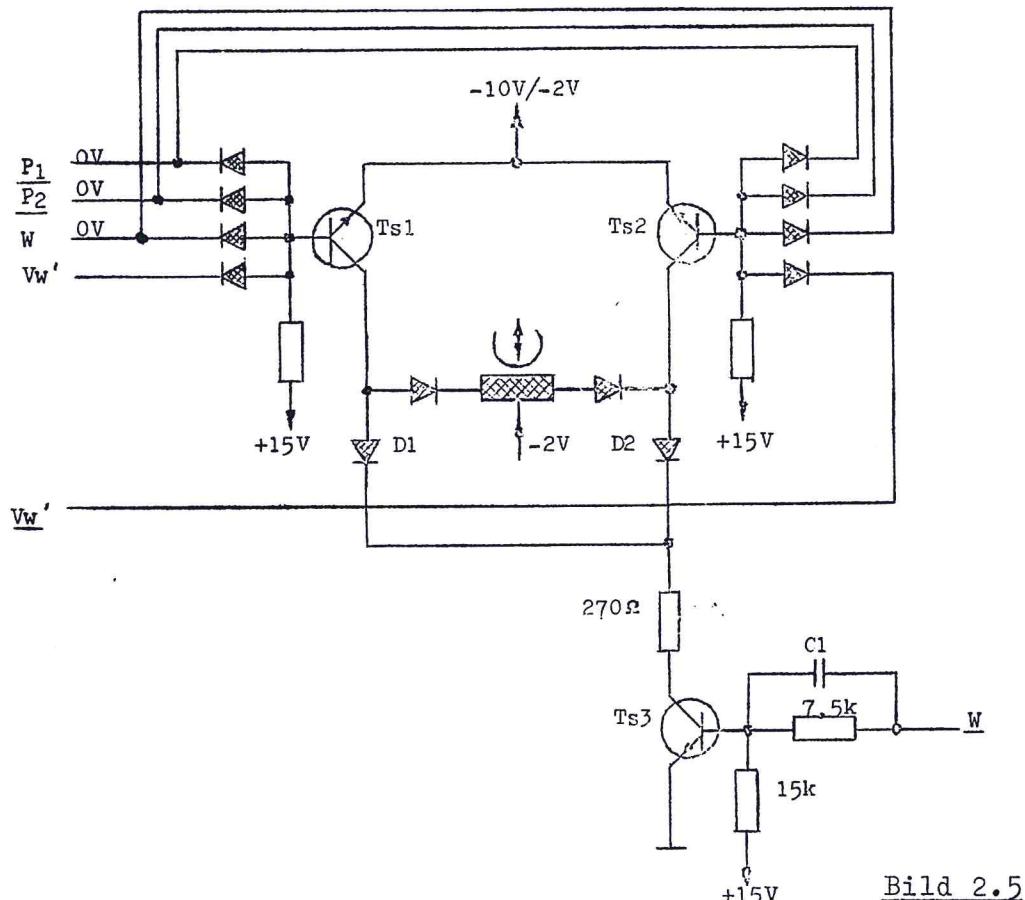


Bild 2.5

Die Schreiberlaubnis W setzt sich zusammen aus:

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

Adreßzeit

Die Schreibinformation Vw' setzt sich zusammen aus:

$$Vw' = L \left[A \underbrace{Q_1}_{R-\text{Befehl}} + A \underbrace{Q_4}_{\substack{\text{Addiere "1" zum} \\ \text{C-Register}}} + \underbrace{Q_1 \ Q_4}_{\substack{\text{A-Information beim Y-Befehl} \\ \text{A-Information beim C-, H-Befehl}}} (K \underbrace{C}_{\substack{\text{Spacerbit-Schreibinformation: "0"} \\ \text{C-Register}}} + \underbrace{K \ C}_{\substack{\text{A-Information beim Y-Befehl} \\ \text{A-Information beim C-, H-Befehl}}}) \right]$$

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

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

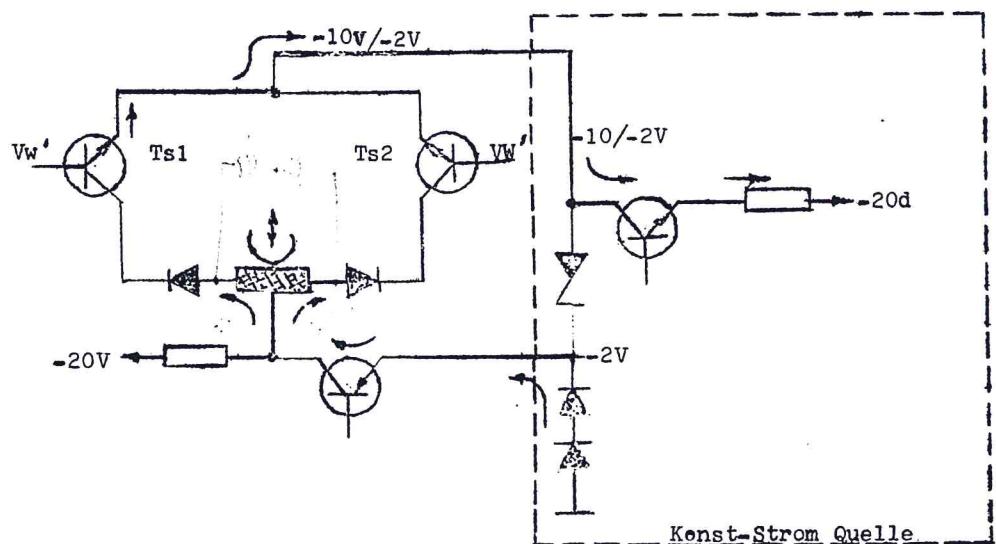


Bild 2.6.

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 Abschaltungsspannungsspitze auf, die zum Zerstören der Schreibtransistoren führen würde, wenn die Abschaltungsspitze 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 angelöscht 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 D₁ und D₂ (Bild 2.7) die Spannung am Eingang des Vorverstärkers auf ca. 0,6V begrenzt. Die Widerstände R₃ und R₄ dienen zur Entkopplung.

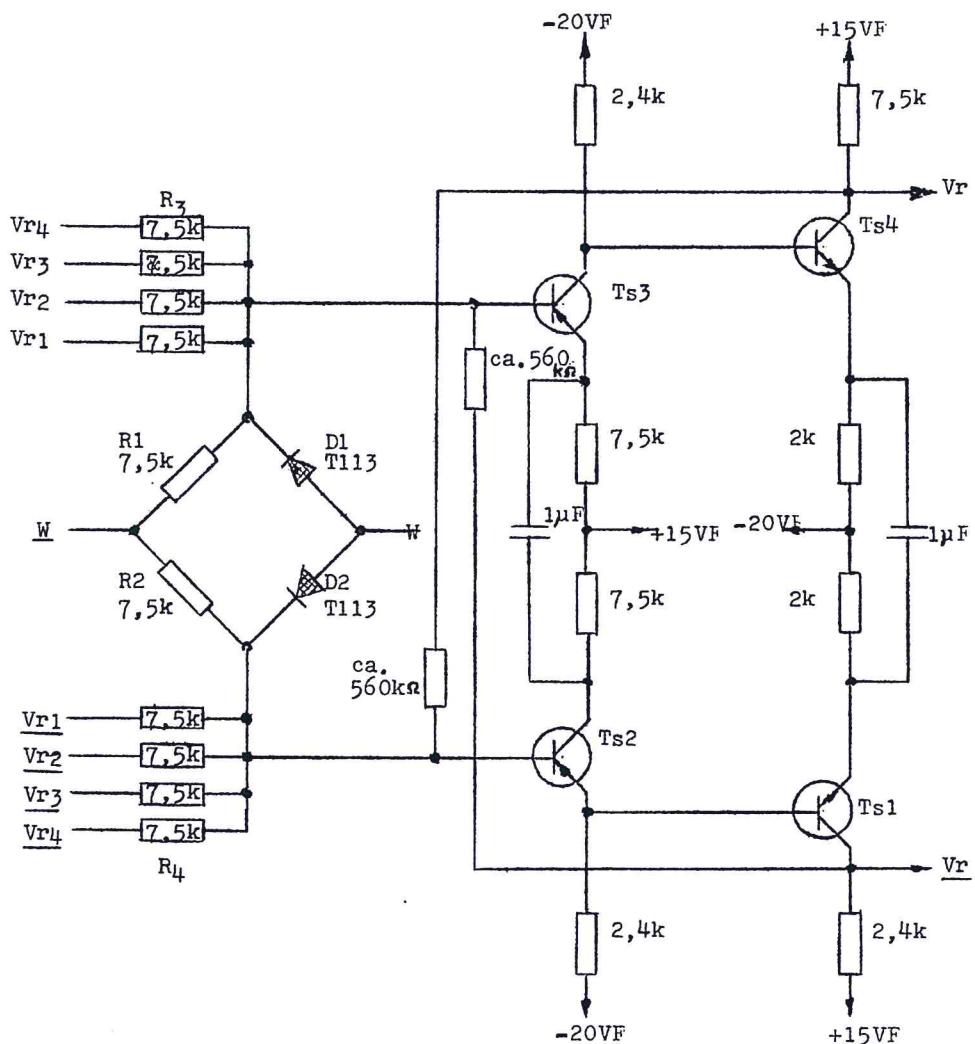


Bild 2.7.

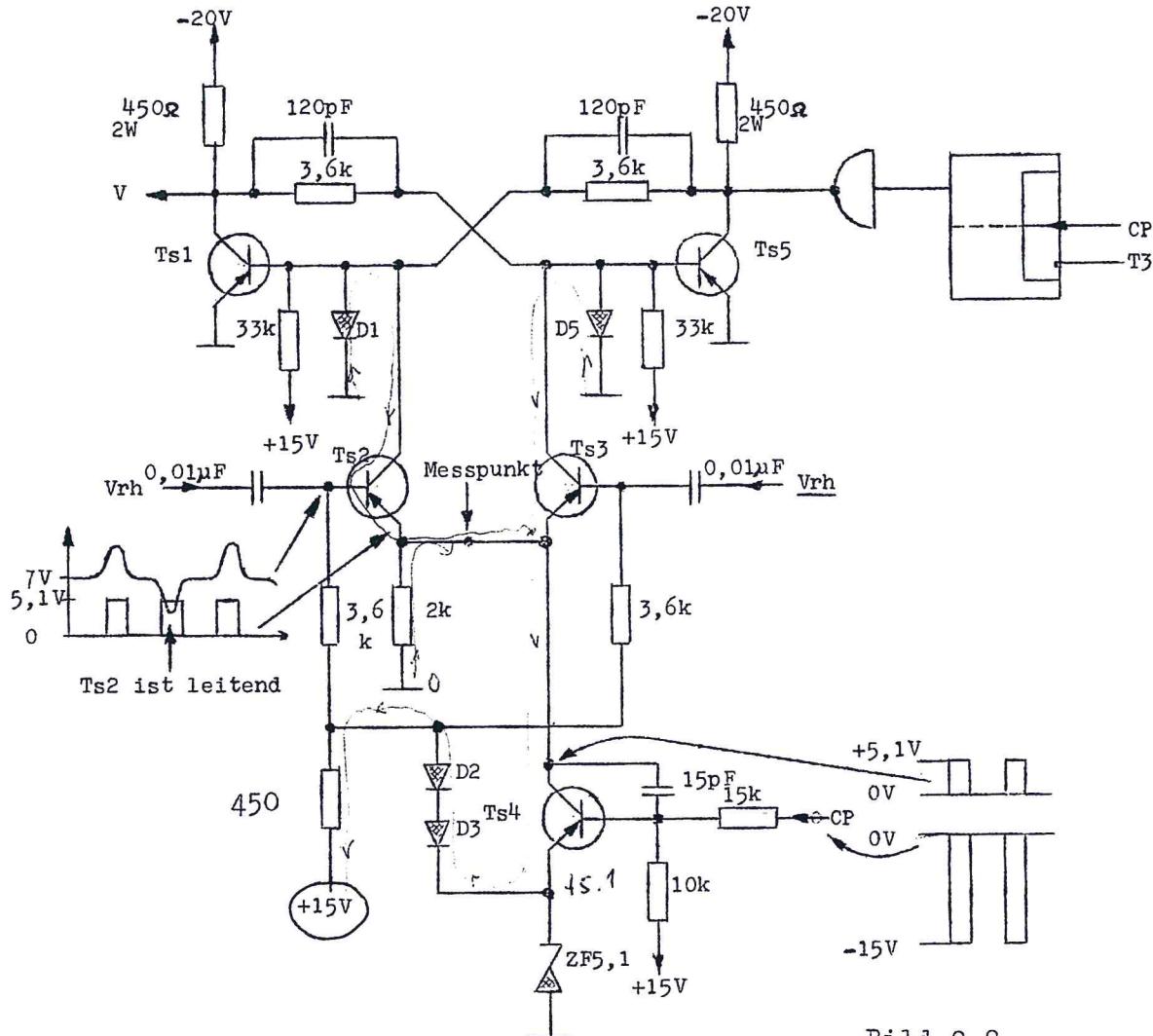


Bild 2.8.

Die gegenseitig um 180° 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. \bar{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.

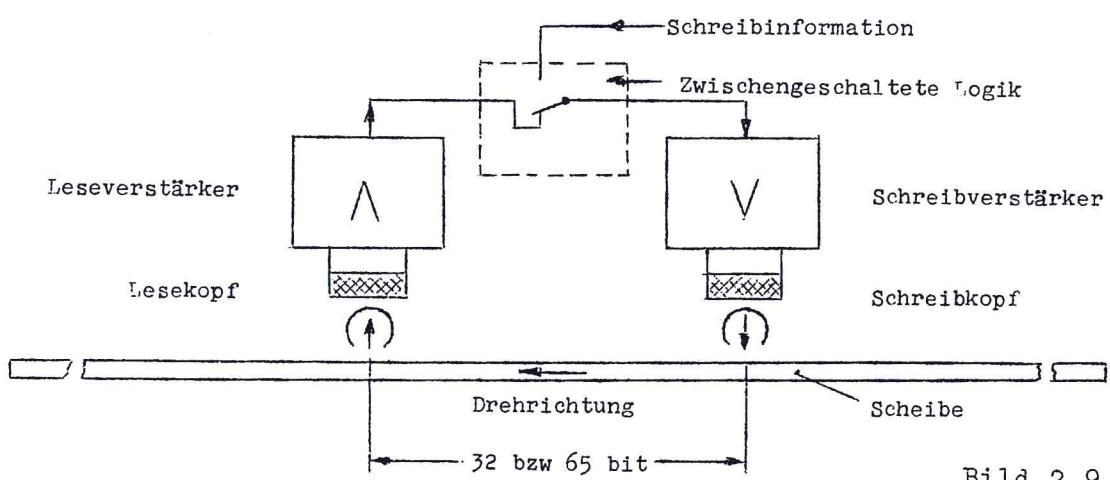


Bild 2.9.

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.

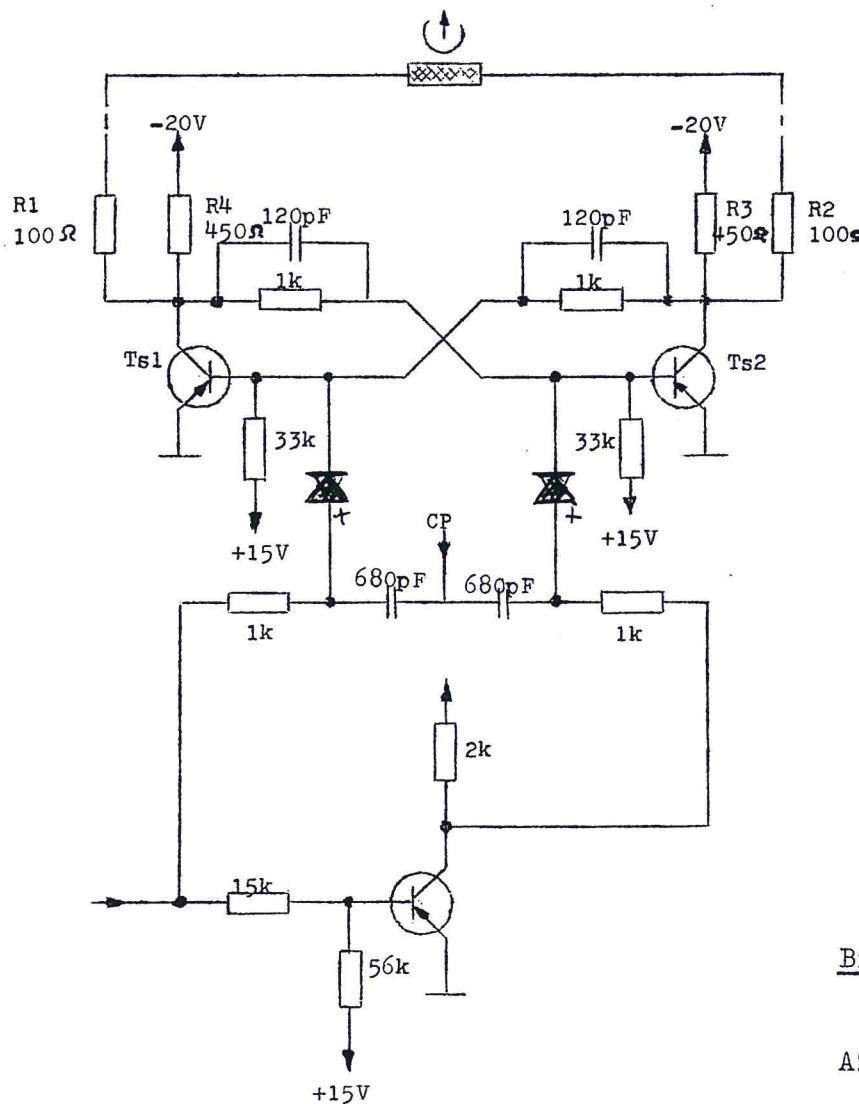


Bild 2.10

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.

