

CBM 1750/1764

RAM Expansion C 128

Schematics	Seite 2-7
Erweiterung der 1764 ab	Seite 8
Erweiterung der 1750 ab	Seite 12
Schematic der Erweiterung	Seite 16

SERVICE SCHEMATICS

1750/1764

RAM EXPANSION

MARCH, 1988

SCHEMATIC #312534
& 311752

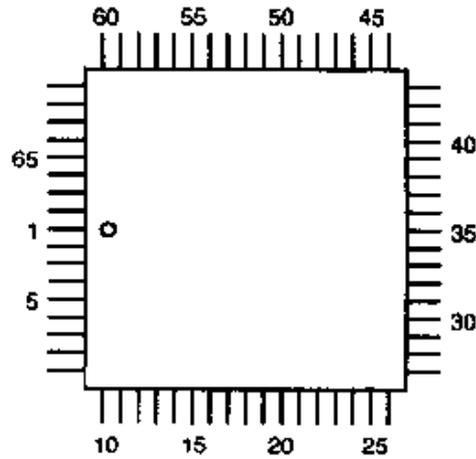
Commodore Business Machines, Inc.
1200 Wilson Drive, West Chester, Pennsylvania 19380 U.S.A.

Commodore makes no expressed or implied warranties with regard to the information contained herein. The information is made available solely on an as is basis, and the entire risk as to quality and accuracy is with the user. Commodore shall not be liable for any consequential or incidental damages in connection with the use of the information contained herein. The listing of any available replacement part herein does not constitute in any case a recommendation, warranty or guaranty as to quality or suitability of such replacement part. Reproduction or use without expressed permission, of editorial or pictorial content, in any matter is prohibited.

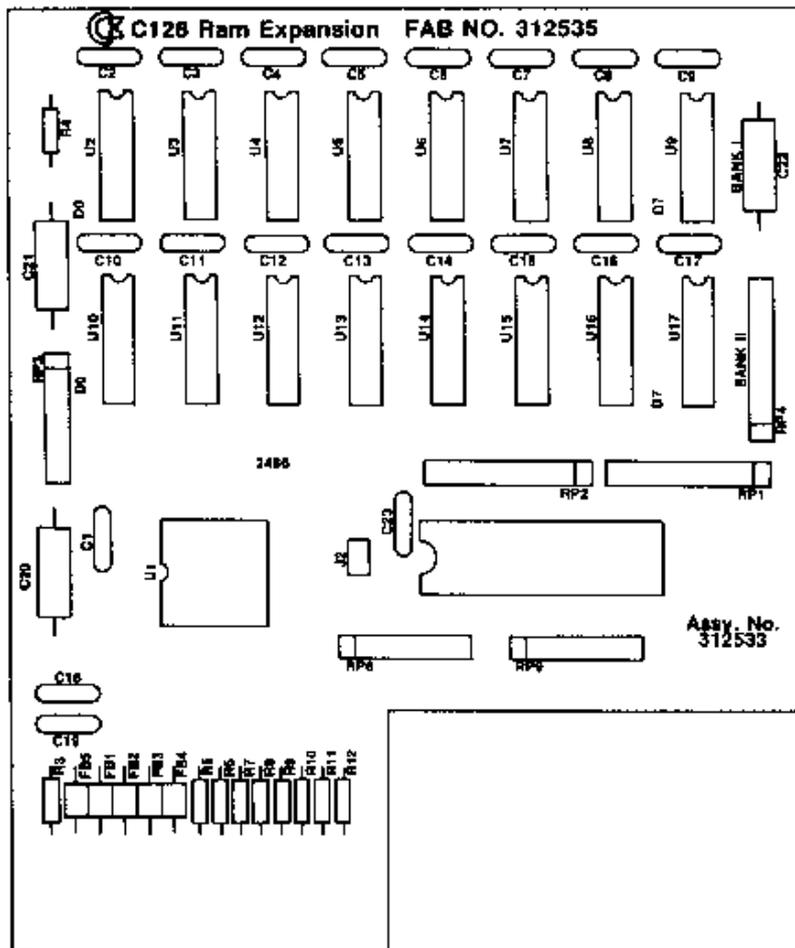
This manual contains copyrighted and proprietary information. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of Commodore Electronics Limited.

Copyright © 1988 by Commodore Electronics Limited.
All rights reserved.

