

# **1MBUniMem-8Bit - Universelle 1Mbyte Speicher für den ECBbus**

**8 Byte-wide-Speicherbausteine mit 32-128 KByte Kapazität**

**Akku-Pufferung für den Einsatz mit CMOSRAMs**

**Adreßdecodierung durch PROM oder PAL Baustein, dadurch hohe Flexibilität**

**Gemischte Bestückung (RAM, EPROM, 32-128k) möglich.**

**Unterstützt den erweiterten ECBbus Adreßraum  
(1 MByte, mit CPUZ280 auch 4 MB).**

Version 1.0  
Stand: 31.1.2024  
Author: Hajo Kitzhöfer

# Inhaltsverzeichnis

1. Vorwort.....	3
2. Überblick.....	4
3. Busanpassung.....	5
4. Technische Beschreibung.....	6
4.1. COMMON Bereich.....	7
4.1.1 ohne COMMON Steuerung.....	7
4.1.2 mit COMMON Steuerung.....	7
5. Bestückungsmöglichkeiten.....	8
6. Speicher - Steckbrücken.....	9
7. Bestückung ohne Akkupufferung.....	11
8. Akkupufferung – mögliche Optionen.....	12
8.1 Onboard Akkupufferung.....	13
8.2 Pufferung des Speichers durch externe Spannung.....	15
9. Bestückungsliste.....	16
10. Schaltplan.....	17
11. PAL Varianten.....	17
Anhang.....	18
ECB-BUS Varianten.....	18
Erläuterung zur PAL Adressdekodierung.....	19

# 1. Vorwort

Im folgenden wird die 1MBUniMem ECBbus Speicherkarte detailliert beschrieben. Diese Speicherkarte basiert auf der 1MCRE Speicherkarten von Elzet80, unterscheidet sich aber in einigen Punkten.

- die PAL/PROM basierenden Adressdekodierung und die COMMON-Bereichssteuerung wurde geändert da die verwendeten PALs und PROMs nicht mehr erhältlich sind. Die Funktionalität wurde erhalten.
- bei der Akkupufferung kann man nun bei der Bestückung zwischen zwei unterschiedlichen Akku Typen wählen oder alternativ eine Batterie von Type CR2032 verwenden.

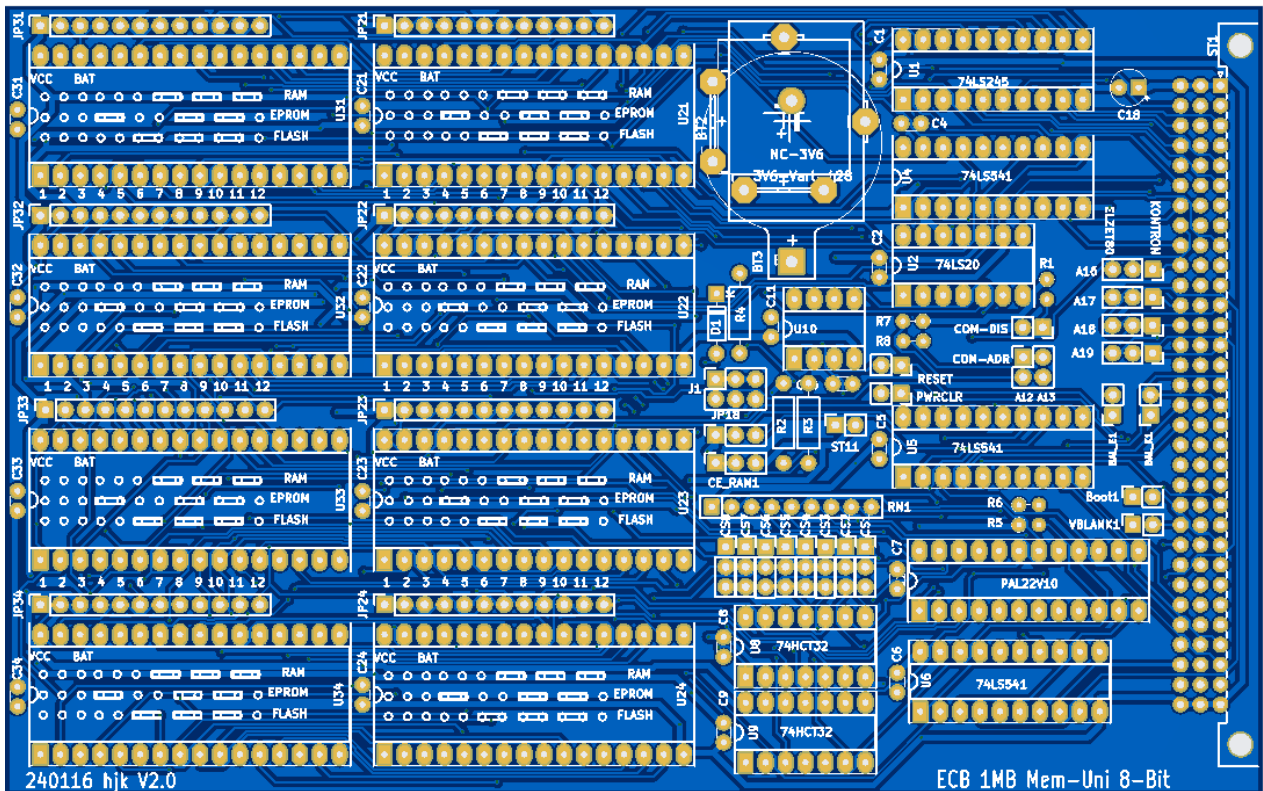


Bild 1: Bestückungsseite

## 2. Überblick

Die 1MBUniMem ist eine äußerst flexible und vielseitige Speicherbaugruppe für den ECBbus. Sie kann in Verbindung mit allen ELZET 80 CPU-Baugruppen und vielen CPU-Baugruppen von anderen Herstellern eingesetzt werden.

Man kann sie zum Speicherausbau von Rechnern mit Standardbetriebssystemen, wie z.B. CP/M 3 verwenden. Durch die Mischbarkeit verschiedener Speichertypen ist sie für den Einsatz in unterschiedlichsten Anwendungsbereichen geeignet.

Durch passende Dekodierung mittels eines PAL-Bausteins lässt sich die Speicherkarte an verschiedene Einsatzzwecke anpassen. Einige mögliche Varianten findet man im Anhang.

Die Speicherkarte lässt sich über Steckbrücken an zwei ECBbus Varianten anpassen. Die zwei Varianten sind:

- Elzet80, Systec, MC oder
- Kontron, MKC, J&K, RetroBrew etc.

Nähere Informationen finden man in Anhang.

### 3. Busanpassung

Die Steckbrücken A16 bis A19 und BAL\_E, BAL\_K dienen der Busanpassung. Wie bereits erwähnt lässt sich die Karten an den

- Elzet80 oder den
- Kontron/RetroBew

Bus anpassen.

Kann man auf die flexible BUS Anpassung verzichten dann lassen sich die Steckkontaktleisten und Jumper durch Draht-Lötbrücken ersetzen.