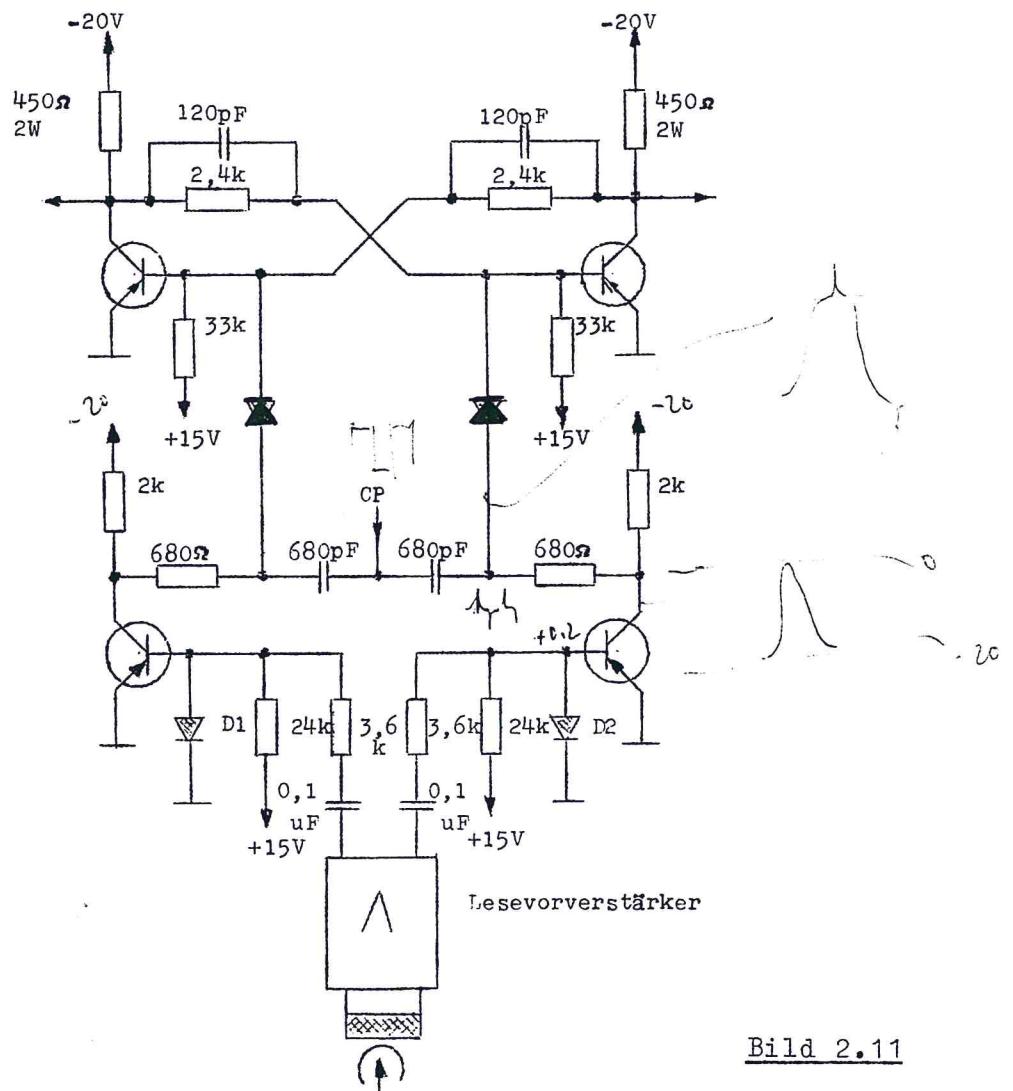


Bild 2.11

2.2.5. Takt- und Zeitspuren

Zur Steuerung des Rechenablaufs im LGP 21 dienen die vier Taktspuren C_p , S_1 , S_2 und S_3 . 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_1 , S_2 und S_3 (s. auch Bild 3.13)

Die Leseverstärker für die Spuren S_1 , S_2 , S_3 sind mit dem Leseverstärker für die Umlaufregister identisch (siehe Abschnitt 2.2.4.2).

2.2.5.2. Leseverstärker für den Grundtakt C_p

(siehe auch Bild 3.13)

Das Lesesignal wird zunächst wieder in einem linearen Differentialverstärker verstärkt und dann über einen Resonanzkreis (C_{19} , R_{43} , L_1) auf die Basis von Ts_{13} gegeben. Durch negative Signale wird Ts_{13} leitend. Sein Ausgangssignal wird mit C_{21} (680 pF) und R_{47} (470 Ω) differenziert. Die positiven Spitzen setzen das nachfolgende One Shot (monostabiler Multivibrator; Taktzeit min 3,5 μ s, max 5 μ s).

Der über R_{49} , R_{50} normalerweise leitend gehaltene Ts_{14} sperrt bei einem positiven Impuls über D_{14} . Über den Emitterfolger (Ts_{15}) koppelt der negative Sprung am Kollektor von Ts_{14} auf die Basis von Ts_{16} . Ts_{16} wird leitend. Über den Kondensator C_{23} (680pF) wird der positive Sprung am Kollektor von Ts_{16} kapazitiv auf die Basis von Ts_{14} zurückgekoppelt. Das Sperren von Ts_{14} wird beschleunigt. Man erhält am Kollektor von Ts_{14} 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 μ 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.

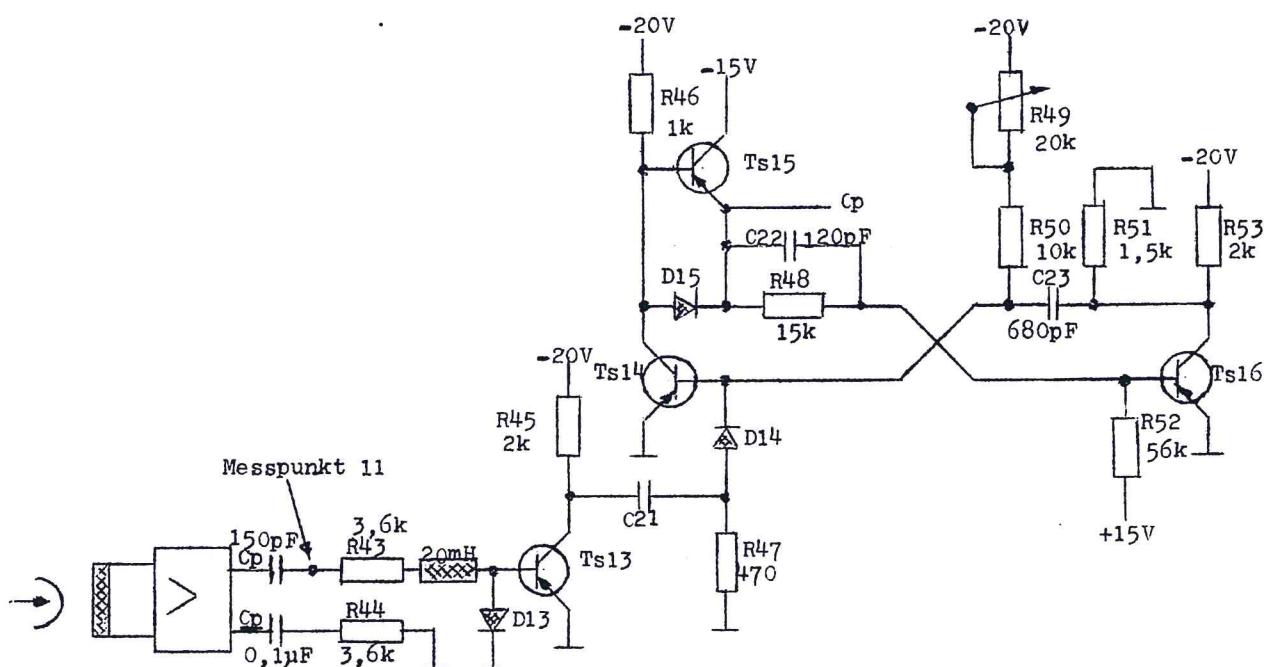


Bild 2.12

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).

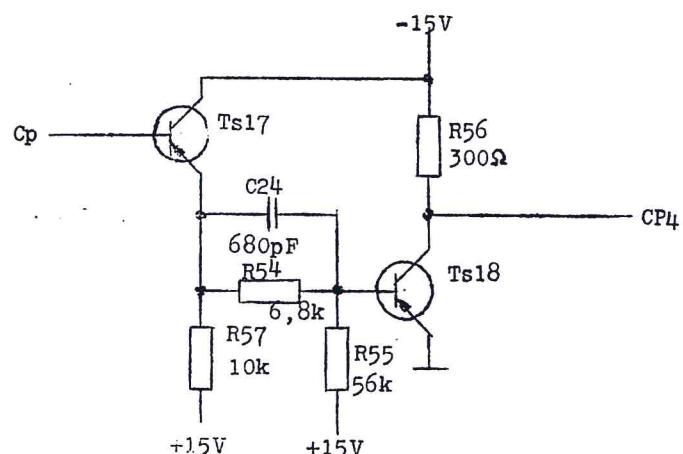


Bild 2.13

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' , A' entgegengesetzte Wertigkeiten haben. Falls keine Setzbedingungen gegeben sind, liegen beide Eingänge auf logisch "0" (= -10V...-20V).

Die zwei Ausgänge A, A des Flip Flops haben grundsätzlich entgegengesetzte Wertigkeit.

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

Das Flip Flop kann nur zur Taktzeit gesetzt werden und auch nur dann, wenn das Setzsignal am Eingang A' , A' ca. 3,3 bzw. 2 μ 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 μ sec bzw. 2 μ 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.

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.

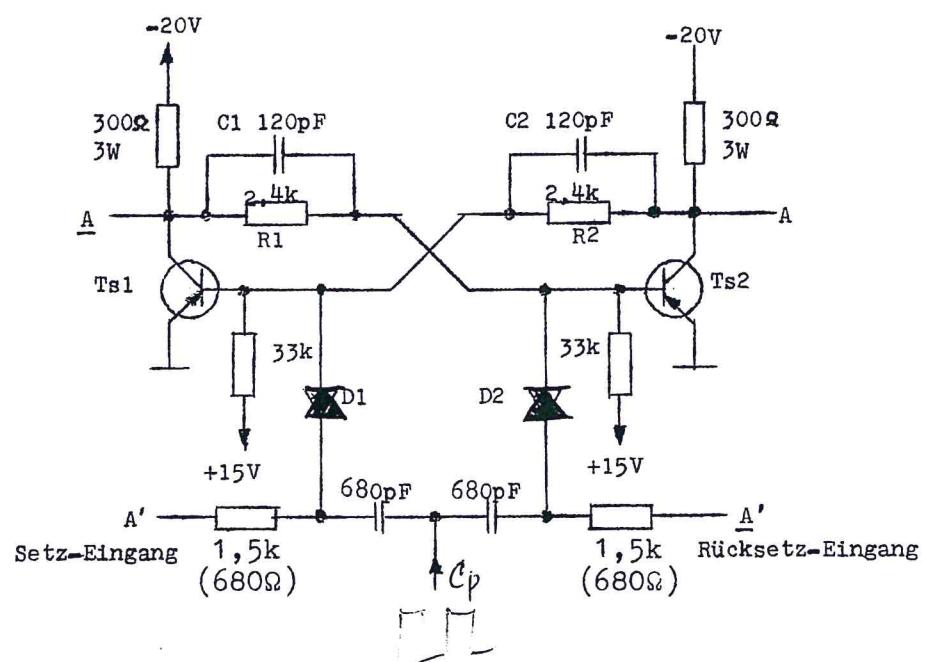


Bild 2.14

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.

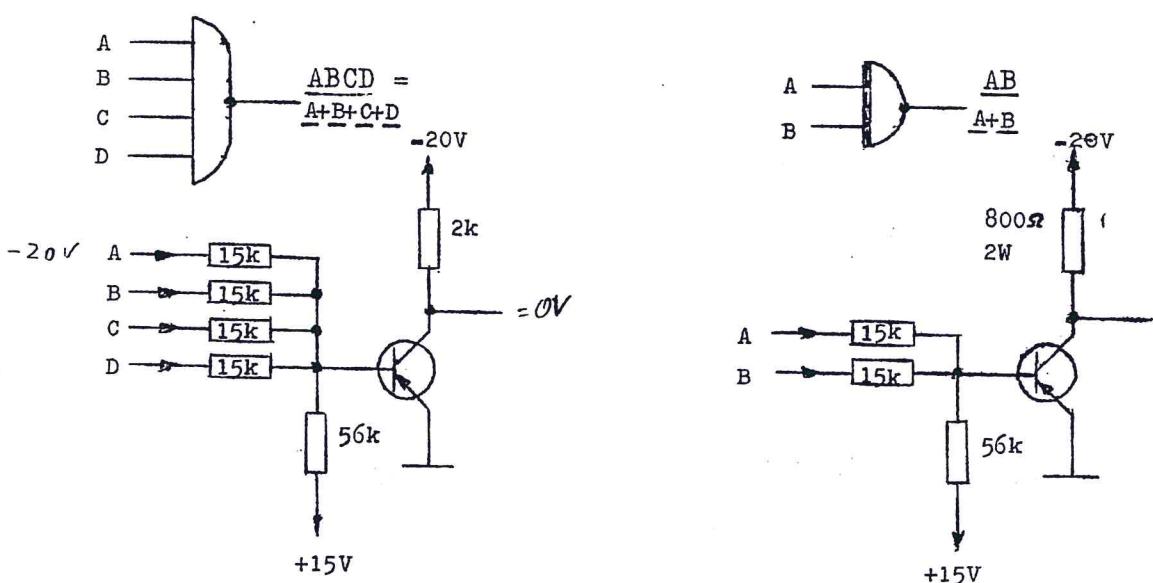


Bild 2.15

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.

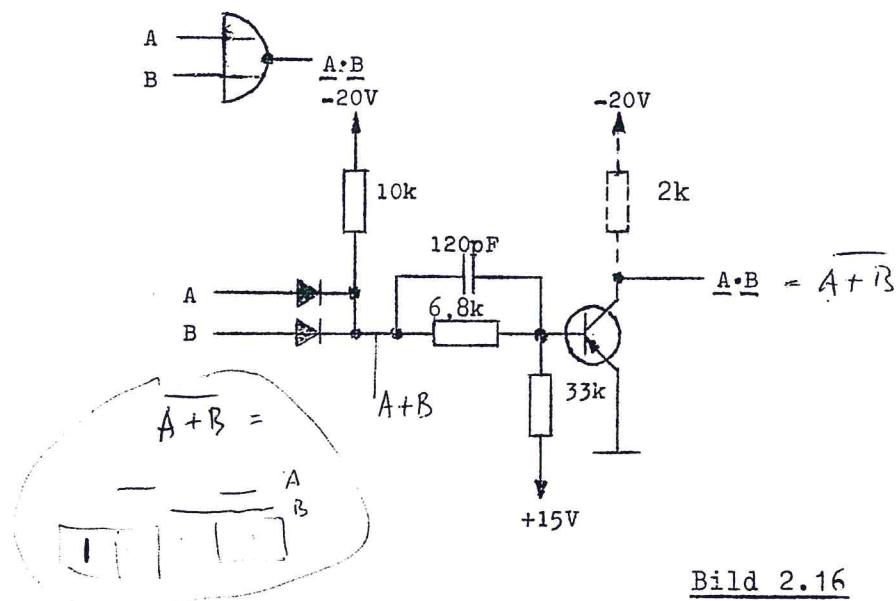


Bild 2.16

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.

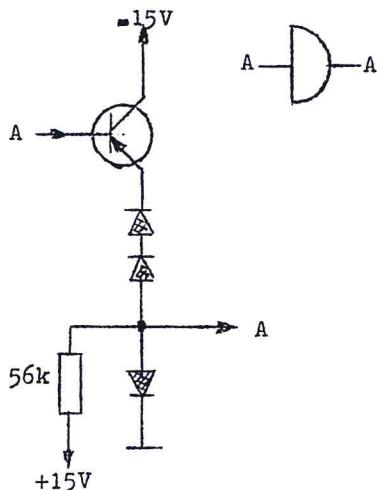


Bild 2.17

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.

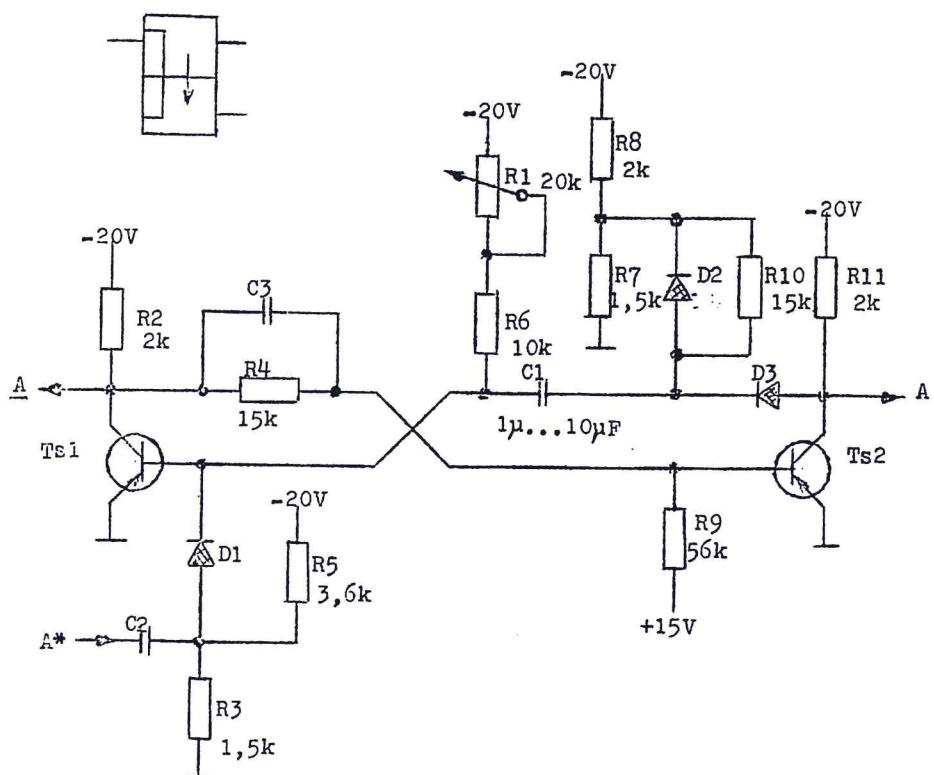


Bild 2.18

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.

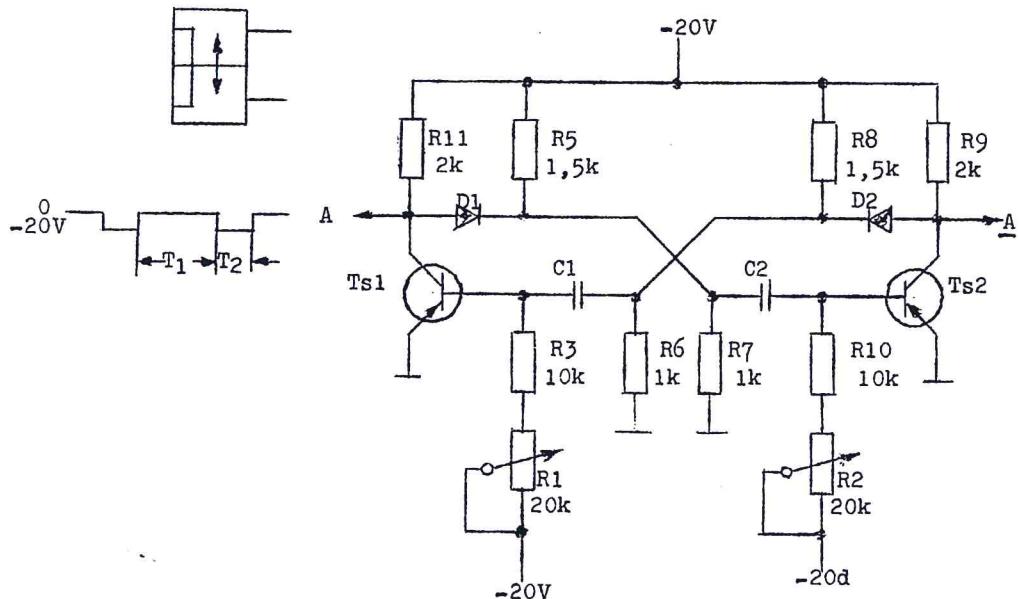


Bild 2.20

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 Tp 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 Leitern. 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.

