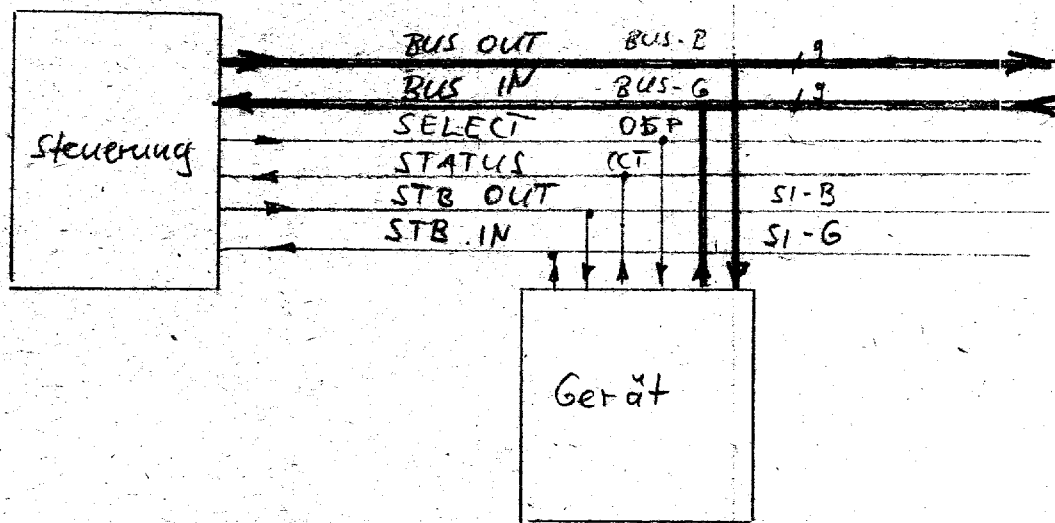
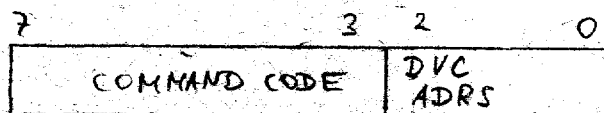


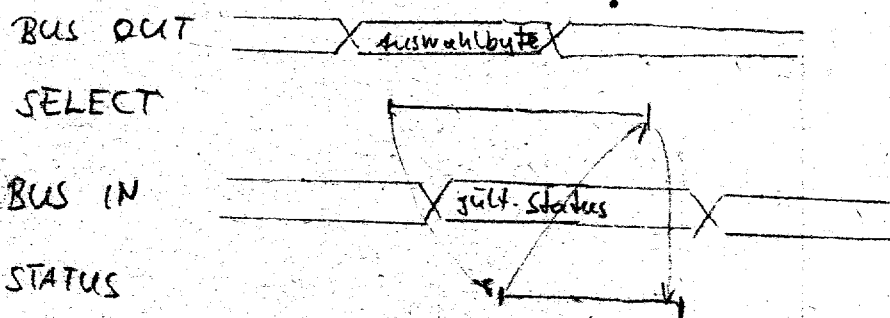
Interface-Leitungen



Auswahl-Information auf BUS OUT



Geräte-Auswahl



Statusabfrage

Grundsätze

1. Die Steuerung beginnt die Verbindungsaufnahme mit SELECT.

Die Verbindung bleibt bestehen, solange SELECT aktiv ist.

2. Das Gerät antwortet auf die Auswahl mit STATUS.

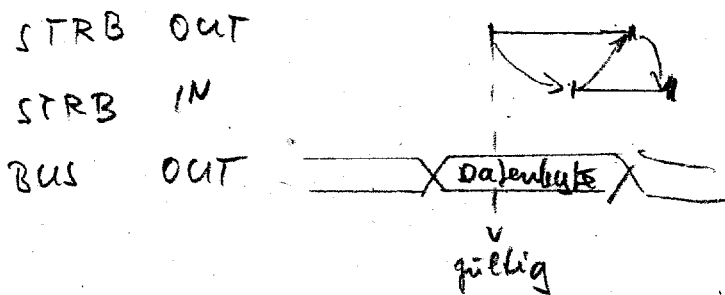
Die Verbindung bleibt bestehen, solange STATUS aktiv ist.

Wenn die Steuerung einen Abbruch herbeiführen will, muß sie SELECT ausschalten.

Wenn das Gerät einen Abbruch herbeiführen will, muß es STATUS ausschalten.

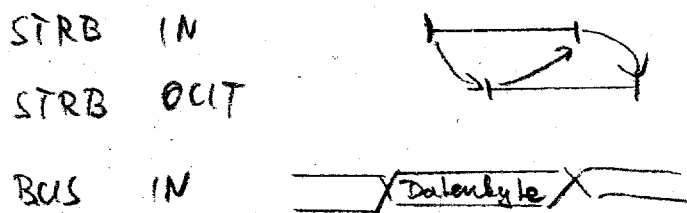
Prinzipien der Handshaking

a.) bei Ausgabe



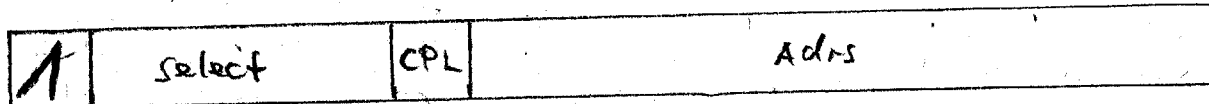
- Gerät erwartet STRB OUT
- schaltet daraufhin STRB IN ein
- STRB IN fällt auf Abfall von STRB OUT

b.) bei Eingabe

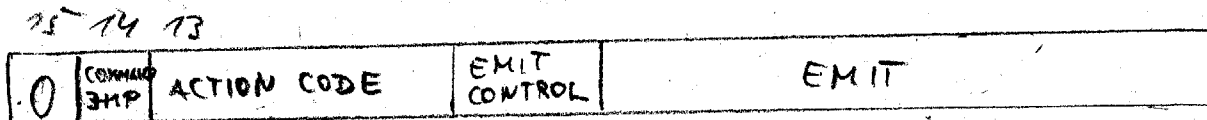


- Gerät schaltet STRB IN ein
- erwartet STRB OUT
- STRB IN fällt auf Ausstieg von STRB OUT

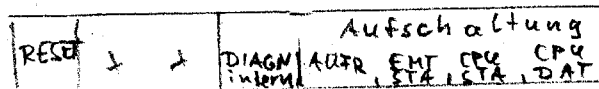
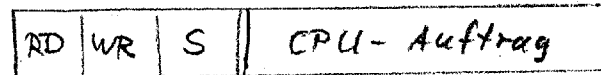
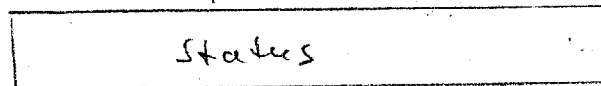
EMP:



ACTION:



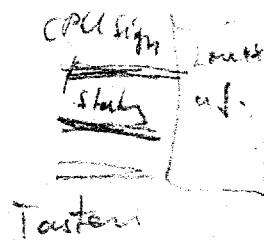
Codierung 00 00H : NOP



Abfragepositionen

- ad. • Adresse erkannt
- atf. • SELECT
- STRB OUT
- Paritätsfehler
- Read } Kommando - stats
- Write }
- Bereitsignal (Sammelsignal aller 4 Geräte)
- ? • Aufzug
- ? • Auftrag akzeptiert
- presence • o.k.
- WAIT
- 4 Signalleit. v. CPU
- 4 Statussign.
- 5-6 Fehlersign.

12 die 16 / 2 mit



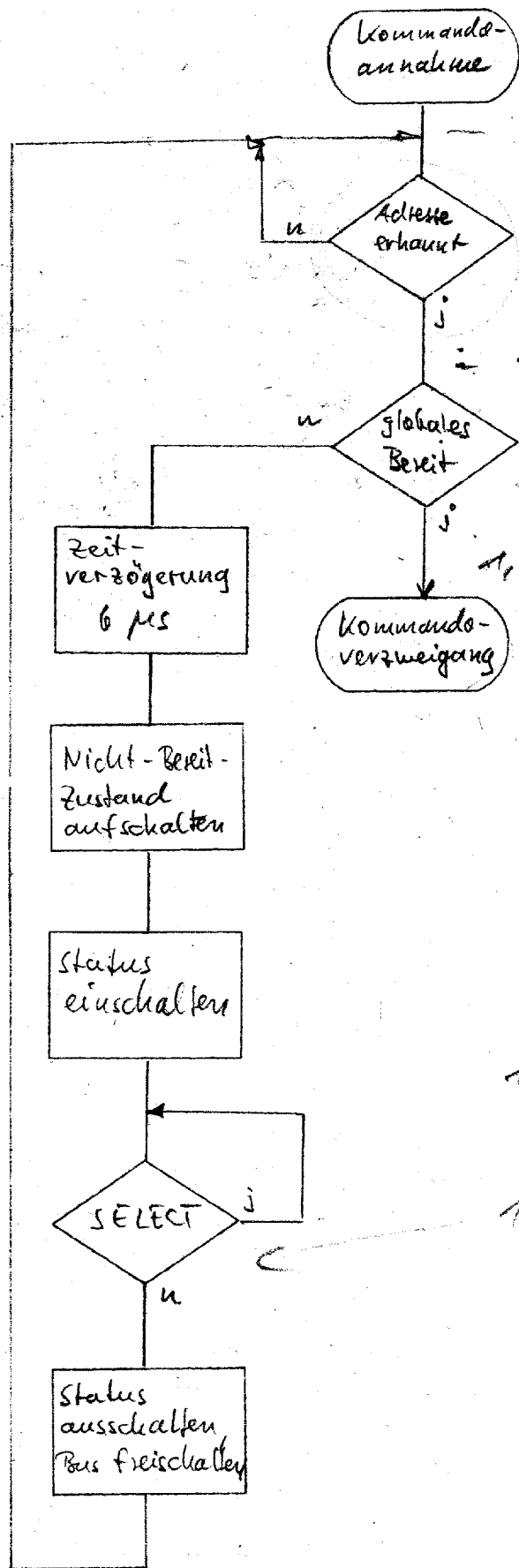
Stellpositionen

- Status einschalten
- Status ausschalten
- RESET senden
- NMI senden
- WAIT ausschalten
- Gerätereister laden
- EMIT Status
- EMIT Zustand + Kommando
- Melden 'erledigt' } entk. über CPU-Kommando
- Melden 'Auftrag' }
- Übernahme in CPU-Datenreg (RD)
- STRB IN signalisieren (einschalten / ausschalten)

Aufschaltmöglichkeiten

auf BUS IN

- Adresscode
- EMIT-Register
- CPU-Status
- CPU-Daten (WA)