	<b>Elzet80</b>	<b>Kontron</b>
A16	19c	10c
A17	17a	12c
A18	12a	13c
A19	12c	14a
BAI	23a	12a
BAO	25a	17a
/BOOT	22a	-
/VBLANK	23c	-

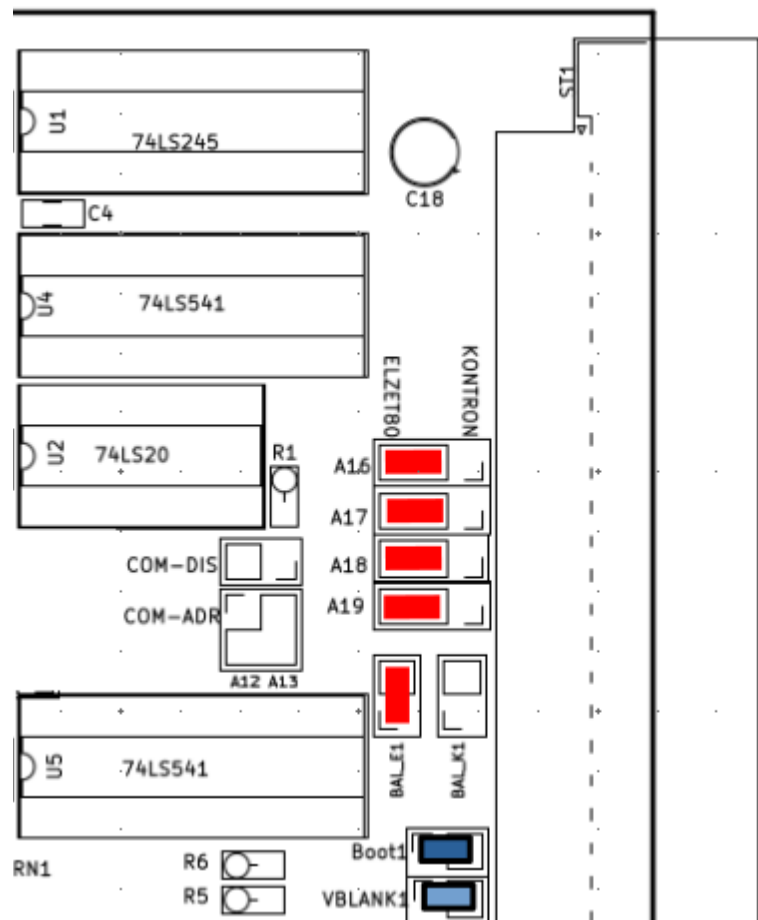


Bild 1: Jumper Einstellung für den Elzet80 Bus

Die Elzet80 spezifischen Signal /VBLANK und /BOOT werden über die Brücken (Boot und VBLANK, blaue Jumper im Bild) geführt um sie einfach unterbrechen zu können, falls sie den Betrieb der Karte oder des Rechners stören.

## 4. Technische Beschreibung

Die **1MBUniMem** Speicherkarte kann mit unterschiedlichen 128 KB Speicherbausteinen bestückt werden, z.B. mit

- 128 KB SRAM oder CMOS Chips wie z.B.
- 128 KB Eproms
- 128 KB EEProms (Flash Speicher)

Der Mischbestückung mit unterschiedlichen großen Speicherbausteinen (32 K, 64 K) sollte möglich sein, wurde aber noch nicht getestet.

Die maximale Gesamtkapazität der Karte beträgt also 1MByte RAM/EPROM/FLASH. Sie eignet sich für alle CPU Baugruppen, die mit einer Adresserweiterung ausgestattet sind, oder CPU bedingt über einen linearen Adressraum von 1MB verfügen (z.B. CPU 68k). Ein Einsatz in Systemen mit mehr als 1MB Adressraum ist möglich, wenn die Signale /BOOT und /BLANK nicht gebraucht, und statt dieser zwei zusätzliche Adressen der Baugruppe zugeführt werden. Selbstverständlich muss dann auch der Decoder-PAL entsprechend programmiert werden.

Aus den Adressen AI2 bis AI9 werden über einen PAL-Baustein (22V10 U7) die Speicherauswahlsignale CS1 .. CS8 generiert. Daraus ergibt sich, daß der Speicher eines Bausteins in Seiten von minimal 4kB über den ganzen Adressraum der **1MBUniMem** verteilt werden kann. Es findet allerdings keine (!) Adressumrechnung statt. Dadurch muss aber das vorher Gesagte etwas eingeschränkt werden, die Seiten müssen auf der korrekten Adresse einer 64kB Seite liegen.

Ein weiterer Aspekt der Dekodierung ist, dass eine Speicherseite auch auf mehreren Adressseiten auftauchen kann! Diesen Effekt nutzt man für den COMMON-Bereich, der in der Regel von Systemen, die Banking unterstützen, benötigt wird.



Bild2: COMMON-Bereich - Beispiel

Setzt man auf der **1MBUniMem** CMOS-RAM Bausteine ein, so ist es möglich, sie über einen opti-

onalen Akku auch bei abgeschaltetem Rechner mit Strom zu versorgen. Dadurch behalten sie auch über einen längeren Zeitraum noch die korrekte Information, was besonders für Anwendungen wie z.B. eine RAM-Floppy interessant ist.

Die Speicherkarte ist geeignet zum Einsatz mit der VIDEO-80 Karte aus dem Elzet80 System, d.h. wenn das Bussignal VBLANK auf LOW gezogen wird, ist der EPROM/RAM-Bereich E000 bis EFFF ausgeblendet.

Die Speicherkarte kann auch in Bootsystemen eingesetzt werden, d.h. wenn das Bussignal BOOT auf LOW gezogen wird, werden die untersten 32KB auf Bank 0 ausgeblendet.

## 4.1. COMMON Bereich

### 4.1.1 ohne COMMON Steuerung

Wird keine COMMON Steuerung benötigt, z.B. Z80 mit 64K Speicher oder 68xxx Systeme mit linearem Adressraum, dann können U2 (74HCT20), die Widerstände R7,R8 und die Pfostenstecker COM-DIS und Com-ADR entfallen. Der PAL22V10 muss entsprechend programmiert sein.

### 4.1.2 mit COMMON Steuerung

Bei der original **1MCRE** Speicherkarte von ELZET-80 wird in einem zusätzlichen PAL die Steuerung für den COMMON Bereich für 128K Speicherbausteine realisiert.

In diesem PAL werden alle Chip-Select Signale der Speicherbausteine ausgewertet und die Größe des COMMON-Bereichs durch zwei Jumper eingestellt.

Bei der **1MBUniMem** wird diese Steuerung durch den PAL 22V10 (U7) in Kombination mit einem 74HCT20 (U7) realisiert. Somit ist kein zweiter PAL erforderlich.

Der COMMON-Bereich lässt in 4 KB Schritten einstellen. Die folgende Tabelle zeigt die möglichen Jumper Einstellungen.

Common	COM-ADR - A13	COM-ADR - A12	COM-DIS
16 KB	offen	offen	geschlossen
8 KB	geschlossen	offen	geschlossen
4 KB	geschlossen	geschlossen	geschlossen
0	-	-	offen

Bei einer Bestückung mit Speicherbausteinen mit einer Kapazität größer als 64K, also hier 128K Speicher, tauchen bei 8-Bit Prozessoren Speicherbereiche zweimal auf (Details dazu im Anhang). Damit diese Überlappung im COMMON-Bereich nicht auftritt muss das Adresssignal A16 durch den PAL Baustein „gesteuert“ werden.