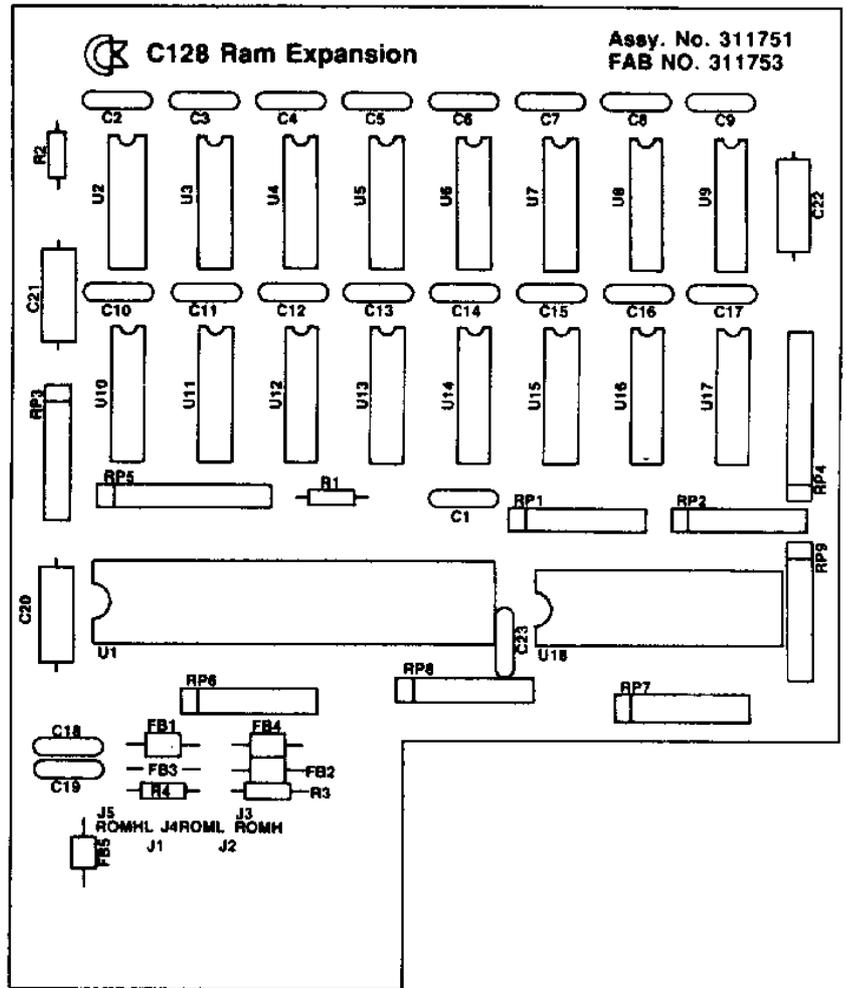
315089-02 8726 QUAD - PACK VERSION



pin 1 — /RESET	pin 18 — VSS	pin 35 — /ROMH	pin 52 — VSS
pin 2 — /IRQ	pin 19 — A15	pin 36 — /ROML	pin 53 — DD7
pin 3 — DOTCLK	pin 20 — A14	pin 37 — /ROMSEL	pin 54 — DD6
pin 4 — R/W	pin 21 — A13	pin 38 — VCC	pin 55 — DD5
pin 5 — 1MHZ	pin 22 — A12	pin 39 — VSS	pin 56 — DD4
pin 6 — /CS	pin 23 — A11	pin 40 — TEST	pin 57 — DD3
pin 7 — /BA	pin 24 — A10	pin 41 — N.C.	pin 58 — DD2
pin 8 — /DMA	pin 25 — A9	pin 42 — MA0	pin 59 — DD1
pin 9 — D7	pin 26 — A8	pin 43 — MA1	pin 60 — DD0
pin 10 — D6	pin 27 — A7	pin 44 — MA2	pin 61 — /DWE
pin 11 — D5	pin 28 — A6	pin 45 — MA3	pin 62 — /RAS0
pin 12 — D4	pin 29 — A5	pin 46 — MA4	pin 63 — /RAS1
pin 13 — D3	pin 30 — A4	pin 47 — MA5	pin 64 — /CAS0
pin 14 — D2	pin 31 — A3	pin 48 — MA6	pin 65 — /CAS1
pin 15 — D1	pin 32 — A2	pin 49 — MA7	pin 66 — BS
pin 16 — D0	pin 33 — A1	pin 50 — MA8	pin 67 — VCC
pin 17 — N.C.	pin 34 — A0	pin 51 — N.C.	pin 68 — N.C.

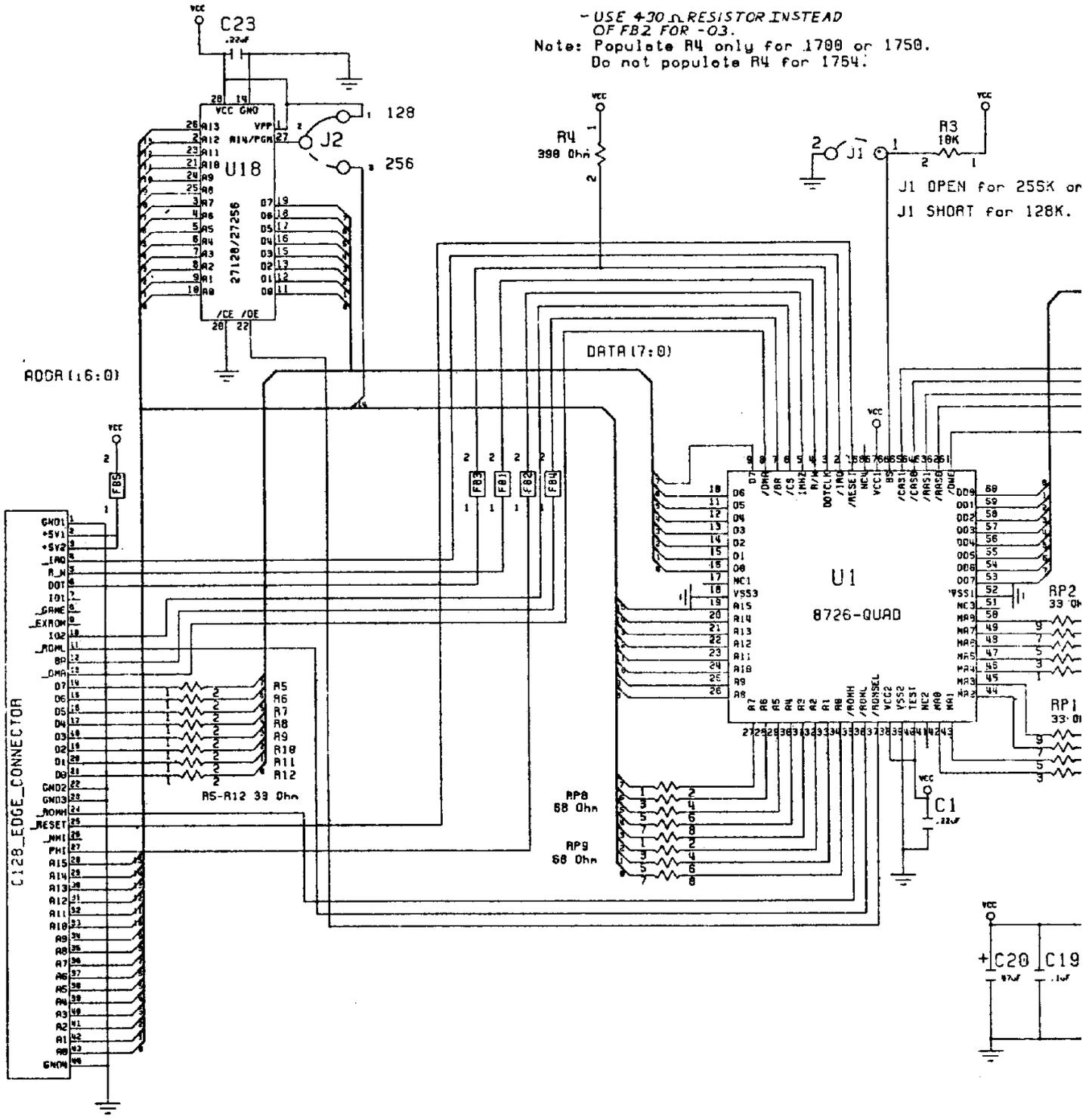


/RESET	1	64	VCC
/IRQ	2	63	BS
DOTCLK	3	62	CAS1
R/W	4	61	CAS0
1MHz	5	60	RAS1
/CS	6	59	RAS0
/BA	7	58	/DWE
/DMA	8	57	DD0
D7	9	56	DD1
D6	10	55	DD2
D5	11	54	DD3
D4	12	53	DD4
D3	13	52	DD5
D2	14	51	DD6
D1	15	50	DD7
D0	16	49	VSS
VSS	17	48	MA8
A15	18	47	MA7
A14	19	46	MA6
A13	20	45	MA5
A12	21	44	MA4
A11	22	43	MA3
A10	23	42	MA2
A9	24	41	MA1
A8	25	40	MA0
A7	26	39	TEST
A6	27	38	VSS
A5	28	37	VCC
A4	29	36	/ROMSEL
A3	30	35	/ROML
A2	31	34	/ROMH
A1	32	33	AO