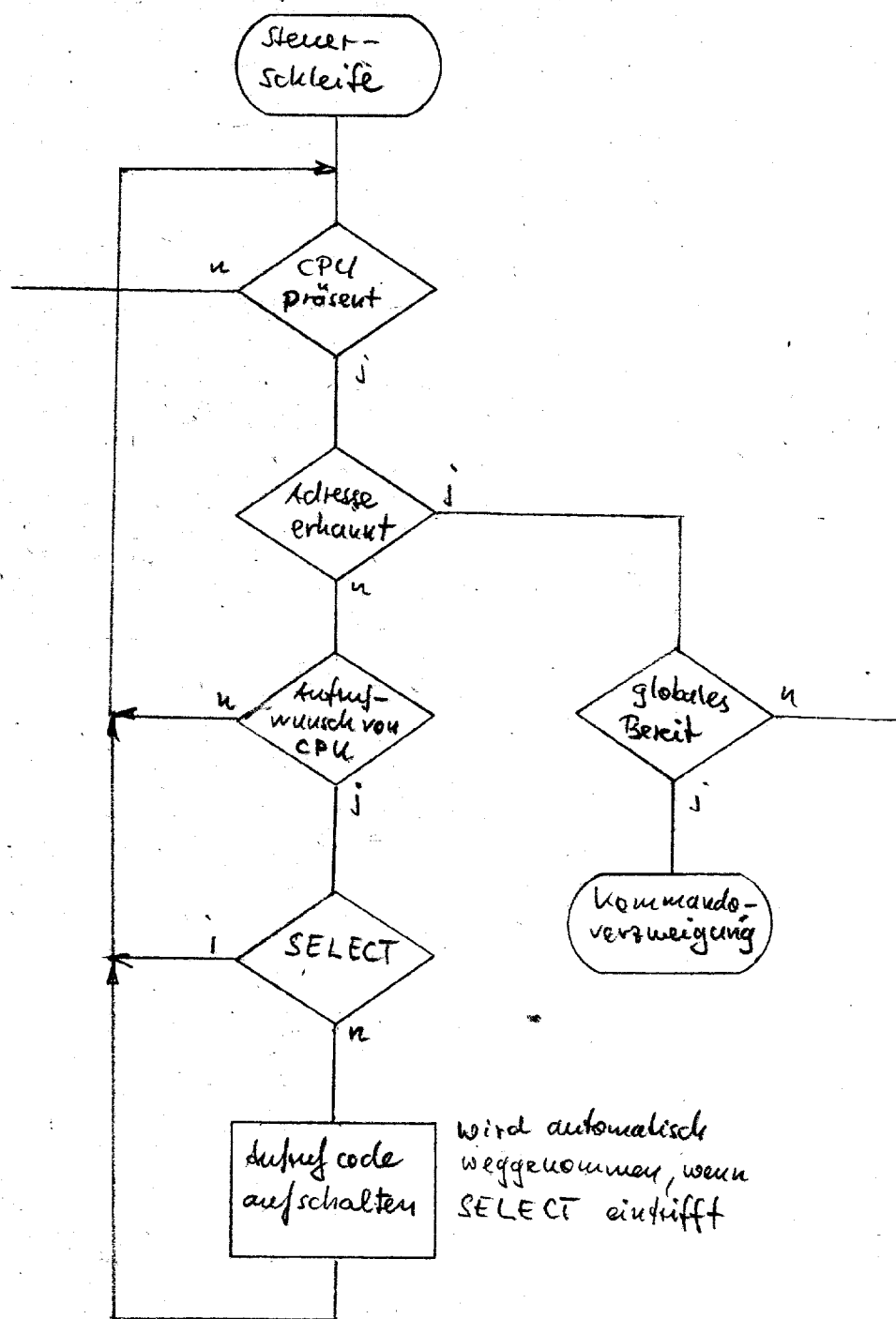


IM sdw. ← 10
 BIT 10
 3RZ 10
 BIT gl.h. 30 2
 12 µs

3RZ St. es.
 Vers.

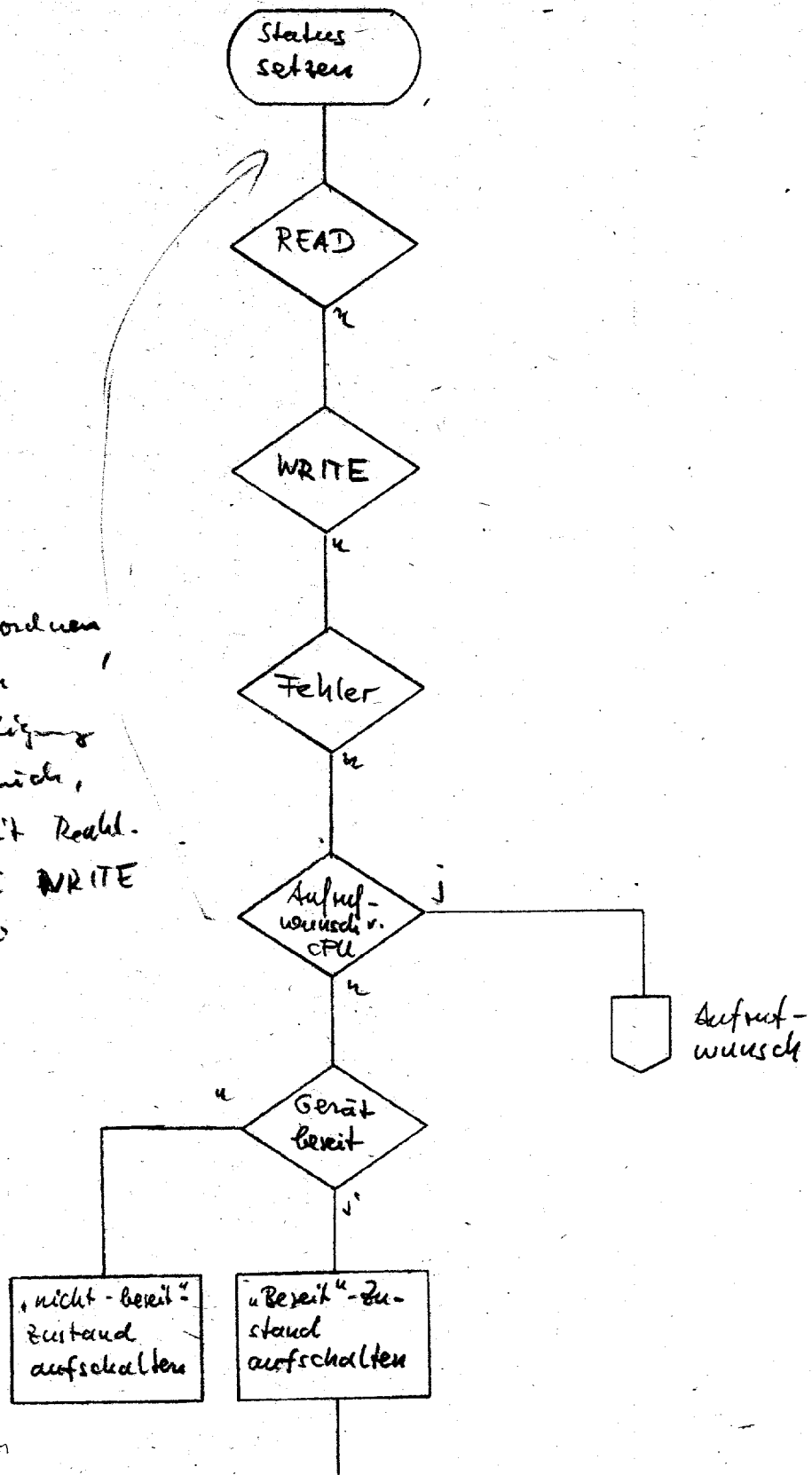
18 LDA,
 20 OUT MBRzust.
 10 LDA -
 40 OUT Status, einde
 20 IN
 20 BIT
 10 SELECT

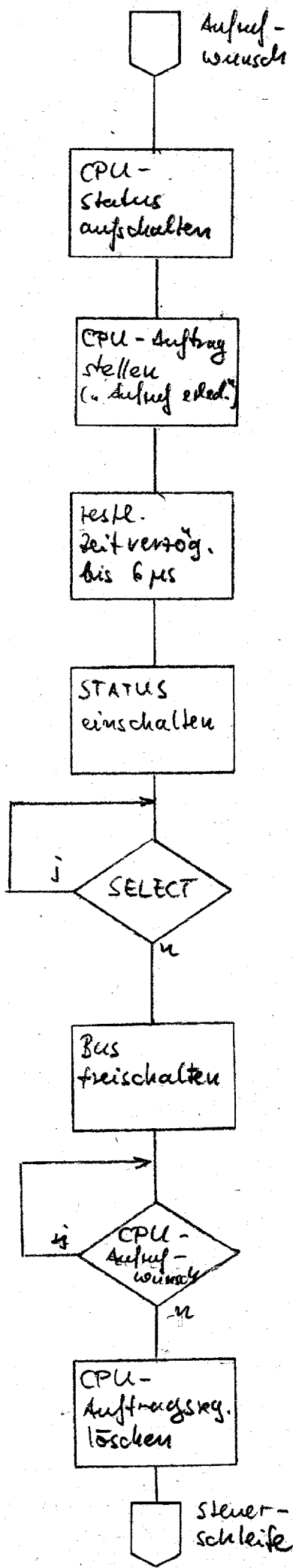
1002 40 µs

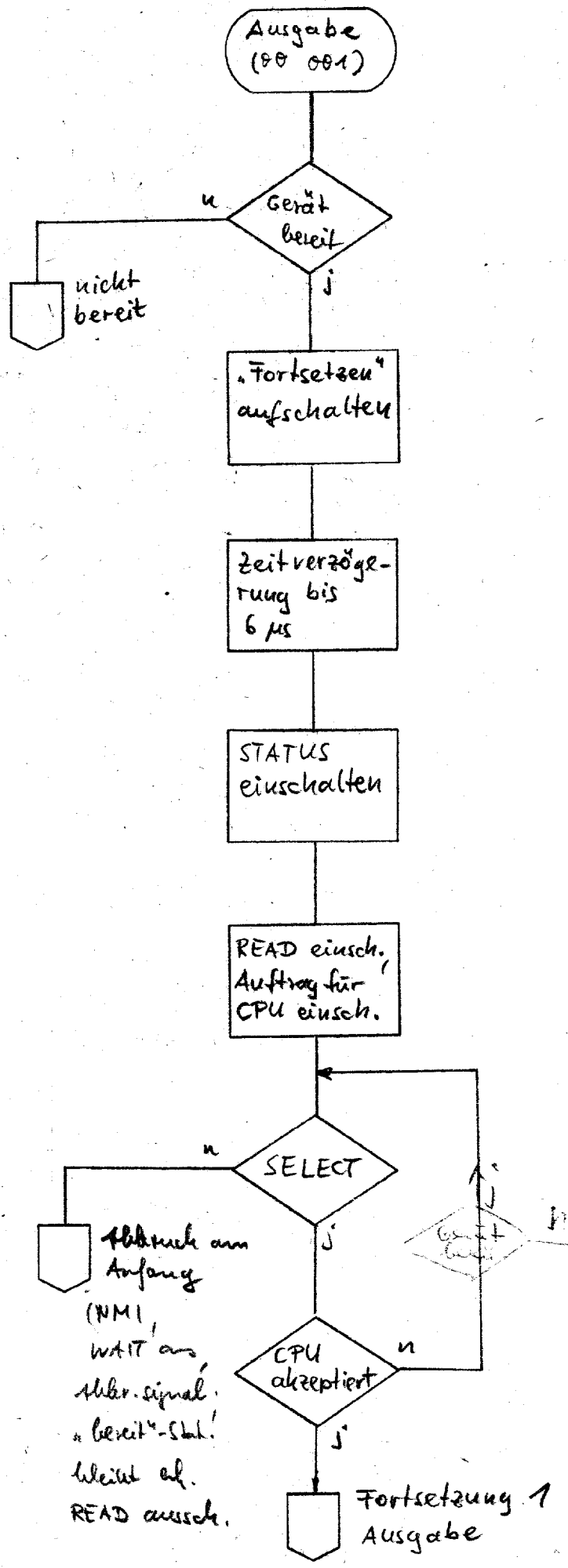


wird automatisch
weggenommen, wenn
SELECT eintrifft

vordem
nach
Erledigung
Zurück,
dann Read.
bei WRITE
o.g.