D₃ soll verhindern, daß sich am Emitter von Ts₂ ein positiveres Potential als +1V einstellt (1V reicht zum sperren von Ts₃), weil sonst Ts₂ nicht über den Spannungsteiler R₇, R₆ gesperrt werden kann.

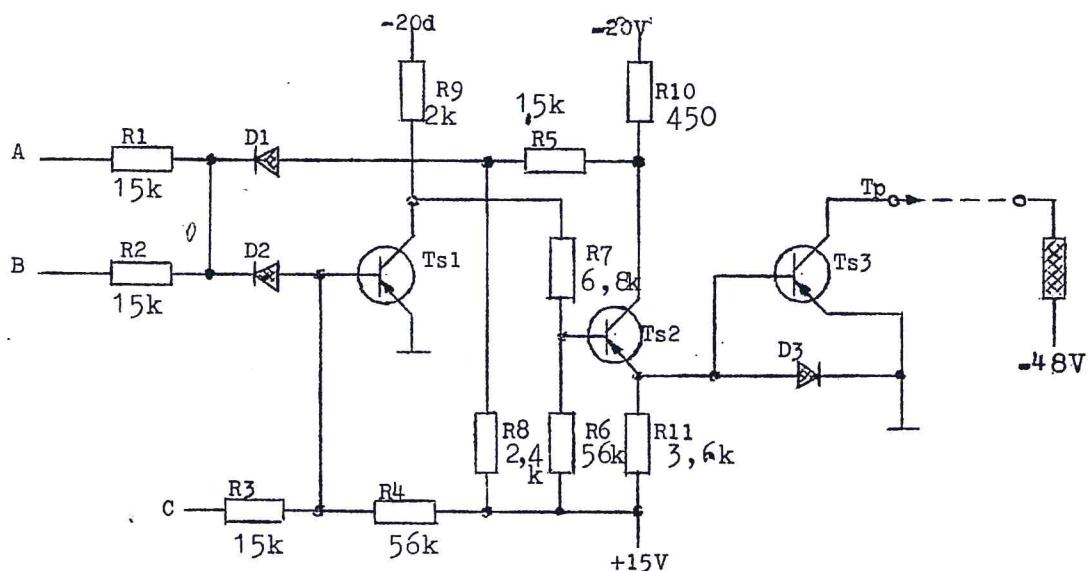


Bild 2.21

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 Ts₁ über den Spannungsteiler R₁₁, R₃, R₄ leitend und damit Ts₂ sowie Ts₃ über den Spannungsteiler Ts₁, R₆, R₉ gesperrt. Wird A OV, ist D₂ noch gesperrt. Die folgende positive Flanke von CP öffnet D₂ und sperrt Ts₁.

Der negative Sprung an seinem Kollektor koppelt über C₃, R₆ auf die Basis von Ts₂. Ts₂ und Ts₃ werden leitend. Die Stufe verharrt in diesem stabilen Zustand (siehe auch Abschnitt 2.3.1.).

Ts₂ und Ts₃ werden gesperrt. Der negative Sprung an seinem Kollektor koppelt über C₂, R₃ auf die Basis von Ts₁, der somit leitend wird. Auch dieser Zustand ist stabil.

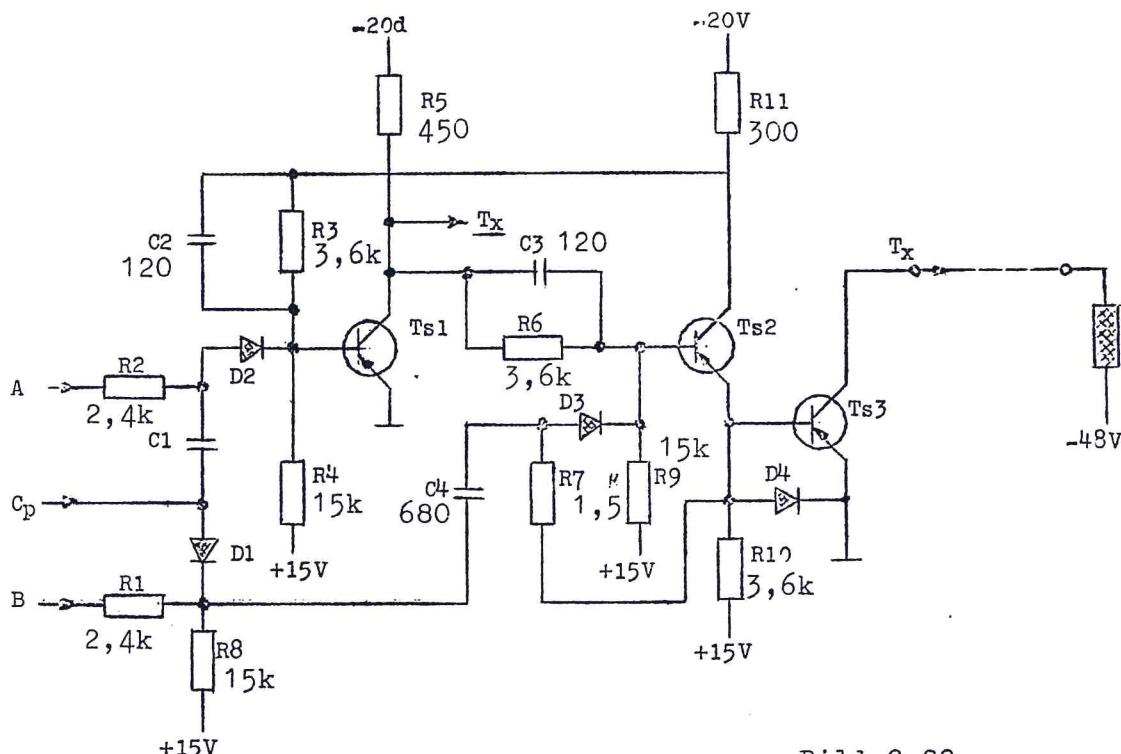


Bild 2.22

2.3.9. Leistungs-Treiber

Der Kupplungstreiber ist ein Leistungsverstärker.

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

Die Dioden D₁ und D₂ sollen bei leitendem Ts₁ das Sperren von Ts₂ gewährleisten. Ist Ts₁ gesperrt, so ist nur R₃ für die Basisstrombegrenzung von Ts₂ maßgebend.

D₃ soll verhindern, daß sich am Emitter von Ts₂ ein positiveres Potential als +1V einstellt (1V reicht zum Sperren von Ts₃), weil sonst Ts₂ nicht über den Spannungsteiler R₅, R₆ gesperrt werden könnte.

(Bild 2.23)

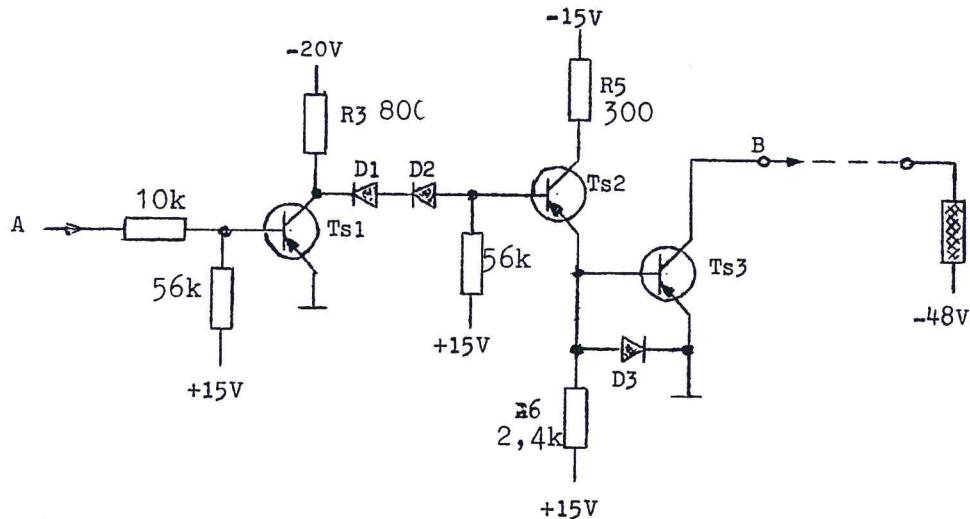


Bild 2.23

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 Sperren von Ts2 gewährleisten. Ist Ts1 gesperrt, so ist nur R1 für die Basisstrombegrenzung von Ts2 maßgebend.

(Bild 2.24)

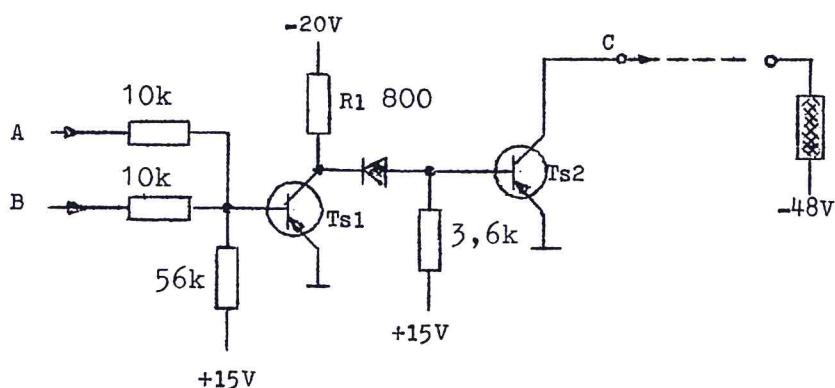


Bild 2.24

2.4. Zeitsteuerung

2.4.1. Zeitmarkierungen

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

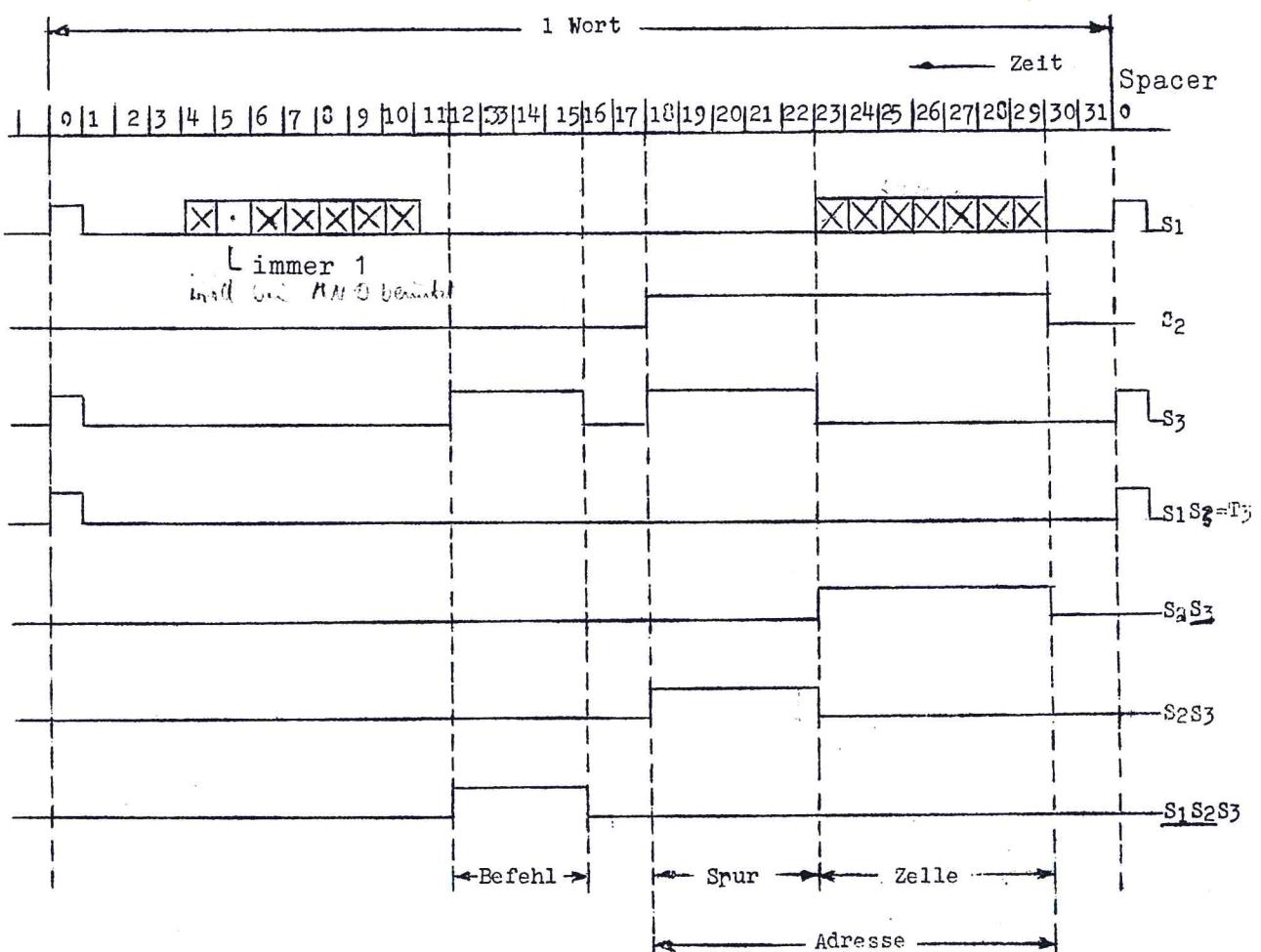


Bild 2.25

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.

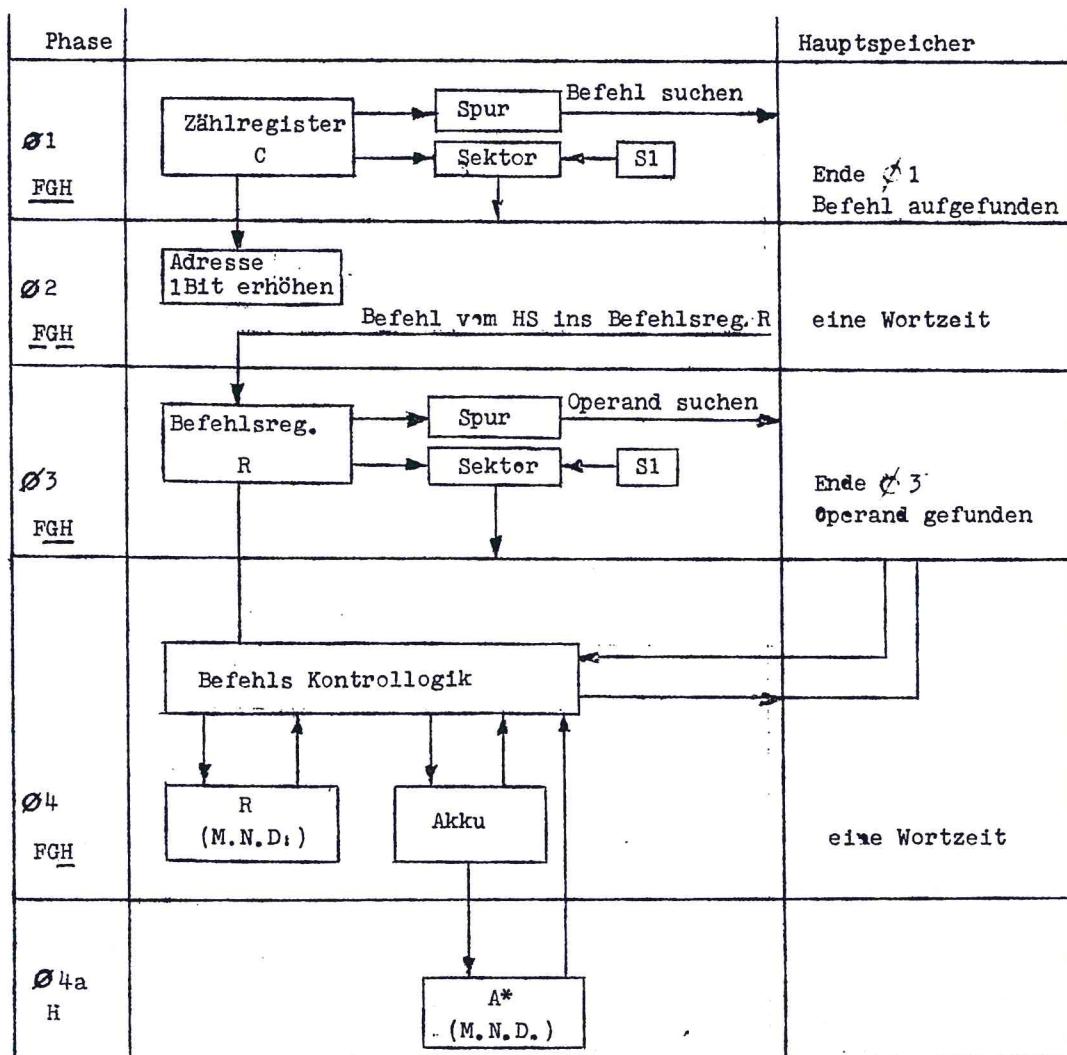


Bild 2.26

Die vier Phasen werden mit den Flip-Flops F, G und H festgelegt. Das H-Flip-Flop ist nur während der verlängerten ϕ_4 gesetzt.

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

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

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

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

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

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

ϕ_1 und ϕ_3 dauert bis zum erfolgreichen Vergleich der Sektoradresse (max 1 Scheibenumdrehung).

ϕ_2 und ϕ_4 dauert eine Wortzeit. Bei der Multiplikation und Division wird die ϕ_4 verlängert in ϕ_{4a} (Zeitdauer siehe M, N, D - Befehl).

2.4.2.1. Phase 1 (ϕ_1)

In ϕ_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 ϕ_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 ϕ_1 beendet und G gesetzt.

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

└ keine Ein- oder Ausgabe

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

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

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

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 ϕ_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.

$$\begin{aligned} P1' &= i \underline{G} r1 + \dots \} \text{ Übernehme } C \text{ zu jeder Bitzeit} \\ \underline{P1'} &= i \underline{G} \underline{r1} + \dots \} \\ &\quad \underline{\underline{L}}_{r1} = \underline{F} C \end{aligned}$$

$$\begin{aligned} P2 &= i P1 \\ \underline{P2} &= i \underline{P1} \quad \text{Übernehme } P1 \text{ zu jeder Bitzeit} \end{aligned}$$

$$\begin{aligned} P3 &= i P2 \\ \underline{P3} &= i \underline{P2} \quad \text{Übernehme } P2 \text{ zu jeder Bitzeit} \end{aligned}$$

usw. für P4 und P5

$$\begin{aligned} i &= \underbrace{S2 \ S3}_{\text{Spur-zeit}} \ \underbrace{H \ G}_{\text{adreß-zeit}} + \dots \\ &\quad \underline{\underline{L}}_{\phi 1 \text{ und } \phi 3} \end{aligned}$$

2.4.2.2. Phase 2 (ϕ_2)

In Phase 2 wird der Befehl, dessen Adresse in ϕ_1 gefunden wurde, vom Hauptspeicher ins R-Register geschrieben.

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

$$\begin{aligned} R'w &= \underbrace{F \ G \ H \ V}_{\text{R-Umlauf in } \phi 1 \text{ und } \phi 3} + R \ \underline{brc} \ (G + \dots) + \dots \\ &\quad \underline{\underline{L}} \text{ nicht Füllen/Löschen} \\ &\quad \text{Schreibinformation } \phi 2 \end{aligned}$$

Der Umlauf in ϕ_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 ϕ_1 , um 1 erhöht.