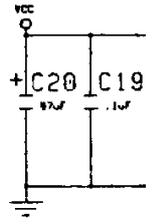


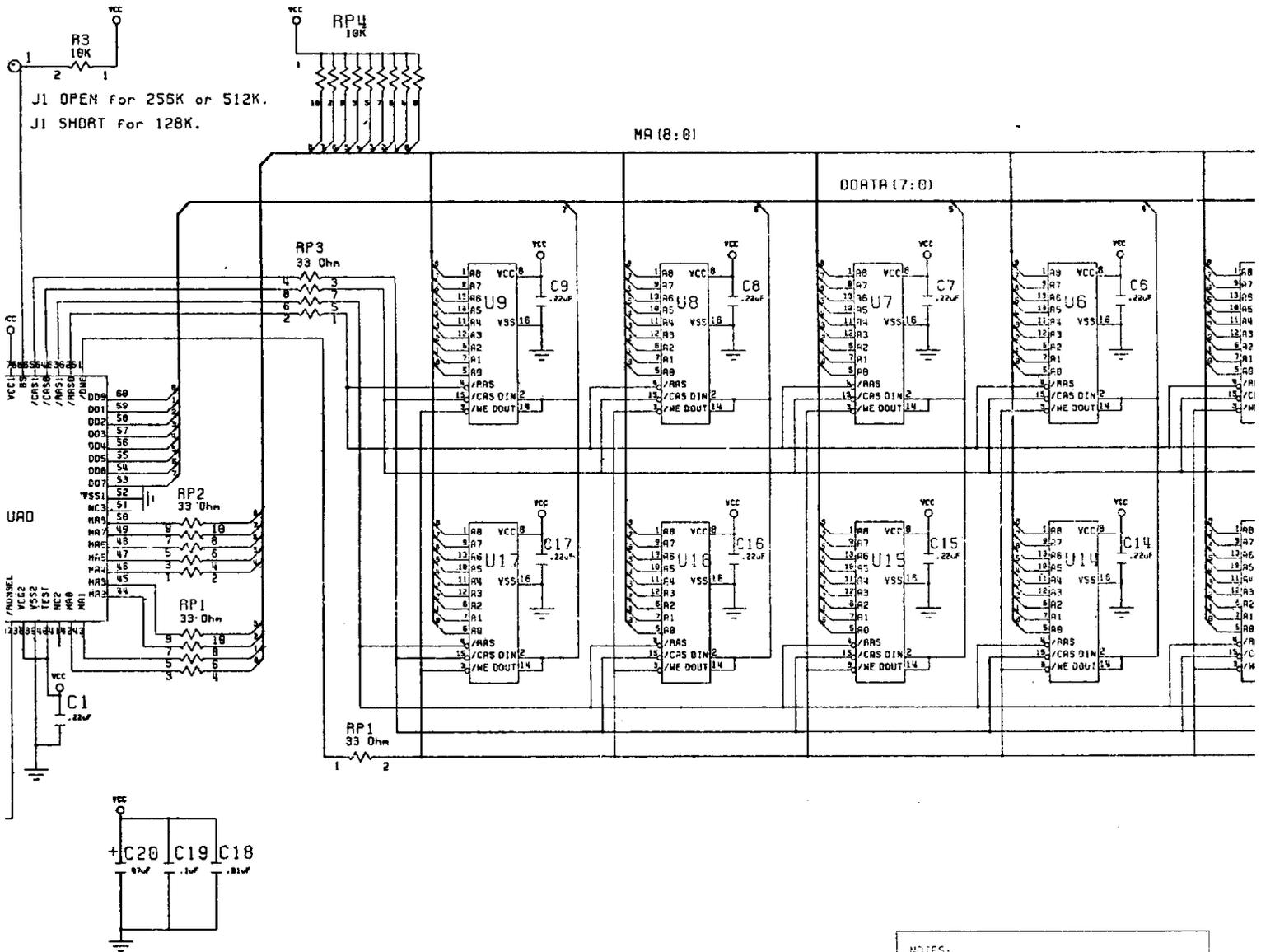
A15-A0	These lines are bidirectional and buffered. They are used by the 8726 to access C128 memory and to detect address FF00. A0-A4 are used by the host processor to select the internal registers.	/IRQ	Output, active low. This line will go low if interrupts are enabled and any of the interrupt conditions have occurred.
D7-D0	Data lines used by the controller to output or input data from C128 memory. These lines are bidirectional and buffered. The host also uses these lines to access control registers.	MA8-MA0	Multiplexed address bus output to DRAMs. Includes full 256K addressing for use with 256K x 1 DRAMs.
R/W	This line is also bidirectional and buffered. It is used by the 8726 to perform either a read or write operation on system memory. The host uses this line to read or write to the controller registers. In both cases, a low on this line signifies a write operation and a high signifies a read.	DD0-DD7	Bidirectional data lines used by the controller to access expansion memory.
1MHz	This input is the system clock and is used to synchronize all data transfers.	/DWE	Output, active low. When low, this line signifies that the controller is writing to expansion memory. This line is high during a read cycle.
/DMA	Output, active low. This signal is used by the 8726 to take control of the system buses prior to initiating a transfer sequence.	/RAS0	Output, active low. This line is used by the controller to access or refresh RAM BANK 0.
/BA	Input, active low. The 8726 monitors this signal to determine whether or not it can use the bus. On the C128, VIC generates this signal.	/RAS1	Output, active low. This line has the same function as /RAS0 except that it controls RAM BANK 1.
DOT CLOCK	This input is used to derive DRAM control signals.	/CAS0	Output, active low. Column address strobe for expansion bank 0.
/CS	Input, active low. This signal selects the 8726. This line must be brought low in order to access any of the internal registers.	/CAS1	Output, active low. Column address strobe for expansion bank 1.
RESET	Input, active low. A low on this line disables the controller and resets all internal registers. DRAM refresh is not interrupted during a reset.	BS	Bank Select. This input allows configuration of the controller for either of the two DRAM combinations.
		/ROMH	Input, active low.
		/ROML	Input, active low.
		/ROMSEL	Output, active low. This output is equal to the anding of /ROMH and /ROML.
		TEST	This input is for test purposes and has an internal pullup so that it may be left unconnected during normal operation.
		VCC	+5 volt supply.
		VSS	Ground.

- USE 430 Ω RESISTOR INSTEAD OF FB2 FOR -03.
 Note: Populate R4 only for 1700 or 1750.
 Do not populate R4 for 1754.



J1 OPEN for 255K or
 J1 SHORT for 128K.





- NOTES:
1. Associate 312534-01 with model 1709. 128K C128 Ram Expansion.
 2. Associate 312534-02 with model 1750. 512K C128 Ram Expansion.
 3. Associate 312534-03 with model 1764. 256K C64 Ram Expansion.
 4. J1 must be OPEN for 512K or 256K. J1 must be SHORTEDED for 128K (default).
 5. Populate R4 for C128 Ram Expansion. Do not populate R4 for C64 Ram Exp.
 6. For 256K only populate U2-U9.

UPGRATING 1764

NOTE: The following article also applies to expanding C1700 128K REU's to 512K with the following changes:

1. You will need to remove the 64x1 DRAMS, and replace them with 16 256x1 DRAMS.
2. You will need to cut the "J1", aka "CUT 512K" jumper.

EXPANDING YOUR 1764 RAM EXPANSION UNIT FROM 256 TO 512 KILOBYTES OF MEMORY

by Scott A. Boydman (ScottB30 on QuantumLink)

This article details the steps required to increase the 1764 RAM Expansion Unit (REU) to a total of 512K. Although the article has been tailored primarily for the GEOS 64 user, it is also applicable to all 1764 REU users who wish a 512K expansion. I have had a great deal of correspondence with people, both by mail and over Quantum-Link, and I would like to share some of the knowledge gained from these rewarding interactions. You should note that since the Japanese "chip embargo" the prices have been inflated and availability is scarce. One of the best sources of chips may be found in the various advertisements in the "COMPUTER SHOPPER" magazine.

