

Vorbemerkung: Dies ist ein Versuch, ein Mikrocomputer-Modul gezeigtmäßig zu planen, zu realisieren und zu testen.

AUFGABENSTELLUNG

(14.2.1984)

Grobe Spezifikation:

Es ist ein universelles EPROM-Programmiergerät zu konstruieren.

Die zu programmierenden Eprom-Typen sind:

| |
|-------|
| 2716 |
| 2732 |
| 2764 |
| 27128 |

laut INTEL-Spezifikation.

Das Gerät ist als Einschub in einen 19"-Rahmen (VERO) mit 3HE und 8TE Frontplattengröße zu konstruieren.

Die Einheit soll auf einer Europakarte mit rückwärtsigen 64 poligen DIN 41612 C-64+0 Steckverbindern (Mensch/Computer) aufgebaut werden. Die Busbelegung entspricht dem MPR-80 Standard. (Genau Kurz Mikroprozessor-Bus)

Auf der Frontplatte soll sich ein 28 pol zero-insertion-force-Stecker befinden, ebenso eine Anzeige-LED, die den Programmierprozess anzeigt. Die EPROM-type-Umschaltung soll per Software möglich sein, ebenso das Programmieren einzelner Speicherzellen (random), sowie das Auslesen des Eproms.

Zum Schutz des Eproms soll sich eine Timeout-Vorrichtung auf dem Board befinden, die ein Überschreiten der Programmierzeit nicht gestattet die als Referenzzeitbasis für den Programmierimpuls genommen wird und den Programmierimpuls automatisch abhält.

Feine Spezifikation:

Der Computer soll das Gerät auf folgenden Adressen ansprechen können:

| | | |
|----|--------------|-----------------|
| | output | input |
| X0 | Adresse Low | — |
| X1 | Adresse High | — |
| X2 | Daten | Daten |
| X3 | EPROM type | busy/ready-flag |

— undefiniert

(X ... 0..F dezimal)
(je nach Einstellung)

Ein Schreibvorgang nach Adresse X0 (I/O) lädt das Adresse-Low-latch mit der dazugehörigen Adresse (nur dann, wenn nicht ein Programmiervorgang stattfindet)

Ebenso X1 das Most significant byte der Adresse.

Ein Schreibvorgang auf X2 leitet den Programmiervorgang ein.

Die Anzeig-LED leuchtet auf und ein 50ms Programmierzyklus wird begonnen. Währenddessen sind alle Schreibversuche gesperrt.

Ein Leservorgang auf X2 liest die Daten vom EPROM ein.

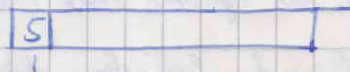
Ein Schreibvorgang auf X3 selektiert einen bestimmten Type von EPROM.

Dabei gilt:

| don't care | a | b | |
|------------|---|---|-------|
| | 0 | 0 | 2716 |
| | 0 | 1 | 2732 |
| | 1 | 0 | 2764 |
| | 1 | 1 | 27128 |

Ein Leservorgang auf X3 holt status-Information.

Dabei ist:



0 — ready
1 — busy

Programmiervorgang beendet
Programmiervorgang findet statt

Bei einem RESET - Vorgang stellt sich folgender Zustand ein:

Adresse: 0000

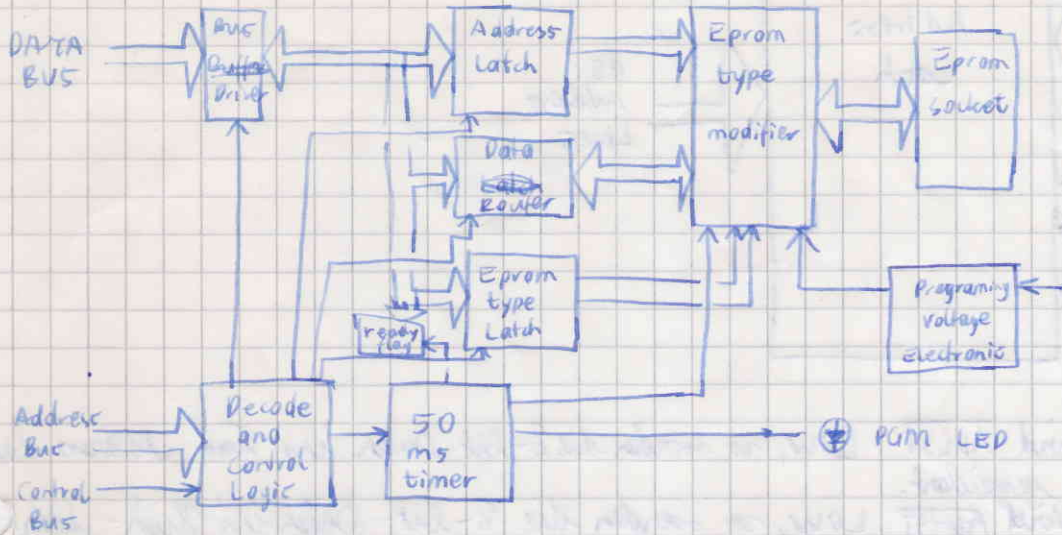
EPROM type: 2716

Status: ready

Daten-Lesen während des Programmierens ist zulässig und liest die derzeit zu programmierenden Daten ein.

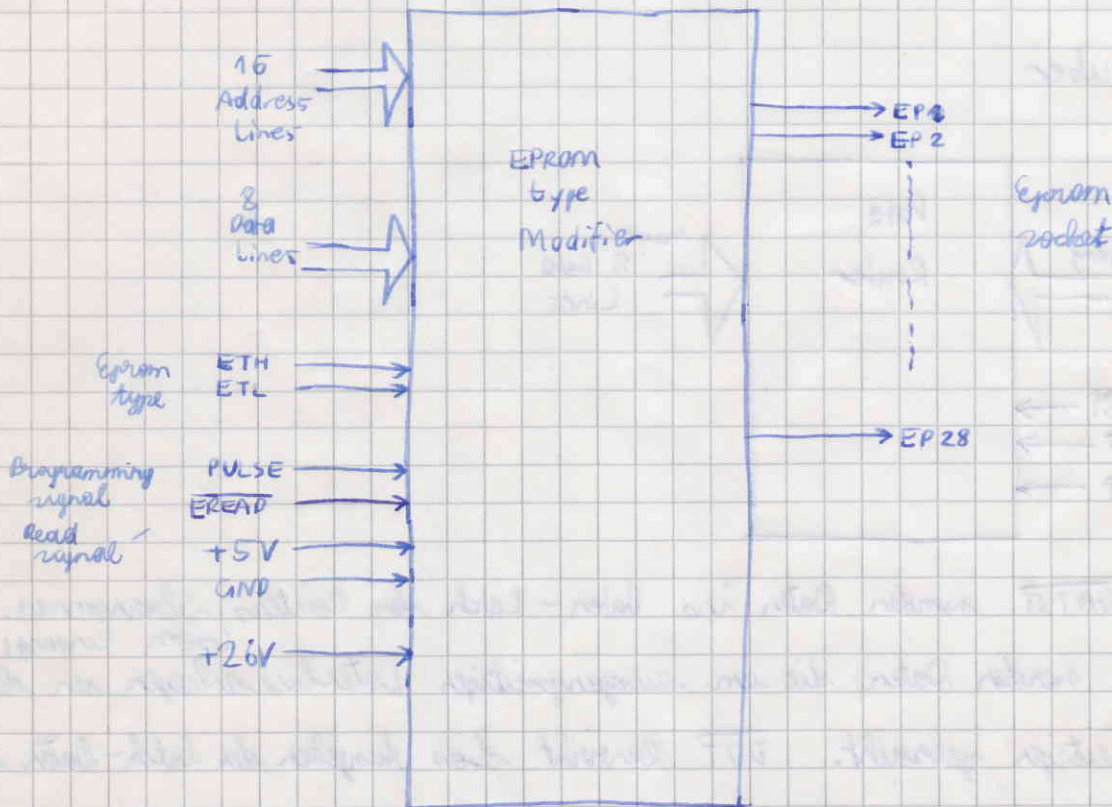
GROBENTWURF (14. 2. 1984)