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

Adreßzeit

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

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

K wird zu Beginn der ϕ_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 ϕ_3 , ϕ_4 , ϕ_1 um:

$$C'w = \underbrace{\text{brc}}_{\text{Umlauf der Adressen}} \ S2 \ \underbrace{\underline{G} \ C}_{\phi_4} + \underbrace{\text{brc}}_{\text{alle Befehle außer U}} \ S2 \ F \ C \ | \ (\underbrace{Q1+Q2+Q3+Q4}_{\text{in } \phi_1 \text{ und } \phi_3}) + \dots$$

ϕ_2 ist nach einer Wortzeit beendet. F wird gesetzt und G zurückgesetzt. Damit beginnt ϕ_3 .

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

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

2.4.2.3. Phase 3 (ϕ_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 ϕ_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.

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

$\underline{\quad}$ keine Ein- oder Ausgabe

$$\underline{K'} = \underline{G} \underline{H} \underline{S2} \underline{S3} \underline{Faf} (\underline{S1} \underline{r1} + \underline{S1} \underline{r1})$$

$(\phi_1)\phi_3$ Sektoradreßzeit

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

$S1r1 + S1r1 = \dots C S1 + \dots C S1$, d.h. dieser Ausdruck ist nur bei Nichtübereinstimmung von R und S1 logisch "1"

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

$\underline{\quad}$ kein Stop

Um auch die ausgewählte Spur noch vor Beginn der ϕ_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.

$P1' = i \underline{G} r_1 + \dots$
 $P1'$ = $i \underline{G} \underline{r_1} + \dots$ übernehme R zu jeder Bitzeit
└ $r_1 = F \underline{H} R$

$P2' = i P1 + \dots$
 $P2'$ = $i \underline{P1} + \dots$ übernehme P1 zu jeder Bitzeit

$P3' = i P2 + \dots$
 $P3'$ = $i \underline{P2} + \dots$ übernehme P2 zu jeder Bitzeit

usw. für $P4'$ und $P5'$

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

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

$Q1' = \alpha_{11} R + \dots$
 $Q1'$ = $\alpha_{11} \underline{R} + \dots$ übernehme R in die Q-Flip-Flops

$Q2' = \alpha_{11} Q1 + \dots$
 $Q2'$ = $\alpha_{11} \underline{Q1} + \dots$ übernehme Q1 nach Q2

usw. für $Q3$ und $Q4$

$\alpha_{11} = \underbrace{F \underline{G} \underline{H}}_{\phi 3} \underbrace{\underline{S1} \underline{S2} \underline{S3}}_{\text{Befehls-zeit}} \underline{ba} \underline{Faf}$
└ keine Ein-/Ausgabe
keine Eingabe von Hand

2.4.2.4. Phase 4 ($\phi 4$)

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 $\phi 1$ bis $\phi 3$ für alle Befehle ohne Änderung um und in $\phi 4$ für die Befehle U, T, H, Y, R, P und Z.

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

$\underbrace{}_{E/A\text{-Taste nicht gedrückt}}$

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:

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

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

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 P_1 gesetzt ist, erfolgt die Rücksetzung.

$$\begin{array}{rcl} \underline{P_1}' & = & G \ T\bar{3} \ P_1 \ P_{1b} \\ & & | \\ & & \underline{\text{Faf}} \end{array}$$

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

$$\begin{array}{rcl} G' & = & \underbrace{F}_{\emptyset 3} \ \underbrace{G}_{\text{Z-(P)Befehl}} \ \underbrace{T\bar{3}}_{\text{be}} \ \underbrace{\underline{Q_2} \ \underline{Q_3} \ \underline{Q_4}}_{\text{Faf}} \end{array}$$

Zur Spuradresszeit ($S_2 \ S_3$) 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 ϕ_1 zurückgesetzt).
Der Z 0100 wird in den P-Flip-Flops zum Z 0000.

Nach der folgenden ϕ_4 (1 Wortzeit) bleibt der Rechner in ϕ_1 stehen. Das G-Flip-Flop wird nicht gesetzt. Das Setzen von G wird durch Q2 entschieden.

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

Zur T3-Zeit der ϕ_4 wird Q2 für alle Befehle außer Z 0000 und Z 0100 gesetzt:

$$Q2' = \begin{array}{l} \underline{F} \underline{G} T3 \underline{Q2} \underline{01} (\underbrace{Q1+Q2+Q3+Q4+P1+P2+P3+P4+P5+P6}_{\text{außer Z0000, Z0100}}) \\ \downarrow \phi_4 \end{array} \\ \text{kein Step, kein Manuell}$$

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

$$Q2' = \begin{array}{l} \underline{F} \underline{G} \underline{H} bs \\ \downarrow \phi_1 \end{array} \downarrow \text{"Start"}$$

2.5.1.2. Z-Befehl mit Spuradressen > 3

Q1 entscheidet in ϕ_2 über die Ausführung des Z-Befehls. Ist Q1 in ϕ_2 gesetzt, wird der nächste Befehl übersprungen. F kann nicht gesetzt werden. Die folgende ϕ_3 wird nicht ausgeführt. Da am Ende von ϕ_2 G jedoch wie immer zurückgesetzt wird, läuft der Rechner wieder in ϕ_1 . Durch die Adressenerhöhung in ϕ_2 ist jetzt ein Befehl übersprungen worden.

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

Q1 wird am Ende von ϕ_2 und ϕ_4 zurückgesetzt. Damit beginnt wieder ein normaler Phasenablauf.

$$\underline{Q1}' = \begin{array}{l} \underline{G} \underline{H} \underline{Q1} T3 + \dots \\ \downarrow \phi_2, \phi_4 \end{array}$$

Die Spuradresse des Befehls steht in $\emptyset 2$ in den P-Flip-Flops. Die Setzbedingung für Q1 lautet also:

$$Q1' = \underbrace{G \ H}_{\emptyset 2(\emptyset 4)} \underbrace{Q1 Q2 Q3 Q4}_{Z-\text{Befehl}} T3 \underbrace{(P1 \underline{Tb1} + P2 \underline{Tb2} + P3 \underline{Tb3} + P4 \underline{Tb4} + \dots)}_{PS-\text{Tasten}} + \dots$$

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 $\emptyset 2$ gesetzt (weiterer Ablauf s. 2.5.1.2.).

$$Q1' = \underbrace{G \ H}_{\emptyset 2(\emptyset 4)} T3 \underbrace{Q1 \ Q2 \ Q3 \ Q4}_{Z-\text{Befehl}} (R \ C + \dots) + \dots$$

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.

$$Cw' = \underbrace{F \ G}_{\emptyset 4} \underbrace{Q1 Q2 Q3}_{A \text{ oder } S} T3 \underbrace{S}_{A-\text{Befehl}} \underbrace{(I1 I2 L + I1 I2 L)}_{Vorzeichenzeit} + \dots$$

S-Befehl

Bei der Ausführung der Division kann, wenn der Dividend größer als der Divisor ist, ein Überlauf während der zweiten Wortzeit der $\emptyset 4a$ 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 $\emptyset 4a$ in A gespeichert und das des Dividenden in P6. Damit wird das C-Register beim Überlauf durch folgenden Ausdruck gesetzt:

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

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

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

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

2.5.2. B-Befehl

$$B \triangleq Q_1 \underline{Q_2} \underline{Q_3} Q_4 \triangleq 0001$$

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 $\emptyset 4$ wird der Umlauf von A unterbrochen und das vom Hauptspeicher gelesene Wort V in den Akkumulator geschrieben.

$$\begin{aligned} Aw' &= \underbrace{F \ G \ H}_{\emptyset 1, \emptyset 2} \ \underbrace{\underline{Q_1} \ \underline{Q_2} \ \underline{Q_3}}_{\emptyset 3} \ Q_4 \quad V + \dots \\ &\quad \emptyset 4 \quad \text{B-Befehl} \end{aligned}$$

Der neue Inhalt läuft in A um.

$$\begin{aligned} Aw' &= (\underbrace{F}_{\emptyset 1}, \underbrace{G}_{\emptyset 2}) \ \underline{H} \ \underbrace{T_0}_{\substack{| \\ \text{Umlauf}}} \ A + \dots \\ &\quad \emptyset 3 \quad \text{--- E/A-Schalter nicht gedrückt} \end{aligned}$$

Am Ende von $\emptyset 4$ werden F und G zurückgesetzt.

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

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

2.5.3. Y-Befehl

$$Y \hat{=} Q_1 \underline{Q_2} \underline{Q_3} \underline{Q_4} \hat{=} 0010$$

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 $\emptyset 4$.

$$\begin{aligned}W &= F G \underline{Q_1} \underline{Q_2} \underline{Q_3} S2 + \dots \\&\emptyset 4 \quad Y(R)-\text{Befehl}\end{aligned}$$

└── Adreßzeit

Die Schreibinformation Vw' wird aus dem Akkumulator geholt.

$$Vw' = L (\underline{Q_4} A + \dots)$$

└── kein R-Befehl

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

$$L' = F G \underline{Q_2} + \dots$$

└── auch beim Y-Befehl

2.5.4. R-Befehl

$$R \triangleq Q_1 \underline{Q_2} Q_3 Q_4 \triangleq 0011$$

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.

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

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 ≥ 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).

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

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 Taktten 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 Shiftwortzeit. 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).

$$\underline{P1'} = \underbrace{G}_{\emptyset_4} \underbrace{T3}_{Faf} \underbrace{P1}_{\emptyset_4a} \underbrace{P1b}_{\emptyset_4a} + H \underbrace{P1}_{\emptyset_4a} \underbrace{T3}_{\emptyset_4a}$$

$$P1' = \underbrace{H}_{\emptyset_4a} \underbrace{P1}_{\emptyset_4a} \underbrace{T3}_{\emptyset_4a}$$

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

$$Rw' = \underbrace{G}_{\emptyset_4} \underbrace{H}_{D\text{-Befehl}} \underbrace{\overbrace{Q1 Q2 Q4} V}_{\emptyset_4}$$

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

$$P5' = \underbrace{GH}_{WZ1} \quad T3 \quad P5a \cdot V$$

Vorzeichen Multiplikand/Divisor
Faf

$$\underline{P5'} = \underbrace{GH}_{WZ1} \quad T3 \quad P5a \quad V$$

Vorzeichen Multiplikand/Divisor

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

Der Inhalt von P6 bleibt jeweils zwei Wortzeiten lang erhalten.

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

Vorzeich. | D-Bef. | Restwert
Dividend . Vorzeichen

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

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

$$Aw' = \underbrace{HA}_{\phi 4} \underbrace{\text{To}}_{\substack{\text{Umlauf} \\ \text{Vorzeichen-Ausblendung}}} \left[\underbrace{T3}_{\substack{| \\ \text{D-Befehl}}} \underbrace{Q1}_{\substack{| \\ }} \underbrace{Q2}_{\substack{| \\ }} \underbrace{(Q4 + \dots)}_{\substack{| \\ }} + \dots \right]$$

Während der weiteren Rechnung ($\phi 4a$) 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.

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

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

$$I_1 = \underbrace{F \underline{G} \underline{P1} \underline{Q3}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} (A + \underbrace{P6 \underline{T3}}_{\begin{array}{l} | \\ | \\ - \\ - \end{array}})$$

Vorzeichen des Dividenden
Dividend
D-Befehl
2. Wortzeit $\emptyset 4a$

$$I_2 = H \underbrace{Q3 \underline{P1}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} R + \dots$$

Divisor
geradzahlige Wortzeit
Division

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

$$L' = (\underline{T3} + \dots) (\underbrace{S \underline{I1} \underline{I2} \underline{L}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} + \underbrace{S \underline{I1} \underline{I2} \underline{L}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}}) + \dots$$

Addition Subtraktion

$$\underline{L'} = \underbrace{T3 \underline{Q3}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} + \underbrace{Q2 \underline{Q1}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} (\underbrace{S \underline{I1} \underline{I2} \underline{L}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} + \underbrace{S \underline{I1} \underline{I2} \underline{L}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}}) + \dots$$

Anfang jeder D-Bef. Subtraktion Addition
Wortzeit beim
D(A,S)-Befehl

Das Ergebnis wird nach A und A* geschrieben.

$$Aw' = A^*w' = H \underbrace{P1}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} (\underbrace{L \underline{I1} \underline{I2} + \underline{L} \underline{I1} \underline{I2} + \underline{L} \underline{I1} \underline{I2} + L \underline{I1} \underline{I2}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}})$$

2. Wortzeit Addierlogik
 $\emptyset 4a$

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).

$$I_1 = \underbrace{G \underline{H} \underline{P1}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} \underbrace{Q3 \underline{A^*} + F \underline{H} \underline{Q3} \underline{A^*}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}}$$

Division 65., 66. Wortzeit
3., 5...63. Wortzeit

$$I_2 = H \underbrace{P1 \underline{Q3}}_{\begin{array}{l} | \\ | \\ - \\ - \\ - \end{array}} R = "0"$$

nicht ungeradzahlig

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

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.

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

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

67. Wortzeit Subtraktion einer 1

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

67. Wortzeit

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

67. Wortzeit Addierlogik

Fall (+a) : (+b)

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

$$I_1 = (P_5 \underset{0}{A^*} + P_5 \underset{1}{A^*}) \stackrel{\wedge}{=} \underset{1}{A^*}$$

$$I_2 = (P_5 \underset{0}{P_6} + P_5 \underset{1}{P_6}) \stackrel{\wedge}{=} 0$$

$$S = \underbrace{F \ G \ H}_{\stackrel{\wedge}{=}} 1$$

$$\text{und damit } Aw' \stackrel{\wedge}{=} \underset{1}{A^*}$$

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

$$I_2 = (P_5 \underset{0}{P_6} + P_5 \underset{1}{P_6}) \stackrel{\wedge}{=} 1$$

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

Fall (-a) : b

Das negative Vorzeichen des Dividenden 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

$$I1 = \left(\begin{array}{c} P5 \\ 1 \end{array} \right) A^* + \left(\begin{array}{c} P5 \\ 0 \end{array} \right) \underline{A^*} \hat{=} A^* \quad \text{und mit } P6=1 \text{ auch}$$

$$I2 = \left(\begin{array}{cc} P5 & P6 \\ 1 & 1 \end{array} \right) + \left(\begin{array}{cc} P5 & P6 \\ 0 & 0 \end{array} \right) \hat{=} 1$$

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

$$P_6 = 0 \text{ auch } I_2 = (\overset{\cdot}{P}_5 \underset{1}{P}_6 + \underset{0}{P}_5 \underset{0}{P}_6) \stackrel{\wedge}{=} 0$$

Fall (-a) : (-b)

Die negativen Vorzeichen werden wieder in der 1. Wortzeit in die Flip Flops P6 und P5 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 ($A_w' \stackrel{\wedge}{=} A^*$).

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 ϕ_4 und ϕ_{4a}

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

Am Ende von ϕ_4 werden F und G für zwei Wortzeiten zurückgesetzt.

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

Am Ende der 3. Wortzeit wird F wieder gesetzt.

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

Am Ende der 64. Wortzeit wird G gesetzt.

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

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.

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

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

Aus der S1-Spur wird zu jeder Wortzeit der ϕ_3 während $\underline{S2} \underline{S3}$ eine (neue) Sektoradresse in das C-Register geschrieben:

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

Ende ϕ_3 steht die Operandenadresse des D (N, M)-Befehles im C-Register und läuft in ϕ_4 um.

$$Cw' = G \underset{\emptyset 3, \emptyset 4}{|} \underline{S2} \underset{\text{Umlauf}}{\downarrow} C \underline{brc} (Q_1 + Q_2 + Q_3 + Q_4 + \dots) + H \underset{\emptyset 4a}{C} \underline{brc} + \dots$$

Nicht Füllen/
Löschen
Umlauf $\emptyset 4a$

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

Am Ende der 66. Wortzeit wird F zurückgesetzt:

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

L gerade Wortzeit

Wortzeit LGP 2.1	Wortzeit Beispiel	F G H P1 P5 P6						I1 I2 S			AW			Akkuinhalt am Wortzeitende A*			Bemerkungen						
		F	G	H	P1	P5	P6	I1	I2	S	A.	T3	(1)	(V)	(A)	X	X						
1	1	1	1	0	X						0	0	1	1	1	0	0	Dividend durch B-Bef. in Akku					
2	2	0	0	1	0	0	0	(A+T3P6)	R	(P5P6+P5P6) = 1	1	$\Sigma = (A+T3P6) - R$	1	1	0	1	1	0	0				
3	3	0	0	1	1	0	0	A*	0	$\Sigma = A^* \cdot T3$	0	1	1	1	0	0	1	1	0	0			
4	4	1	0	1	0	0	1	A*	R	= 0	0	$\Sigma = A^* \cdot P1 \cdot R$	0	0	1	0	1	0	0				
5	5	1	0	1	1	0	1	A*	0	$\Sigma = A^* \cdot T3$	0	1	1	0	0	0	1	1	0	0			
6	6	1	0	1	0	0	0	A*	R	= 1	0	$\Sigma = A^* - R$	0	0	0	0	0	0	1	0	0		
7	7	1	0	1	1	0	0	A*	0	$\Sigma = A^* \cdot T3$	0	1	1	0	0	0	0	0	0	0			
8	8	1	0	1	0	0	0	A*	R	= 1	0	$\Sigma = A^* - R$	1	0	1	0	0	0	1	0	0		
9	9	1	0	1	1	0	0	A*	0	$\Sigma = A^* \cdot T3$	0	1	0	0	1	0	0	0	0	0			
10	10	1	0	1	0	0	1	A*	R	= 0	0	$\Sigma = A^* \cdot P1 \cdot R$	1	0	1	1	0	0	0	1	0	1	
11	11	1	0	1	1	0	1	A*	0	$\Sigma = A^* \cdot T3$	0	0	0	0	1	0	0	0	1	1			
12	12	1	0	1	0	0	1	A*	R	= 0	0	$\Sigma = A^* \cdot P1 \cdot R$	1	0	1	1	0	0	0	0	1	1	
13	13	1	0	1	1	0	1	A*	0	$\Sigma = A^* \cdot T3$	0	0	0	0	1	1	0	1	0	0	0		
14,16...62	14	1	0	1	0	0	1	A*	R	= 0	0	$\Sigma = A^* \cdot P1 \cdot R$	1	0	1	1	0	0	0	0	1	1	
15,17...63	15	1	0	1	1	0	1	A*	0	$\Sigma = A^* \cdot T3$	0	0	1	0	1	1	0	1	0	0	0		
64	16	1	0	1	0	0	1	A*	R	= 0	0	$\Sigma = A^* \cdot P1 \cdot R$	1	0	1	1	0	0	0	0	1	1	
65	17	1	1	1	1	0	1	A*	0	$\Sigma = A^* \cdot T3$	0	1	0	0	1	1	1	0	1	0	0		
66	18	1	1	1	0	0	1	A*	R	$(P5P6+P5P6) = 0$	0	$\Sigma = A^* \cdot P1 \cdot R$	1	0	1	1	0	0	0	1	1	1	
67	19	0	1	1	1	0	1	(P5A*+P5A*) (P5P6+P5P6)	1	$\Sigma = A^* - 0$	0	$\Sigma = A^* - 0$	0	1	1	0	0	0	0	1	0	0	0
$b > /a/$		a		:		b		:		0.1 0 1 0 0 0 0		= 0.1 1 0 0 0 0 0		Divisionstabelle		Divisionstabelle							
Dezimal:		15 \textcircled{A} 5		:		5 \textcircled{Q} 3		= \textcircled{R} 2 : 3		Divisor		LGP 21		LGP 21									

2.5.6. N-Befehl

$$N \stackrel{?}{=} Q_1 \ Q_2 \ Q_3 \ Q_4 \stackrel{?}{=} 0110$$

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
$a \times b \stackrel{?}{=} a \times b \stackrel{?}{=} a \cdot b$		$+ab \stackrel{?}{=} ab$
$-a \times b \stackrel{?}{=} (2-a) \times b \stackrel{?}{=} 2b - ab$		$-ab \stackrel{?}{=} (2-ab)$
$-a \times (-b) \stackrel{?}{=} (2-a) \times (2-b) \stackrel{?}{=} 4-2b-2a+ab$		$+ab \stackrel{?}{=} ab$
$a \times (-b) \stackrel{?}{=} a \times (2-b) \stackrel{?}{=} 2a-ab$		$-ab \stackrel{?}{=} (2-ab)$

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 ≥ 1 nicht darstellbar sind. Es wird aber nicht gewidmet.