Daher wird die Signalleitung A16 nicht direkt mit dem Speicherbausteinen verbunden sondern „über“ das PAL geführt.

## 5. Bestückungsmöglichkeiten

Die **1MBUniMem** kann wahlweise mit folgenden Speicherbausteinen (auch gemischt) bestückt werden:

	Typ	Kapazität
<b>EPROM</b>		
	27256	32 KB
	25512	64 KB
	27C1001	128 KB
<b>RAM</b>		
	62256	32 KB
	KM681000	128 KB
<b>EEPROM/FLASH</b>		
	28C256	32 KB
	SST39SF010	128 KB

Der Mischbestückung mit unterschiedlichen großen Speicherbausteinen (32 K, 64 K) sollte möglich sein, wurde aber noch nicht getestet.

Selbstverständlich sind auch alle kompatiblen Speicherbausteine einsetzbar.



## 6. Speicher - Steckbrücken

Die **1MBUniMem** besitzt eine Reihe von Steck- und Lötbrücken, die der Anpassung der Karte an unterschiedliche Anforderungen und Speicherbausteine dienen. Im folgenden werden diese Steck- und Lötbrücken im Einzelnen vorgestellt und ihre Funktion beschrieben.

Steckleisten JP21 .. JP24, JP31 .. JP34

Diese identischen Steckleisten dienen der Anpassung der Karte an unterschiedliche Speicherbausteine.

Die Pinbelegung der Byte-wide-Speicherbausteine ist nicht exakt genormt, so dass gewisse Inkompatibilitäten mit diesen Steckbrücken ausgeglichen werden können. Außerdem wird hier eingestellt, ob ein Baustein akkugepuffert werden soll oder nicht. Da die Karte gemischt (EPROM / RAM / FLASH) bestückt werden kann, muss auch für jeden (Speicher-) Steckplatz eine Steckbrückenleiste vorhanden sein, wobei die Nummern von Baustein und Steckleiste die Zugehörigkeit klären (JP21 zu U21, JP22 zu U22 usw.).

Die Steckleisten haben je 12 Kontakte. Die Position der Steckleiste auf der Karte ist dem Bestückungsplan zu entnehmen. Für die möglichen Bausteine sind die zugehörigen Jumperstellungen auf der Platine aufgezeichnet.

Das folgende Bild zeigt die Belegung der 12 Kontakte.

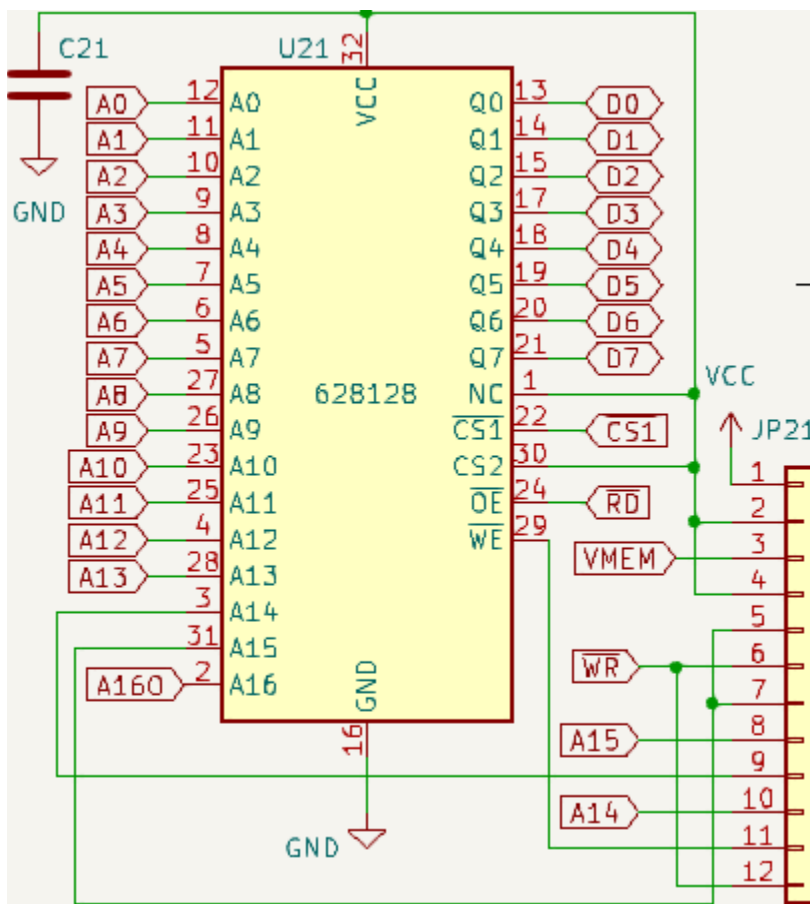


Bild 3: Belegung der Speicher-Steckleiste

Die folgenden Bilder zeigen die Jumperstellungen für die drei Speichertypen RAM, EPROM und Flash.

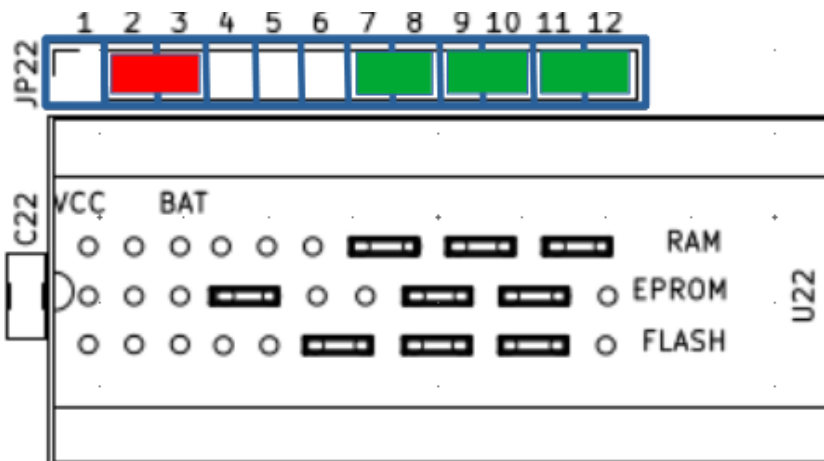


Bild 4: Jumperstellung für RAM Speicher die über die Akkupuffer versorgt werden; Brücke 2 - 3

