COMMODORE®

1700

RAM EXPANSION MODULE

Increases your RAM
by 128K Bytes

1750

RAM EXPANSION MODULE

Increases your RAM
by 512K Bytes

1700/1750 RAM Expansion Module
User's Guide

For Your Commodore 128™

Commodore 128 and C128 are trademarks of Commodore Electronics Limited.

Contents

CP/M is a registered trademark of Digital Research, Inc

Chapter 1   INTRODUCTION

What a RAM Expansion Module Is

A RAM expansion module is a device that allows you to increase the available memory of your computer. Your computer has two kinds of memory areas, RAM and ROM. RAM, short for Random Access Memory, is for temporary storage. You can put programs or other data in RAM and retrieve them later. ROM, or Read Only Memory, stores the operating system of your computer and may not be changed. The 1700 and 1750 RAM expansion modules contain RAM, in an arrangement called banks.

The 1700 RAM expansion module increases the memory of your Commodore 128 computer by 128 Kilobytes or K (for a total of 256 K). The RAM space is arranged in two 64 K banks. The 1750 RAM expansion module increases the available memory by 512 K (for a total of 640 K). In the 1750 RAM Expansion Module there are eight 64 K banks. All the instructions for using the two RAM expansion modules are the same. The only difference between the two modules is the amount of RAM they contain.

The RAM expansion module gets attached directly to your computer's expansion port. Once installed, you give instructions which access the expansion RAM banks. The data can be transferred immediately from the RAM expansion module into the C128 computer's main memory. No time is lost waiting for a disk drive to find and load a file. In CP/M mode of the C128 computer, for example, file loading is 10 to 15 times faster with the RAM expansion module than with a conventional disk drive.

The RAM expansion module contains **indirectly** accessible RAM. This means that programs stored in the RAM expansion module cannot be executed by the C128 computer until they are transferred from the RAM expansion module into your computer's main memory. The main advantage of the RAM expansion module is rapid retrieval of data.

Remember RAM is not permanent storage. If you turn your computer off any data stored in the RAM expansion module will be lost. Any information stored in the RAM expansion module should therefore also be saved on disk, to avoid loss of your work.

Two Ways to Use the Ram Expansion Module

You can use the RAM expansion module in two ways:

1) In CP/M® mode
2) In C128 mode

The techniques used to access the RAM expansion module in each of the two modes are different, but the end result is the same — you get extra memory.

In CP/M mode, the RAM expansion module acts exactly like a very rapid disk drive. Large application programs, such as word processors or business analysis programs, can be stored in the RAM expansion module at the beginning of a work session. From then on, you indicate to the computer to find program utilities in the RAM, rather than on an external disk drive. Because the RAM expansion module stores the programs in a form the computer can use almost instantly, operation of these programs is rapid and efficient. You will find this ability particularly time-saving with programs that access the disk drive often, called "disk intensive" programs, such as dBaseII®.

The CP/M system controls the RAM expansion module. You do not have to be concerned with the way data is being stored, or where in the RAM banks of the RAM module a particular utility is stored. CP/M remembers all this for you, and can display a directory of currently stored programs or data files.

In C128 mode, you use BASIC or machine language to perform data transfers between the main memory of your computer and the RAM expansion module's RAM banks. Several 7.0 BASIC commands are designed specifically to access the RAM expansion module. Machine language is helpful in using the more advanced functions of the RAM expansion module in C128 mode. Using the RAM expansion module in C128 mode requires the user to remember what has been stored in the RAM expansion module, and where in the RAM banks the data lies.

CP/M is a registered trademark of Digital Research, Inc.
dBaseII is a registered trademark of Ashton-Tate, Inc.

What comes with your RAM expansion module

Along with your RAM expansion module and this manual are two disks.

One disk contains a BASIC diagnostic test for your RAM expansion module and several demonstration programs. This disk is provided so you can be assured that your device operates according to its design.

The second disk is a CP/M system disk that supports the RAM expansion module. You may need to use this disk to boot CP/M when you plan to use the RAM expansion module. See Chapter 3, **How to Use the RAM Expansion Module in CP/M Mode**, for complete information about this disk.

Suggestions on Reading This Guide

Here is a brief description of what is contained in the remaining chapters of this guide:

Chapter 2 — Installation and Demonstration Programs gives illustrated step-by-step instructions for installing the RAM expansion module and describes how to use the test program and demonstrations provided.