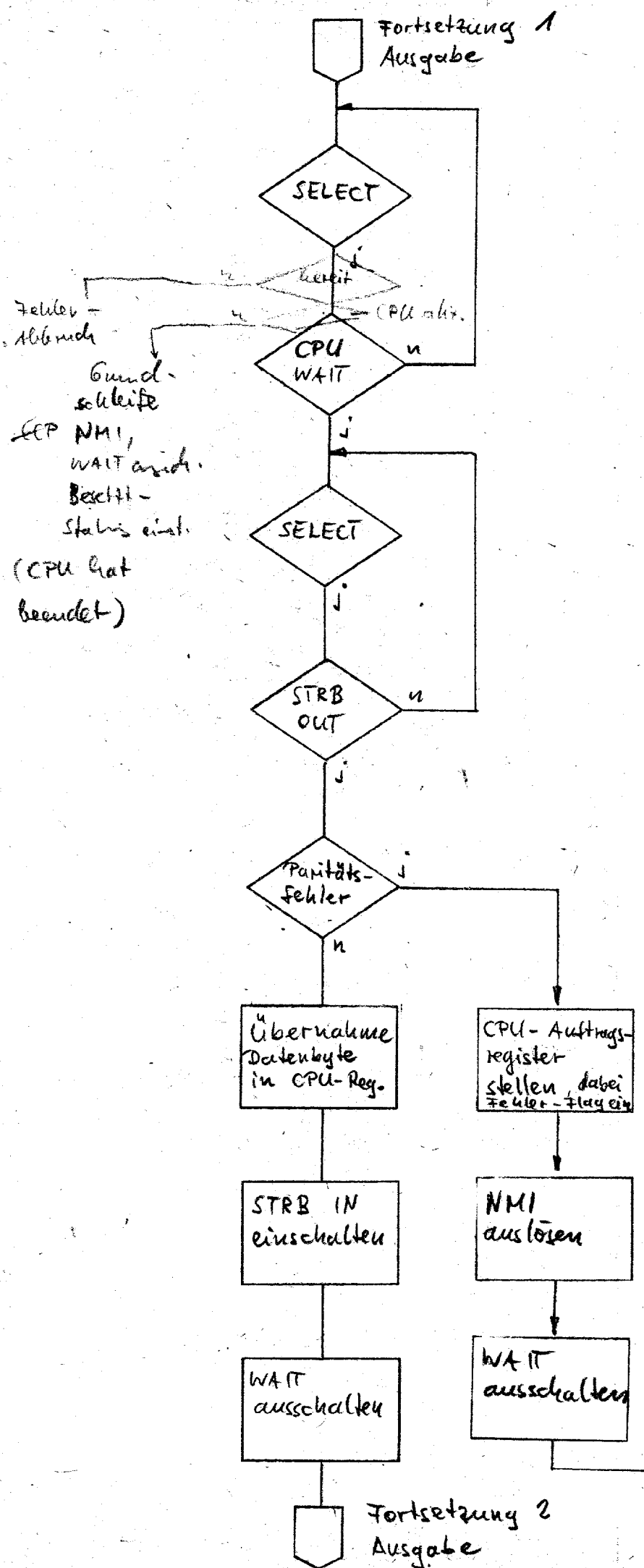






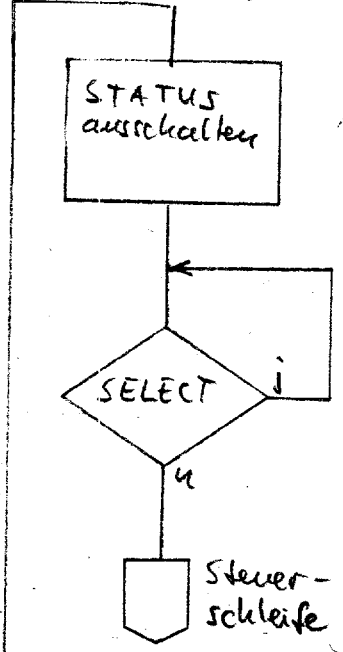
Abbruch am
Anfang
(NMI,
WAIT ans,
Abbr. signal,
"bereit"-Stk.
bleibt ab.
READ aussch.

Fehlerr. Flag
setzen, READ aus-
schalten, an-
melden
CPU empfangen
halten



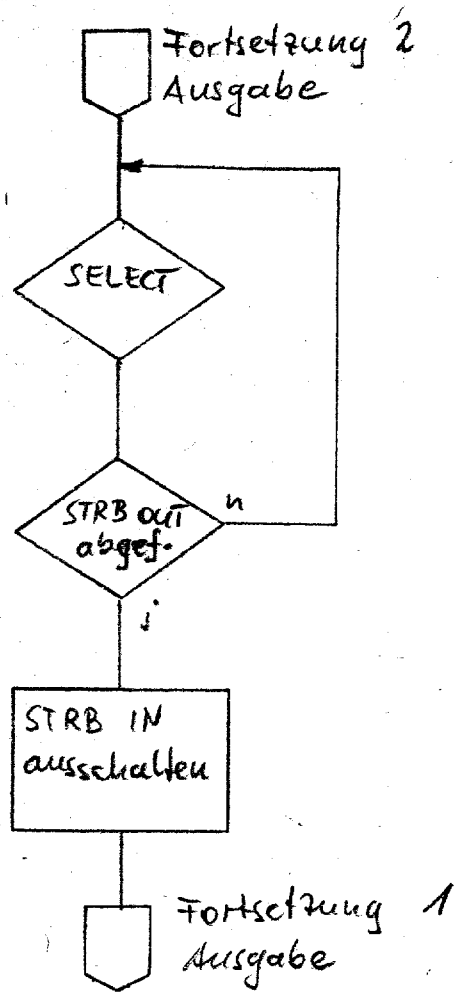
Fehler-Abbruch
 Grundschleife
 IOP NMI, WAIT anst.
 Bechtel-Status einst.
 (CPU hat beendet)

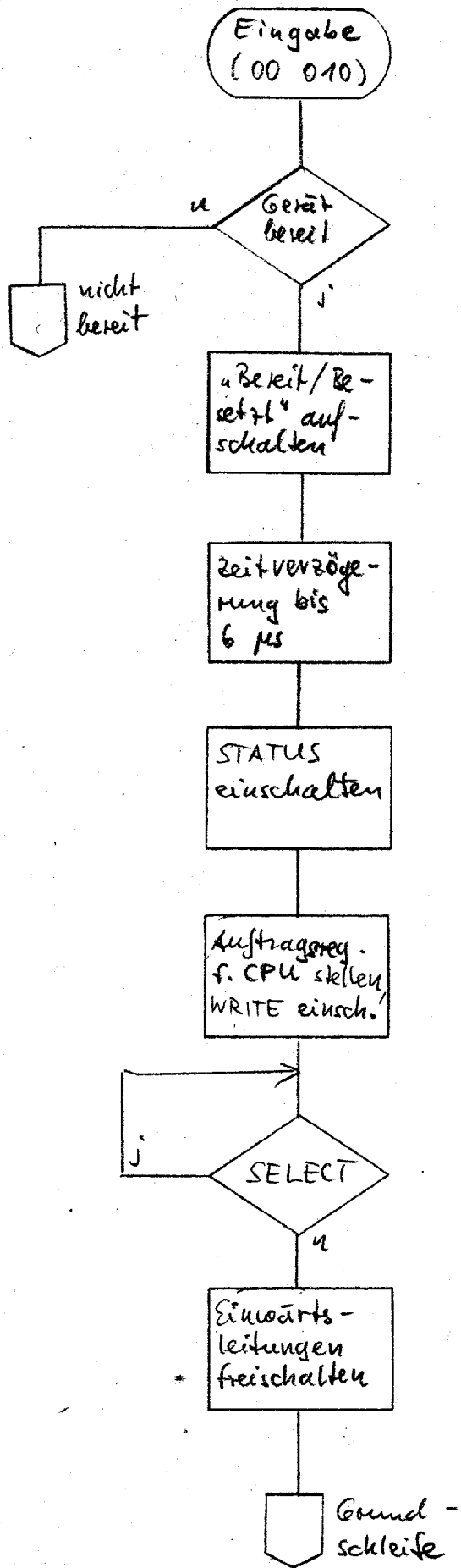
Abbruch bei Fehler:
 (Parity check)



Hinweis:
 CPU stellt Fehler-Zustand in ihrem Statusreg ein; bei Zustandsabfrage wird infolge der Fehler-Flag dieses Reg. auf den Bus geschaltet

Fortsetzung 2
 Ausgabe

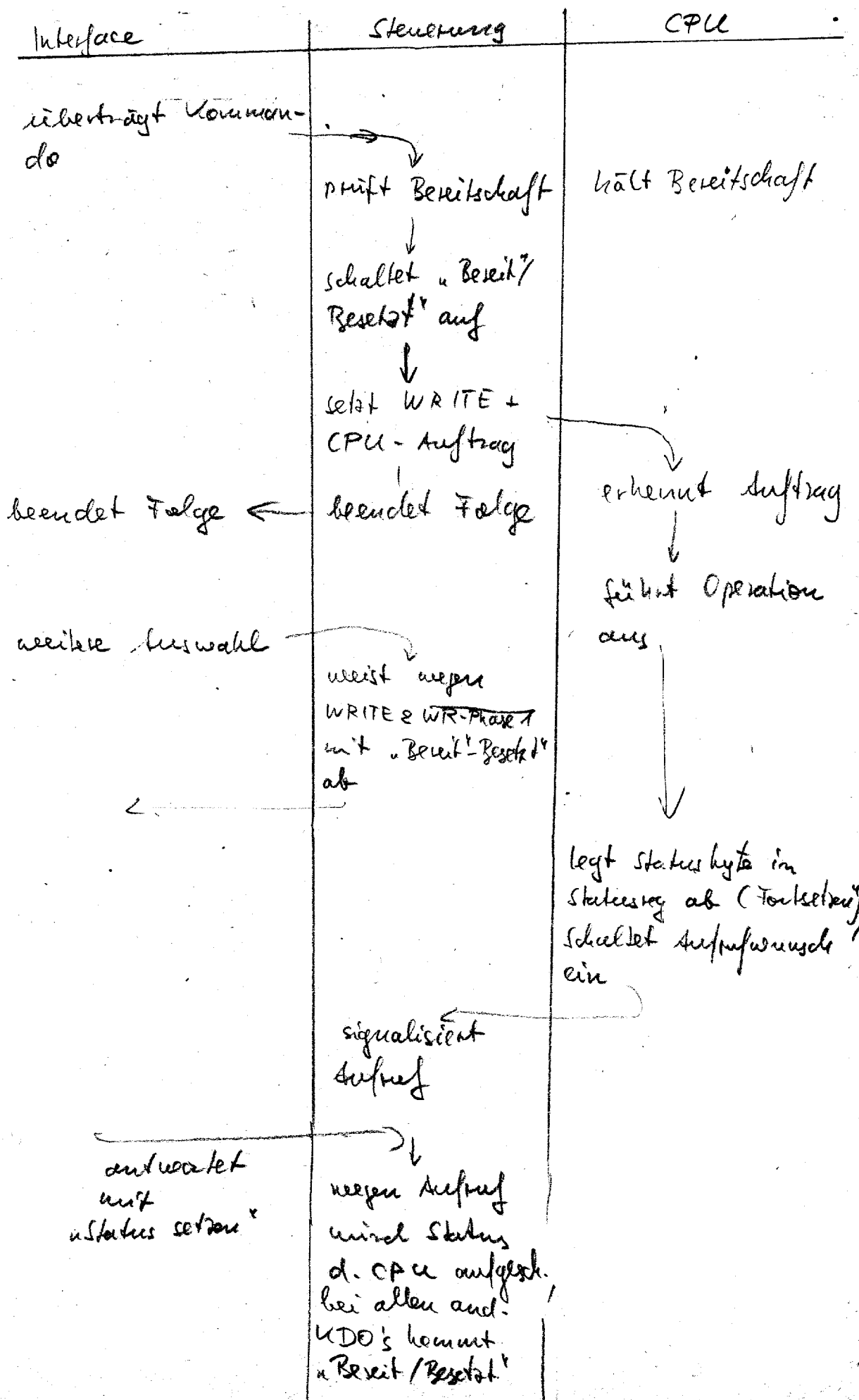




Hinweis:

CPU verlangt Signali-
sierung von „Fortsetzen“,
um die Datenüber-
tragung zu beginnen.