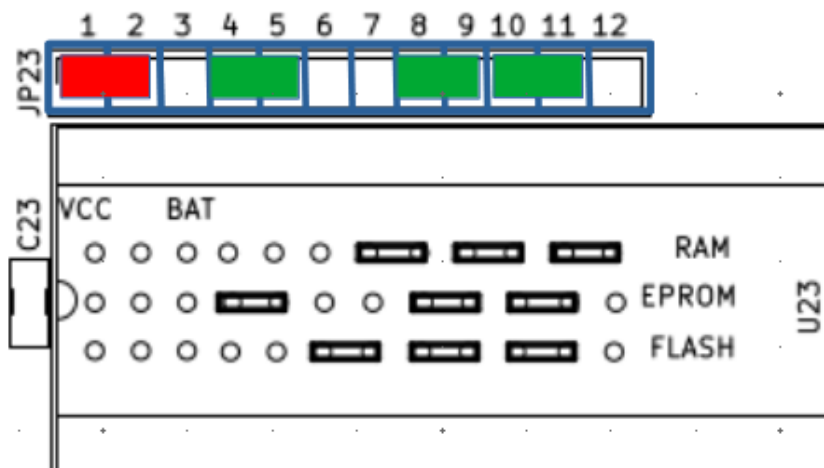


Bild 5: Jumperstellung für EPROM Speicher. Spannungsversorgung = VCC Brücke 1 - 2

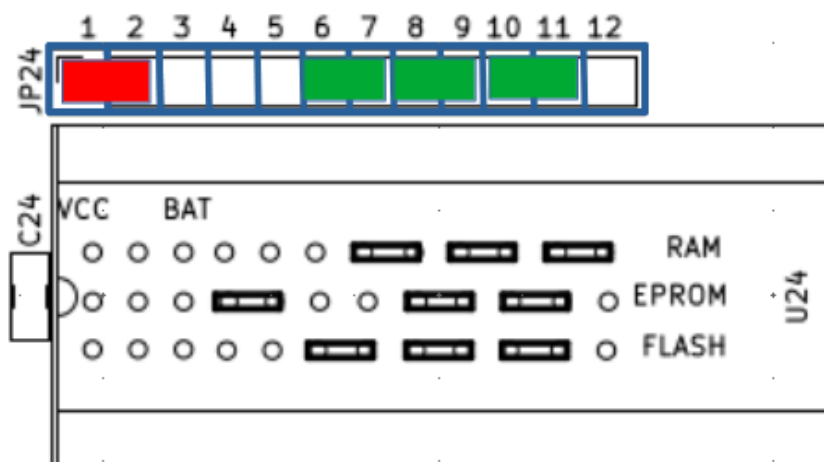


Bild 6: Jumperstellung für FLASH Speicher. Spannungsversorgung = VCC Brücke 1 - 2

## 7. Bestückung ohne Akkupufferung

Wird diese Pufferung nicht benötigt so kann man einen Teil der Platine unbestückt lassen. Welche Bauteile entfallen entnimmt man der Bestückungsliste.

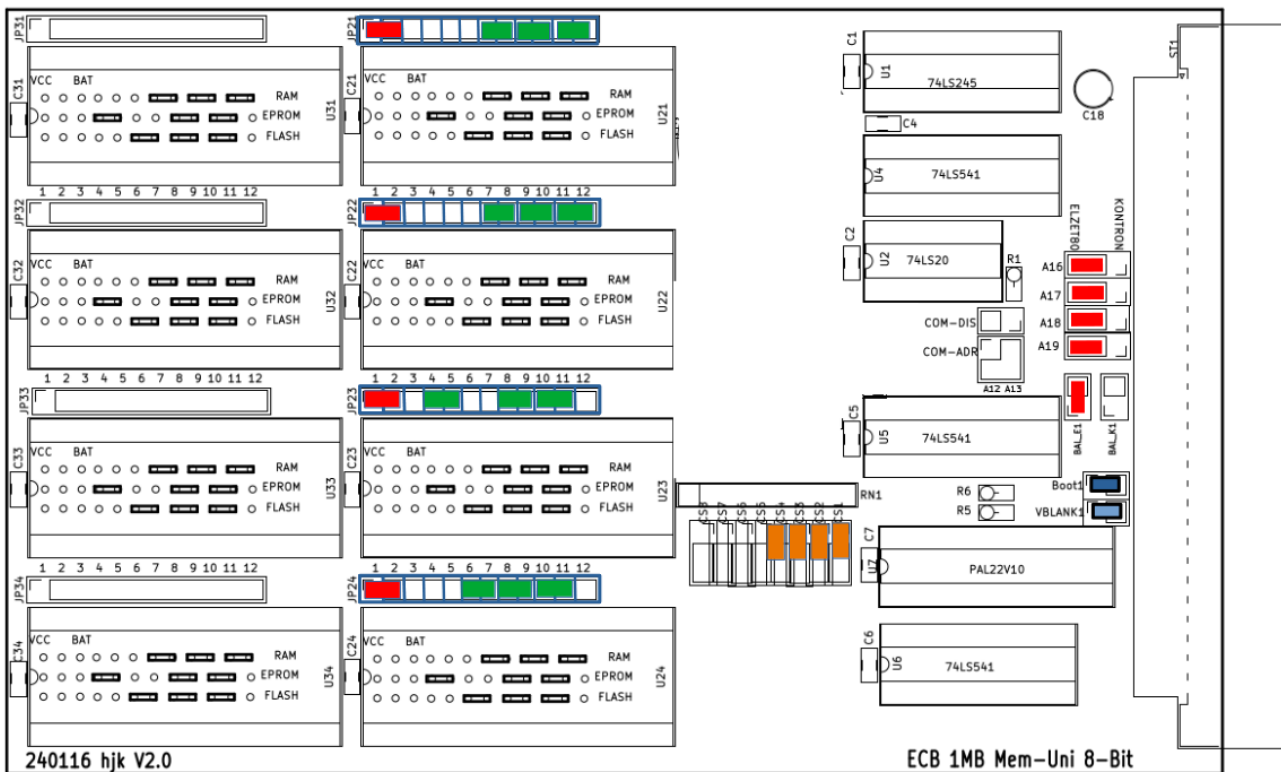


Bild 7: Beispiel Bestückung mit 4 Speichern ohne Akkupufferung

### Wichtig - Jumperstellung:

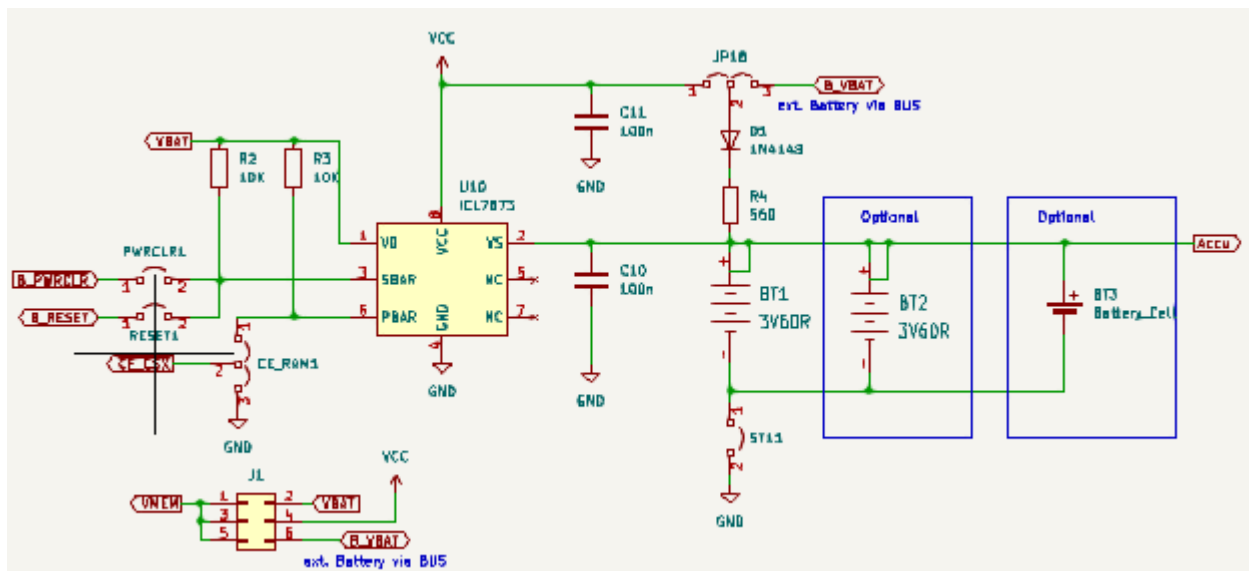
1. Der ChipSelect Jumper (Orange Jumper) verbindet PIN1 mit PIN2; obere Position
2. Bei der 12-poligen Jumperleiste direkt am IC muß Pin1 mit Pin 2 verbunden werden (roter Jumper); siehe Bild.