The 1764 REU consists of a plug-in cartridge, a deluxe power supply, and a disk containing software for using the REU with both BASIC (side 1) and GEOS (side 2.) The deluxe power supply is required with the C-64, since the original was not designed to meet the power requirements of the many chips present in the REU. The deluxe power supply is rated for 2.5 amps DC, which is sufficient to meet the power requirements of the computer, a modem, and an REU with up to 512K RAM. The C-128 power supply already has enough amps to handle any REU. The REU itself is nonfunctional without appropriate software. The front side of the disk contains Commodore's RAMDOS software for use with BASIC. The back side of the disk contains a GEOS upgrade which installs the GEOS Kernal V1.3 on your system boot disk, as well as other upgraded applications and input/printer drivers.

The latest release of GEOS (V2.0) is a two-disk system and already supports the REU. If you have Kernal V1.2 or lower, you will have to upgrade by one of two methods. The first is to run the GEOS Upgrade program on the back of the disk supplied with the REU. This will upgrade your GEOS System disk to Version 1.3. However, this technique does not always work properly, so many people have upgraded directly through Berkeley Softworks. The newest version of GEOS (V2.0) supports the REU as well as Commodore's new 1581 3.5 inch disk drive. Note that the GEOS and BASIC RAMDISKS are not compatible, therefore, GEOS RAMDISK files are not accessible from BASIC. The original REU test program on the BASIC side of the Test/Demo disk was designed to test the 256K of random access memory that is supplied in the original 1764, but Commodore has since released newer software (available for downloading over Quantum-Link, or directly from Commodore) that also supports the C-128, and any size REU from 128K to 512K.

The 1764 REU itself consists of a printed circuit (PC) board inside a plastic and metal housing, which plugs into the expansion port of the C-64 or 64-C computers. The PC board is the same as that used in the 1700 (128K) and 1750 (512K) REU's, intended for the C-128, but with the following changes:

1. The 1764 has only one bank of eight 256K Dynamic RAM chips installed. Each chip holds 256K bits of information; therefore eight chips are required to obtain 256K bytes (since eight bits make up one byte).
2. There is an empty row of eight chip positions labeled BANK II on the PC board. The traces for additional memory chips are present, and the holes for the pins on the chips are presoldered.
3. There is a 390 Ohm resistor in the 1700 and 1750 REU's that is not present in the 1764. The resistor's position is located in the upper-left corner of the board, and labelled R4 in the 1764. According to Fred Bowen of Commodore, this resistor "tweaks" the signal that is supplied to the C-128 computer. There are C-64 users that have installed unmodified 1750 REU's without reporting

problems. In these installations, larger power supplies rated at 2.5 amps DC are used. Also, there are C-128 owners who have used 1764 REU's without difficulty. Therefore, the presence or absence of this resistor seems to be a non-critical factor for most C-64 or C-128 computers, at least when using the REU with GEOS.

4. A jumper, labelled with the notations J1 and CUT 512K is located on the bottom side of the circuit board. The jumper is cut in the 1750 and 1764 REU's, but left intact in the 1700 REU. Apparently, the cut jumper tells the Ram Expansion Controller (REC) that 256K RAM chips are being used, rather than the 64K RAM chips used in the 1700.

Commodore's BASIC RAMDOS software (included with the 1764) is not compatible with many commercial software packages. GEOS, however, allows the expansion to operate as a fully functional disk drive in RAM. The current versions of both GEOS 64 and GEOS 128 have provisions for both 1541 and 1571 RAMDISKS. Another available feature is the shadow mode. GEOS is able to store data to the REU when reading files from the physical 1541 disk drive, so that further references of that file occur directly from RAM. This results in a significant speed-up by eliminating multiple disk reads of the same file. All writes go to both the physical drive and the REU, which eliminates the RAMDISK's primary disadvantage of being a nonpermanent storage device. In the shadow mode, your data is always current and up-to-date on your real disk. Of course, when using the RAMDISK, files must be copied to a physical disk, or be lost when the computer is turned off.

The 256K REU does not contain the amount of memory needed to shadow two physical 1541 drives, or shadow one drive while supporting the RAM 1541, or support a RAM 1571. For this reason many people have desired their 1764's to have 512K RAM. It is not very difficult for someone with basic mechanical and soldering skills to install additional chips on the 1764 PC board. I recommend soldering 16-pin DIP (Dual Inline Package) sockets to the board, which prevents possible heat damage from direct soldering of the memory chips. The chips are simply pushed carefully into the sockets. Radio Shack stocks the 16-pin DIP sockets, and the chips may be available from mail order houses.

NOTE: The steps required to install the chips should be undertaken by someone with some skill in handling and soldering PC boards. Also, some mechanical skill is needed to remove and install the PC board in its housing. Opening the REU housing will void Commodore's warranty and is done entirely at the owner's risk.

1. The housing of the 1764 must be opened carefully. There are four pins, one in each corner, holding the plastic case together with friction. Starting at the expansion port opening, gently but firmly pull the top and bottom halves apart. The plastic case will begin to separate. Continue around the perimeter of the case until the top can be removed from the bottom.
2. Remove the metal shield from the plastic casing by pulling it out at each corner. The PC board is inside the metal shield. Open the shield, also starting at the connector. You may need a small screwdriver to pry the metal slightly where it is held together. When the shield opens, spread it only as much as needed to remove the PC board. Then slide the cardboard liner off from the expansion connector. This metal shield serves as a barrier to radio frequency (RF) interference as required by the FCC. When testing your expansion, the PC board can be safely plugged into the computer prior to final reassembly of the REU.
3. Examine the PC board. BANK I is the row of eight chips along the top of the board. The RAM Expansion Controller (REC) is the square chip in the lower portion of the board. The empty area labelled BANK II can be seen just below BANK I.
4. Solder eight DIP sockets into the locations in BANK II. This is the most difficult step, because it is hard to position all 16 pins at once. I recommend that a small, round toothpick be used to open the 16 holes on the PC board for each socket position. Using a soldering iron, heat the hole position until the toothpick can be inserted. By rotating the toothpick during cooling, a hole will be left upon its removal. An alternative method would be to use a desoldering tool or wick (available from Radio Shack) to remove the solder from the holes. Note however that removal of ALL the solder can make it difficult to achieve good contact when soldering the DIP sockets to the board. After doing this for all 16 holes, the DIP socket can be inserted into the holes, and then each pin soldered individually. Use care that each solder point makes a secure connection on both sides of the PC board. When opening the holes, be certain the metal traces printed on the circuit board are not disrupted, otherwise your REU may not function at all when tested.

5. Insert the chips carefully into the sockets, checking to maintain the correct chip orientation. The notch or dot on the chip points toward the top of the PC board. Be certain that all pins are inserted into the socket, and that none bend or break.
6. Before putting the REU together again, plug the PC board into the computer (make sure the power is off), turn the computer on, and boot GEOS V1.3 or higher. If necessary, run the UPGRADE conversion program on the disk that came with the 1764 REU, or 1351 MOUSE to convert your V1.2 to V1.3, but be sure to keep a V1.2 backup copy should anything go wrong. If you have a two-disk GEOS system, you already have a compatible version. From the deskTop, run the CONFIGURE program. The RAM EXPANSION box that appears should indicate 512K if your work was successful. If this is the case, reassemble the REU. If 256K or NONE is displayed, you should check the board for improperly soldered or disrupted traces, a chip that is not securely plugged in, or rarely a defective chip. Recheck all the connections systematically. You may have to desolder and remove one or more sockets to locate a bad connection point or points. Then rerun CONFIGURE to see if the problem has been corrected. If you are not a GEOS user, use Commodore's RAMTEST program to check out your expanded REU (see below.)