Kommunikation bei Eingabe



Interface

Steuerung

CPU

signalisiert
"Auftrag erledigt"
an CPU,
WR-PHASE 1 einsch.

nimmt
Aufnahmewunsch
zurück, setzt
"akzeptiert"

nimmt Status
weg (50 µs ??)

beginnt mit
Übertragung

erwartet
"wait"

legt Datenbyte
auf Bus

Strobe-Signal-
spiel

schaltet "wait"
aus

nimmt SELECT
weg

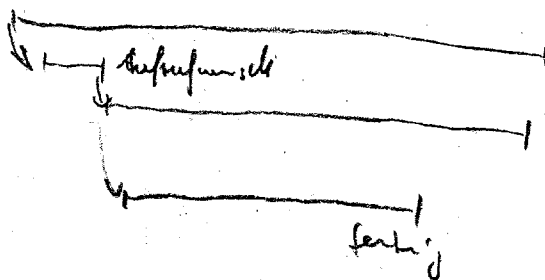
nimmt
"akzeptiert"
weg, wenn
keine Daten
mehr anzuholen
sind

Kommando

WR

WR-Phase 1

akzeptiert



bei Neuauswahl
"Status setzen"

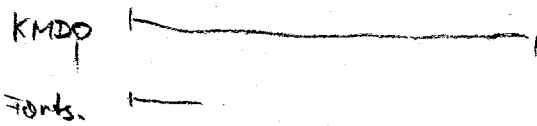
wenn WR- $\overline{WRPH1}$:
Bet./Resetzt

wenn WR- $\overline{WRPH1}$:
akzeptiert; Folgebite;
Datenbyte übertr.

wenn WR- $\overline{WRPH1}$:
als "Resetz" bis
Ende

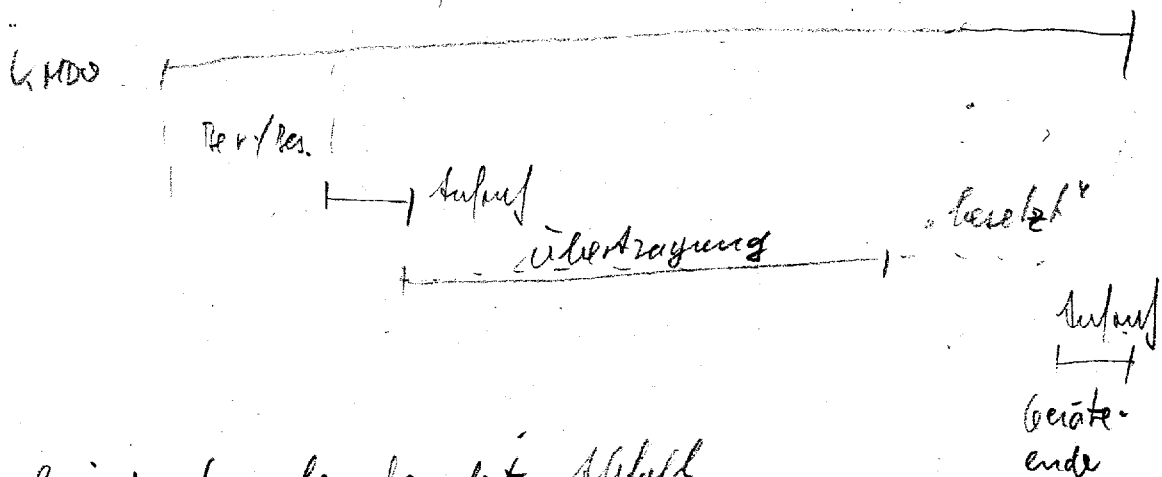
Zustände allgemein

1. Ruhe,
mögl. Zust: "bereit" o. "nicht bereit"
2. Kommando - Annahme
"Fortsetzen"



dann "Besetzt" bei jeder anderen Anfrage,
wenn CPU nicht bereit ist

3. Eingabekommando



Bei burst mode beendet Abfall
von SELECT die Datenübertragung

Synchronisation evtl. so; die CPU muß in WAIT gehen
dann zuvor muß sie "akzeptiert" entspr. stellen.
demgemäß wird übertragen o. ein NMI
ausgelöst.

bleibt
hessen
wie alt

warten
da während
RD / WR
Auftrag kommen
können

CPU-Status
aufschalten

Warten-
bereit-
Zustand

(Mut-
Betrieb)