## 8. Akkupufferung – mögliche Optionen

Wie bereits erwähnt unterstützt die Platine die Akku/Batterie Pufferung von CMOS RAMs, d.h. auch nach Ausschalten der Versorgungsspannung bleiben die Speicherinhalte in den CMOS RAMs erhalten.

Eine Akkupufferung ist nur mit CMOS RAMs sinnvoll, da diese einen ausreichend geringen Stromverbrauch haben. Dadurch ist es möglich das auch nach Ausfall oder Abschalten der Versorgungsspannung die Speicherinhalte der CMOS RAMs erhalten bleiben. Der folgende Ausschnitt des Schaltbilds zeigt das die Speicherkarte einige Kombinationen zur Akkupufferung unterstützt.

Man kann zwischen einer externen Versorgung der RAMs und drei Onboard Varianten wählen; das sind zwei unterschiedliche Akku Bestückungen oder eine Batterievariante.



### Bild 8: Akkupuffer – Schaltung

Der Ausschnitt aus dem Schaltplan zeigt welche Bestückungs- und Jumper-Optionen sinnvoll sind.

Bei Akkupufferung der CMOS-RAMs muß der entsprechende Jumper für die Versorgungsspannung des Speicherbausteins (12-polige Jumperleiste, JP21..JP24, JP31..JP34) Pin2 mit Pin3 verbinden.

Zusätzlich muss der entsprechende ChipSelect Jumper nahe den ICs U8,U9 Pin2 mit Pin3 verbinden (Jumper in die untere, dem IC8 nahe Stellung).

Dadurch wird ein Schreibschutz wirksam, wenn die Versorgungsspannung zusammenbricht.

**Wichtig Jumperstellung für alle Varianten:**

1. Jumper J1 verbindet Pin1 mit Pin2 (VMEM – VBAT), Jumper links (Rot)
2. Jumper CE\_CSX verbindet Pin2 mit Pin3 (gelb)
3. Jumper Csx für Ux verbindet Pin2 mit Pin3 (orange Jumper)

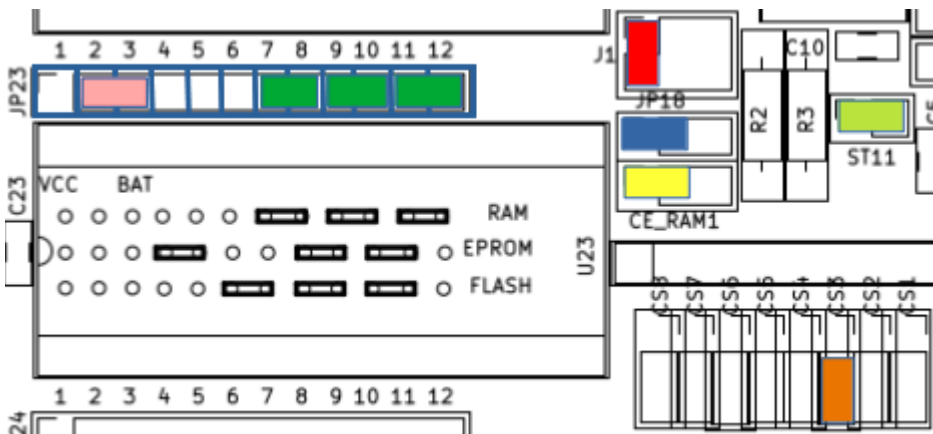


Bild 9: Jumperstellung am Beispiel von CMOS-RAM U23

## 8.1 Onboard Akkupufferung

**Wichtig – Jumperstellung:**

1. Jumper JP18 Pin1 – Pin2 (blau), siehe Bild 9
2. Jumper ST11 – schliessen (hellgrün)

Der auf der Karte befindliche Baustein U10 (ICL7673) überwacht die Akkuspannung. Er erzeugt an Pin3 (SBAR) ein LOW Signal wenn die Schwellspannung unterschritten wird.

Diese Signal kann man nutzen um den Prozessor bei einem Zusammenbruch der Versorgungsspannung in einen definierten Zustand zu bringen (RESET).

Bei einigen Implementierungen wird der RESET über Pin 31c des VG-Steckers an die CPU weitergeleitet und das Signal PWRCLR wird von dort an alle anderen Busteilnehmer weiter geleitet.

Hier ist also das **RESET (Pin 31c)** Signal ein **Open-Kollektor Eingang** und **PWRCLR (Pin 26c)** ein **Ausgang**.

Bei anderen Implementierungen ist **PWRCLR (Pin 26c)** ein **Eingang** um die CPU rückzusetzen.

Um beide Varianten abzudecken sollte man einen der beiden Jumper

- PWRCLR oder
- RESET

setzen, um den Speicher vor unkontrollierten Zugriff zu schützen. Siehe folgendes Bild

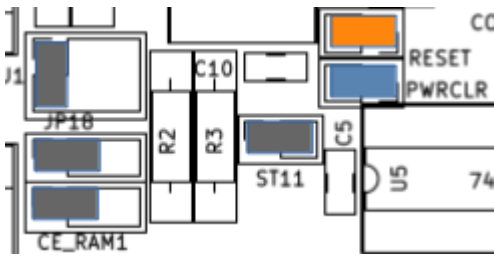


Bild 10: Reset Jumper

Wie beschrieben können die CMOS-Bausteine von den Akku auf der Karte versorgt werden. Die Datenhaltung ist aber nur für ca. 48 Stunden garantiert.

Will man die Daten auch bei einem längeren Spannungsausfall erhalten dann sollte die Spannung der CMOS-Bausteine über den Sonderanschluss VCMOS (Pin 24a ECBbus, im Schaltplan V\_BAT) versorgen.

Hier noch einmal ein Zusammenfassung der Funktion einiger Jumper.