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 $\emptyset 4$ wird der Multiplikand v in das R-Register geschrieben, um ihn während der Ausführung des Befehls bei jeder Addition sofort zur Verfügung zu haben.

$$Rw' = \underbrace{G \underline{H} V \underline{Q1 Q2 Q3}}_{\emptyset 4} + \underbrace{R \underline{brc H}}_{\text{Multiplikation} \quad \text{Umlauf in } \emptyset 4a}$$

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

$$P5' = \underbrace{G \underline{H} T3 V}_{\emptyset 4} P5a + \dots$$

E/A-Verriegelung
- Multiplikand

$$\underline{P5'} = \underbrace{G \underline{H} T3 V}_{\emptyset 4} P5a + \dots$$

+ Multiplikand

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

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

↓ ↓ ↓
 φ1, φ2 φ3
 -Umlauf A

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

$$Aw' = \underline{H} \underline{To} A \underline{T3} \underline{\underline{Q1 Q2 Q3}} + \dots$$

↓
 N-Multiplikation
 ↓
 Ausblendung des Vorzeichens

$$P6' = \underbrace{\underline{F} \underline{G} \underline{H}}_{\phi 4} \underbrace{\underline{T3} \underline{A}}_{\begin{matrix} | \\ -A \end{matrix}} P6a + \dots$$

↓
 E/A-Verriegelung

$$\underline{P6'} = \underbrace{\underline{F} \underline{G} \underline{H}}_{\phi 4} \underbrace{\underline{T3} \underline{A}}_{+A(\text{Multiplikator})} P6a$$

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.

$$\underline{P1'} = \underbrace{G}_{\phi 4} \underline{T3} P1 \underbrace{P1b}_{Faf} + \underline{H} \underline{T3} \underbrace{P1}_{\phi 4a}$$

$$P1' = \underbrace{H}_{\phi 4a} \underline{T3} \underline{P1}$$

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)).

I1 = 0

I2 = G H P1 P6 R

↘
 Multiplikand
 ↙
 Multiplikator

S = F H Q3

↗
 2 WZ. ↘ M, N-Befehl
 ↗
 Ø4a

L' = (T3 + H P1 Q3) (S I1 I2 L, + ...)

WZ 2 - MN-Befehl |
 Übertragslogik

L' = T3 H + Q2 (Q3 + Q1) (S I1 I2 L, + ...)

↗ auch MN-Befehl Übertragslogik
 ↗ Rücksetzer am Ende aller Ø

Aw' = H P1 (L I1 I2 + L I1 I2 + L I1 I2 + L I1 I2)

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 geschiftet, und es verschwindet die zuletzt ausmultiplizierte Stelle. Die neu zu multiplizierende Stelle wird ausgeblendet und in P6 (alle ungeradzahligen Wortzeiten) gespeichert.

P6' = H T3 P1 Q3 A* + ...

ungeradzahlige WZ

P6' = H T3 P1 Q3 A*

Um das Vorzeichen des Multiplikanden zu berücksichtigen, wird bei negativem Multiplikanden (P5 = "1") eine "1" von jedem Bit des Multiplikators subtrahiert. Bei positivem Multiplikanden (P5 = "0") wird von jedem Bit des Multiplikators eine "0" subtrahiert (Korrekturfaktor).

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

┌─────────┐
 M, N-Befehl
 └─────────┘ ungerade Wortzeiten Ø4a

$$I_2 = \underbrace{\underline{F} \ H \ P_1}_{3 \cdot WZ} \ Q_3 \ P_5$$

┌─────────┐
 Vorzeichen Multiplikand (R)
 └─────────┘ M, N-Befehl

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

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

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

Addierlogik

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

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

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

M, N-Befehl

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

$$\underline{L}' = T_3 P_1 + Q_2 Q_3 \underline{S} \underline{I_1} \underline{I_2} L + \dots$$

$$Aw' = Aw^{*'} = (H \underline{T_3+HP_1+\dots})(LI_1I_2 + \underline{LI_1I_2} + \underline{LI_1I_2} + \underline{LI_1I_2})$$

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.

$$\underline{F}' = F \underline{G} \underline{H} T_3 \quad \underline{F} \underline{G} \underline{H} = 2. \text{ und } 3. \text{ Wortzeit}$$

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

Am Ende der 3. Wortzeit wird F wieder gesetzt.

$$\underline{F}' = \underline{F} \underline{G} H P_1 T_3 \quad \underline{F} \underline{G} \underline{H} = 4. \text{ bis } 64. \text{ Wortzeit}$$

Am Ende der 64. Wortzeit werden F und H zurückgesetzt

$$\underline{F}' = F H T_3 \underline{Q_4} K$$

↓
Sektorkoinzidenz
nach 64 Wortzeiten

$$\underline{H}' = H T_3 \underline{Q_4} K$$

↓
N-Befehl

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.

$$K' = T_3 \underline{Faf} (F + G + R + Q_3 + Q_4 + \dots)$$

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

∅4a	Anti- koinzidenz	Bit 1...1, 16, 17, 30, 31
-----	---------------------	---------------------------

Aus der S1-Spur wird zu jeder Wortzeit der ∅3 während S2 S3 eine (neue) Sektoradresse in das C-Register geschrieben:

$$Cw' = \underbrace{G \underline{H} \underline{S2} \underline{S3}}_{\emptyset 3} S1 + \dots$$

Ende ∅3 steht die Operandenadresse des M,N(D)-Befehls im C-Register und läuft in ∅4 um.

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

∅3, ∅4	Umlauf	nicht "Füllen/ Löschen"
--------	--------	----------------------------

Umlauf ∅4a

Nach der Sektorkoinzidenz werden F und H zurückgesetzt, und der Rechner steht wieder in ∅1.

2.5.7. M-Befehl

$$M \triangleq Q_1 \ Q_2 \ Q_3 \ Q_4 \triangleq 0111$$

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:

$$I_1 = F \ G \ Q_3 \underline{P_1} \ A + \dots$$

$$I_2 = G \ H \ Q_3 \underline{P_1} \ A + \dots$$

$$A_{W'} = H \underline{P_1} (L \ I_1 \ I_2 + L \ \underline{I_1} \ \underline{I_2} + L \ I_1 \ \underline{I_2} + L \ \underline{I_1} \ I_2)$$

Wortzeitgruppen

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

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

Die 65. Wortzeit ist gegeben mit

$G \ F \ H \ P1.$

Die 66. Wortzeit ist gegeben mit

$F \ G \ H \ \underline{P1}$

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

$\underline{F'} = F \ G \ T3 \ \underline{P1} + \dots$

$\underline{G'} = G \ H \ T3 \ Q3 \ \underline{P1}$

$\underline{H'} = G \ H \ T3 \ Q3 \ \underline{P1}$

Wortzeit LGP 21	Wortzeit Beispiel	F	G	H	P1	P5	P6	I1	I2	S	AW	AN	Akkuinhalt am Wortzeitende		Bemerkungen
		X	X	(A)	(V)	X	A ₀ .T3	0 1 0 1 0 0 0 0	0 1 0 1 0 0 0 0 0	A	A*				
1	1	1 1 0 X				0	P6R	1	$\sum = 0$	0 0 0 0 0 0 0 0	0 1 0 1 0 0 0 0 0				Multiplikator durch B-Befehl in Akku
2	2	0 0 1 0	0	0	0	0	P5	1	$\sum \cdot T3 = A^* \cdot T3$	0 0 1 0 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
3	3	0 0 1 1	0	0	A*	P5	1	$\sum = A^* \cdot T3$	0 1 1 0 0 0 0 0	0 0 0 1 0 0 0 0 0					P6 geht aus für WZ 6 und 7
4	4	1 0 1 0	0	1	A*	P6R	0	$\sum = A^* P1 R$	0 1 1 0 0 0 0 0	0 0 1 0 0 0 0 0 0					
5	5	1 0 1 1	0	1	A*	P6P5	0	$\sum \cdot T3 = A \cdot T3$	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	0	$\sum = A^*$	1 1 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0					
7	7	1 0 1 1	0	0	A*	P6P5	0	$\sum \cdot T3 = A \cdot T3$	0 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0 0					
8	8	1 0 1 0	0	1	A*	P6R	0	$\sum = A^* P1 R$	1 1 1 0 0 0 0 0	0 0 0 0 0 0 0 0 1					
9	9	1 0 1 1	0	1	A*	P6P5	0	$\sum \cdot T3 = A^* \cdot T3$	0 0 0 0 0 0 0 1	1 1 1 0 0 0 0 0 0					
10	10	1 0 1 0	0	0	A*	P6R	0	$\sum = A^*$	1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 1 1					
11	11	1 0 1 1	0	0	A*	P6P5	0	$\sum \cdot T3 = A^* \cdot T3$	0 0 0 0 0 0 0 1	1 1 0 0 0 0 0 0 0					
12	12	1 0 1 0	0	0	A*	P6R	0	$\sum = A^*$	1 0 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	0	$\sum \cdot T3 = A^* \cdot T3$	0 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	0	$\sum = A^*$	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1					
15,17...63	15	1 0 1 1	0	0	A*	P6P5	0	$\sum \cdot T3 = A^* \cdot T3$	0 0 0 0 0 1 1 1	0 0 0 0 0 0 0 0 0					
64	16	1 0 1 0	0	0	A*	P6R	0	$\sum = A^*$	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 0				N endet hier	
65	17	1 1 1 1	0	0	A*	P6P5	0	$\sum \cdot T3 = A^* \cdot T3$	0 0 0 1 1 1 0	0 0 0 0 0 0 0 0 0					
66	18	1 1 1 0	0	0	A	A	0	\sum	0 0 1 1 1 0 0	0 0 0 1 1 1 1 0 0				M endet hier	

Multiplicationstablette
LGP 21

Dezimal
bei q
5
5
3
3
 \oplus
 \oplus

$$0.1010000 \times 0.1000000 = 0.0111100$$

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.

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

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.

$$G' = \underbrace{F \underline{G} T_3}_{\emptyset 3} \underbrace{Q_1 \underline{Q_2} Q_3}_{\substack{\text{be} \\ \text{U(T)-Befehl}}} + \dots$$

$\emptyset 4$ | Taste Ausführen nicht gedrückt

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

 | Inhalt des R-Registers

2.5.10. T-Befehl

$$T \triangleq Q_1 \underline{Q_2} Q_3 Q_4 \triangleq 1011$$

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 $\emptyset 3$.

$$\underline{Q4'} = \underline{\underline{Q1 \underline{Q2} Q3}} \quad T3 \text{ A} + \underline{\underline{Q1 \underline{Q2} Q3 \quad T3 \text{ R}}} \quad Tc$$

$\underbrace{\text{T-Befehl}}_{\text{A negativ?}} \quad \underbrace{-\text{T-Befehl}}_{\text{Signal der PST-Taste}}$

2.5.11. H-Befehl

$$H \triangleq Q_1 \ Q_2 \ \underline{Q_3} \ \underline{Q_4} \triangleq 1100$$

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

Die Schreiberlaubnis W erfolgt in ϕ_4

$$W = \underbrace{F \ G \ H}_{\phi_4} \ \underbrace{Q_1 \ Q_2 \ \underline{Q_3}}_{(C)H-\text{Befehl}} + \dots$$

Die Schreibinformation Vw' lautet:

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

$$L' = W + \dots$$

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

Der Umlauf in A erfolgt über:

$$Aw' = \underbrace{\underline{H} \ To \ A}_{\text{Umlauf in } \phi_1-\phi_3} \left[\underbrace{F+G+Q_1 \ Q_3 \ Q_4}_{\text{Umlauf in } \phi_4 \text{ für alle Befehle außer C}} + T_3 \ Q_1 \ Q_2 (Q_3+Q_4) + Q_2 (Q_3+Q_4) \underline{Faf} \right]$$

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 ϕ_4 :

$$W = \underbrace{F \ G \ H}_{\phi_4} \ \underbrace{Q_1 \ Q_2 \ \underline{Q_3}}_{C(H)-\text{Befehl}} + \dots$$

Die Schreibinformation in V_w' lautet:

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

$$L' = W + \dots$$

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

Der Umlauf in A wird beim Löschen unterbrochen.

$$A_w' = \underbrace{H \ To \ A}_{\begin{array}{l} \text{Umlauf in } Q_1, \phi_2, \phi_3 \\ \text{Umlauf in } \phi_4 \text{ für alle} \\ \text{Befehle außer C} \end{array}} \left[\underbrace{F+G+Q_1 \underline{Q_3} Q_4 + T_3 Q_1 Q_2 (Q_3+Q_4)}_{\text{Umlauf in } Q_1, \phi_2, \phi_3} + \underbrace{Q_2 (Q_3+Q_4) Faf}_{\text{Umlauf in } \phi_4} \right]$$

2.5.13. A-Befehl

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

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 $\emptyset 4$ die Überträge. L wird am Ende einer jeden Wortzeit (ausgenommen $\emptyset 4a$) 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.

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

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

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

$$\underline{L'} = T_3 \ \underline{H} + \underbrace{L \ \underline{S} \ \underline{I_1} \ \underline{I_2}}_{\text{unwichtig für A-Befehl. Verhindert das Rücksetzen bei einem Y-, R-, H- und C-Befehl}} \ \underbrace{Q_2 \ (Q_3 + \dots)}_{\text{Rücksetzer für A und V beide "0"}} + \dots$$

Das Signal S gibt die Addition an

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

T₃ in der Setzbedingung für L verhindert, daß L zur T₃-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:

$$Aw' = (\underbrace{FG \ Q_1 Q_2 Q_3 + \dots}_{\emptyset 4 \ A(S)-Befehl}) (\underbrace{LI_1 I_2 + LI_1 \underline{I_2} + \underline{L} I_1 I_2}_{\text{Addierlogik}}) + \dots$$

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 ≈ -14

2.5.14. S-Befehl

$$S \doteq Q1 \ Q2 \ Q3 \ Q4 \doteq 1111$$

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.

Beispiel 1:	A = 30	$\doteq 11110$	$+27 \doteq 11011$
	V = -27	$\doteq 00101$	-27 $\doteq 00100$ invers
			00101 Komplement
	3	00011	$\doteq 3$

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.

Beispiel 2:	A = 30	$\doteq 11110$	invertiert: 00001
	V = -27	$\doteq 11011$	$+11011$
		3	11100
			invertiert: 00011 = 3

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

S = Q4 H = Angabe des Subtraktionsbefehles

I1 = A H

I2 = V H

L' = T3 H S L I1 I2
 | Invertierung von A

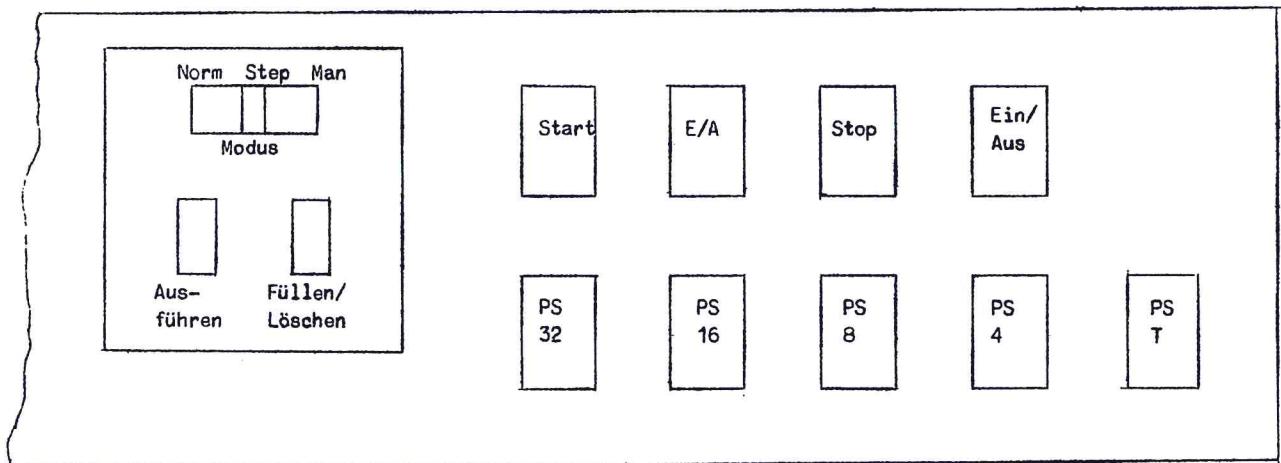
L' = T3 H + L S I2 I1 Q2(Q3+...)
 ist für den S-Befehl unwichtig

Die Schreibinformation Aw' ergibt sich dann wie folgt:

$$Aw' = (\underbrace{FG \ Q1 \ Q2 \ Q3}_{\emptyset 4} + \dots) (\underbrace{LI1 \ I2 + LI1 \ I2 + LI1 \ I2 + LI1 \ I2}_{(A)S-Befehl \quad Addierlogik})$$

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, $K_c = "1"$)

$$K_c = \underbrace{F}_{\emptyset 1} \underbrace{G}_{-} \underbrace{Q2}_{\text{"Stop"}} \quad \begin{array}{l} \text{wird Relais B (Bedienungsfeld)} \\ \text{nicht angesteuert.} \end{array}$$

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.

$$A_{W'} = \underbrace{T_0}_{\text{E/A-Signal}} \underbrace{H}_{\substack{\text{Umlauf}}} \underbrace{A}_{\substack{[F + G + \dots]}}$$

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

Die E/A-Taste leuchtet, solange ein Anwahl-Flip Flop gesetzt ist (Faf $\hat{=}$ "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 Abwechselung mit der Stop-Lampe. Sie leuchtet, wenn Kc nicht erfüllt ist $\hat{=}$ 0 Volt

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

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.).

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

$\emptyset 1$ nicht Start

Q2 wird nach jeder Befehlausführung zurückgesetzt.

$$\underline{Q2}' = \underbrace{\frac{G}{\emptyset 4} \underline{T3} \underline{01}}_{Step} \underline{Q1} + \dots \underbrace{\underline{01} \underline{H'}}_{\begin{array}{l} \text{für M,N,D-Befehl nach} \\ \text{Ablauf von } \emptyset 4a \end{array}} + \dots$$

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:

$$\underline{Q2} = G \ T3 \ \underbrace{\underline{Q1} \underline{Q2} \underline{Q3} \underline{Q4}}_{\begin{array}{l} I-\text{Befehl} \\ Step \end{array}} \underline{01} + \dots$$

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.