Fortsetzen
Schreib-
schleife

STATUS
SETZEN

READ

WRITE

Auftakt
signal

Gerät
bereit

Bereit-
Zustand
aufschalten

"Fortsetzen"
aufschalten

Zeitverzög.
6 μs

STATUS
einschalten

"wait"
von CPU

Datenbyte
auf Bus
schalten

SELECT

Fortk. Schreiben

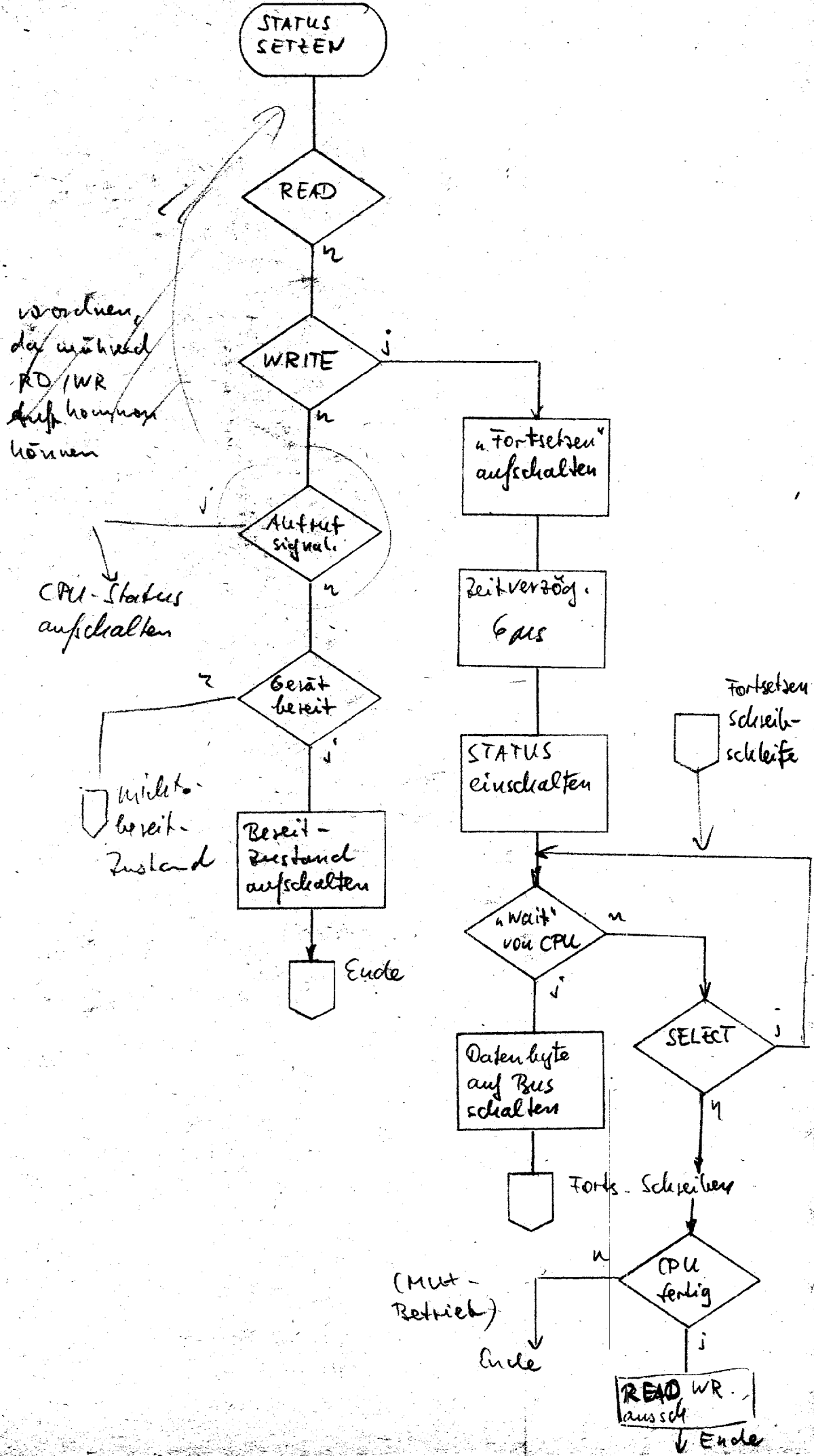
CPU
fertig

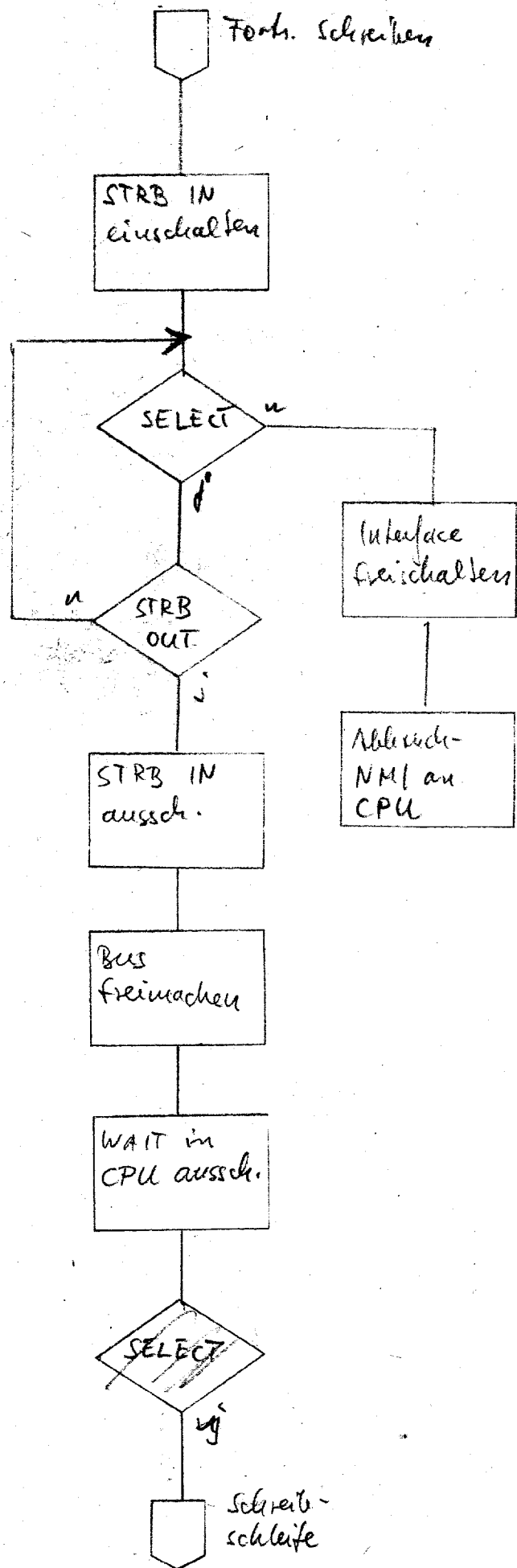
READ WR.
austausch

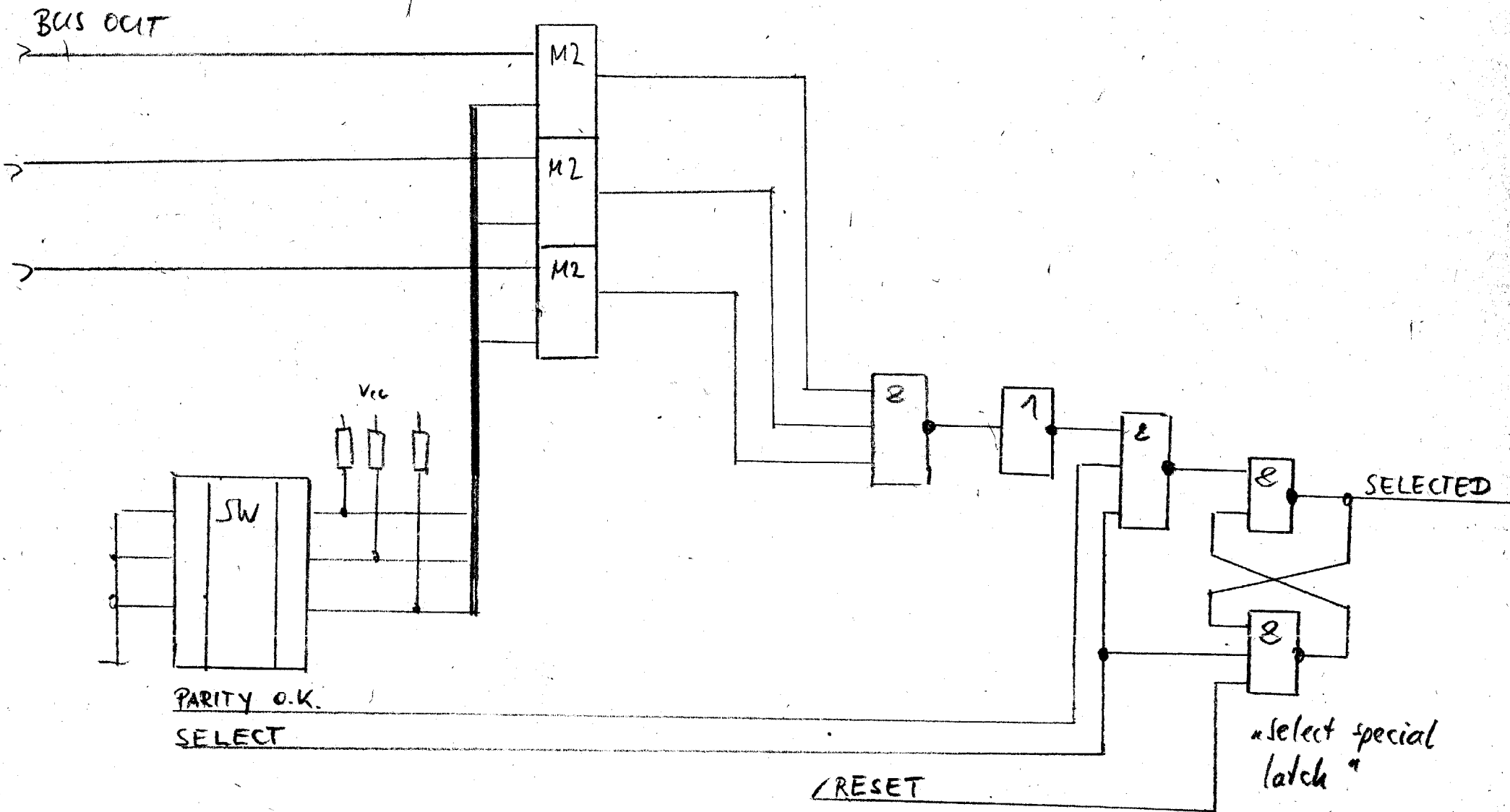
Ende

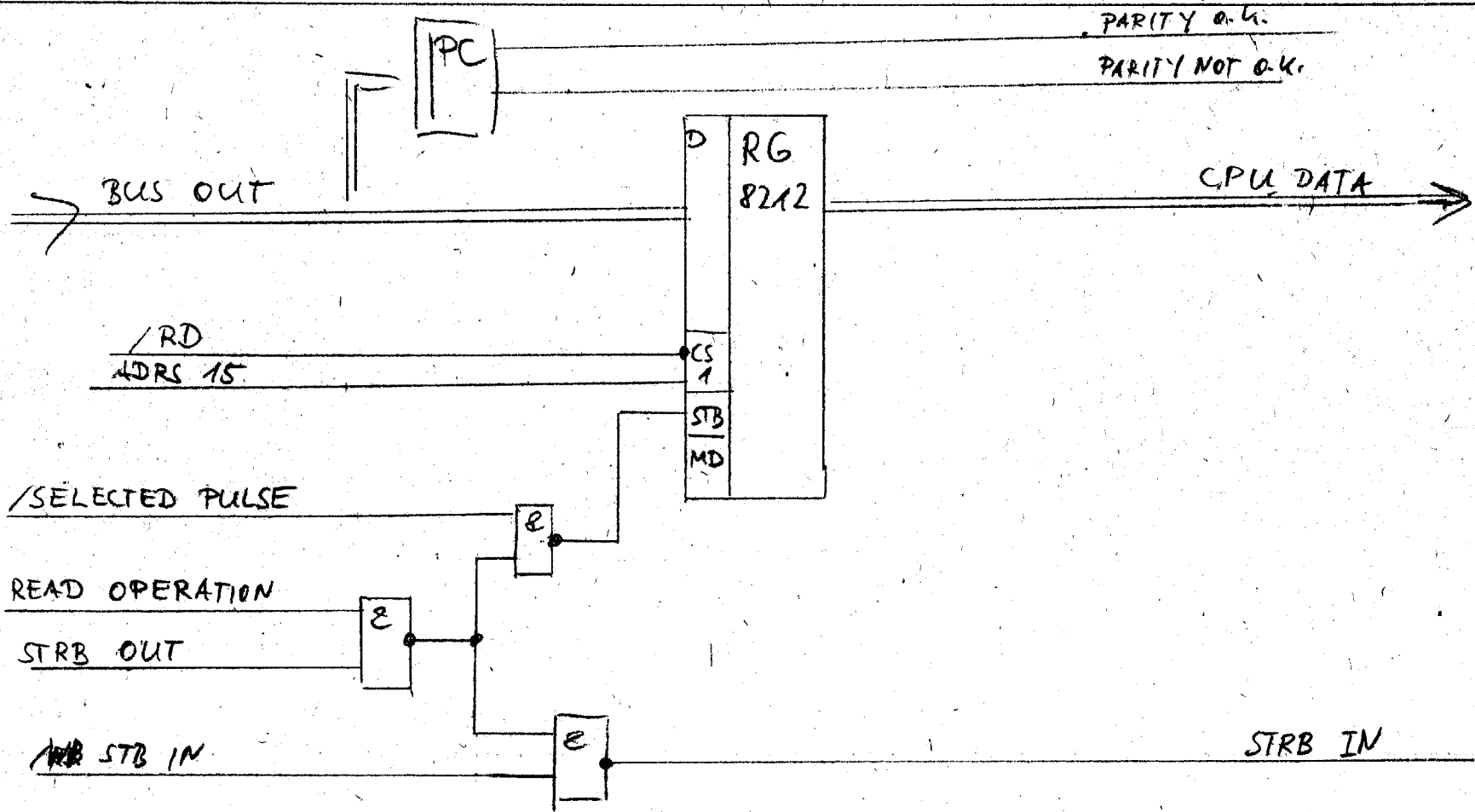
Ende

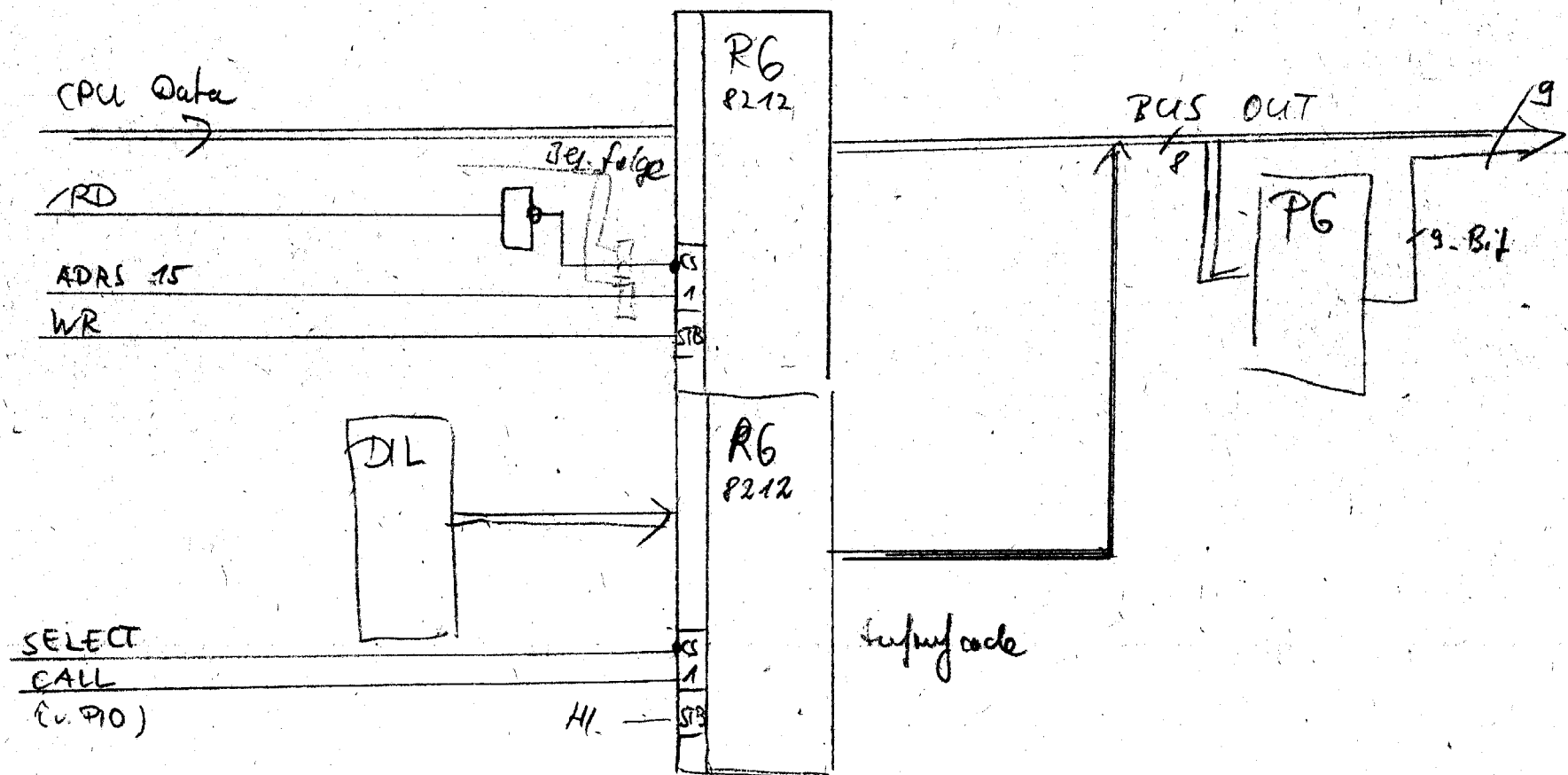
Ende

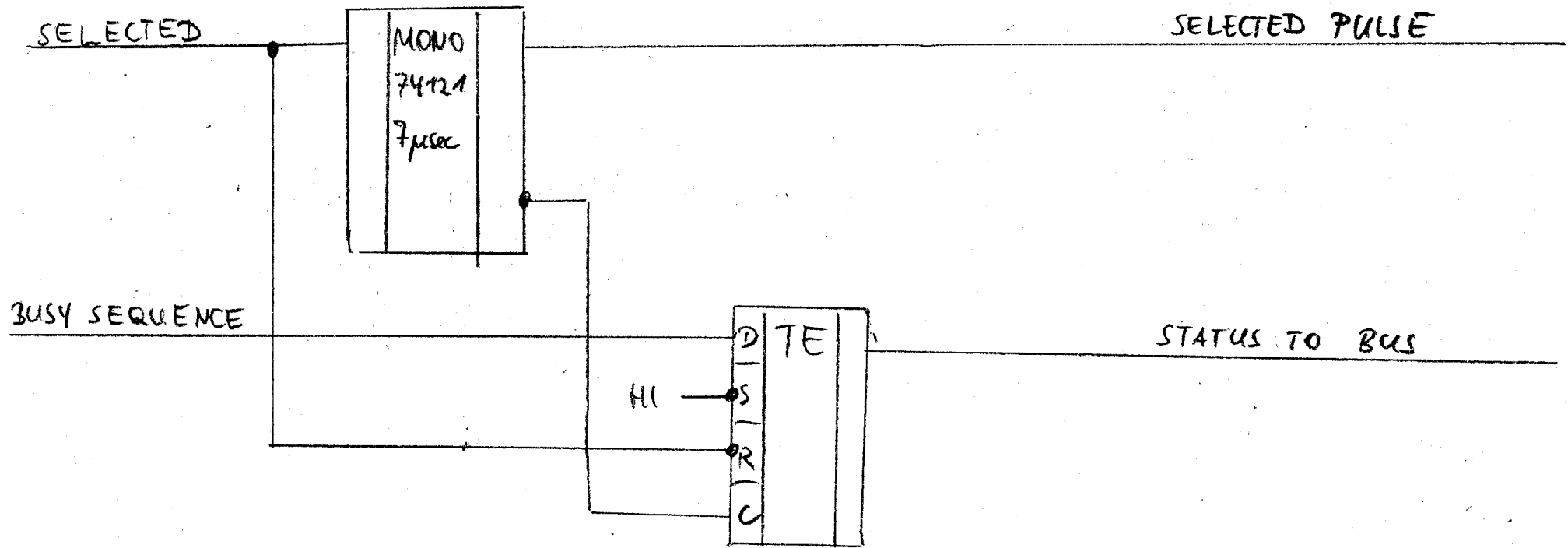


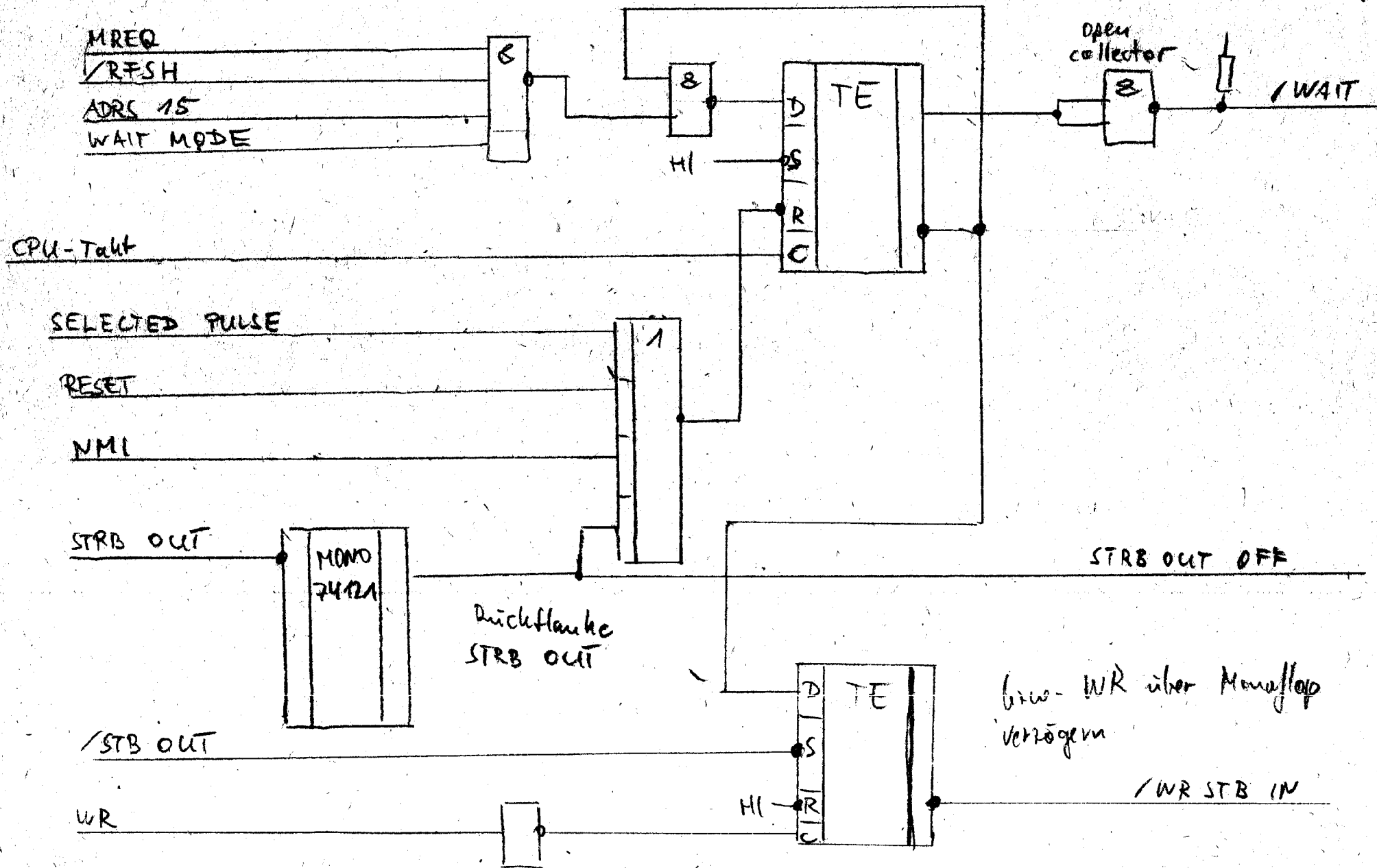


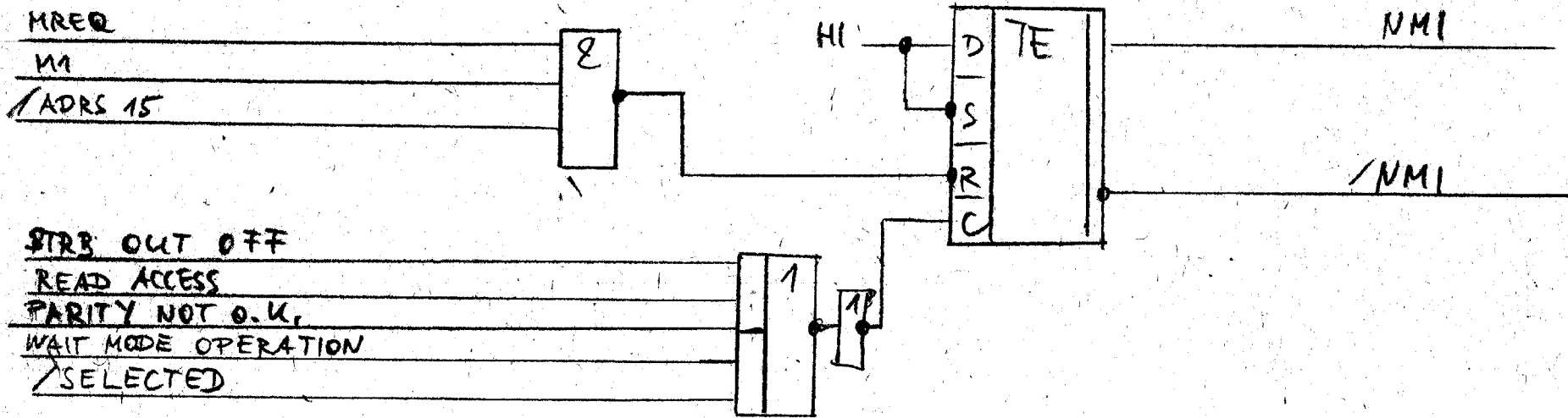