Chapter 3 — How to Use the RAM Expansion Module in CP/M Mode contains instructions on how to determine which CP/M system disk you should use and explains how to use the RAM module with CP/M.

Chapter 4 — How to Use the RAM Expansion Module in C128 Mode describes the BASIC commands SWAP, STASH, and FETCH and shows how to use a memory map to keep track of information stored in the RAM expansion module.

Chapter 5 — Technical Specifications; this section provides information for advanced users.

Note: It is important to read and follow the installation instructions in Chapter 2. Even if you intend to use the RAM expansion module in CP/M, run the diagnostic test and demo programs.

Chapter 2   INSTALLATION and DEMONSTRATION PROGRAMS

Installing the RAM Expansion Module

1) Turn off your computer and any peripherals.

2) Carefully remove your RAM expansion module from its protective foam packaging. Locate the connector on the edge of the device. See Figure 2-1.

Figure 2-1, The RAM expansion module

3) Locate the expansion port on the back of your C128 computer (see Figure 2-2).

RAM module is inserted here

Figure 2-2, C128 computer expansion port

4) Carefully insert the connector on the RAM expansion module into the expansion port on your computer. Insert the module firmly but do not force it, as this might damage the connectors (see Figure 2-3.)

Figure 2-3, RAM expansion module installed

5) Turn on all your equipment. Your RAM expansion module is now ready to use.

Using The Demonstration Programs

Included with your RAM expansion module is a test/demonstration disk. It contains a diagnostic program to test your RAM expansion module to see if it is operating correctly, and demonstrations providing an example of how you might use the RAM expansion module to aid your BASIC programming.

Note: A 40-column monitor is necessary to run the demonstration programs. An 80-column monitor is sufficient for the diagnostic test.

Booting the Demonstration Disk

1) Turn off your computer.
2) Place the disk in the drive and close the disk drive door.
3) Turn on the computer. The message **BOOTING STARTUP** appears on the screen, and the program loads automatically.

   Note: You can also load STARTUP by typing **LOAD"STARTUP",8** and pressing **(RETURN)** or by typing **BOOT (RETURN)**

   STARTUP is a program that controls the demonstration programs. It displays a menu of the programs on the disk. STARTUP reboots automatically when the disk is in the drive and you press the reset button.

Using the STARTUP Menu

The first program on the STARTUP menu is HELP.BAS. This is a file that explains what the other programs on the disk do.

To load HELP.BAS, choose option 1 from the STARTUP menu.

A series of information screens appears. After you read each screen, press any key to continue. When you have read the entire file, press the reset button. The disk will reboot, and you will be returned to the STARTUP menu.

The second program on the STARTUP menu is RAMTEST. This is the diagnostic test.

To load RAMTEST, choose option 2 from the STARTUP menu.

The program loads and this message appears:

RAM EXPANSION TEST   SIZE = 128K (512K)
TEST IS RUNNING WHEN UPPER LEFT
LOCATION ON SCREEN IS FLICKERING.

The test takes about five minutes to run. If an error is located, an error message appears and the test stops.

If no error is located, the message **TEST COMPLETED SUCCESS-FULLY** appears on the screen at the end of the test.

Press the reset button, and the STARTUP program is rebooted so you can choose another program.

If you receive an error message, reset the computer and run the program again. If several repeated efforts to run the test fail, take your RAM Expansion Module back to the dealer who sold you the module.

The rest of the programs on the disk are demonstration programs. They are described in the HELP file. Choose the corresponding number from the STARTUP menu to load each program.

Some of the demonstration programs take several minutes to load, but begin automatically once loaded. Some programs give you the option of stopping (using the RUN/STOP key) then starting again without reloading the RAM expansion module. On screen directions describe the various options presented by each program. Pressing the reset button will always reboot the disk and return you to the STARTUP menu.

Once you have run the RAMTEST and seen the demonstration programs you should be confident your RAM expansion module works. Continue with Section 3, How To Use The RAM Expansion Module In CP/M Mode or Section 4, How To Use The RAM Expansion Module in C128 Mode.

CHAPTER 3 HOW TO USE THE RAM EXPANSION MODULE IN CP/M MODE

Booting CP/M

If you are a CP/M user, you are familiar with the CP/M system disk packed with your C128 computer. Another version of the CP/M system came with your RAM expansion module. The reason for two CP/M disks is this: early releases of the CP/M software packed with the C128 computer do not support the RAM expansion module. A special CP/M system disk, that supports the RAM expansion module, was prepared and packaged with your RAM expansion module.