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

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

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.

$$F' = be \ T3 + \dots \quad (\text{Springen nach } \phi 3)$$

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) IO000-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 an-
 gewählt (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.

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

\vdots
 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

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

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.

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

keine unlesbaren Zeichen

kein Stop Code

$P1 = P1c$

$P2 = P2c$

usw. bis P6

$P1 = P1d$

$P2 = P2d$

Der Rechner steht in ϕ_3 bis SC7 wieder umschaltet und -20V an JL11 legt. Hierdurch wird G gesetzt und ϕ_4 eingeleitet.

$G' = G_c = JL11 \underbrace{F \quad G}_{\phi_3} T3$

Die P-Flip Flops werden für die Dauer der ϕ_4 durch das i-Signal als Schiftregister geschaltet.

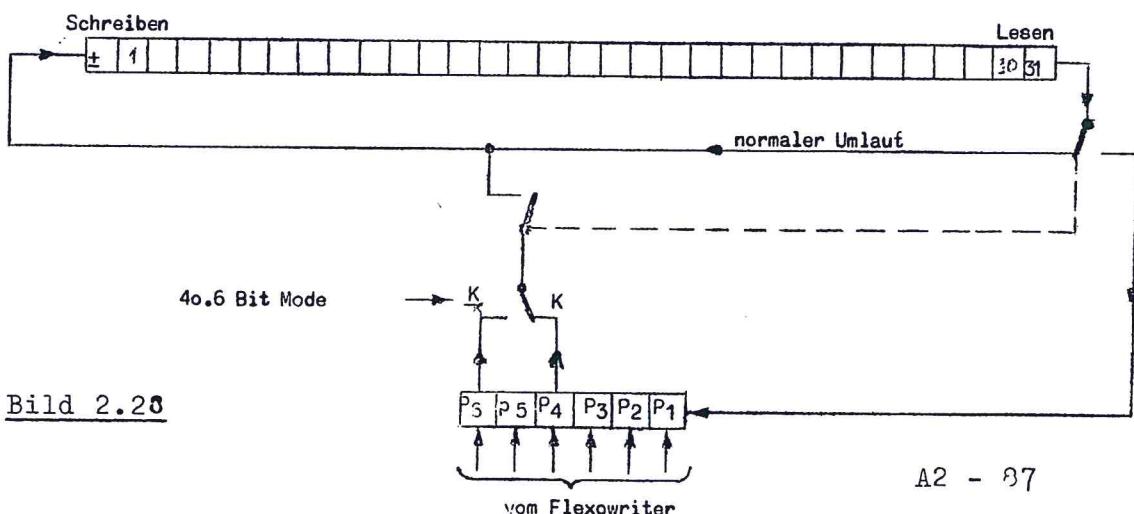
$i = ic + \dots = Faf \underbrace{G \quad F}_{\phi_4}$

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

$P1' = G_i A \quad P2' = G_i P1 \quad \text{usw. bis } P6$

$P1' = G_i A$ $P2' = G_i P1$

$Aw' = Ac1 = \underbrace{F \quad G}_{\phi_4} Q1 \quad Faf \quad K \quad P4$
Eingabe Übernahme P4 (4 Bit)



Die ϕ_4 wird nach einer Wortzeit beendet, und der Rechner wartet in ϕ_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 ϕ_1 , da ϕ_3 und damit auch ϕ_4 nicht gesetzt werden können.

Die 4 Bit-Information wird von P4 in die meistbedeutende Stelle von A eingeschrieben und wandert im Laufe der ϕ_4 bis zur wenigstbedeutenden Stelle (Schiften um 32 Bit des erweiterten Akkumulators), d.h. die Information von P4 steht nach der ϕ_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 ϕ_4 läuft der Inhalt von A wieder normal um. Es folgt die ϕ_1 und ein weiteres Zeichen kann eingegeben werden. Dann verschiebt sich in ϕ_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 ϕ_1 . Die Taste "Manuell" wird zurückgesetzt, d.h. bq wird -20V. Der Flexowriter wird abgewählt.

Ff' = F G T3 Q3 Xs Sr' bQ
 ϕ_1 Eingabe

2.7.3. Eingabe über Flexowriter-Leser Q1 Q2 Q3 Q4 = 0100
(siehe auch 2.10.5.)

Mit einem (800) IO200-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. Gleichzeitig werden die P-Flip Flops zurückgesetzt.

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

$$\underline{P1d} = \underline{P2d} = \underline{P3d} = \underline{P4d} = \underline{P5d} = \underline{P6d} = \\ \text{Rücksetzer } P1...P6$$

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

\emptyset_3 wird nach einer Wortzeit beendet, da keine Operandenadresse 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 Akkumulators eingeschlossen.

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

$$\begin{array}{ll} P1' = G i A & P2' = G i P1 \\ \underline{P1}' = G i \underline{A} & \underline{P2}' = G i \underline{P1} \end{array} \quad \begin{array}{l} \text{usw. bis} \\ P4 \text{ bzw. } P6 \end{array}$$

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)

Ø4 dauert eine Wortzeit, danach beginnt wieder Ø1.

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 Ø3 ein.

F' = Fc = JL12 SC Sk T3
 |
 | keine unlesbaren Zeichen
 |
 | kein Stop Code

P1* = P1c P2* = P2c
P1* = P1d P2* = P2d usw. bis P6

Der Rechner steht in Ø3 bis SC7 wieder umschaltet und -20V an JL11 legt. Hierdurch wird G gesetzt und Ø4 eingeleitet.

G' = Gc = JL11 F G T3

Die P-Flip Flops werden für die Dauer der Ø4 durch das i-Signal als Schiftregister geschaltet und das P-Schiftregister in den A-Umlauf eingeschlossen (s. Bild 2.28)

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

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

Übernahme von P4 bzw. P6

Die Ø4 wird nach einer Wortzeit beendet, und der Rechner wartet in Ø1 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.

Ff' = F G Q3 T3 Sr' Xs 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 $\emptyset 1$, da $\emptyset 3$ und damit auch $\emptyset 4$ 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 $\emptyset 3$ 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
 $\emptyset 4$ 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.

ϕ_1 und ϕ_2 verlaufen normal. ϕ_3 wird auf eine Wortzeit begrenzt, da keine Operandenadresse gesucht zu werden braucht.

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

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

ϕ_4 dauert eine Wortzeit.

Die P-Flip Flops werden in ϕ_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.

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

$$\begin{array}{lll} P1' = G \ i \ A & P2' = i \ P1 & P3' = i \ P2 \\ \underline{P1'} = G \ i \ \underline{A} & \underline{P2'} = i \ \underline{P1} & \underline{P3'} = i \ \underline{P2} \end{array} \quad \text{usw. bis } P6$$

$$Aw' = \underbrace{H \ A \ To}_{\substack{| \\ \text{Umlauf}}} \ \underbrace{Q_1 \ Q_3 \ Q_4}_{P(H)\text{-Befehl}}$$

Am Ende ϕ_4 wird zur Kennzeichnung des P-Befehls für den weiteren Ablauf Q3 gesetzt.

$$Q3' = G \ T_3 \ \underbrace{Q_1 \ Q_2 \ Q_3 \ Q_4}_{\substack{| \\ \text{P-Befehl}}}$$

Der eigentliche Druckvorgang beginnt in der folgenden ϕ_1 .

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

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

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

$$P1, P2, P3, P4, P5, P6 \stackrel{\Delta}{=} \underline{\underline{XX}} X10 \\ \text{beliebig}$$

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

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

Das X-Flip Flop löst den Druckvorgang aus und hält den Rechner in ϕ_1 bis der Druckvorgang ausgeführt ist, wenn ein neuer Druckvorgang folgt.

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

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

$$TP1 = X P1$$

$$TP2 = X P2$$

$$TP6 = X P6$$

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

Die Druckoperation wird ausgeführt, damit kann die Ausgabeauswahl 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 μ 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 Translаторkupplung 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) IO000-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
 $\emptyset 3$ I-Befehl

$\emptyset 3$ wird nach einer Wortzeit beendet, da keine Operandenadresse gesucht zu werden braucht.

G wird gesetzt und der Rechner steht in $\emptyset 4$.

G' = F G T3 Q1 Q2 Q3 Q4 be Faf

Die P-Flip Flops (alle "0") werden für die Dauer von $\emptyset 4$ 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 Q1 Q2 Q3 Q4 (K P4 + K P6)

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

$\emptyset 4$ dauert eine Wortzeit, danach beginnt wieder $\emptyset 1$.

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

Di' = F G Ft Xp
 $\emptyset 1$ | 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

$R_p = Tr$
|
Signal für die Tally-Leser Kontakte

Solange D_i gesetzt ist, verläuft das Signal R_p , R_p , Tr 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 :

$F_c = \underline{F} \underline{G} \underline{T_3} \underline{S_c} \underline{C_d} \underline{S_k} \underline{F_t} R_p$
|
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 $Tr = \underline{R_p} = -20V$ die entsprechenden P-Flip Flops gesetzt.

$P_1 = P_{1c}$
 $P_1 = P_{1d}$ 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 ϕ_4 durch das i-Signal als Schiftregister geschaltet und das P-Schiftregister in den A-Umlauf eingeschlossen (s. Bild 2.28).

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

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

Die Phase 4 wird nach einer Wortzeit beendet und der Rechner wartet in ϕ_1 auf das nächste Zeichen vom Leser.

Wird ein Stop Code, ein Code delete oder ein Maschinenzeichen gelesen, wird F zur T3-Zeit der ϕ_3 zurückgesetzt und der Rechner steht in ϕ_1 .

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

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

$$\underline{Ft}' = \underbrace{F \text{ G}}_{\phi_1} \underbrace{T3}_{\text{Stop Code}} \underbrace{Sc}_{\text{E/A-Taste}} \underbrace{Xp}_{\text{}} + To$$

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 $\phi_1 \dots \phi_{4a}$ 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 ϕ_{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 ϕ_1 und ϕ_3 .

Übertrags-Flip Flop in ϕ_2 bei der Addition von 1 zum C-Register.

Übertrags-Flip Flop in ϕ_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 Translаторtreiber
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 Eingabe 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
B5	Definiert Drucken in Ø1 bei 4 Bit-Mode.

2.9. Logische Funktionsgleichungen

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

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

Ende Ø4 + WZ 66, D, M-Befehl
 $F' = \underline{F} \underline{G} \underline{H} \underline{T3}$ + $\underline{F} \underline{G} \underline{T3} \underline{P1}$ +
 Ø4 Ø2a

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

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

+ gehe in Ø4 nach Sektorvergleich +
 + $\underline{F} \underline{G} \underline{H} \underline{T3} K Ga be$ +
 Ø3 | Faf Sektorvergleich

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

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

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

+ gehe in Ø4 nach 1 WZ bei Z,P-Befehl Ende Eingabezyklus
 + $\underline{F} \underline{G} \underline{T3} \underline{Q2} \underline{Q3} \underline{Q4} be Faf$ + 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 + Ø2 oder WZ 67 beim D-Befehl
 Ø4

+ 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
 + β 6
 - = G H T3 Q3 P1

Ende WZ 66 beim M-Befehl
β 6 = 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 Ø4 (Q3+Q4)
 Hauptspeicher-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' = L (Q1 Q4 K C + Q1 Q4 K C +
 R-Befehl R-Befehl /
 schreibe 0 zur Spacerzeit Addition von 1

übernehme A beim C,H-Befehl + übernehme A beim Y-Befehl
 + Q1 A + Q4 A)
 | A-Inhalt | A-Inhalt
 - C,H-Befehl - Y-Befehl

Schreiberlaubnis in Ø4 beim H,C-Befehl +
 W = F G H Q1 Q2 Q3 +
 Ø4 H,C-Befehl

Schreiberlaubnis zur Adresszeit in Ø4 beim Y,R-Befehl
 + F G S2 Q1 Q2 Q3,
 Ø4 Y,R-Befehl
 Adresszeit

übernehme C in Ø1 + übernehme C in Ø4a + übernehme R in Ø3
 r1 = F C + H C + F H 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 Faf (F + G + R + Q1+Q2+Q3+Q4) -
 Manuell Ø1,Ø2 Ø4 jeder Befehl außer +P
 -Befehl

bei einem -Befehl, nicht beim +E/A-Befehl
 • (F + G + R + Q1 + Q2 + Q3 + Q4)
 Ø1,Ø2 Ø4 jeder Befehl außer +P
 -Befehl

K' = (siehe Rechenwerk 80571, Seite A2-113)

Schalte die P-Flip Flops in $\emptyset 4$ als Schiftregister für

$$Ac1 = \frac{F}{\emptyset 4} \frac{G}{I-\text{Befehl}} \frac{Q1}{n.U.} \frac{BL}{E/A-\text{Mode}} (Faf + \frac{Q1}{I- und Schiftbefehl} \frac{Q2}{} \frac{Q3}{} \frac{Q4}{})$$

4- oder 6-Bit Mode

$$\bullet (\frac{K}{I-\text{Befehl}} \frac{P4}{Übernehme P4} + \frac{K}{+(I)\text{Befehl}} \frac{P6}{Übernehme P6 (6 Bit)})$$

Schalte die P-Flip Flops in $\emptyset 4$ als Schiftregister bei einem I-Befehl

$$ic = \frac{F}{\emptyset 4} \frac{G}{I-\text{Befehl}} \frac{BL}{n.U.} (Faf + \frac{Q1}{E/A-\text{Mode}} \frac{Q2}{I- und Schiftbefehl} \frac{Q3}{} \frac{Q4}{})$$

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

Rücksetzer für P's beim I-Schiftbefehl zur T3-Zeit $\emptyset 3$

$$= \frac{F}{\emptyset 3} \frac{G}{I-\text{ und Schiftbefehl}} \frac{T3}{E/A-\text{Verriegelung}} \frac{Q1}{} \frac{Q2}{} \frac{Q3}{} \frac{Q4}{}, \frac{be}{\text{nicht "ausführen"}} \frac{Faf}{}$$

E/A-Verriegelung

$$P1b = P5a = P6a = ia = Ga = \underline{Faf}$$

E/A-Verriegelung

Vorzeichenzeit

$$T3 = S1 S3$$

$$Aw' = \frac{F \quad G \quad H}{\emptyset 4} \quad \underline{Q2 \quad Q3 \quad Q4} \quad V \quad (Q1 + A) \quad +$$

B,E-Befehl

$$+ \quad A\text{-Umlauf in } \emptyset 1 \dots \emptyset 3, \text{ in } \emptyset 1 \dots \emptyset 4 \text{ beim P,H,U,T,Y,R,Z-Befehl} \quad +$$

$$+ \quad \underline{H} \quad A \quad To \quad [\underline{F} \quad + \quad \underline{G} \quad + \quad \underline{Q1 \quad Q3 \quad Q4} \quad +$$

nicht $\emptyset 4a$ E/A nicht $\emptyset 1, \emptyset 2$ $\emptyset 3$ P,H-Befehl

$$+ \quad \text{und beim D,N,M-Befehl außer Vorzeichen} \quad +$$

$$+ \quad \underline{T3} \quad \underline{Q1} \quad Q2 \quad \alpha 9 \quad + \quad \underline{Q2} \quad (Q3 + Q4) \quad \underline{Faf} \quad] \quad +$$

D,N,M-Befehl (Q3+Q4) U,T,Y,R,Z-Befehl E/A-Verriegelung

$$+ \quad \text{Schiftregister in } \emptyset 4 \text{ bei Eingabe} \quad +$$

$$+ \quad Ac1 \quad +$$

$$+ \quad \text{übernehme die Addierlogik} \quad +$$

$$+ \quad \beta 1 \quad [(L \quad I1 \quad I2 + L \quad I1 \quad I2 + L \quad I1 \quad I2 + L \quad I1 \quad I2)] \quad +$$

s. nächste Gleichung Addierlogik

$$\beta 1 = \underline{H} \quad \underline{T3} \quad + \quad \emptyset 4a \text{ keine Vorzeichenzeit} \quad + \quad \emptyset 4a \text{ gerade WZ} \quad +$$

$$H \quad P1 \quad +$$

$$+ \quad WP 67, D-Befehl \quad + \quad \emptyset 4 \quad A,S-Befehl$$

$$+ \quad \underline{F} \quad G \quad H \quad + \quad \underline{F} \quad \underline{G} \quad \underline{Q1 \quad Q2 \quad Q3}$$

$\emptyset 4$ A,S-Befehl

$$Cw' = F \quad S2 \quad C \quad brc \quad (Q1+Q2+Q3+Q4) \quad +$$

$\emptyset 3, \emptyset 4$ nicht Fü/Lö kein U-Befehl

Adressezeit

$$+ \quad \text{Adressenumlauf in } \emptyset 1, \emptyset 3 \quad + \quad C\text{-Umlauf in } \emptyset 4a$$

$$+ \quad \underline{G} \quad S2 \quad C \quad brc \quad + \quad H \quad C \quad brc \quad +$$

$\emptyset 1, \emptyset 3$ nicht Fü/Lö $\emptyset 4a$ nicht Fü/Lö

Adressezeit

$$+ \quad \text{C-Umlauf außer Adresse, außer Vorzeichenzeit, außer -Z-Befehl} \quad +$$

$$+ \quad \underline{G} \quad \underline{S2} \quad C \quad brc \quad (\underline{F} \quad + \quad \underline{T3} \quad + \quad \underline{R} \quad +$$

$\emptyset 2, \emptyset 4$ nicht Fü/Lö kein -Befehl

nicht Adressezeit

+ $Q_1 + Q_2 + Q_3 + Q_4$, + 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* +

 WZ3,5...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 +

 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' = G H S2 C Faf (F + Q3) +

 Ø2,Ø4 E/A-Verriegelung | Ø2 R-Befehl

+ keine Sektorkoinkidenz 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

$$\begin{array}{ccc} \beta_3 & (\alpha_5 & + \alpha_7) \\ = \underline{T3} + H \underline{Q3} \underline{P1} & = \underline{S} \underline{I1} \underline{I2} \underline{L} & = \underline{S} \underline{I1} \underline{I2} \underline{L} \\ & \text{Addition} & \text{Subtraktion} \end{array}$$

T3-Zeit Ø1...Ø4, D-Befehl, ungerade WZ M,N-Befehl

L' = β_3 +
T3 (H + Q3 + P1)

+ Übertragslogik

$$\begin{array}{ccc} Q2 (\underline{Q1} + \underline{Q3}) (\alpha_6 & + \alpha_8) \\ D, M, N, A, S - Befehl & = \underline{S} \underline{I1} \underline{I2} \underline{L} & = \underline{S} \underline{I1} \underline{I2} \underline{L} \\ & \text{Subtraktion} & \text{Addition} \end{array}$$

Subtrahiere + bei D-Befehl bei Vorzeichengleichheit

S = H Q4 + H Q3 P5 P6 + H Q3 P5 P6 +
Ø4a 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

$\alpha_6 = \underline{S} \quad \underline{I1} \quad \underline{\underline{I2}} \quad L$