Grobes Blockschaltbild des Gerätes:



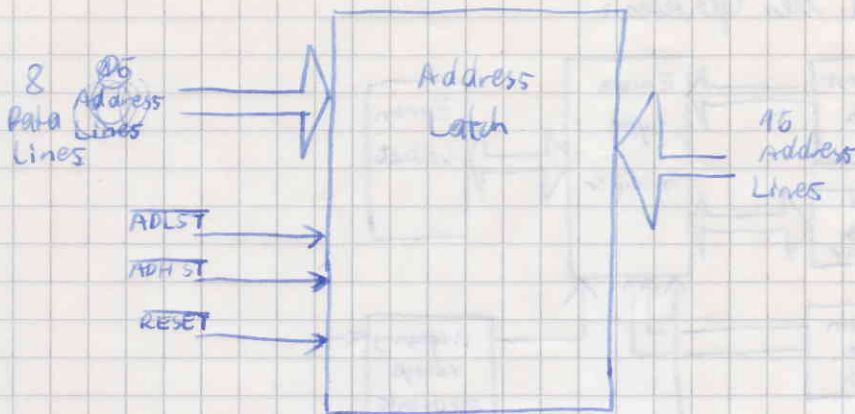
Einzelteil - Entwurf:

1) EPROM-Type-Modifizier:



Startet den Programmiervorgang ein (je nach EPROM type), wenn **PULSE** auf HIGH geht. Liest das Eprom, wenn **EREAD** auf LOW geht.

2) Address Latch



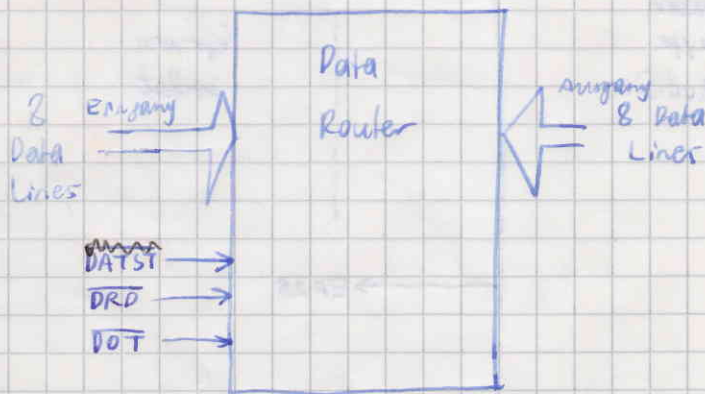
Wird \overline{ADLST} LOW, so werden die 8-bit-Daten ins Low-Address-Latch gespeichert.

Wird \overline{ADHST} LOW, so werden die 8-bit-Daten ins High-Addr-Latch gespeichert.

Wird \overline{RESET} LOW, so werden beide Latches gelöscht.

Die Ausgangs des Latches liegen unmittelbar an den 16 Address Lines an.

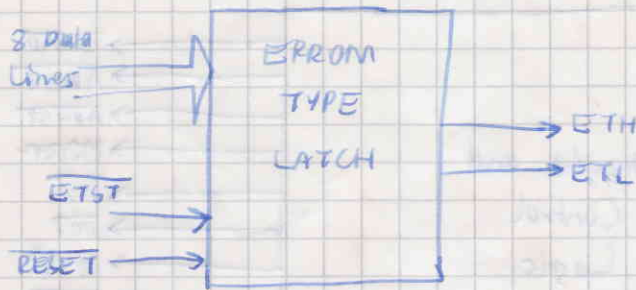
3) Data Router



Mit \overline{DATST} werden Daten ins Daten-Latch des Routers übernommen. (vom Eingang)

Mit \overline{DRD} werden Daten, die am ausgangseitigen Datenbus anliegen an den eingangseitigen gebracht. \overline{DOT} bewirkt das Abgeben der Latch-Daten am Ausgang.