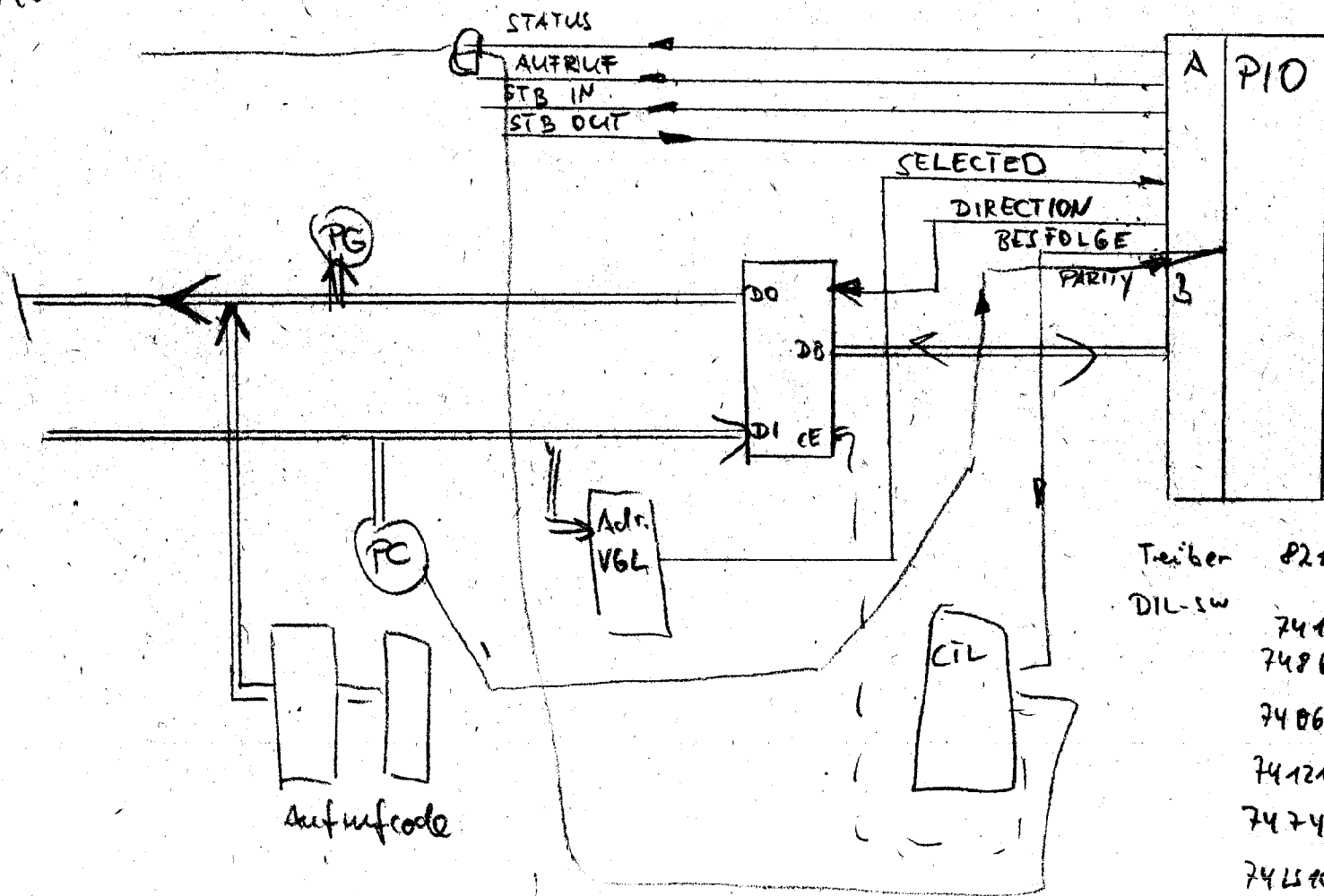


u BUSY SEQ. & RESET COMMAND

P10

- READ ACCESS
- WRITE ACCESS
- WAIT MOD OP.
- STATUS (COCT)
- RESFRCGE

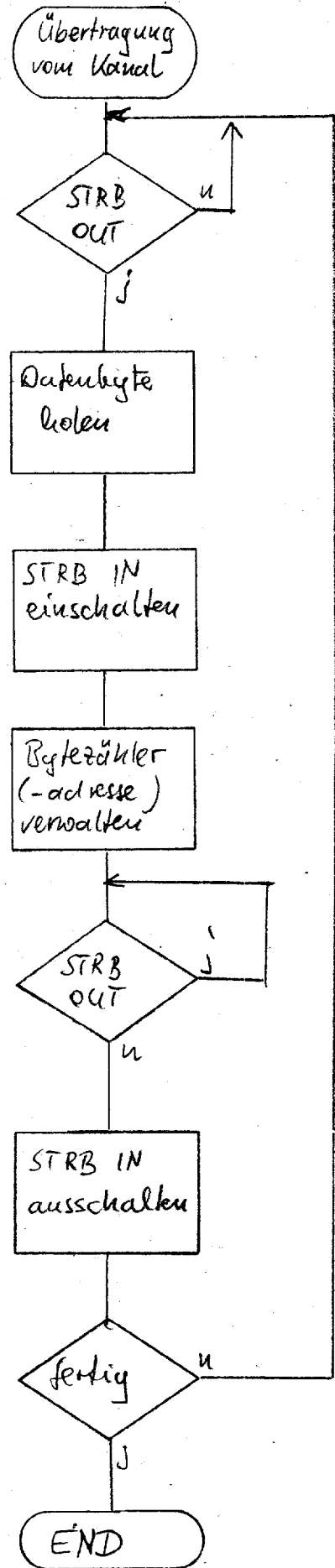
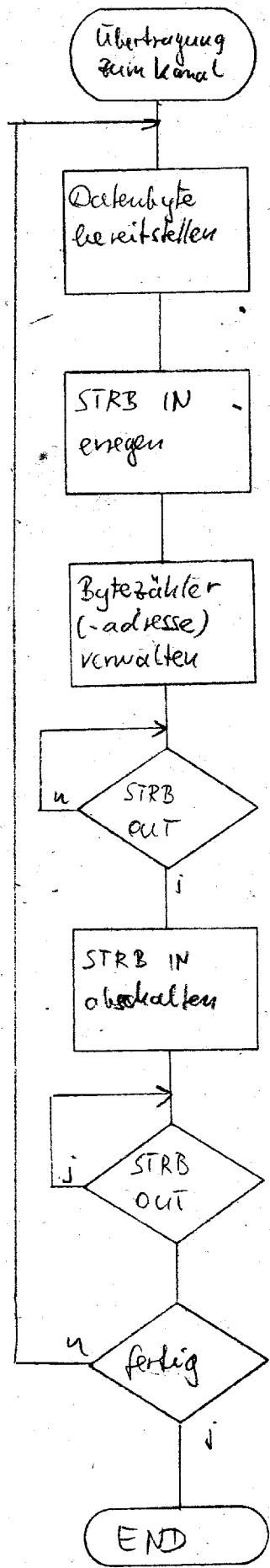
7406



Teilen	8216:	4
DIL-sw	74100	2
	7486:	2
	7406:	1
	74121:	3
	7474:	1
	741510:	1
	741500:	1

Minimalversion

1615



Interrupt - Zyklus:

aktueller Befehl bis Ende: ≈ 4 Zyklen
 M1 - Acknowledge-Zyklus: 6 Zyklen
 PC retten: 6 Zyklen
 RETI: 14 Zyklen

 30 Zyklen
 $\approx 12 \mu\text{sec}$

einfache Abfrage:

\rightarrow IN 10 Zyklen
 BIT, A 8 Zyklen
 JPE (JRE) 10 Zyklen (7 bei JRE)

 28 Zyklen \dots 25 Zyklen