### **JUMPER J1**

Pin1 – Pin2 VMEM – VBAT interne Akkupufferung, Standard Einstellung  
 Pin3 – Pin4 VMEM – VCC Akkupufferung wird ausser Kraft gesetzt (für Testzwecke)  
 Pin5 – Pin6 VMEM – B\_VBAT (VCMOS Pin 26a) externe Spannungsversorgung

VMEM ist mit Pin3 aller 12-poligen Steckerleisten der Speicherbausteine verbunden.

### **JUMPER ST11**

Mit dieser Steckbrücke kann der Akku aus dem Betrieb genommen werden, wenn die Karte nicht genutzt wird und nur gelagert werden soll.

Ist die Steckbrücke geschlossen, ist der Akku in Betrieb, sonst ist er abgeschaltet.

Ist der Akku vorhanden, muss für den Betrieb der Karte ST11 geschlossen sein !!

### **Jumper JP18**

Hier wird eingestellt, ob der Akku auf der Karte über die Leitung VCMOS (B\_VBAT, Pin 26a) oder von der Versorgungsspannung aufgeladen wird.

Pin1 – Pin2 aufladen des Akkus über VCC, Standard  
 Pin2 – Pin3 aufladen des Akkus über VCMOS

### **CE\_RAM**

Hat man alle Komponenten der Akkupuffer bestückt und will sie nicht nutzen, dann kann man die Spannungsüberwachung und damit das Steuern der Chipselect Signal durch setzen des CE\_RAM Jumper von Pin1 – Pin2 nach Pin2 – Pin3 ausser Kraft setzen.

## RESET / PWRCLR

Soll bei einem Zusammenbruch der Versorgungsspannung den Prozessor zurückgesetzt werden so sollte einer den beiden Jumper RESET oder PWCCLR gesetzt werden.

Wie beschrieben können die CMOS-Bausteine von den Akku auf der Karte versorgt werden. Die Datenhaltung ist aber nur für ca. 48 Stunden garantiert.

Will man die Daten auch bei einem längeren Spannungsausfall erhalten dann sollte die Spannung der CMOS-Bausteine über den Sonderanschluss VCMOS (Pin 24a ECBbus, im Schaltplan V\_BAT) versorgen.

## 8.2 Pufferung des Speichers durch externe Spannung

Die einfachste Lösung ist die externe Akkupufferung.

In diesem Fall wird verbindet man Jumper JP18 Pin2 mit Pin3. Jumper ST11 kann entfallen. Siehe folgendes Bild.

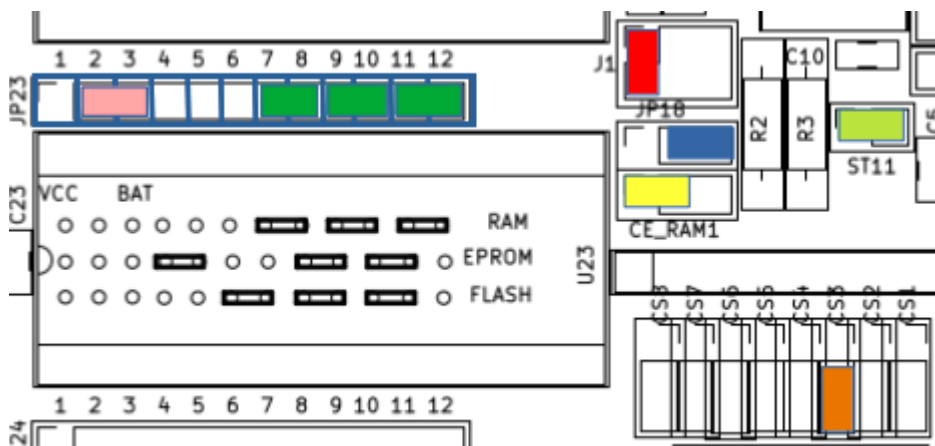


Bild 10: Jumperstellung für externe Spannungsversorgung

### Wichtig - Jumperstellung:

**Jumper JP18 Pin2 – Pin3 (blau)**

## 9. Bestückungsliste

Schaltplan Bezeichnung	Beschreibung	Wert	Raster	Stck	Lieferant	Best-Nr
<b>IC-Sockel</b>						
U21-U24, U31-U34	IC-Sockel 32-polig			8		
U1,U4,U5,U6	IC-Sockel 20-polig			4		
U5	IC-Sockel 24-polig Reihenabstand 7,62 mm			1		
<b>Bauelement</b>						
U1	IC	74HCT245		1		
U4, U5, U6	IC	74HCT541		3		
U5	PAL	22V10		1		
RN1	Widerstandsarray SIL9	8x10K		1		
JP21-JP24,JP31-JP34	Pfostenleiste 12-polig		2,5	8		
A16,A17,A18,A19	Pfostenleiste 3-polig		2,5	4		
BAL_E1, BAL_K1	Pfostenleiste 2-polig		2,5	2		
BOOT	Pfostenleiste 2-polig		2,5	1		
VBLANK	Pfostenleiste 2-polig		2,5	1		
CS1,CS2,CS3,CS4,CS5,CS6,CS7,CS8	Pfostenleiste 8-polig		2,5	3		
C18	Tantal Kondensator	33uF	2,5	1		
R1	Widerstand ¼ Watt	10K		1		
R5,R6	Widerstand ¼ Watt	10K		2		
C1, C4-C7	Vielschicht Kondensator	100nF	2,5	4		
CS21-C24,C31-C34	Vielschicht Kondensator	100nF	2,5	8		
<b>Optional – Common-Logik</b>						
U2	IC-Fassung 14-polig			1		
U2	IC	74HCT20		1		
R7,R8	Widerstand ¼ Watt	10K		2		
C2	Vielschicht Kondensator	100nF	2,5	1		
COM-DSI	Pfostenleiste 2-polig		2,5	1		
COM-ADR	Pfostenleiste 2x2 polig		2,5	1		
<b>Optional – Akkupuffe-rung</b>						
U8,U9	IC-Sockel 14-polig			2		
U10	IC-Sockel 8-polig			1		
U8,U9	IC	74HCT32		2		
U10	IC	ICL7673		1		
D1	IC	1N4148		1		
R2,R3	Widerstand ¼ Watt	10K		2		
R4	Widerstand ¼ Watt	560		1		



C8,C9,C10,C11	Vielschicht Kondensator	100nF	2,5	4		
J1	Pfostenleiste 3x2 polig		2,5	1		
JP18	Pfostenleiste 1x3 polig		2,5	1		
CE RAM	Pfostenleiste 1x3 polig		2,5	1		
ST11	Pfostenleiste 1x2 polig		2,5	1		
RESET	Pfostenleiste 1x2 polig		2,5	1		
PWCLR	Pfostenleiste 1x2 polig		2,5	1		
<b>Altern. Akku Bestückung</b>						
Bt1	Akku Varta 3/V80H 70mAh LxBxH 18 x 16 x 16			1	Reichelt	3GP-60
Bt2	Akku Varta 3/V150H 140mAh LxBxH 26.8 x 15.2 x 18.2			1	Reichelt	3/100 DKO
Bt3	Batterie Halter für CR2032		22	1	keystone	103

## 10. Schaltplan

Siehe externe Datei 1MBMemUin-8Bit.pdf

## 11. PAL Varianten

Siehe externe Datei 1MB-PALs.doc

# Anhang

## ECB-BUS Varianten

Der ECB Bus ist leider nicht standardisiert, d.h. es gibt unterschiedliche BUS (PIN) Belegungen. Die ursprüngliche Belegung wurde im Jahr 1977 von der Firma Kontron festgelegt und war auf den Z80 Prozessor zugeschnitten.