If you purchased your computer at the same time you purchased your RAM expansion module, it is likely that the CP/M disk supplied with your computer supports the RAM expansion module.

Commodore 128
Personal Computer

1700/1750 RAM
Expansion Module

CP/M 3.0

May or may not support
RAM expansion, depending
on release date

CP/M 3.0

Supports RAM
expansion
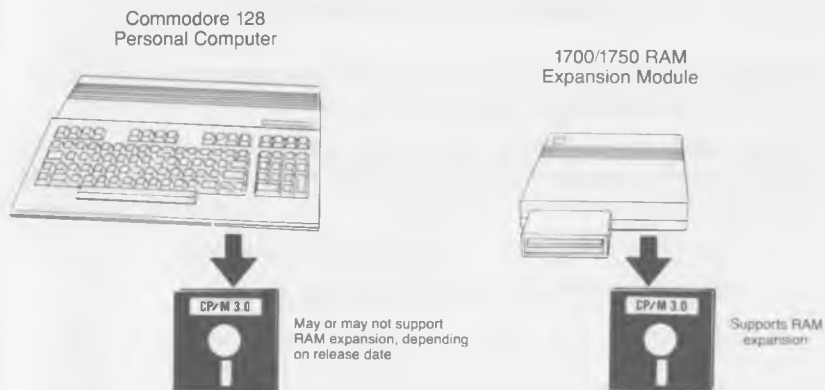
FIGURE 3-1, CP/M system disks

9

If you purchased your C128 computer earlier, and later decided to purchase a RAM expansion module, then you will probably need to use the version of CP/M packed with your RAM expansion module.

To determine which is the most recent version of CP/M, boot each system using the instructions below. At the top of the screen a title and date appears. Choose the most recent version of the CP/M system for use with the RAM expansion module, regardless of which machine it was packaged with.

The CP/M system packaged with your RAM expansion module is booted the same way as your original CP/M system was booted. You have two options:

1) Turn off the computer. Place the disk in the drive, then turn on the computer. CP/M will boot automatically.

2) Start in C128 mode, with equipment turned on. Put the CP/M system disk in the disk drive and type the BASIC command **BOOT (RETURN)**

Once the system is booted, CP/M will function in the usual manner with up to four external disk drives labelled A through D and the virtual drive E. Follow the instructions in your C128 Computer System Guide for using CP/M without the RAM expansion module. Provided below are instructions for using the RAM expansion module like a fifth drive.

Using CP/M Commands

Once the system is loaded, you can use the RAM expansion module like a fast external drive simply by referring to drive **M** (Memory drive).

For example, to save time during a work session, you might want to store some of the more commonly used transient utilities in the RAM expansion module. Here is the method:

1) With the CP/M utility disk in the external drive A, type:

A> **PIP M: = A:FORMAT.COM [v] (RETURN)**

In this example, the PIP utility would search the disk in drive A, locate the FORMAT.COM file, and store it in the RAM expansion module. The **[v]** verifies that the transfer occurred correctly. You can also use **DIR M:** to determine what is stored in the RAM expansion module.

Any filename could be used in place of FORMAT. Be sure to use the correct filetype in addition to the filename if necessary.

2) When the computer displays the **A>** prompt, the transfer is complete. Now the FORMAT utility is stored in the RAM disk.

3) To compare the speed of execution from an external drive and from the RAM expansion module:

Type A>**FORMAT (RETURN)**

The disk drive will whir and eventually instructions for formatting a disk will appear on screen. Abort the format by pressing (RETURN), then any key, then answering **N** to the question, **"Do you want to format another disk?"**

Now load the FORMAT command from the RAM expansion module:

Log on to drive M by typing **M:** at the **A>** prompt.

Type: M>**FORMAT (RETURN)**

Almost instantly the format command is retrieved from the RAM expansion module and displayed on the screen. Alternately, you can choose to execute the FORMAT command without logging to drive M> by typing **M:FORMAT** at the **A>** prompt.

This example should illustrate how the RAM expansion module can speed up your operations. You can copy a number of files into the RAM expansion module with a series of PIP commands. Or copy the entire contents of a disk:

A> **PIP M: = A: *.***   will copy all the files from the disk in drive A into the RAM expansion module.

Note that a full single-sided disk holds 160 K of data. If your disk is completely full, the entire contents cannot be copied into the 1700 RAM Expansion Module.