 $\approx 10 \mu\text{sec}$

bei Anschluss an Speicheradressenraum

LD HL, ...
 \rightarrow BIT, M 12 Zyklen
 JPE (JRE) 10 Zyklen (7 bei JRE)

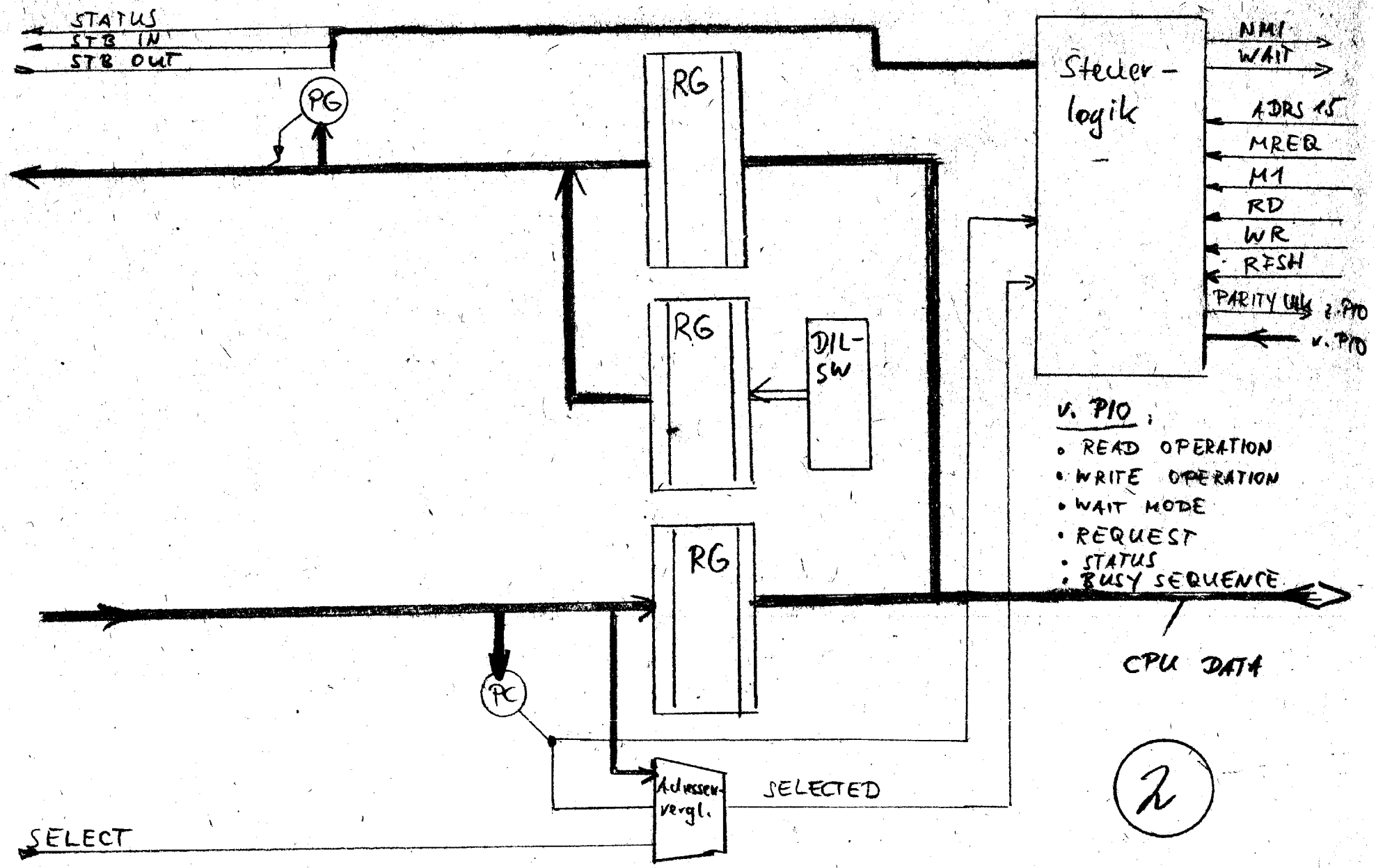
 22 Zyklen \dots 19 Zyklen
 $\approx 8 \mu\text{sec}$

NMI:

aktueller Befehl bis Ende: ≈ 4 Zyklen
 PC retten 6 Zyklen
 RETN 14 Zyklen

 24 Zyklen $\approx 10 \mu\text{sec}$

T-17



Anschluß-
varianten
T-IF

* weitere Eigenschaften:

- erweiterungsfähig
- kommandospezifische Reaktionen ohne Mithilfe der CPU möglich
- sehr weitgehende Diagnose möglich

1

Minimal-
ausführung

- 15 Schalthreise
- Datenrate um 10 kHz
- Antwortzeit bei Auswahl > 40 µs (von OBP bis CCT)