When you arrive at this point, CONGRATULATIONS! GEOS makes use of the 512K REU in the following ways; the first 64K holds a copy of the entire C-64 RAM, which allows for fast memory access and rebooting GEOS from the REU. If you select a RAM 1541, 165K plus directory space are reserved. This amount is also allocated if you are shadowing a real 1541 drive. GEOS V2.0 also allows for a 1571 RAMDISK. The BASIC RAMDISK software included with the REU also uses the expanded amount of REU memory, but the original version of Commodore's RAMTEST program was only designed for 256K. If you have this version, it can be modified to check the entire 512K in the expanded 1764 as follows:

1. Copy the files 1764 RAMTEST.BAS and 1764 RAMTEST.BIN from the demonstration disk onto another formatted disk. If the files have different names than those noted above, then you may not have to alter the test program (your disk may be an updated version of Commodore's RAMDOS). You may use the FILE COPY utility included on side 1 of the disk to do this, or even copy the files from the GEOS deskTop.
2. From BASIC, type the following: **NEW (RETURN)**
LOAD "1764 RAMTEST.BAS",8 (RETURN)
501 REM LINES 502 & 503 POKE THE CODES TO TEST 512K! (RETURN)
502 POKE 33093,7: POKE 33099,33 (RETURN)
503 POKE 36003,53: POKE 36004,49: POKE 36005,50 (RETURN)
3. List lines 501-503 to confirm the changes have been entered. Then, list line 100 and space over the beginning characters 100 REM (to erase them) and press . This will resave the altered version of 1764 RAMTEST.BAS to the disk, and delete the original version. Reload 1764 RAMTEST.BAS and relist lines 500-503, rechecking for errors. Then, run 1764 RAMTEST.BAS, and the program should test all 512K in the expansion. If this does not happen, recheck the poke codes in lines 502 and 503 for errors.

You should be aware of some bugs in the original Commodore RAMDISK software:

1. The DOS NEW command does not seem to work (a syntax error is returned). However, the SCRATCH command using the wildcard symbol (S0:"*") will do the same job. This bug has been corrected in subsequent revisions of the Commodore RAMDOS. In fact, the RAMDOS has now been upgraded for use in C-128 mode. The latest revisions of RAMDOS are version 4.2 for C-64 mode, and version 4.3 for C-128 mode.
2. With the 256K REU, the BLOCKS FREE message in the original RAMDOS directory reports an incorrect number (2014) of blocks, as if the expansion contained 512K. This number remains about the same after the 512K is installed. This bug has been corrected in the newest releases of RAMDOS.
3. In version 4.2 and 4.3 of RAMDOS (downloaded from Quantum-Link), the unit (or device) number of the RAMDISK cannot be changed with the CHANGE UNIT program. The only way to change unit numbers (i.e. from 9 to 8) is to reinstall RAMDOS specifying the new unit number, but without reinitializing the REU memory (in other words, files in the REU memory are retained).

The operation of the 1764 with GEOS seems flawless, although I have heard of occasional incompatibilities with user-programmed software. This, however, is a very rare case. It would be nice to be able to exit GEOS and directly utilize the GEOS RAM 1541 files in the BASIC environment; but due to the differences between the GEOS and Commodore RAMDOS formats, this isn't possible.

In the area of troubleshooting, the majority of people who have upgraded their 1764's have done so without problems. Of the people who have experienced difficulty, the usual defect has been poor connections at the soldering points, or disruption of the PC board traces occurring at the time the presoldered holes were opened. In most cases, a careful inspection and reheating of the solder points will reestablish the proper connection(s). Use of a continuity tester (available from Radio Shack) may be helpful in isolating any bad connections. In the case of an extremely hard-to-identify problem, it may be best to desolder and remove the sockets and reinspect the PC board, checking for disruptions on both sides of the board. A careful, meticulous job of soldering is the best way to avoid trouble.

Regarding the compatability of the 1764 with the C-128, in most cases, no alteration seems required. However, the addition of a 390 Ohm resistor to the empty R4 position of the 1764 circuit board makes it virtually identical to the 1750 REU. Conversely, removal of this resistor makes the 1750 equivalent to the 1764.

I hope this information will be helpful to all 1764 owners who desire to have 512K RAM rather than the 256K supplied by the factory. If you need assistance, feel free to contact:

Scott A. Boydman, 25447 Bryden Road, Beachwood, Ohio 44122
or send E-Mail to ScottB30 on Quantum-Link.

[Marko Mäkelä](mailto:Marko.Makela@HUT.FI) (*Marko.Makela@HUT.FI*)

UPGRATING 1764

Note: To build this expansion, you need a 512 kilobyte REU. If you only have 128 or 256 kilobytes on your REU, **you must upgrade your REU to 512kB first.**

From: Artie.Stevens@p2.f158.n633.z3.fidonet.org (Artie Stevens)

Newsgroups: comp.sys.cbm

Subject: 1750 Expansion

Date: Sun, 23 May 1993 07:50:00

Thanks goes out to STAN LOGAN for sending the text-converted file of the geowrite-file in which the original article was written.

The material contained in this document is of an experimental nature. As such, all consequences resulting from the use of the information is the express responsibility of the user. The author shall not be held liable, and implies no warranties or guarantees what so ever. The author does certify that the information presented was correct, to his knowledge, at the time of the first distribution of the document. This document and the information it contains is the sole property of the author, who retains all copyrights, and patents to the material presented. The author does grant a limited distribution right to all if the following conditions are satisfied: The document is distributed in complete and unaltered form, with this legal form included. No commercial distribution (i.e.. publishing) of the document is entered into without the written consent of the author. Lastly, no monies shall be exchanged for the distribution of the document, except for reproduction and shipping charges, without the written permission of the author. With these conditions fulfilled, the limited distribution license grants the holder of the document to otherwise freely distribute, and store it in any manner.

Beyond 512 kb: The Two Megabyte REU

By Andrew E. Mileski, Copyright October 3, 1989.

In Volume 9, Issue 6 of the Transactor, Paul Bosacki showed us a miracle; the one megabyte C64. Paul had developed an elegant method of allowing the C64 to access this extra memory out of the C64's own memory map. Unfortunately, this extra RAM is rather difficult to access, not all of it is available to the user, and C128 owners cannot expand their machines in the same manner. The circuit necessary for this feat allows a C64 to use 256 k RAM chips, instead of the 64 k RAM chips it was designed to use. Although a brilliantly simple circuit, it is over-kill when one already has a Ram Expansion Unit (REU) that directly supports 256 k RAM chips! Expanding a REU is relatively simple, and if you already know how to program a REU you can easily take advantage of an expanded one. In fact, all your software that uses a REU is completely compatible! Now are fellow C128 owners can expand their machines easily too, since the extra RAM is accessed out of a REU, and not off a modified mother-board! In a 512 kb REU, there are two dynamic RAM chip banks on the circuit board. They each consist of eight, 1 bit by 256 k dynamic RAM chips, which gives us our 512 k bytes. The REU's internal bank register at \$DF06 works in 64 kb increments, and has only eight of these REU banks. Banks 0 to 3 are accessed out of chip bank one (U2 to U9), and banks 4 to 7 are accessed out of chip bank two (U10 to U17). Please note the distinction between chip banks, and REU banks. Note: signals preceded by an asterisk indicate that they are active low Dynamic RAM Basics.