Once the RAM expansion module has been loaded with a program, use of the program is no different than if the program were in an external drive. If you are saving data files for later use, however, you may want to save them on an external drive instead of in the RAM expansion module.

Using the reset button does not clear the expansion RAM banks, only the C128 computer's main memory.

To avoid accidental loss of your work, periodically save programs or data files in progress on disk, as well as in the RAM expansion module. A power surge, or an accidentally pulled electric cord, can have a devastating effect on your work if you only store it in the RAM expansion module.

CHAPTER 4   HOW TO USE THE RAM EXPANSION MODULE IN C128 MODE

BASIC Commands

There are three BASIC commands built into BASIC 7.0 that were designed especially for use with the RAM expansion module. These commands handle the three types of data transfers possible with the RAM expansion module:

1) STASH — Stores data in the RAM expansion module
2) FETCH — Retrieves data stored in the RAM expansion module and places them in main memory
3) SWAP — Exchanges data stored in the RAM expansion module with data stored in main memory

Each of these command words is followed by four parameters that define how large a block of data is being transferred, where in main memory the data is currently stored, where in the RAM expansion module (i.e. what address) the data should be transferred, and what expansion RAM bank is being used.

The parameters are the same for all three commands:

```
FETCH
STASH        number of bytes,intsa,expsa,expba
SWAP
```

where bytes = number of bytes involved in the transfer
    intsa = starting address of host RAM (0 - 65535)
  expsa = starting address of expansion RAM (0 - 65535)
    expb = expansion RAM bank number (0 - 1 on the 1700)
                                        (0 - 7 on the 1750)

Note: FETCH, STASH, and SWAP only accept decimal numbers.

When you use any of these commands, you are transferring blocks of data from one area of RAM to another. It is important that you know what you are transferring, and where you are putting it. If you transfer data into an area that already is occupied by a program or other data, the data you transfer into that area will overwrite the original data. This can have an undesired effect on the outcome of the transfer.

Memory Locations

In order to guarantee a safe transfer, you need to be aware of the memory map for your computer. It is a diagram, see Figure 4-1 on next page, that shows what each section of your computer's memory stores. Certain areas are set aside for specific functions, and other areas are free for your programs. The entire contents of the RAM expansion module is up to you; no space is preallocated.

When you perform any operation with your computer, data is stored in your computer's memory. Each memory location is given an address which, like the street addresses of houses, can be used to identify a particular location. The specific contents of memory locations determines the output of programs, what the screen looks like, and other computer functions.

The addresses of memory locations accessible by the user range from 0 to 65535. Many of these locations, however, have special purposes. For example the addresses between 4608 and 4864 are reserved for storing absolute values for BASIC programs.

One important note: the BASIC commands STASH, FETCH and SWAP can only be used to transfer or retrieve data in Bank 0 of the C128 computer's internal RAM. See the section entitled **Notes for BASIC Programmers** in Chapter 5 for details on accessing Bank 1.

The 1700 RAM expansion module contains two RAM banks labelled 0 and 1. The 1750 RAM expansion module has eight RAM banks labelled 0 to 7. All the banks in the RAM expansion modules are available with the BASIC commands STASH, FETCH, and SWAP. The RAM expansion module banks bear no relation to the C128 computer RAM banks.

C128 RAM

Decimal	Hex	
65535	FFFF	
65530	FFFA	NMI RST IRQ
65488	FFD0	CP/M RAM Code
		Krnl RAM Code
65285	FF05	MMU Configuration Register
65280	FF00	
		BASIC TEXT AREA (Basic text begins at $1C00 if bit map unallocated)
16384	4000	

C128 ROM

Decimal	Hex		
65535	FFFF		
65357	FF40	Kernal Jump Table & Hardware Vectors	
65285	FF05	Kernal Interrupt Dispatch Code	
65280	FF00	MMU Configuration Registers	
64640	FC80	ROM Reserved for Foreign Lang. Versions	HIGH ROM
64128	FA80	Editor Tables	
57344	E000	Kernal ROM Code	
53248	D000	I/O Space	
49152	C000	Editor ROM Code	
45056	B000	Monitor ROM Code	MID ROM
32768	8000		
16384	4000	Basic ROM Code	LOW ROM