S-Befehl Addierer-Eingänge

$\alpha_7 = \underline{S} \quad \underline{\underline{I1}} \quad I2 \quad \underline{L}$

S-Befehl

$\alpha_8 = \underline{S} \quad \underline{\underline{I1}} \quad \underline{\underline{I2}} \quad L$

A-Befehl

$\beta_3 = \underline{T3} + H \quad Q3 \quad \underline{P1}$

Schalte die P-Flip Flops als Schiftregister

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

Faf Ø1, Ø3 Spurzeit

s. Phasensteuerung A2-110

$$P1' = \begin{array}{l} \text{übernehme C oder R} \\ \underline{G} \quad i \quad r1 \\ | \quad | \quad | \\ \emptyset 1, \emptyset 3 \quad \emptyset 1=C, \emptyset 3=R \end{array} + \begin{array}{l} \text{übernehme A beim I,P-Befehl} \\ G \quad i \quad A \\ | \quad | \quad | \\ \emptyset 4 \end{array} +$$

$$\begin{array}{lll} + & \text{ungerade WZ, M,N,D-Befehl} & + \text{Setzer von Eingabegerät} \\ + & H \quad \underline{P1} \quad T3 \\ | & | & | \\ \emptyset 4a & & \end{array}$$

$$P1' = \begin{array}{l} \text{übernehme C oder R} \\ \underline{G} \quad i \quad \underline{r1} \\ | \quad | \quad | \\ \emptyset 1, \emptyset 3 \quad \emptyset 1=R, \emptyset 3=R \end{array} + \begin{array}{l} \text{übernehme A beim I,P-Befehl} \\ G \quad i \quad \underline{A} \\ | \quad | \quad | \\ \emptyset 4 \end{array} +$$

$$\begin{array}{lll} + & \text{gerade WZ, M,N,D-Befehl} & + \text{Rücksetzer von Eingabegerät} \\ + & H \quad P1 \quad T3 \\ | & | & | \\ \emptyset 4a & & \end{array}$$

$$\begin{array}{lll} + & 1. \text{ WZ, M,N,D-Befehl} \\ + & G \quad P1 \quad T3 \quad P1b \\ | & | & | \\ \emptyset 4 & & Faf \end{array}$$

$$P2' = \begin{array}{l} \text{übernehme P1} \\ i \quad \underline{P1} \\ | \quad \text{Shiftsignal} \end{array} + \begin{array}{l} \text{Setzer von Eingabegerät} \\ P2c \end{array}$$

$$P2' = \begin{array}{l} \text{übernehme P1} \\ i \quad \underline{P1} \\ | \quad \text{Shiftsignal} \end{array} + \begin{array}{l} \text{Rücksetzer v. Eingabegerät} \\ P2d \end{array}$$

$$P3' = \begin{array}{l} \text{übernehme P2} \\ i \quad \underline{P2} \\ | \quad \text{Shiftsignal} \end{array} + \begin{array}{l} \text{Setzer v. Eingabegerät} \\ P3c \end{array}$$

$$P3' = \begin{array}{l} \text{übernehme P2} \\ i \quad \underline{P2} \\ | \quad \text{Shiftsignal} \end{array} + \begin{array}{l} \text{Rücksetzer v. Eingabegerät} \\ P3d \end{array}$$

übernehme P3 + Setzer v. Eingabegerät
P4' = i P3 + P4c
 Schifftsignal

übernehme P3 + Rücksetzer v. Eingabegerät
P4' = i P3 + P4d
 Schifftsignal

übernehme P4 + Setzer v. Eingabegerät
P5' = i P4 + P5c +
 Schifftsignal

Vorzeichen von V bei M,N,D-Befehl + Setze P5 bei 4 Bit Ausgabe
 + G H T3 V P5a + β_5
Ø4 Hauptsp. Faf siehe weiter unten
 (E/A-Verriegelung)

übernehme P4 + Rücksetzer v. Eingabegerät +
P5' = i P4 + P5d +
 Schifftsignal

Vorzeichen v. V bei M,N,D-Befehl
 + G H T3 V P5a
Ø4 Hauptsp. Faf (E/i-Verriegelung)

übernehme P5 + Setzer v. Eingabegerät +
P6' = i P5 + P6c +
 Schifftsignal

Vorzeichen von A in Ø4 bei M,N,D-Befehl +
 + F G H T3 A P6a +
Ø4 Akku Faf (E/A-Verriegelung)

Vorzeichen v. A (laufender Rest) D-Befehl + Multiplizierbit beim N,M-Befehl
 + H T3 P1 Q3 A + H T3 P1 Q3 A*
Ø4a D-Befehl Ø4a M,N-Befehl

übernehme P5 + Rücksetzer v. Eingabegerät +
 $P_6' = i \underline{P_5} + P_{6d} +$
 Schifftsignal

+ 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
 + B 5
 | s. weiter unten

übernehme R + Ø4 Z-Befehl, PS-Tasten Koinzidenz oder
 $Q_1' = \alpha_{11} R + G \underline{H} T3 Q_1 Q_2 Q_3 Q_4 Faf T_0$
 | s. weiter | Ø4 | Z-Befehl | E/A nicht gedrückt
 unter | | | | | | | | E/A-Verriegelung

- Z-Befehl, Überlaufbit
 $(R : C + P_1 \underline{Tb_1} + P_2 \underline{Tb_2} + P_3 \underline{Tb_3} + P_4 \underline{Tb_4})$
 Überlauf PS-Tasten Koinzidenz
 - Z-Befehl

übernehme R + Manuell, Eingabe + Rücksetzer nach Z-Befehl
 $Q_1' = \alpha_{11} \underline{R} + bq + G \underline{H} T3 Q_1$
 | s. weiter unten | Ø2, Ø4

übernehme Q1 + beende Stop +
 $Q_2' = \alpha_{11} Q_1 + F \underline{G} \underline{H} bs +$
 | s. weiter unten | Ø1 | "Start"

+ Setzer für Q2 für alle Befehle außer Z0000, Z0100
 + F G T3 Q1 Q2 (Q1+Q3+Q4+P1+P2+P3+P4+P5+P6)
 Ø4 nicht "Step" alle Befehle außer Z0000, Z0100

übernehme Q1 + Stop, nach jeder Befehlausführung in Step +
Q2' = $\alpha_{11} \underline{Q_1} + \underline{H'} \underline{01} + G T_3 Q_1 \underline{01} +$
 ↓ s. weiter unten ↓ Step ↓ Ende D,M,N-Befehl ↓ Step ↓ P,E,U,T,H,C,A,S-Befehl
Ø4

+ Stop, nach Eingabe in Step oder Manuell + E/A-Rücksetzer
 + G T_3 Q1 Q2 Q3 Q4 01 + To
Ø4 Eingabe

übernehme Q2 + Druckbefehl
Q3' = $\alpha_{11} \underline{Q_2} + G T_3 \underline{Q_1} \underline{Q_2} \underline{Q_3} \underline{Q_4}$
 ↓ s. weiter unten ↓ Ø4 Druckbefehl

übernehme Q2 + Manuell Eingabe
Q3' = $\alpha_{11} \underline{Q_2} + bq$
 ↓ s. weiter unten

übernehme Q3
Q4' = $\alpha_{11} \underline{Q_3}$
 ↓ s. weiter unten

übernehme Q3 + Manuell Eingabe +
Q4' = $\alpha_{11} \underline{Q_3} + bq +$
 ↓ s. weiter unten

+ Testbefehl, A negativ + - Testbefehl
 + T_3 Q1 Q2 Q3 A + T_3 Q1 Q2 Q3 R Tc
 ↓ T-Befehl ↓ T-Befehl ↓ PST-Taste
 ↓ A negativ ↓ - T-Befehl

$\alpha_9 = Q_3 + Q_4$

Schalte Q-Flip Flops als Schiftregister
 $\alpha_{11} = F \underline{G} \underline{H} \underline{S1} \underline{S2} \underline{S3}, bq Faf$
Ø3 Befehlszeit ↓ E/A-Verriegelung
 nicht "Manuell"

Ø1 Druckbefehl in 4 Bit Mode
 $\beta_5 = F \underline{G} K Q_3 Faf$
Ø1 Ausgabe ↓ E/A-Verriegelung
 4 Bit Mode

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
Ø3 0000 I-Befehl | E/A-Verriegelung
nicht "ausführen"

Rücksetzer für Tally beim Stop Code + E/A-Taste

Ft' = F G T3 Sc Xp + To
Ø1 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
Ø1 - Multivibrator (Zykluszeit)

Di' = Faf

E/A-Verriegelung

Vorbereitung Flexowriter Eingabe

Sr* = F Q1 Xs bQ Ff
Ø4 I-Befehl | nicht "Manuell"
nicht in Eingabe-Zustand

Ausgabebefehl zum Flexowriter

X' = F G T3 Tx JL33 Q3 Ff
Ø1 Kupplungstreiber | Ausgabe-Indikator
nicht gesetzt Translator Cam

$\underline{X}' = F + G + \underline{T3} + Tx + JL33 + \underline{Q3} + \underline{Ff}$
 Ø3, Ø4 Ø2 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

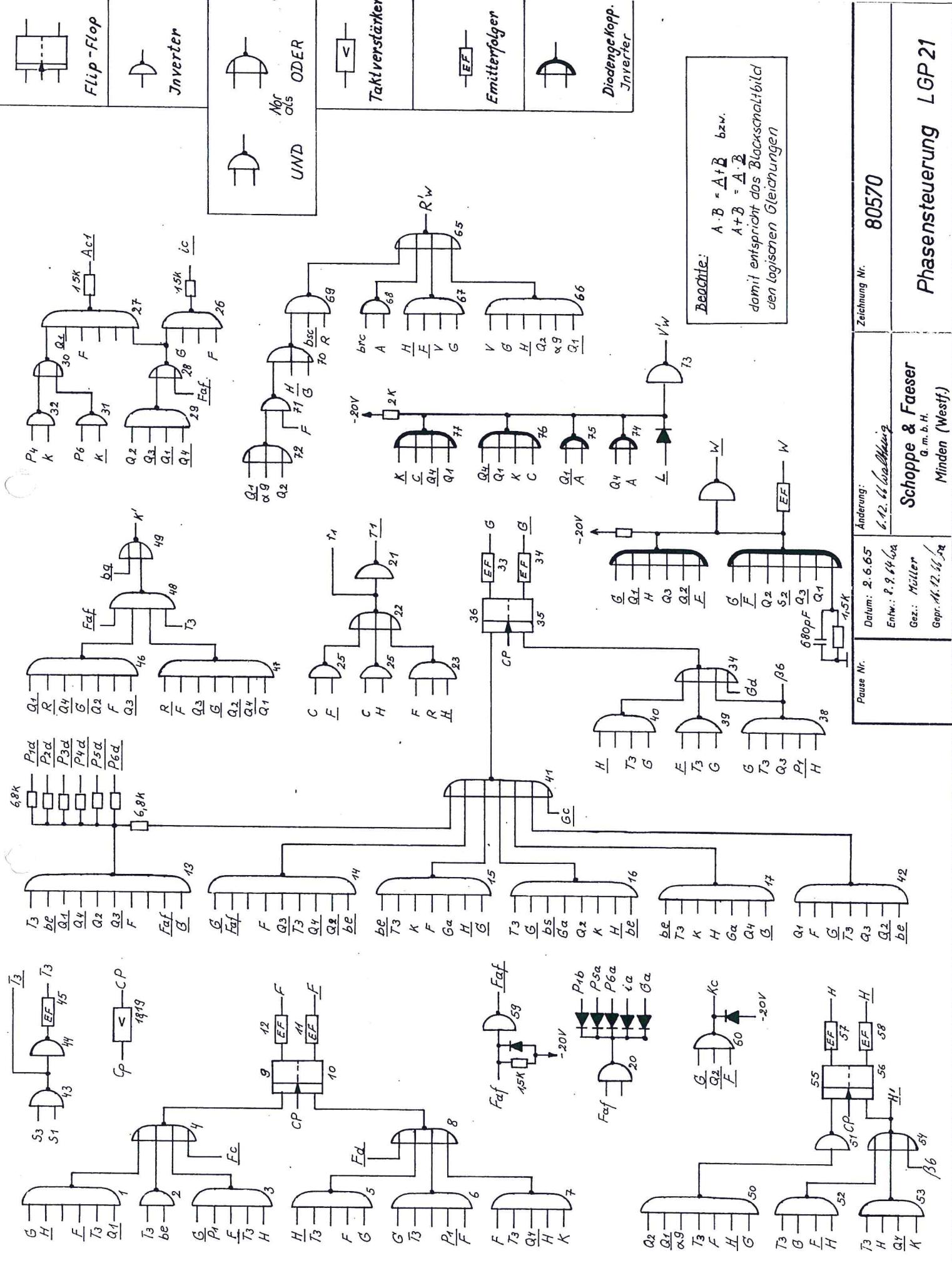
Drucke

Tx = X = JL7

Maschinenezeichen

Sk = P5 P6 K

4 Bit Modus



Jan. 1967
Herrn. Dr. Ing. H. Müller
für die Firma Faeser
GmbH & Co. KG
Wuppertal 1

80570

Zeichnung Nr.

Phasensteuerung LGP 21

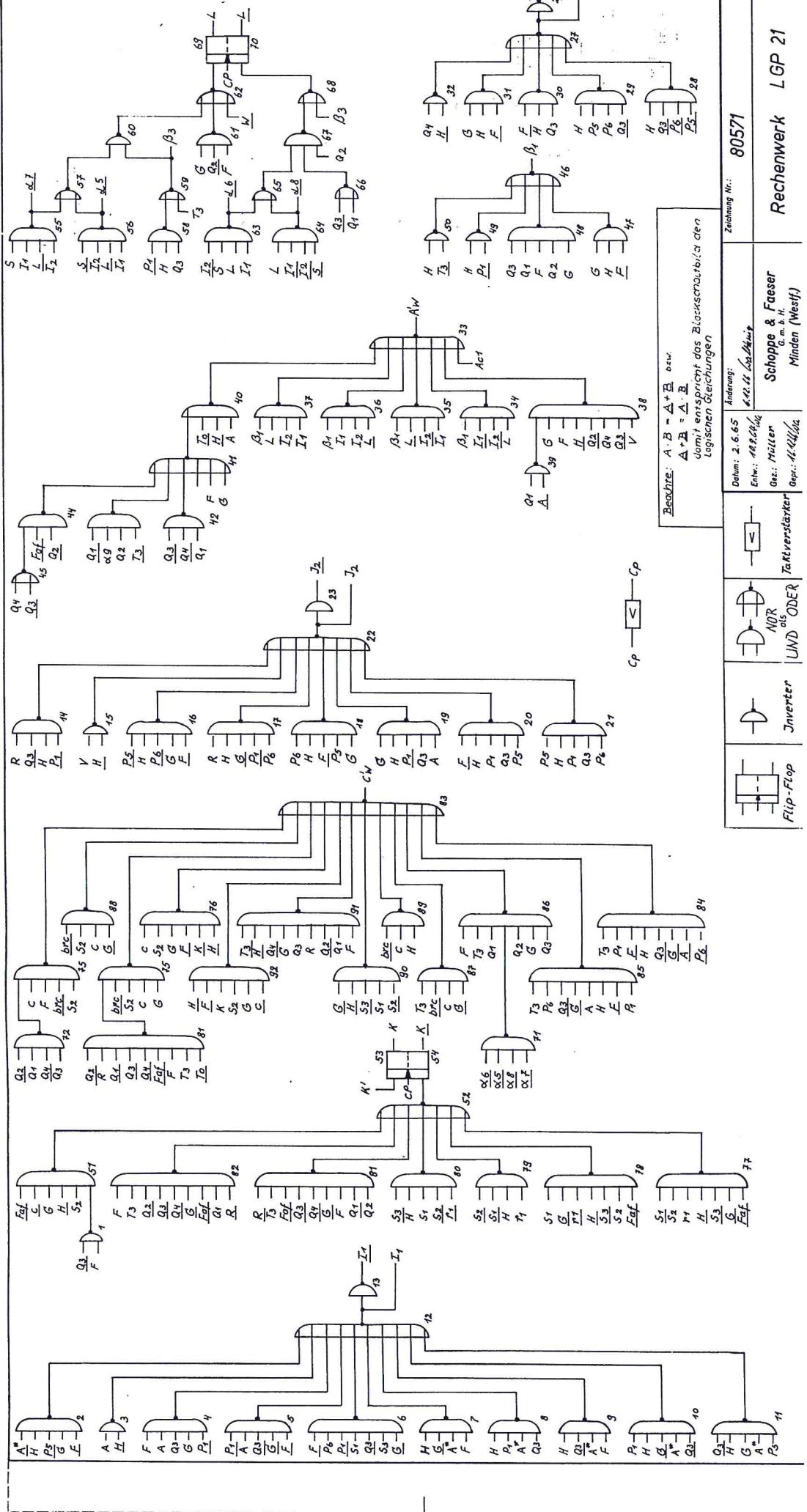
Schoppe & Faeser
a.m.b.H.
Mindnen (Westf.)

Pause Nr. Datum: 2. 6. 65 Änderung: 6. 12. 66 Wiedergabe

Entw.: 2. 9. 64 v.a.

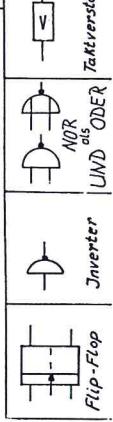
Gez.: Müller

Gegr. M. 12. 66 /



Beschriftungen: $A \cdot B = A + B$ bzw.
 $\bar{A} + \bar{B} = \bar{A} \cdot \bar{B}$
 damit entspricht das Blockschaltbild den
 logischen Gleichungen

Änderung: 2. 6. 6. 5
 Datum: 14. 2. 1974
 Entw.: Müller
 Ge.: Müller
 Rechenwerk LGP 21
 Zeichnung Nr.: 80571
 Schoppe & Faeser
 G.m.b.H.
 Minden (Westf.)



LGP 21

Schoppe & Faeser
G.m.b.H.
Mindien (Westf.)

Kommandowerk LGP 21

80572

Zeichnung Nr.

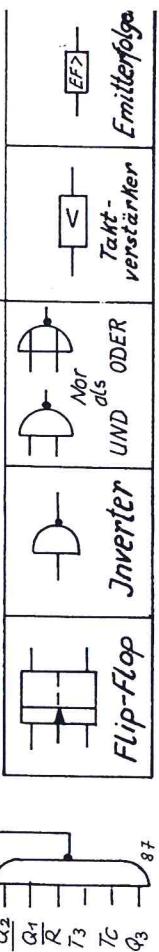
Pause Nr.

Datum: 2.6.65 Änderung:
Entw.: 2.9.64 v.a. 1.12.66 Weitläng

Eaz.: Müller
Gepr.: H. Müller v.a.