Dynamic RAM memory chips are constructed with multiplexed address inputs. This means that they carry different information at different times. To access a particular memory cell, we first supply the chip with half of the actual address, a row address. Next we latch this address into the chip by asserting the Row Address Strobe (*RAS). Now we supply the chip with the other half of the address, the column address. Once again we latch this address into the chip by asserting the Column Address Strobe (*CAS). Depending on the state of the Write Enable (*WE) signal, a read or write memory cycle will occur. Unfortunately dynamic RAMs forget everything unless they are reminded, or refreshed. This must be done about every 4 milliseconds! To refresh the memory a RAS only cycle must be done for every row address. This means that only the row address is latched into the chip; the column address is not needed here. The entire row of data stored in the chip will be refreshed, and our data will be safe for another few milliseconds. In a REU the RAM Expansion Controller (REC) chip takes care of all this for us.

Fooling the REU

Since a REU was only designed to access 512 kb of expansion memory, we must fool it in order to access more. We do this by letting the REU think that it has only 512 kb available at any one time. This means we need some way to switch between the extra banks of memory. The circuit shown in the schematic helps us to do this by adding two bits to the REU bank select register. This two bit output port lets us select one of four banks of 512 kb. So in other words, are REU now thinks it is four separate 512 kb REUs. Now we can easily access up to 2 Mb out of a REU!

How the circuit works

The secret to expanding dynamic memory lies in the *CAS signal. Since all the RAM chips need to be refreshed with *RAS, we don't do anything to this signal and pass it to all memory chips. This leaves us with manipulating *CAS. Manipulation of *CAS is the job of IC1, a dual two to four line decoder/demultiplexor. It is used to direct the REU signals *CAS0 (chip bank one), and *CAS1 (chip bank two), to the correct bank of eight RAM chips. When one of the REU CAS signals is asserted, the CAS signal of the selected bank is asserted. The bank selection is done with the two select inputs S0 and S1 of IC1. IC2 and IC3 form a two bit write-only register, whose outputs are the bank select inputs to IC1. Bits 3 and 4 of the REU bank register at \$DF06 are latched into IC2, two D type flip-flops, on the negative edge of the system clock (theta2). By mapping our new two bit register to these normally unused bits, the extra memory appears to the user as extra 64k banks beyond the normal maximum of 512 kb. IC3 is a 3 to 8 line decoder/demultiplexor, which is used to decode the lowest three bits of the I/O2 page (\$DF00 to \$DFFF) address. The IC3 signal *O6 is used as the clock signal for the two flip-flops, which latches bits 3 and 4 of the data bus on the positive edge. This happens whenever \$DF06 is written to. On a read memory cycle to \$DF06, IC3 is disabled and the REU's regular internal register appears on the data bus. A switch pulls the *CLR inputs to each of the flip-flops low when it is closed (position 1). This forces them to select bank zero of 512 kb at all times, which is present in all 512kb REUs. When the switch is open (position 2), the flip-flops can freely take on the values of bits 3 and 4 of \$DF06. This allows complete software compatibility with a 1750 (512 kb) REU. Lastly, the two Light

Emitting Diodes (LEDs), which are optional, simply show us (in binary) what bank of 512k we are using. They allow us to quickly check software compatibility, as explained later.

Installation

All the expansion hardware fits inside the REU case. You will not be able to use the RF shield, and it will be a very tight fit with all 2 Mb installed (don't use IC sockets!). Start by expanding your REU to 512 kb, or in other words you need a 1750 REU. See Volume 9, Issue 5 of the Transactor, or the article by ScottB30 on Quantum Link for expanding 1764 and 1700 REUs. Take the usual static precautions, and of course, any modification to the REU will void the warranty! Proceed at your sole risk! On the component side of the REU, with the edge connector towards you, locate Resistor Package RP3 on the left side near the middle of the board. Flip the board over to the solder side, again with the edge connector towards you. Locate RP3 again on the right side of the board. From the top of the board, find pin 7 of RP3 and cut the trace leading away from the pin. Solder a wire to this pin (pin 7); this is the *CAS0 signal. Find pin 3 of RP3 and again cut the trace leading away from the pin. Solder a wire to this pin (pin 3); this is the *CAS1 signal. Flip the board over to the component side with the edge connector towards you, and locate ram chips U2 and U10 on the top left side of the board. Flip the board over to the solder side and again locate these chips. To pin 15 of U2 solder a wire; this is the *CASBANK0 signal. Solder a wire to pin 15 of U10; this is the *CASBANK1 signal. Run the four wires you now have, down to the right side of the edge connector and secure them in the corner with a piece of electrical tape. This completes all the solder connections to the solder side of the board. Locate the fifth pin from the right on the edge connector and follow the trace to a component leg; note the placement. Flip the board to the component side for the last time, and locate the component. It is labelled FB2, and should be a Ferrite Button, but is a 430 Ohm resistor (yellow, orange, brown, gold bands) on my REU. To the opposite end of the component, away from the edge connector, solder a wire; this is the system clock signal theta2. Locate the thirteenth pin from the right on the edge connector. Follow the trace to a pass-through, and solder a wire into it; this is the */O2 signal. Locate the eighteenth pin from the right of the edge connector, and follow the trace to Ferrite Button FB1. To the side away from the edge connector solder a wire; this is the R/*W signal. Locate the empty pinout (U18) next to the square REC chip. Solder wires into the holes for pins 8, 9, 10, 14, 15, 16, and 28. These are the signals A2, A1, A0, Ground, D3, D4, and +5 volts respectively. This completes the signal hunting. Lay a 2 inch strip of double-sided foam tape down the centre of the missing IC (U18) pattern. Try to leave room near the right edge of the case so you can mount a switch. Mount the ICs, in order from left to right, to the tape UPSIDE DOWN (pins sticking up) with the notches (or dots) towards the top of the board. Place them as far apart as you can on the tape. Now connect the ICs using (carefully!) point to point soldering, or wire-wrap DIRECTLY on the IC pins (yes, it can be done). If you do wire-wrap, only 4 wrap per connection are necessary, and "over wrapping" (wrapping on top of wrapping) is recommended for the second connection to a pin. Keep the connections as short as possible e.g.. +5 volts to pin 14 of IC1 in mine is 1/4 of an inch long. Mount the switch in the top half and on the edge of the REU's case in a convenient place. I recommend placing it near the bottom right corner, near the plastic post since there is nothing in the way here. A slide switch is neater than a toggle switch, but it isn't very much fun trying to make a square hole for a the slide switch! Mark (or just note) the position of the switch on the case (open: 2 Mb, closed: 512 kb) with a marker (Sanford "Sharpie" writes permanently on anything!) or use "Letraset" type lettering. The next step, and last step, is to wire and mount the two optional LED indicators. Solder wires to all the leads of both the LEDs. Drill holes for the at the top edge, of the top half of the case, so they're visible with the REU installed. Make sure to put LED1 on the left and LED0 on the right, so you can read the bank of 512 kb in binary (0=un-lighted, 1=lighted). Your REU now works the same as before, at least it should! Plug it in, move the switch to position 1 (switch closed, the optional LEDs should both be un-lighted) and give the REU a spin with any test program of your choice. If things look bad, power down and recheck all connections! If it does work, congratulations! You now have an REU that can be expanded to 2 Mb by adding a meagre (48) RAM chips.