Die Steckkarten für den ECB Bus haben Europaformat (160mm x100mm) und nutzen einen einen zweireihigen Steckverbinder der Ausführung DIN 41612.

Die ursprüngliche Kontron-Belegung ließ 12 der 64 Signal-Pins unbelegt.

Diese wurde dann später auch belegt um zum einen eine DMA Daisy-Chain zu realisieren und den Adressbereich in einem ersten Ansatz 64K auf 1MB zu erweitern was 4 zusätzliche Adress-Signale erforderte.

In der Zwischenzeit hatten aber andere Anbieter ebenfalls Produkte für ECB-Bus angeboten. Sie hatten aber die freien Signalleitungen anders belegt. Somit gab es dann im wesentlichen 2 unterschiedliche Ausprägungen des ECB Bus;

Die Kontron Variante und die Elzet80 Variante

Hier die Unterschiede:

<del>Kontron</del>	<del>J&amp;K MKC</del>	<del>O+R</del>	<del>ELZET80</del>	<del>Systec</del>	<del>LAKOSA</del>	<del>MC</del>
A16	A16	A16				
/BAI	/BAI	/BAI	A18	A18	A18	A18
A17	A17	A17	A19	A19	A19	A19
A18	A18	A18				
A19	A19	A19	-12V	-12V	-12V	
-5V	-5V	-5V	-5V	-5V	-5V	-5V
2CLK	2CLK	2CLK	2CLK	2CLK	2CLK	2CLK
/BAQ	/BAQ	/BAQ	A17	A17	A17	A17
A21			A16	A16	A16	A16
A22						
A23	/RDY	/RDY	/BOOT	/BOOT	/BOOT	-12V
DPR	/PFL	/PFL	/BAI	/BAI	/BAI	
A20	/SEL		/VBLANK	/VBLANK	/VBLANK	
nCLK			/BAQ	/BAQ	/BAQ	
/WREN	/DESELECT	/DESELECT	/RDY	/RDY	/RDY	

## Erläuterung zur PAL Adressdekodierung

Im Folgenden soll verdeutlicht werden wie durch die PAL Adressdekodierung die einzelnen Speicherbausteine angesprochen werden.

Durch die Verwendung der Adressleitungen A12 ... A19 wird der Gesamtadressraum von 1MByte in 256 Speichersegmente von 4KB Größe aufgeteilt.

Eine Zeile in der PAL-Gleichung beschreibt einen Bereich eines Speicherbausteins der adressiert werden soll.

Beispiel für ein 128K RAM:

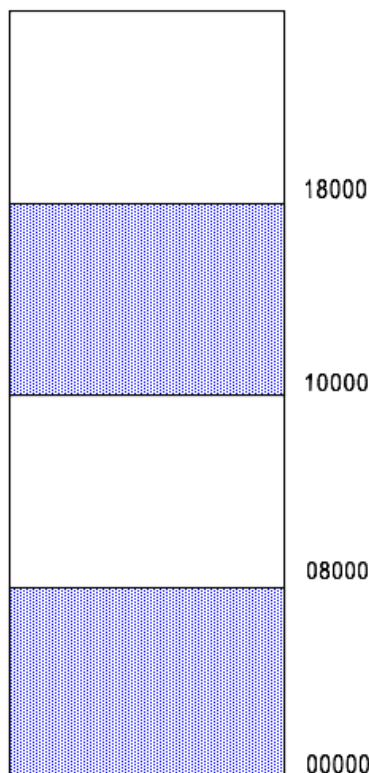
```
CS = /A19*/A18*/A17*  /A15*      /MREQ* BOOT
CS = A19*/A18*/A17*   A15*/A14*   /MREQ
CS = A19*/A18*/A17*   A15*A14*/A13 /MREQ
CS = A15*A14*A13*/A12  /MREQ *BLANK
CS = A15*A14*A13*A12   /MREQ
```

### Zeile 1

```
CS = /A19*/A18*/A17*  /A15*      /MREQ* BOOT
```

Die Gleichung adressiert zwei 32K Bereiche in dem Adressraum eines 128K RAMs (blaue Bereiche).

Das sieht wie folgt aus:



Da in der Gleichung A16 nicht dekodiert wird „taucht“ der 32K Bereich zweimal auf !

Unterschieden werden beide Bereiche durch A16:

A16 low - unterer 32K Block

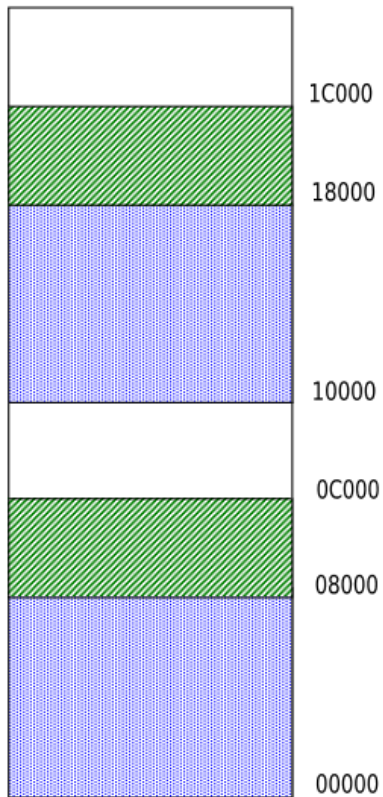
A16 high - oberer 32 Block

Dieser Gleichungsteil sorgt dafür das im BOOT Betrieb die unteren 32K im Normalbetrieb nutzbar sind und im BOOT Betrieb **ausgeblendet** werden.

## Zeile 2

CS = A19\*/A18\*/A17\*    A15\*/A14\*    /MREQ

Diese Zeile adressiert zwei 16K Bereiche im 128K Adressraum (x8000-xBFFF, grüne Bereiche).  
Das sieht wie folgt aus:



Auch hier wird A16 nicht dekodiert und die 16K Bereiche „tauchen“ zweimal auf.

### Beachte:

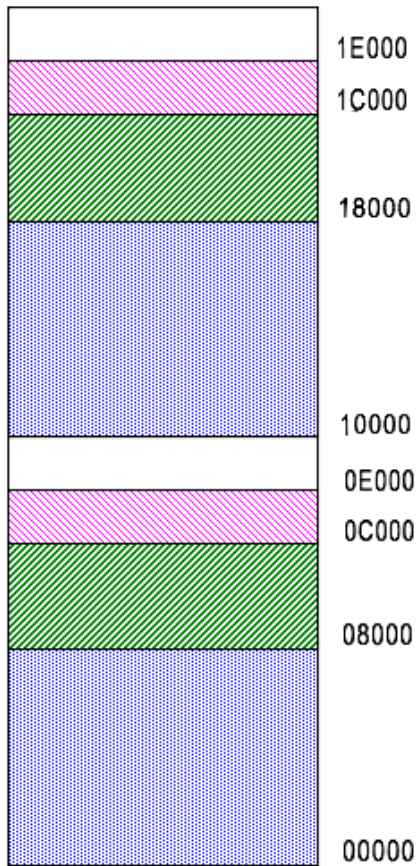
Boot wird **nicht** dekodiert, d.h. Dieser Bereich ist immer eingeblendet.