C128 RAM

Decimal	Hex	
16384	4000	VIC BIT-MAP Screen
8192	2000	VIC BIT-MAP Color (Vm #2)
7168	1C00	Reserved for Applications
6912	1B00	
6656	1A00	
6400	1900	
6144	1800	Reserved for Applications
5120	1400	Reserved for Applications
4864	1300	Basic Absolute Variables
4608	1200	Basic DOS/VSP Variables
4360	1108	CP/M Reset Code
4352	1100	Function Key Buffer
4096	1000	Sprite Definition Area
3840	0F00	
3584	0E00	RS-232 Output Buffer
3328	0D00	RS-232 Input Buffer
3072	0C00	(Disk Boot Page)
3008	0BC0	Cassette Buffer
2816	0B00	Monitor & Kernal Absolute Variables
2560	0A00	Basic Run-Time Stack
2304	0900	
2048	0800	VIC Text Screen (VM #1)
1024	0400	Basic RAM Code
896	0380	Kernal Tables
828	033C	Indirects
764	02FC	Kernal RAM Code
674	02A2	Basic & Monitor Input Buffer
512	0200	System Stack
329	0149	Basic DOS Using
272	0110	F BUFFER
256	0100	Kernal Z P
144	0090	Basic Z P
2	0002	
0	0000	

Figure 4-1, C128 computer memory map

BASIC Examples

Let's create a program that accesses the RAM expansion module.

1) Reset your computer in 40 column mode so the start-up message is visible. The screen should display:

COMMODORE BASIC V7.0 122355 BYTES FREE
(C) 1985 COMMODORE ELECTRONICS, LTD
(C) 1977 MICROSOFT CORP.
ALL RIGHTS RESERVED

2) Move the cursor up to the start-up message. Type some changes: perhaps replace the number of bytes free, or put your name in place of Commodore. Make all your changes with the cursor and INST/DEL keys; be sure not to press (RETURN).

3) When you are satisfied with the changes, move the cursor to the bottom line of the screen.

Type **STASH1000,1024,0,0 (RETURN)**

This command stores 1000 bytes from the computer's main memory, beginning at address 1024 in the RAM expansion module. Address 1024 marks the beginning of the text screen, the part of memory where the current screen contents are stored.

4) PRESS: **(SHIFT) and (CLR/HOME)** to clear the screen. Type this program:

10 FOR X = 1 TO 25
20 FETCH 1000,1024,0,0
30 SLEEP 1
40 PRINT"(SHIFT) and (CLR/HOME)"
50 PRINT X: SLEEP 1: NEXT X

5) Type **RUN (RETURN)**. The altered startup message is fetched from the RAM expansion module, displayed, then a counter flashes and the image is fetched again.

This example gives you a starting point for more creative uses of the RAM expansion. Let's try a simplified version of the graphic demos included on the demonstration disk packed with your RAM expansion module.

Turn the machine off and then on to clear the RAM expansion module's memory. Note that using the reset button does **not** clear the expansion RAM banks, only the C128 computer's main memory.

1) Clear the screen using (SHIFT) and (CLR/HOME).

2) Create a picture using graphic characters. Try a smiling face, or other simple design.

3) Cursor down to the bottom of the screen.

   TYPE: **STASH 1000,1024,0,0 (RETURN)**

4) When the computer responds READY, clear the screen.

   TYPE: **FETCH 1000,1024,0,0 (RETURN)**

   Your smiling face returns. Move the cursor up to the picture, and alter it in some way. Perhaps change the smile to a frown, or add eyebrows. Delete any unnecessary characters (such as a READY message) by using the spacebar to go over them. Don't use the INST/DEL key as this can change the arrangement of your picture. Be careful to align the picture in the same location it originally was. Use the spacebar to add lines of space and move the picture up or down a row at a time if necessary.

6) Move the cursor to the bottom of the screen.

   TYPE: **STASH 1000,1024,1000,0 (RETURN)**

   Notice that the starting address given for the RAM expansion module is 1000, not 0 as in previous trials. This is because we don't want to overwrite the existing screen with the smiling face.

7) Clear the screen once again, and type this program:

   10 FOR X = 1 TO 10
   20 FETCH 1000,1024,O,O:REM SMILE FACE
   30 SLEEP 1
   40 FETCH 1000,1024,1000,0:REM FROWN FACE
   50 SLEEP 1
   60 NEXT X

8) Type **(RUN)** to view the program.