Adding memory

Expand the memory by piggy-backing the existing RAM chips in chip banks one or two, starting with chip bank 1. Bend pin 15 up, to a 45 degree angle, on each of the RAM chips to be added. Connect a wire to this pin before soldering it to the other RAM chip. Once the chip is soldered in place, connect the wire to pin 15 of the next RAM chip to be added. Solder another wire to its pin 15, and solder the chip in place next to the other. Continue in this way until all the RAMs in a chip bank are piggy-backed, and all pins 15 are connected in a daisy-chain fashion in each chip bank. Solder a wire to pin 15 of the rightmost RAM chip in the bank; this is the *CASBANKx signal. Now use an ohmmeter or continuity tester to see that all the chips in the newly added bank share the same signals on pins 1, 3 to 13, and 15 and 16. All that is left is to connect the wire *CASBANKx to an appropriate numbered signal on IC1.

Odd numbered CASBANK signals are for RAM chips piggy-backed in chip bank one, even numbered ones are for RAM chips piggy-backed in chip bank two. Choose the next available signal when adding a new bank of RAMs. *CASBANK0 and *CASBANK1 are reserved for the two chip banks already in a 512 kb REU. Connect the *CASBANKx signal, and you now have another 256 kb of memory! Repeat for chip bank two for 512 kb extra RAM. Add just as much RAM as you need in 256 kb increments!

Programming an expanded REU

You don't need to do anything different to use the extra memory in the REU. Just remember that you now have more 64 kb banks that you can access through the bank register at \$DF06 when in 2 Mb mode (switch in position 2). When in the 1750 emulation (512 kb) mode (switch in position 1), remember you only have access to banks 0 to 7 of 64 kb. Keep in mind the peculiarities of the 2 Mb mode as well. That's all there is to it! The number of banks depends on how much memory you added; 0 to 15 with 1 Mb, and 0 to 31 with 2 Mb for example. Note that all software that tests for a REU will only find 512 kb available, so a slightly different memory test is necessary to check on how much expansion memory we have. Since the REU can be expanded up to 2 Mb in 256 kb increments at the users discretion, a slightly different test is mandatory to figure out just how big the REU is. See listing 1 for an example. Peculiarities

There are a few very minor inconveniences with this expansion project detailed as follows:

1. The REU will not wrap internally between banks of 512 kb, instead the REU will wrap to the beginning of the same 512k bank. For example, saving 2 bytes to \$FFFF in bank 7 will put one byte at \$FFFF in bank 7, and the other at \$0000 in bank 0 NOT bank 8 as expected. This is because the REC chip does not recognise more than 512 kb directly; we have fooled it to use more RAM.
2. Bank register at \$DF06 is write-only where bits 3 and 4 are concerned. If \$DF06 is read, bits 3 and 4 (as well as bits 5, 6, and 7) will always be one no matter what bank of 512 kb the REU is in.
3. Bits 3 and 4 of the bank register at \$DF06 are now significant, but in an unexpanded REU they are ignored. This is why we have a switch to disable all but 512k of memory when we run into non-compatible software. See the next section for details.

- The 512 kb bank select bits are memory mapped by only the three lowest address bits. This means there are images of these bits at \$DF0E, \$DF16, \$DF1E,

etc. This is a trivial matter since no sane person uses image addresses!

Compatibility

There is only one source of software incompatibility in this modification; the 2 bit 512 kb bank select register we had added to \$DF06. Since these two bits are now significant and didn't used to be, we could have a problem; software that doesn't set these bits to the same value at all times, or sets the bits to a bank of 512 kb that isn't installed yet, won't work. But this is what the switch is for! The 2 Mb modification is 100% compatible with any software written for a 1750 (512kb) REU, when in the switch selectable 1750 emulation mode (position 1, switch closed). Unfortunately none of the extra memory beyond this can be accessed when in this mode. This is of no consequence, since the software can't make use of any additional memory anyway. When the 2 Mb mode is switch selected (position 2, switch open), we have complete access to how ever much memory we have added. The switch should be in position 2 whenever possible! Most software can function in this mode, including GEOS! The two optional LEDs are very useful to determine if the software is compatible in the 2 Mb mode. If you are using software meant to be used with a 1750 REU, you can use the software in the 2 Mb mode if: during a REU transfer the LEDs are always showing the same bank of 512 kb is being accessed (LEDs not flickering), and they indicate a bank that is installed (bank 0 of 512 kb is always present). Without the LEDs, it is simply a matter of saying "It works", or "It doesn't work" when you use software for a 1750 REU.

Final words

Don't let the length of this article discourage you; it only seems difficult in print! Your REU can now contain as much memory as an average IBM PC. So be nice to your IBM buddies, and don't brag too much! I can be reached on Q-link (screen name Recursion), or you can write me directly if you have any questions comments, or construction ideas.

Andrew E. Mileski 210-180 Lees Avenue Ottawa, Ontario Canada, K1S 5J6

Listing 1: Recommended type of REU memory check

It is implemented here in BASIC with a REU wedge for the C64, but can easily be translated into assembly. This check will always return the number of 64 k byte banks that are useable in the REU. FETCH and STASH do exactly what you might think. The operands are bytes to transfer, computer base address, REU base address, and REU bank (of 64 kb) number.

```
2000 REM ****INITIALISE ****
2010 REM Put a single status byte, that shows an unchecked bank, in each bank
of 64kb.
2020 POKE 49152,136
2030 FOR I=0 TO 255
2040 !STASH 1,49152,0,I
2050 NEXT
2060 REM **** TEST ****
2070 REM Test for # banks present by seeing if bank checked yet. If not,
change status of
2080 REM bank to checked, and add 1 to the # of banks. Else, bank is checked
already and
2090 REM we're done! Note that we only test one byte per bank to speed things
up.
2100 POKE 49152,27
2110 I=0
2120 !FETCH 1,49152,0,I
2130 IF PEEK(49152)<>136 THEN 2190
2140 POKE 49152,27
2150 !STASH 1,49152,0,I
2160 I=I+1:IF I<256 THEN 2120
2170 REM **** DONE ****
2180 REM Number of banks of 64kb is now in I.
2190 IF I=0 THEN PRINT "REU NOT PRESENT, OR NOT SEATED PROPERLY"
2200 IF I=2 THEN PRINT "THIS IS A 1700 (128 K) REU"
2210 IF I=4 THEN PRINT "THIS IS A 1764 (256 K) REU"
2220 IF I=8 THEN PRINT "THIS IS A 1750 (512 K) REU"
2230 IF I>8 THEN PRINT "THIS IS AN EXPANDED"; 64*I ;"K REU"
```

Parts List and Miscellaneous Data

Required parts

IC1: 74F139 Dual 2 to 4 line decoder/demultiplexor.
IC2: 74F74 Dual D-type flip-flops.
IC3: 74F138 3 to 8 line decoder/demultiplexor. FAST series components
recommended!

3.3k Ohm Resistor, 1/4 watt and 5% tolerance.

SPST micro-mini toggle, or slide switch.

Two inch strip of double-sided foam tape.

Supply of 30 gauge wire.