4) EPROM TYPE LATCH



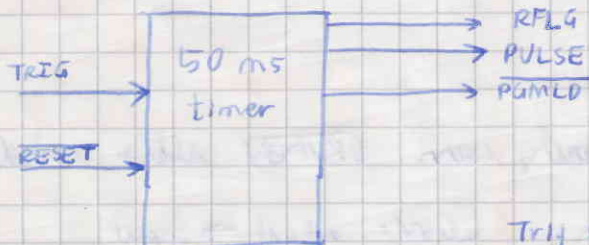
Mit ETST werden die Daten ins Latch übernommen. Die Bits 0 und 1 liegen danach am Ausgang ETH und ETL an.
RESET setzt die Ausgangs auf LOW

5) READY FLAG



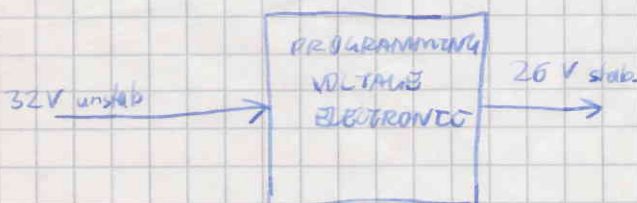
Mit RFRST wird das Bit 7 von Data Lines so gesteuert wie die Leitung RFLG anliegt. Sonst ist Bit 7 im High-Impedance state.

6) 50 ms timer



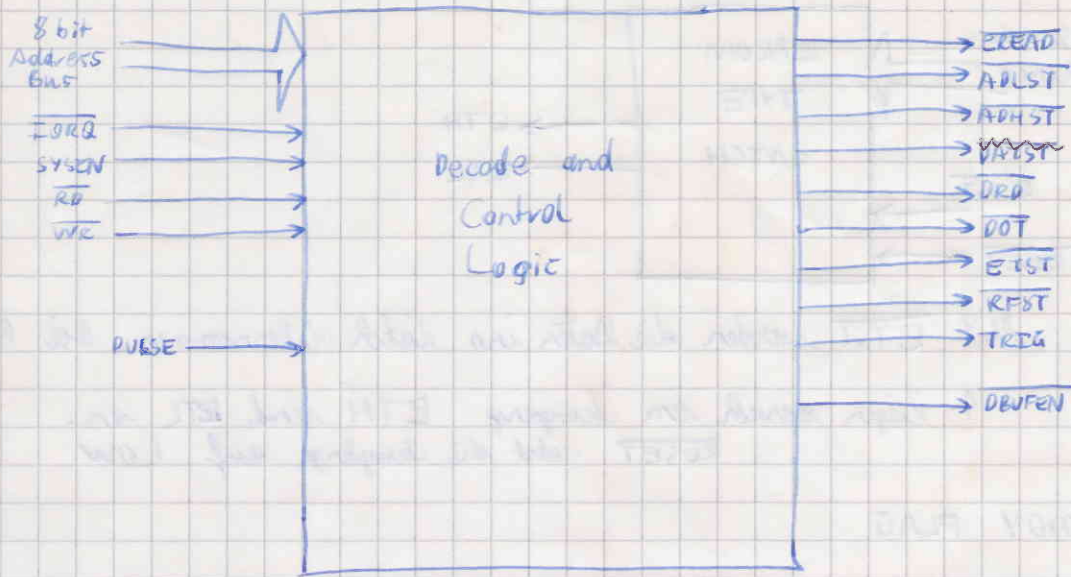
Trig startet den Timer an, der darauf auf RFLG, PULSE und PULSE einen 50ms-Impuls abgibt. RESET unterbricht den Timer.