These examples make use of an area of RAM in the C128 computer that is easy to access. Since the text screen always begins at address 1024, and is exactly 1000 bytes long, we can safely create as many text screens as we like. Each screen can be stored in the RAM expansion module, in addresses 1000 bytes apart, and later fetched without any problem.

Storing other types of data, such as programs, requires being able to use machine language to view other addresses. Detailed information about machine language and the machine language monitor is contained in the **C128 Personal Computer System Guide.**

Chapter 5   Technical Specifications

It is possible to use the RAM expansion module in the ways described in chapters 3 and 4 of this manual without being aware of the RAM Expansion Controller (REC). The REC is the processor inside the RAM expansion module that makes communication between the C128 computer and the RAM expansion module possible. Both CP/M and the built-in BASIC commands automatically control the transfer of information. In order to gain more precise control over the RAM expansion module, details about the REC are needed.

Operating Modes

The REC has four primary operating modes. These are:

1) Transfer a block of data from main memory to expansion memory (BASIC command: STASH)

2) Transfer a block of data from expansion memory to main memory (BASIC command: FETCH)

3) Exchange a block of main memory with a block of expansion memory (BASIC command: SWAP)

4) Verify a block of main memory with a block of expansion memory

The REC has several internal registers which can be used to set up a particular operating mode. Modes can be selected by setting the appropriate bits in the command register (see Figure 5-1). The starting C128 address, expansion RAM address, expansion RAM bank, and number of bytes are all programmable values.

REC Register Description

ADDRESS	BITS	FUNCTION	
$00	7-0	Status Register — Read Only	
		7 - Interrupt Pending;	I = Interrupt waiting to be serviced
		6 - End of Block;	I = Transfer complete
		5 - Fault;	I = Block verify error
		4 - Size;	0 = Total expansion = 128K
			I = Total expansion = 512K
		3-0 - Version;	
		Note: Bits 7-5 are cleared when this register is read	
$01	7-0	Command Register — Read/Write	
		7 - Execute;	I = Transfer per current config.
		6 - Reserved;	
		5 - Load;	I = Enable AUTOLOAD option
		4 - FF00	I = Disable FF00 decode
		3 - Reserved;	
		2 - Reserved;	
		1,0 - Transfer type;	00 = transfer C128 —> RAM module
			01 = transfer C128 <— RAM module
			10 = swap C128 <-> RAM module
			11 = verify C128 — RAM module
$02	7-0	C128 Base Address, LSB — Read/Write	
		Lower 8 bits of base address, C128	
$03	7-0	C128 Base Address, MSB — Read/Write	
		Upper 8 bits of base address, C128	
$04	7-0	Expansion RAM address, LSB — Read/Write	
		Lower 8 bits of base address, expansion RAM	
$05	7-0	Expansion RAM address, MSB — Read/Write	
		Upper 8 bits of base address, expansion RAM	
$06	2-0	Expansion RAM bank — Read/Write	
		Expansion RAM bank pointer.	
		Bits 2 (MSB) to 0 (LSB) are significant	
$07	7-0	Transfer length, LSB — Read/Write	
		Lower 8 bits of the byte counter	
$08	7-0	Transfer length, MSB — Read/Write	
		Upper 8 bits of the byte counter	
$09	7-5	Interrupt mask register — Read/Write	
		7 - Interrupt enable	I = Interrupts enabled
		6 - End of Block mask	I = Interrupt on end of block
		5 - Verify error	I = Interrupt on verify error
$0A	7-6	Address control register — Read/Write	
		0,0 = Increment both addresses (default)	
		0,1 = Fix expansion address	
		1,0 = Fix C128 address	
		1,1 = Fix both addresses	

Figure 5-1

Note: The Base address of the REC registers in the C128 memory map is DF00 in I/O space. The addresses of the REC registers can be obtained by adding the value in the Address column in Figure 5-1 to the base address of DF00.

Direct Memory Access Operation

Direct Memory Access (DMA) is the process the REC uses to transfer data to and from the C128 computer's memory. During DMA the C128 computer processor is temporarily halted so the REC may access the C128 computer's memory.

There are two different events which cause the REC to begin a transfer sequence

1) If the FF00 option in the command register ($01) is disabled, the DMA begins right after the execute bit is set in the command register.

2) If the FF00 option in enabled, the DMA begins right after a write to address FF00 is detected — providing the execute bit was set in the command register.