And other standard tools and equipment.

Optional Two LEDs T-1 size.

Two 390 Ohm resistors, 1/4 watt and 5% tolerance.

Pin connections

IC1 Pin 16: +5 Volts
1: *CAS0
15: *CAS1
2: IC2 pin 5
14: IC1 pin 2
3: IC2 pin 9
13: IC1 pin 3
4: *CASBANK0
12: *CASBANK1
5: *CASBANK2
11: *CASBANK3
6: *CASBANK4
10: *CASBANK5
7: *CASBANK6
9: *CASBANK7
8: Ground IC2 Pin 14: +5 Volts

1: **1**
13: **1**
2: D3
12: D4
3: IC3 pin 9
11: IC2 pin 3
4: +5 Volts
10: +5 Volts
5: IC1 pins 2,14
9: IC1 pins 3,13
6: LED0 **2**
8: LED1 **2**
7: Ground IC3 Pin 16: +5 Volts

1: A0 15: No connection
2: A1 14: No connection
3: A2 13: No connection

4: *I/O
12: No connection
5: R/*W
11: No connection
6: Theta2
10: No connection
7: No connection
9: IC2 pins 3,11
8: Ground

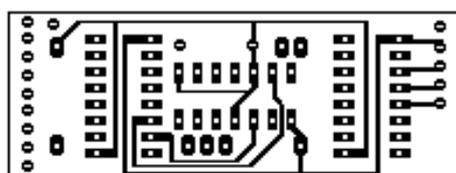
1 Connect this pin to one side of the switch. There should be two pins connected to the same side of the switch. To the same side of the switch connect a single 3.3k Ohm resistor. Connect the free end of the resistor to +5 Volts. Connect the other side of the switch to ground.

2 Connect this pin to the cathode of the LED shown. Connect the anode of each LED to its own 390 Ohm resistor. Connect the free end of each resistor to +5 Volts. Top of ICs are indicated by a dot or notch. When the chips are installed up-side-down, the pin numbering is clockwise from the top right pin! Since the RAM chips are installed right-side up, their pin numbering is counter clockwise from the top left pin. Don't mix up the numbering!!!

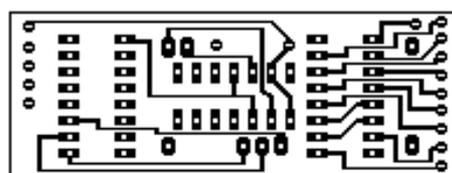
Note This file was converted from the original geoWrite 2.1 file by the author. The schematic in geoPaint format has been reduced into the pin connection table shown above. If you have GEOS 2.0 I recommend you obtain the original ARCed files from Q-Link ("BEYOND512KB.SDA" by Recursion)

* OLX 2.1 TD * Venturing Inn - with Heath - Commodore Network Magazine
* Origin: Commodore Network - Melbourne Australia +61-3-802-3852
(3:633/158.2)

This design for use in REU's with square REC **ONLY!!** Beyond512k plans takes the place of the wirewrapping/taping of support IC's. All inputs/outputs are brought out to the edge of the board for easy access. Thanks to Andrew Mileski for this project and uploads of the schematics.



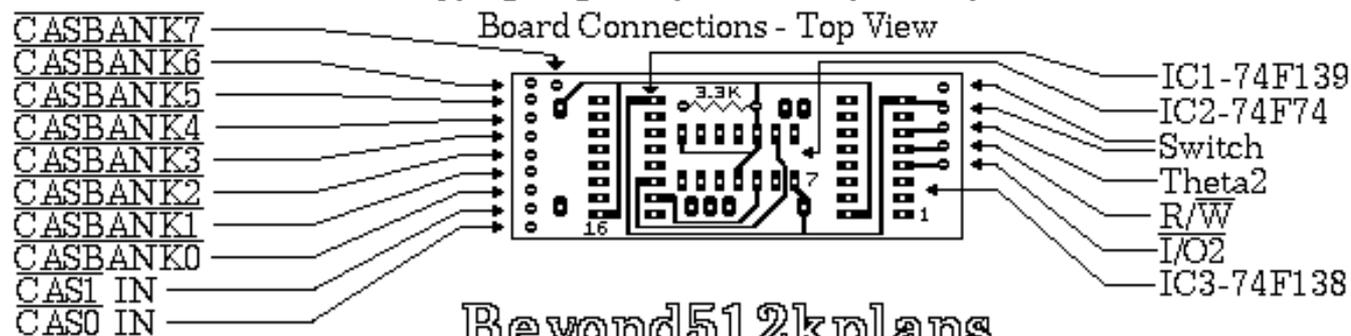
Top



Bottom

NOTES: Solder in a 28 pin socket where the empty IC space is. Large donut holes are for straight-thru connects to the socket. Make straight thru connectors short enough so that the top of support chips will not interfere with the RF shield.

Do NOT install sockets for the support chips. Use Epson QuadStrike Printer Driver for best results, then transfer to your favourite medium such as Tec-200, Direct Etch, PhotoNegative Etch, etc. This artwork Copyright © May 25, 1990 by Wesley J. Wiese.



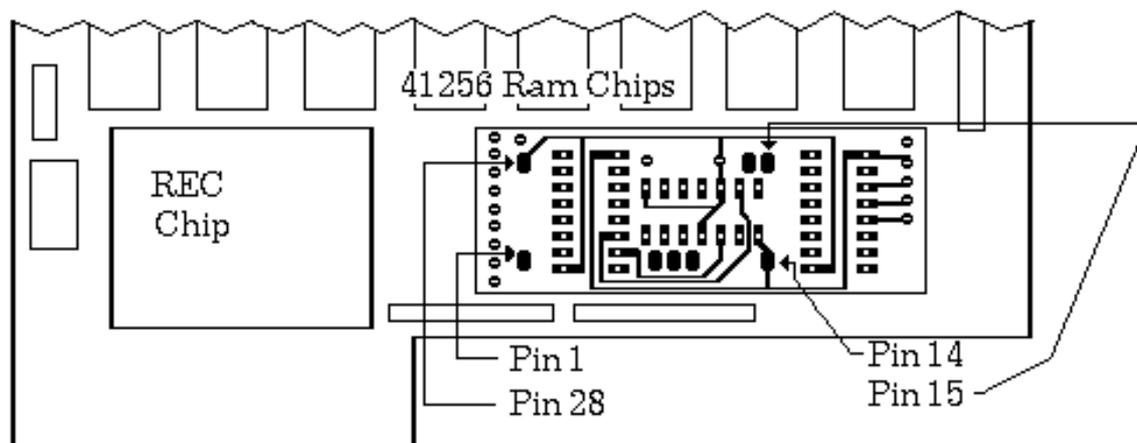
Beyond512kplans

Questions, comments: Leave email to Wes256. Files needed - Beyond 512k.sda which includes Beyond 512k, Schematic and Tech Font, uploaded by Recursion. Also needed- REU, Geos 128 or 64 V2, and Printed Circuit Board construction knowledge. HAVE FUN !!!

If you build this and you haven't uploaded Georep Jim's Configure 2.1 file for Geos, I suggest you do- It's one of the reasons I layed this out. A 1581 RAM Disk.

This layout processed under the Geos 128 V2 Kernal, 128 column mode.

Placement Guide



Converted to GIF
by ME ∞ PK.