Beachte: $A \cdot B = A + B$
 $A + B = A \cdot B$

damit entspricht das
Blockschaltbild den
logischen Gleichungen

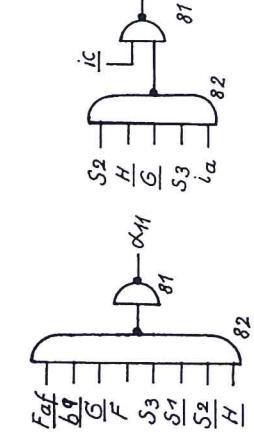


Flip-Flop
Inverter
UND
Oder

NOT
als
Oder

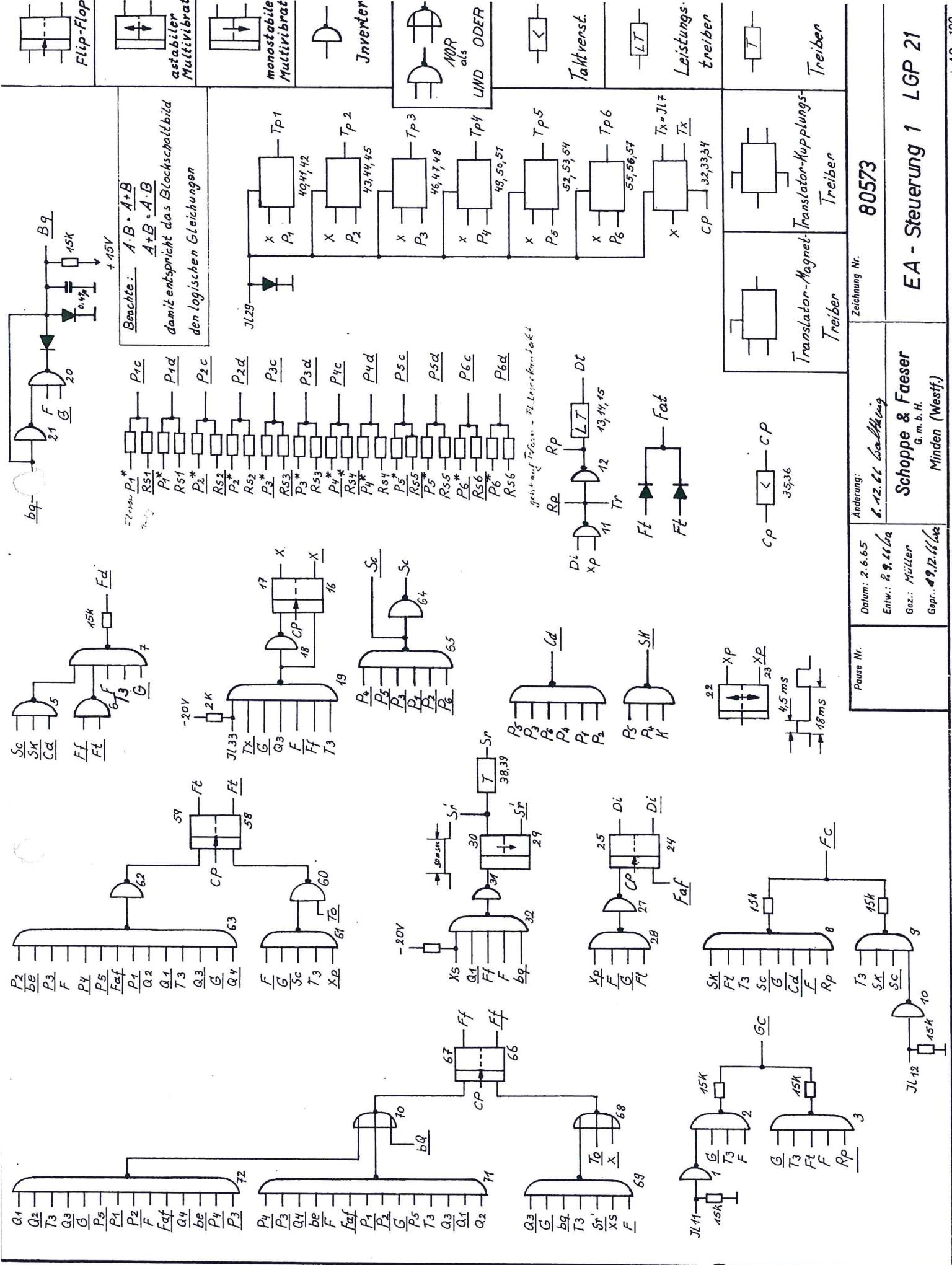
Takt-
verstärker

Emitterfolge



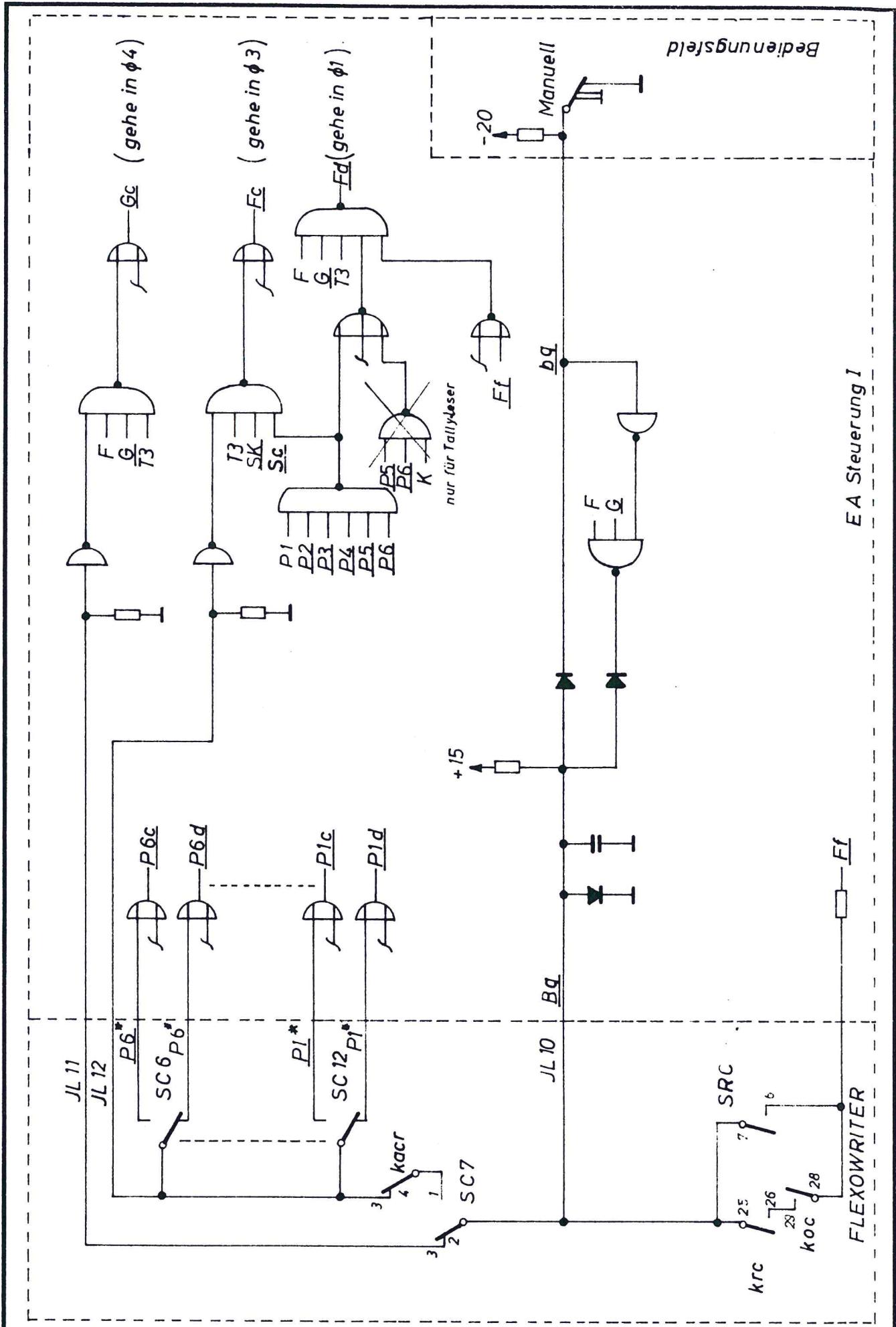
24. Okt. 1967

20.10.1967

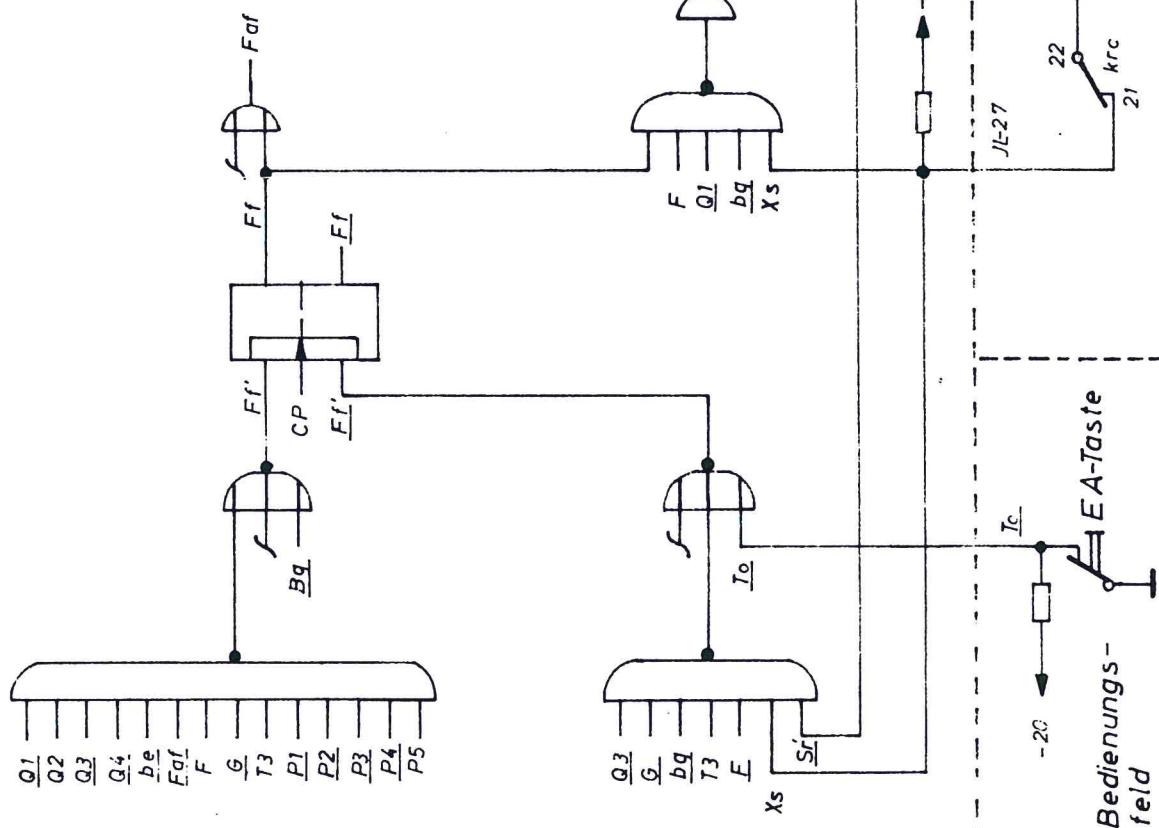


2. Jan. 1967
Schoppe & Faeser
G.m.b.H.
Westf. Minden (Westf.)

Pause Nr.	Datum: 2.6.5	Änderung: 6.12.66 Schaltung	Zeichnung Nr. 80573
	Entw.: 6.9.66/2		EA - Steuerung 1 LGP 21
	Gez.: Müller		
	Gef.: 6.9.12.66/2		



EA Steuerung I



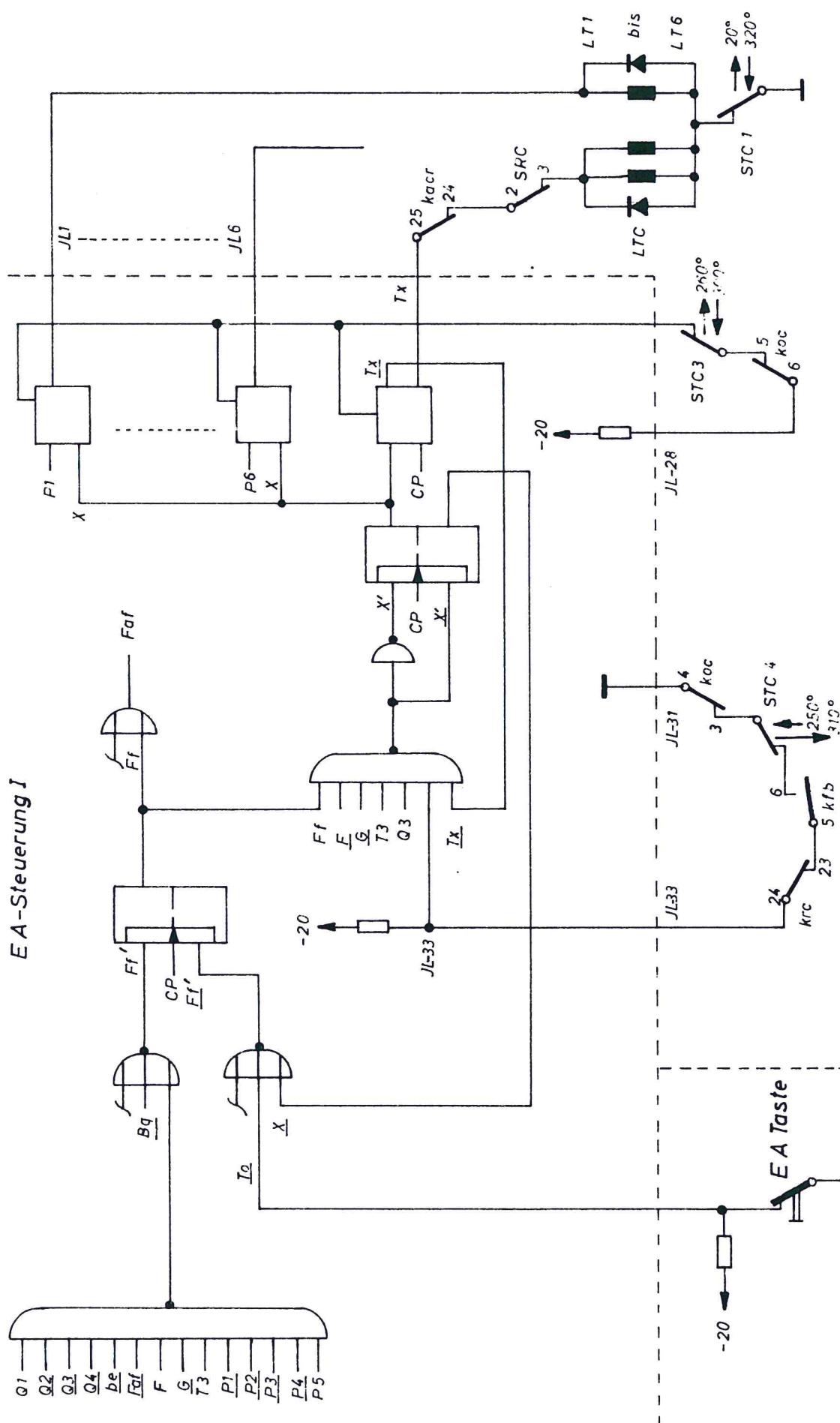
8.12.1966
Schoppe & Faeser
G. m. b. H.
Minden (Westf.)

Eingabe über Flexowriter
Blockschatzbild

Blattzahl: 2
Blatt: 2

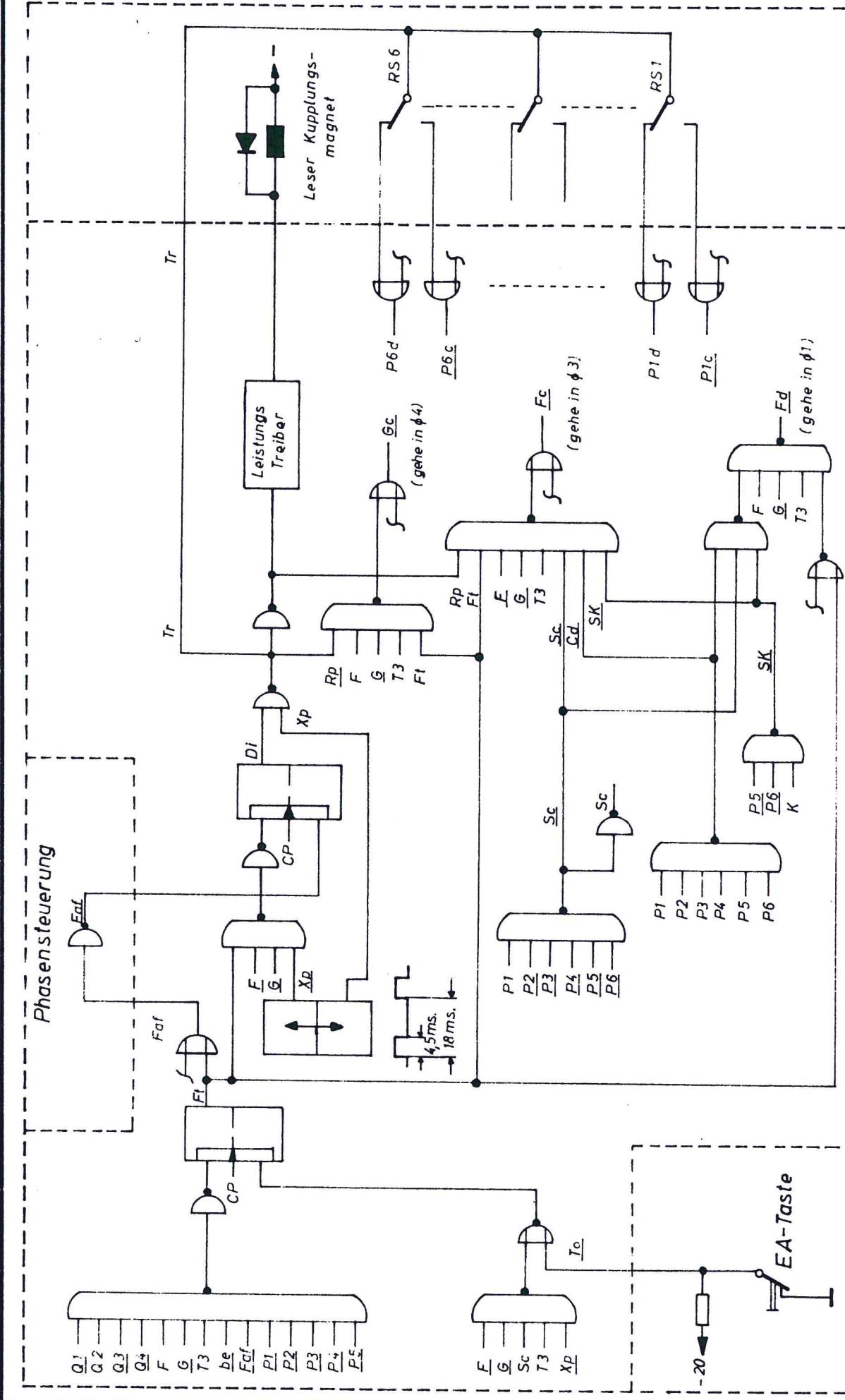
A2 - 133

EA-Steuerung I



Bedienungsfeld

Flexowriter



8.12.1966

Wa

Schoppe & Faeser
G. m. b. H.
Minden (Westf.)

Eingabe über Tally Leser
Blockschatzbild

A2 - 137

Bedienungsfeld

EA-Steuerung I

Tally Leser