2

TTL-Steuer-
werk Anpass.
an CPU

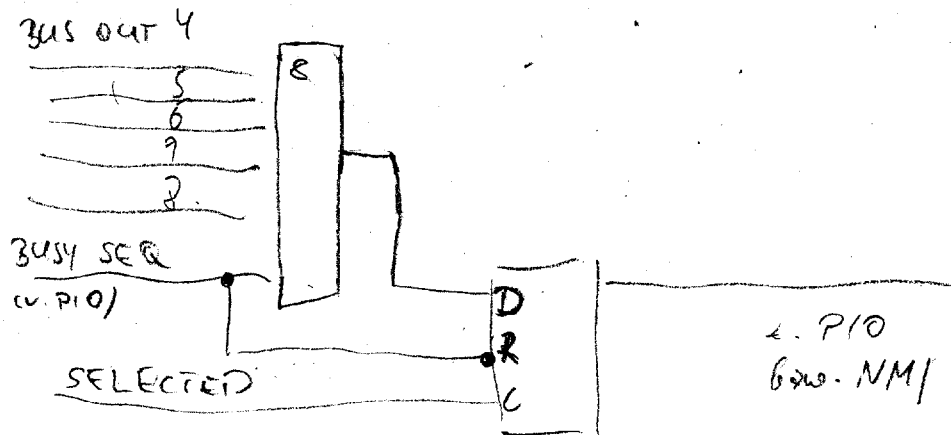
- 20 ... 30 Schalthreise
- Datenrate > 30 kHz (theoret. bis x 100 kHz)
- Antwortzeit bei Auswahl > 20 µs

Mikropro-
gramm-
steuerwerk*

- um 40 Schalthreise
- Datenrate > 30 kHz (theoret. bis x 100 kHz)
- Antwortzeit bei Auswahl < 10 µs

Zusätzliche Probleme

- a.) Erkennung des RESET-Kommandos während der Besatzfolge

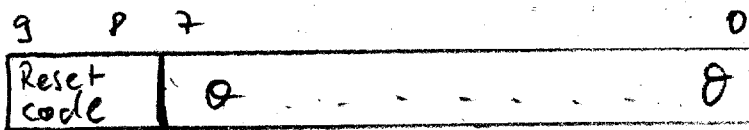


kann bei Variante 1 mit PARITY geadert werden, um im d. PIO einen Interrupt auszulösen (im Bit-Mode); kostet 1-2 IC's mehr!

- b.) Wie wird "Stoßbetrieb" gewährleistet?
(dadurch, daß SELECT aufrecht erhalten wird)?
kann das Gerät "Stoßbetrieb" erzwingen, indem es STATUS aktiv hält?

- c.) Zeiten der "Anerkennung"
- was kann in dieser Hinsicht erwartet werden?

Festadressen

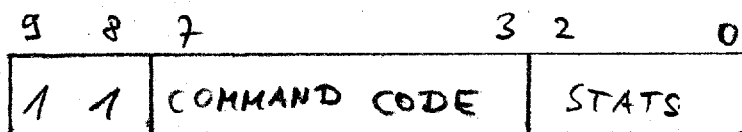


0 0 : Anfangsrücksetzen

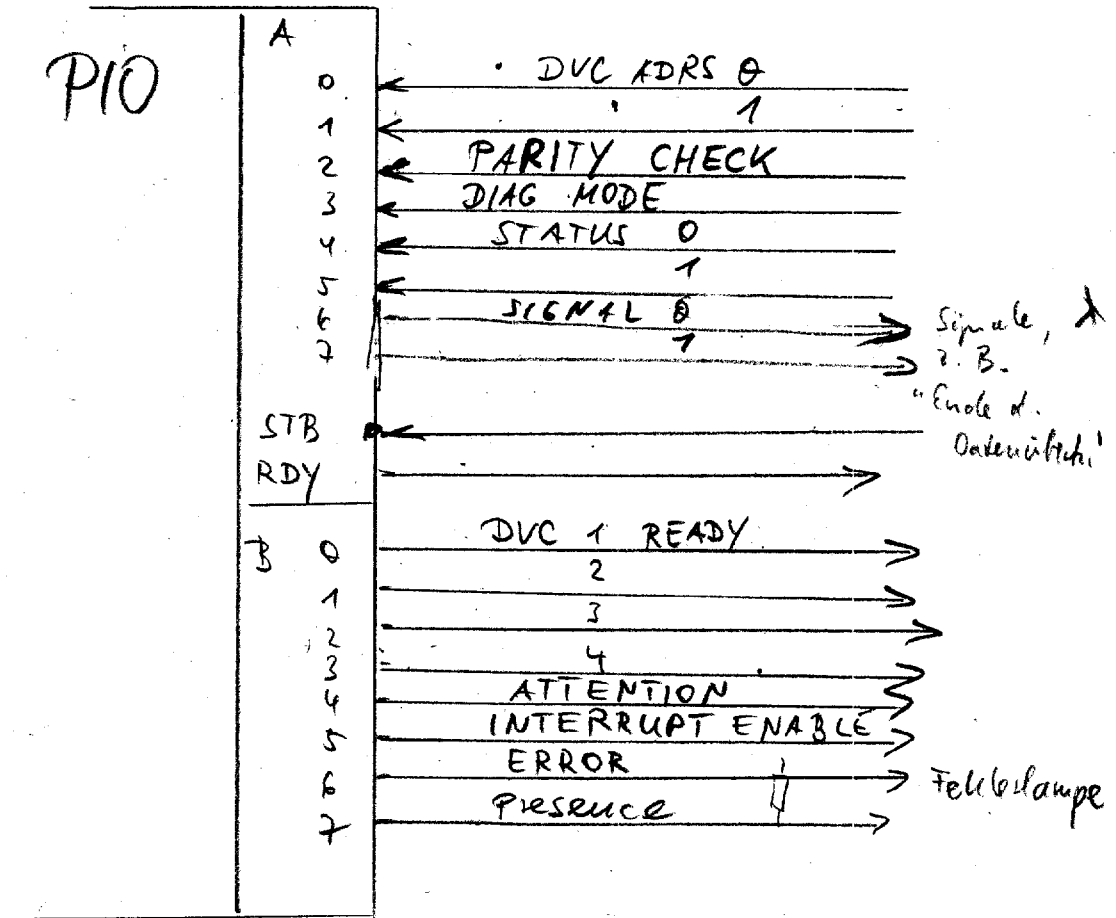
0 1 : Beginn der Auswahl

1 0 : Ende der Auswahl

Command jump



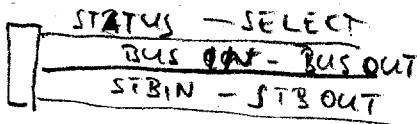
PIO



* oder Signalregister
an Datenbus

oder READY-Sig an
Datenb; Signalis. (4 bit)
über PIO

Kreuzschlüsselstecker



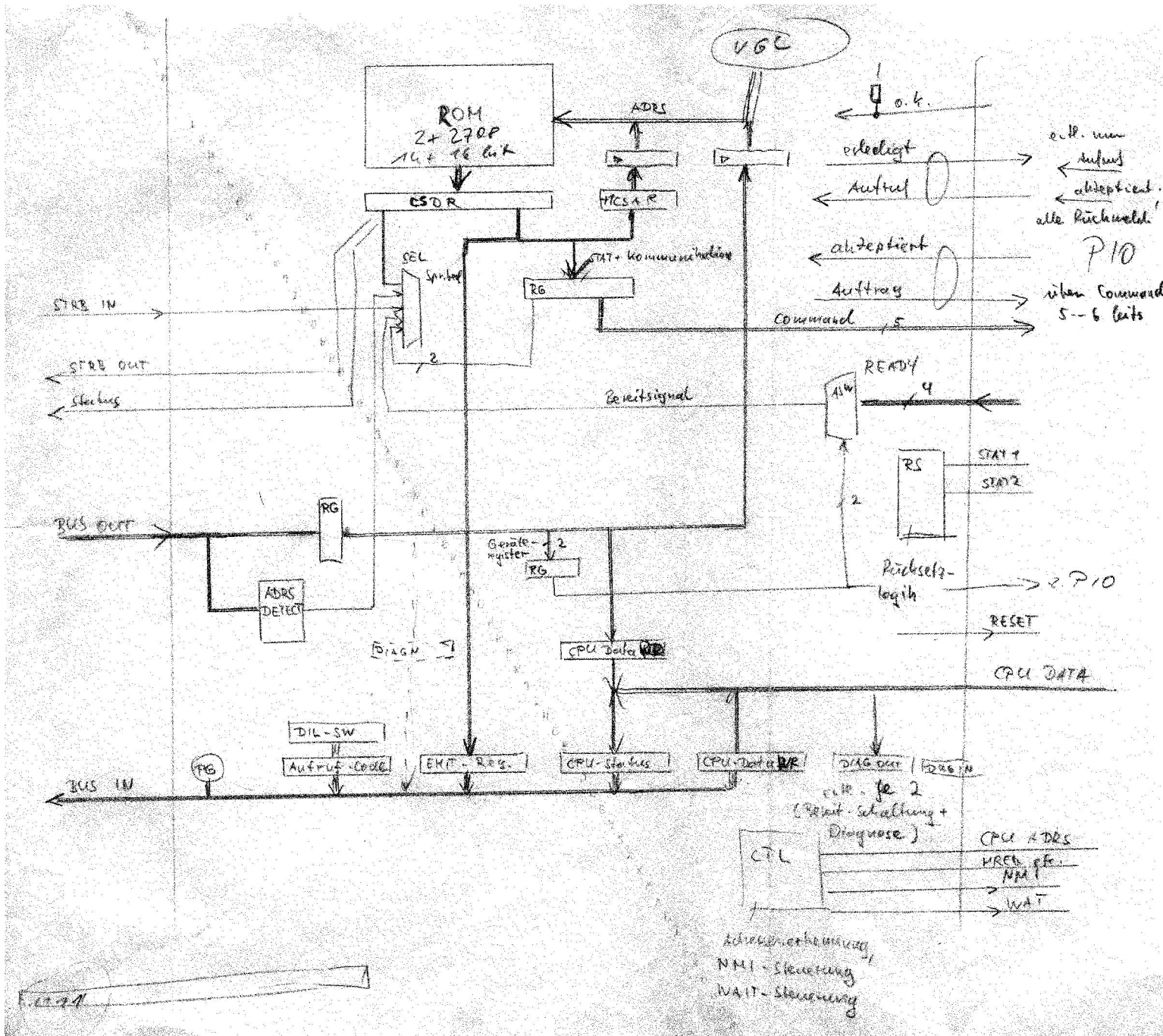
Multiassipale

- ~~Adress Statusreg IF~~
- ~~u u PIO~~
- ~~u Geräteadr-reg.~~
- Setzen Stat 1
- Löschen 1
- S Slot 2
- L
- S Slot 3
- L
- Löschen wait-IF
- CPU Info → BUS IN durchsteuern
- ~~CPU-Bus-Out-Register Adress~~

Tests

gesamtdurchlauf

- ROM-Test ZVE
- RAM - " "
- RAM - " Zusatz-RAM
- ROM-Test Zusatz-ROM
- Access loop ZVE - Zus. ROM "
- " " " RAM "
- " " " Floppy-Ad. "
- positionieren " "
- head load " "
- device select " "
- Access loop TIF-Controller "
- Interrupt from TIF - Contr.
- Plug test TIF - Contr.
- Aufzugtest " "
- Plug - Aufzugtest " "
- - 0 card 1 : 1
 - 0 card 0 : 0
 - 1 card 1 : 0
 - 1 card 0 : 1



Adressenbelegung
 NMI - Steuerung
 WAIT - Steuerung