7) PROGRAMMING VOLTAGE ELECTRONIC:



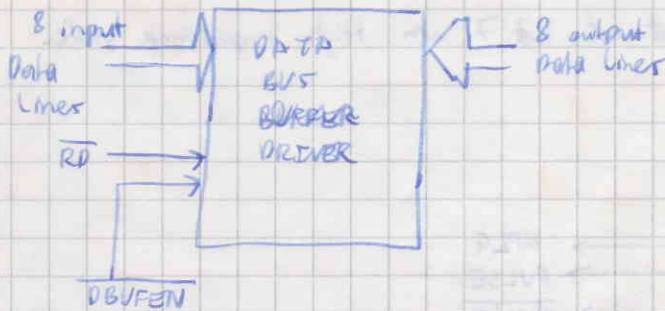
EPROM PROGRAMMER G-4

8) DECODE AND CONTROL LOGIC:



Steuert den Datenfluss zwischen den Latches und stellt den Programmierbeginn an. Spernt Schreibempfe während des Programmierens.

9) DATA BUS BUFFER



Schaltet Daten durch, wenn $\overline{D/UFEN}$ aktiv wird \overline{RD} gibt

die richtung an: LOW: output \rightarrow input
HIGH: input \rightarrow output

FEINENTWURF, algebraische Spezifikation der Module:

1) EPROM TYPE MODIFIER:

Input: 16 Address Lines: ADL 0 ... ADL 15
8 Data Lines: DAL 0 ... DAL 7 (bidirectional)
2 Eprom type lines: ETH, ETL
Programming pulse PULSE
Read signal $\overline{\text{EREAD}}$
Power supply +5V, +26V, GND all stabilized

!!! invalid state: $\overline{\text{EREAD}}$ LOW, PULSE HIGH !!!

Output: 28 EPROM lines EP 1 ... EP 28

able to be in one of the following states:

| | |
|-------|----------------|
| High | +5V |
| Low | 0V (GND) |
| Pgm | +25V |
| Float | High impedance |

T matches all conditions

EP 1 = (PULSE \rightarrow Pgm, T \rightarrow High)

EP 23 = (2716 \wedge PULSE \rightarrow Pgm, T \rightarrow ADL 11)

EP 2 \Leftrightarrow ADL 12

EP 24 = ADL 9

EP 3 \Leftrightarrow ADL 7

EP 25 = ADL 8

EP 4 \Leftrightarrow ADL 6

EP 26 = (27128 \rightarrow ADL 13, T \rightarrow +5V)

EP 5 \Leftrightarrow ADL 5

EP 6 \Leftrightarrow ADL 4

EP 27 = \neg PULSE

EP 7 \Leftrightarrow ADL 3

EP 8 = ADL 2

EP 28 = +5V

EP 9 = ADL 1

EP 10 = ADL 0

EP 11 \Leftrightarrow DAL 0

EP 12 \Leftrightarrow DAL 1

EP 13 \Leftrightarrow DAL 2

EP 14 = ~~DAL~~ GND

EP 15 \Leftrightarrow DAL 3

EP 16 \Leftrightarrow DAL 4

EP 17 \Leftrightarrow DAL 5

EP 18 \Leftrightarrow DAL 6

EP 19 \Leftrightarrow DAL 7

2716 = \neg (ETH \vee ETL)

2732 = \neg ETH \wedge ETL

27128 = ETH \wedge ETL

EP 20 = $\overline{\text{EREAD}} \wedge (2716 \equiv \text{PULSE})$

EP 21 = ADL 10

EP 22 = (2732 \rightarrow (PULSE \rightarrow Pgm, T \rightarrow $\overline{\text{EREAD}}$), T \rightarrow $\overline{\text{EREAD}}$)

\Leftrightarrow direct connection

EPROM PROGRAMMER

F-2

8) DECODE AND CONTROL LOGIC

Inputs: Address bus 8 lines A0 ... A7

\overline{IORQ}
 \overline{SYSEN}
 \overline{RD}
 \overline{WR}
PULSE

Outputs:

\overline{EREAD}
 \overline{ADLST}
 \overline{ADHST}
 \overline{DATSTB}
 \overline{DRD}
 \overline{DOT}
 \overline{ETST}
 \overline{RFST}
TRIG
 \overline{DBUFEN}

Eprom read
Address low strobe
Address high strobe
Data latch strobe
Data read
Data out to Eprom
Eprom type strobe
Read flag strobe
trigger timer
enable data buffer

$$\overline{DBUFEN} = \neg(\overline{SYSEN} \wedge (A4 \dots A7 = X4 \dots X7) \wedge A2 \wedge A3 \wedge \overline{IORQ} \wedge (\overline{RD} \vee \overline{WC}))$$

$$\overline{WC} = \neg \overline{WR} \wedge \text{PULSE}$$

$$\overline{ADLST} = \neg(\overline{WC} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\overline{ADHST} = \neg(\overline{WC} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\overline{DATSTB} = \neg(\overline{WC} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\overline{DRD} = \neg(\overline{RD} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\overline{DOT} = \neg \text{PULSE}$$

$$\overline{ETST} = \neg(\overline{WC} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\overline{RFST} = \neg(\overline{RD} \wedge A0 \wedge A1 \wedge \overline{DBUFEN})$$

$$\text{TRIG} = \neg \overline{DATSTB}$$

$$\overline{EREAD} = \neg(\overline{RD} \wedge \neg \text{PULSE})$$

alle anderen Module scheinen trivial

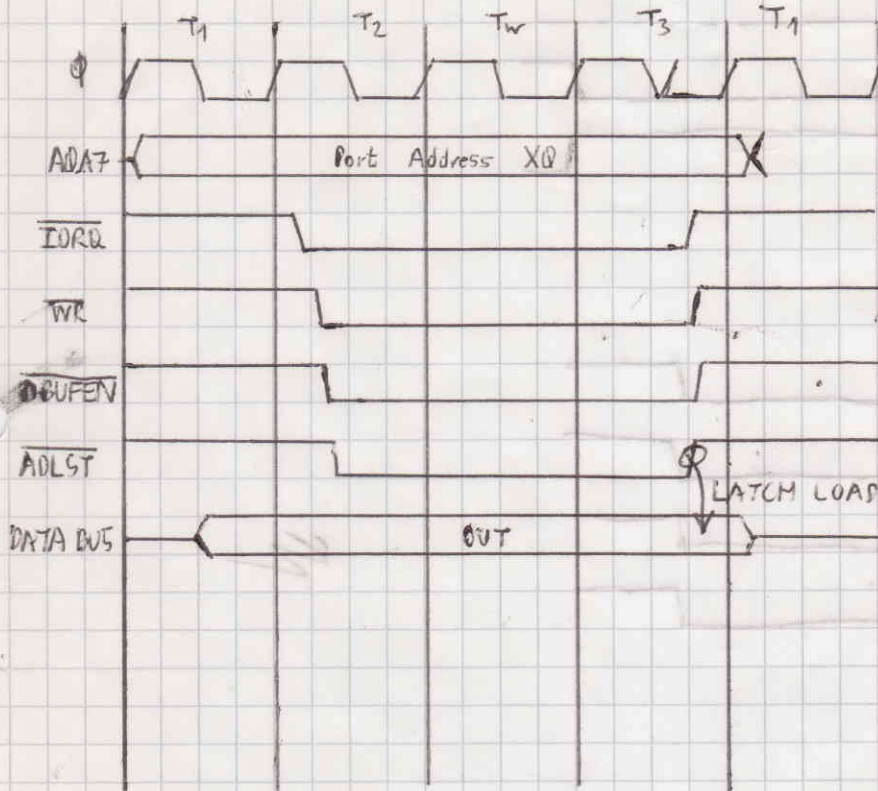
EPROM PROGRAMMER T-1

TEST:

1) Pre-Construction Desk Check:

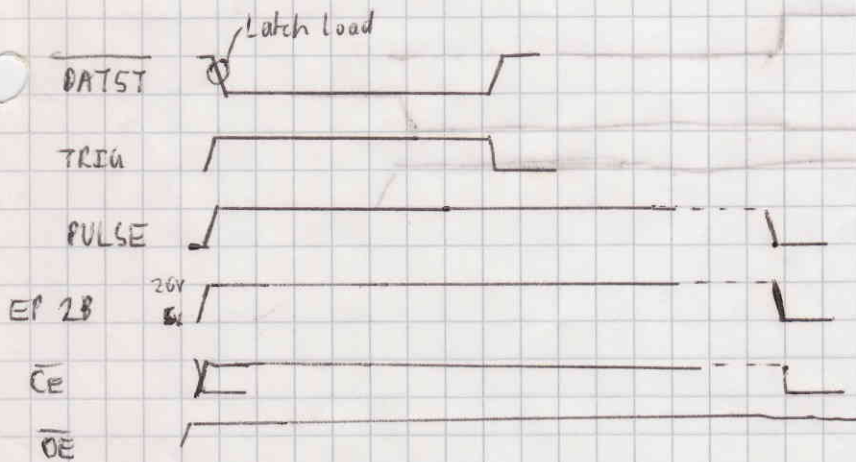
Address Latch Write Cycle:

SEN assumed High, PULSE assumed LOW

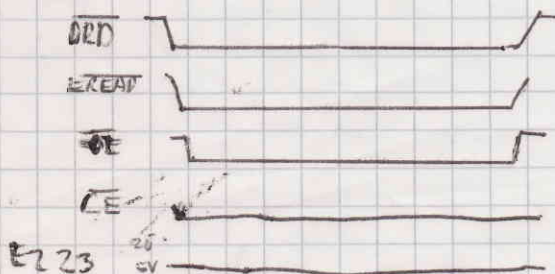


same with
Address High Write
Eeprom type write
Data Write

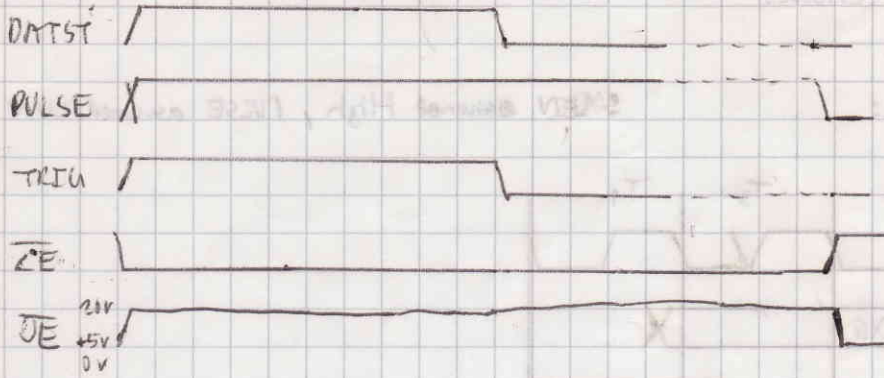
2716 Program Cycle



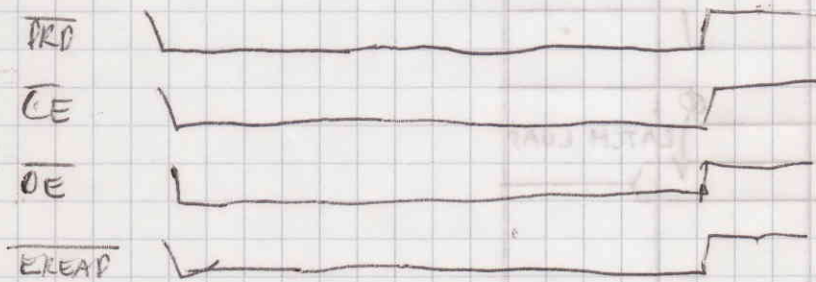
2716 Read Cycle



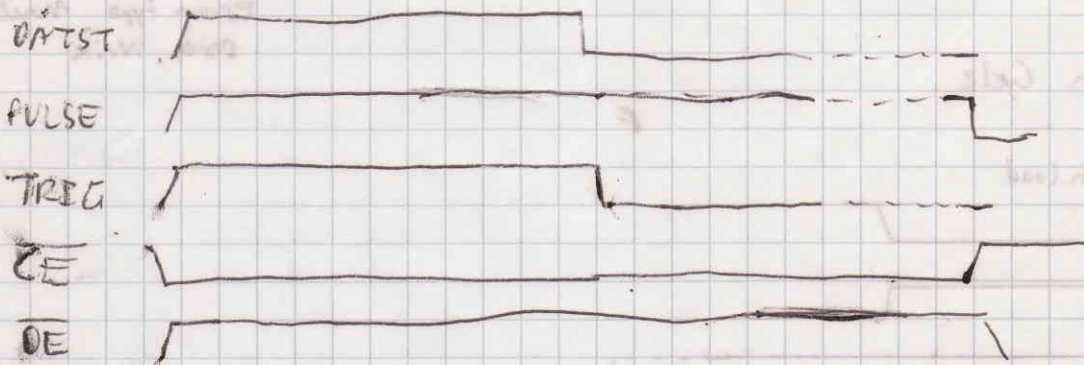
2732 Program Cycle



2732 Read Cycle



2764 Program Cycle



EPROM PROGRAMMER Parts List

| | | |
|-----|----|----------------------|
| ✓ 2 | IC | 74 LS 273 |
| ✓ 1 | IC | 74 LS 374 |
| ✓ 1 | IC | 74 LS 244 |
| ✓ 1 | IC | 74 LS 85 |
| ✓ 1 | IC | 74 LS 125 |
| ✓ 1 | IC | 74 LS 122 |
| ✓ 2 | IC | 74 LS 138 |
| ✓ 1 | IC | 74 LS 245 |
| ✓ 2 | IC | 74 LS 00 |
| ✓ 2 | IC | 74 LS 02 |
| ✓ 1 | IC | 74 LS 04 |
| ✓ 1 | IC | 74 07 |
| ✓ 1 | IC | 74 LS 32 |
| 1 | IC | 74 LS 74 |

| | | |
|-----------------|-----------------|--------|
| ✓ 5 | Fassungen IC | 20 pol |
| ✓ 3 | Fassungen IC | 16 pol |
| ✓ 10 | Fassungen IC | 14 pol |
| ✓ 2 | Fassungen IC | 28 pol |
| ✓ 3 | Diöden AA | 118 |
| 4 | Transistoren BC | 238 |
| ✓ 1 | Zenerdiode | 27V |
| ✓ 1 | LED 3mm rot | |
| 10 1 | Transistor BC | 237 |

| | | |
|-----|------|---------------|
| ✓ 4 | Wid. | 4,7k Ω |
| ✓ 1 | Wid. | 47k Ω |
| ✓ 1 | Wid. | 150 Ω |
| 5 | Wid. | 10k Ω |

| | | |
|-----|----|-------------|
| ✓ 2 | Co | 10 μ F |
| ✓ 1 | Co | 0,1 μ F |
| ✓ 2 | Co | 47 μ F |

1 Trimpot 10k Ω

| | | |
|-----|------------------|--|
| ✓ 1 | Euroboard | |
| ✓ 1 | DEW-Connecter | |
| ✓ 1 | Frontplatte | |
| ✓ 1 | 24pol ZIF-Socket | |
| ✓ 1 | LED-Prüfbuchse | |
| ✓ | Fädelschlinge | |
| ✓ | Fädeldraht | |
| ✓ | Weg-Draht | |