### Zeile 3

CS = A19\*/A18\*/A17\*    A15\*A14\*/A13    /MREQ

Diese Zeile adressiert zwei 8K Bereiche im 128K Adressraum (xC000 – xDFFF, lila Bereiche)

Das sieht wie folgt aus:



#### Zeile 4

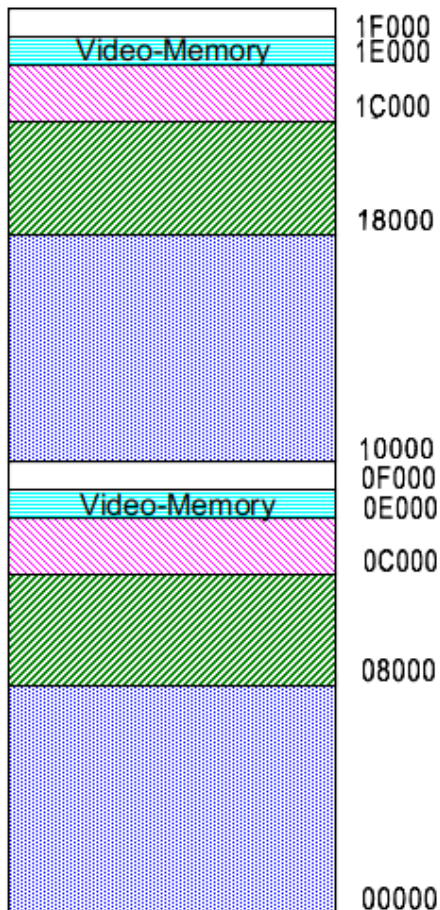
CS = A15\*A14\*A13\*/A12 /MREQ \*BLANK

Diese Zeile adressiert zwei 4K Bereiche im 128K Adressraum (xE000-xEFFF, türkise Bereiche)

Das ist bei Elzet80 der Bereich in dem der Video-Speicherbereich eingeblendet wird.

Ist das BLANK Signal low, dann wird dieser Speicherbereich ausgeblendet und man hat Zugriff auf den Speicher der Video Karte.

Das sieht wie folgt aus:



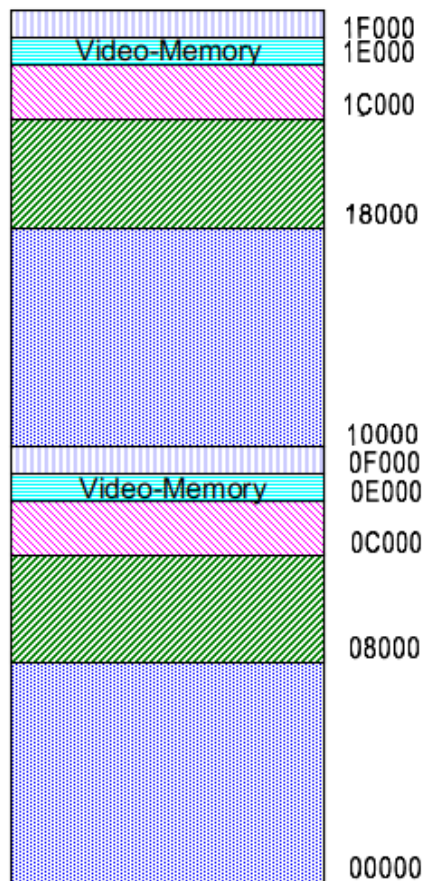
Die oberen 4 Adressbits (A16-A19) werden nicht dekodiert, d.h. Dieser Bereich wird im COMMON-Bereich eingeblendet !!!

## Zeile 5

CS = A15\*A14\*A13\*A12 /MREQ

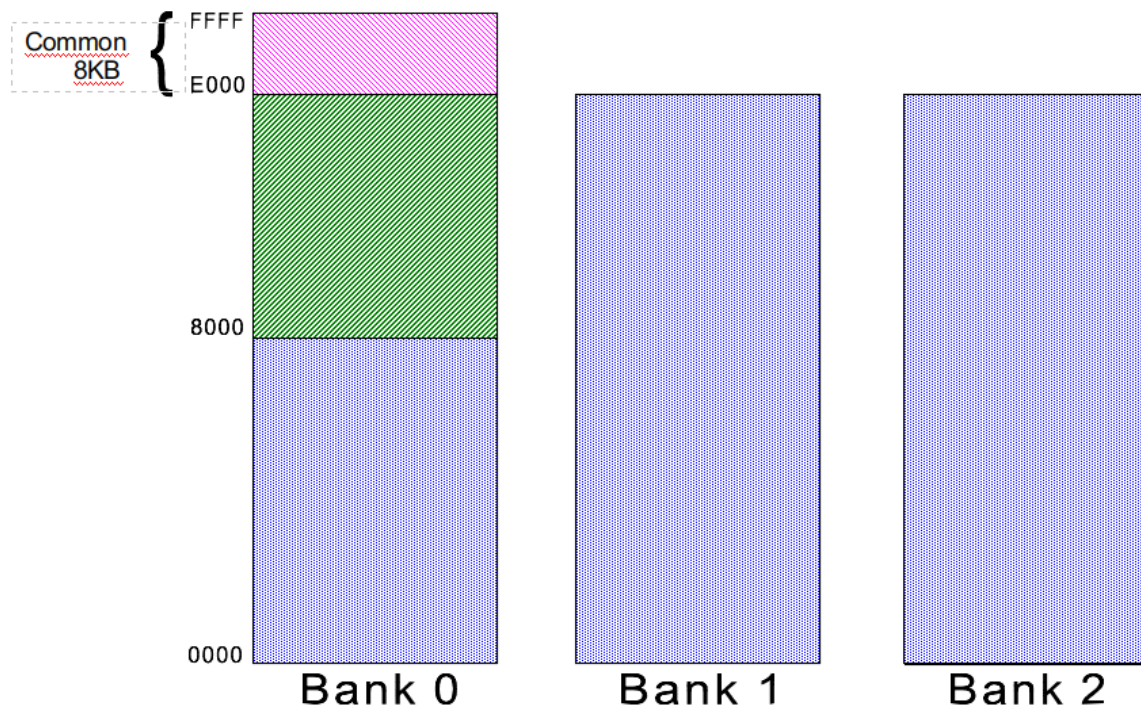
Diese Zeile adressiert zwei 4K Bereiche im 128K Adressraum (xF000-xFFFF).

Das sieht wie folgt aus:



Auch hier werden die oberen 4 Adressbits (A16-A19) werden nicht dekodiert, d.h. Dieser Bereich wird ebenfalls im COMMON-Bereich eingeblendet!!!





Zur Verdeutlichung noch einmal ein Bild für 3 Bänke mit einem 8Kb Common-Bereich