The purpose of the FF00 option is to allow the REC to access the RAM beneath I/O space. The REC registers are I/O mapped on the C128 expansion bus which means that, without the FF00 option, I/O space would be enabled when the DMA was initiated. Address FF00 is the configuration register within the C128's MMU. This option, therefore, allows the user to reconfigure the C128's memory before the DMA takes place. The FF00 option is cleared each time it is used.

Special Features

AUTOLOAD — One special feature is AUTOLOAD. Selecting this option in the command register automatically reloads the C128 base address registers, the external memory base address registers, the external memory bank, and the byte counter registers at the end of the transfer.

This is useful if one operation is to be executed repeatedly on one particular block of data. Note that in verify mode, if AUTOLOAD is selected, the address where the verify error occurred is lost. Ordinarily upon finding a verify error, the REC halts the DMA cycle and both address registers and the bank register point to one location above the address that failed.

Address Control — Another special feature of the REC is the ability to hold a source and/or destination address fixed in any mode. This is accomplished by setting the appropriate bits in the address control register ($0A). The default is that addresses will increment for both source and destination. It is possible, with a fixed C128 address, to DMA to any C128 I/O device except the 8563.

Interrupts — An interrupt can be generated on an end of block condition or verify error by enabling interrupts and by setting the corresponding mask bits in the interrupt mask register ($09). In either case, the corresponding flag is set in the status register. Interrupt flags are cleared upon reading the status register. The interrupt mask register bits stay as they were before the interrupt/clear sequence. For proper operation, the status register ($00) must be read at least once before a DMA is initiated with interrupts enabled. It is important to clear out the previous status before any mode is used with interrupt option.

Operation Description

If a specified address range exceeds the address range in the selected expansion RAM bank, a "wrap" will occur into the next bank. This will happen whenever the given byte counter exceeds the number of bytes left in the selected bank. If the selected bank is equal to the maximum for a particular memory configuration, then the wrap is to bank 0. This will give the appearance of contiguous memory. The C128 side always wraps to the beginning of the same bank.

Under normal operation (no AUTOLOAD, addresses increment), both address registers will point to the next sequential memory location outside the selected transfer range at the end of the transfer. This is true for any mode and applies to both base address pointers except one that is held fixed. The bank pointer will also be updated if necessary at the end of the transfer. Also note that under normal operation, the byte counter decrements to the value 1. Care should be taken, therefore, to check the transfer complete bit in the status register and not rely on the byte counter value to indicate an end of transfer condition. A byte counter of 0 results in a transfer of a full 64 K bytes. Again, wrapping occurs in all modes of operation.

Important operating notes

1) The Z80 must not be in control when a DMA is requested.

2) The REC cannot reconfigure the MMU during a DMA cycle.

3) On the C128 the DMA bank is set by the VIC bank pointer in the MMU, not the processor bank pointer.

4) The C128 computer must be in 1Mhz mode when a DMA is requested.

Timing

The effective transfer rate is 1 Mbyte/sec for uni-directional transfers via STASH and FETCH, and 500 Kbytes/sec for bi-directional transfers accomplished with SWAP. This transfer rate does not include VIC DMA time. The VIC chip has highest priority on the C128 computer unless it is disabled, so VIC DMA's take precedence over REC DMA's.

Bank Configuration

Bank Select (BS) is provided as a means of configuring the controller so it can direct RAS and CAS to the appropriate bank.

BS	Bank I	Bank II	Total
0	64K x 1	64K x 1	128K
1	256K x 1	256K x 1	512K

This bit in the status register reflects the state of an external jumper which was preinstalled in the factory.

Notes for BASIC Programmers

The BASIC commands STASH, FETCH, and SWAP can only be used to transfer and retreive data in Bank 0 of the C128 computer's internal RAM. To access Bank 1 of the C128 computer it is necessary to alter a special control register on the C128. This register is the RAM control register in the MMU, address $D506 Bits 6 and 7 of this register control both the VIC RAM bank pointer and the C128 DMA bank pointer.

Address	Bits	Function
$D506	6-7	00 C128 Bank 0
		10 C128 Bank 1

Be sure to restore these bits to their original values or your 40-column screen will contain the wrong information.

Note that I/O is enabled for the three BASIC commands in Version 0 of the C128 kernal ROM. In Version 1 of the ROM I/O can be disabled by using the BANK command. See the C128 Personal Computer System Guide for details on the BANK command.

The kernal ROM version can be determined by PEEKing location $FF80. Version 0 contains a 0 while Version 1 contains a 1.

COMMODORE