

Memory Management

(Someone please integrate the texts on this page into one...)

Without Cartridges

The low 3 bits of \$01 control the mapping of specific regions of memory. The rules are kind of fiddly:

Name	Bit	Region	0	1	Notes
LORAM	0	\$A000-BFFF	RAM	BASIC	If KERNAL isn't mapped in, then BASIC won't map in either and this region stays mapped to RAM.
HIRAM	1	\$E000-FFFF	RAM	KERNAL	
CHAREN	2	\$D000-DFFF	CHARROM	I/O	If HIRAM and LORAM are both set to 0, then this bit is ignored and the area also maps to RAM. This allows for 3 mappings of this region: RAM, CHARROM, or I/O.

- All other memory locations (\$0000-9FFF, \$C000-CFFF) always map to RAM.
- Writes to a ROM-mapped region are applied to the underlying RAM at the same address.
- I/O includes the registers for the VIC-II, SID, and CIA chips; color RAM; and two external I/O pages that reach out the expansion port.
- The VIC-II always sees the CHARROM at \$1000-1FFF and \$9000-9FFF, and RAM everywhere else, regardless of these bits.

The mappings from combining these 3 bits are listed below. Higher bits of location \$01 are used for other purposes and default to %00110xxx.

\$01 value	\$A000-BFFF	\$D000-DFFF	\$E000-FFFF	Notes
\$30 +48 %000	RAM	RAM	RAM	
\$31 +49 %001	RAM	CHARROM	RAM	
\$32 +50 %010	RAM	CHARROM	KERNAL	
\$33 +51 %011	BASIC	CHARROM	KERNAL	
\$34 +52 %100	RAM	RAM	RAM	
\$35 +53 %101	RAM	I/O	RAM	
\$36 +54 %110	RAM	I/O	KERNAL	
\$37 +55 %111	BASIC	I/O	KERNAL	Default

The cartridge port also has the GAME and EXROM pins, which meddle with the memory map even further. See the Programmer's Reference Guide for that.

- White Flame

Related: from Graham's page

In the C64/C128 series of computers, slightly modified versions of the 6502 were used. The

modifications did not affect the functional part of the processor itself. Only a so-called processor port was added. This port, in combination with an external PLA, was used to map ROM and I/O areas into the 64KB RAM of the C64. Also, some bits of the port were used for the legendary Datasette.

The port can be accessed through memory addresses \$0000 and \$0001, while \$0001 is the port itself, and \$0000 is the data direction register for it.

Explanation for the bits of \$0001:

```
7 - unused (Flash 8: 0=8MHz/1=1MHz)
6 - unused (C128: ASCII/DIN sense/switch (1=ASCII/0=DIN))
5 - Cassette motor control (0 = motor on)
4 - Cassette switch sense (0 = PLAY pressed)
3 - Cassette write line
2 - CHAREN (0=Character ROM instead of I/O area)
1 - HIRAM ($E000-$FFFF)
0 - LORAM ($A000-$BFFF)
```

If HIRAM or LORAM is set, the I/O area is mapped to \$D000-\$DFFF.

\$0000 should always be set to \$2F (%00101111)

Note to bit 6: This bit is used to select either the ASCII or the DIN character ROM of a C128. When data direction is set to INPUT, the charset is selected externally with the ASCII/DIN key.

From C64 Programmers Reference Manual

The following is a Chapter from the C64 Programmers Reference Manual, can be found at the external Link Section.

MEMORY MANAGEMENT ON THE COMMODORE 64

The Commodore 64 has 64K bytes of RAM. It also has 20K bytes of ROM, containing BASIC, the operating system, and the standard character set. It also accesses input/output devices as a 4K chunk of memory. How is this all possible on a computer with a 16-bit address bus, that is normally only capable of addressing 64K?

The secret is in the 6510 processor chip itself. On the chip is an input/output port. This port is used to control whether RAM or ROM or I/O will appear in certain portions of the system's memory. The port is also used to control the Datasette(TM), so it is important to affect only the proper bits.

The 6510 input/output port appears at location 1. The data direction register for this port appears at location 0. The port is controlled like any of the other input/output ports in the system... the data direction controls whether a given bit will be an input or an output, and the actual data transfer occurs through the port itself. The lines in the

6510 control port are defined as follows:

NAME	BIT	DIRECTION	DESCRIPTION
LORAM	0	OUTPUT	Control for RAM/ROM at \$A000-\$BFFF
HIRAM	1	OUTPUT	Control for RAM/ROM at \$E000-\$FFFF
CHAREN	2	OUTPUT	Control for I/O/ROM at \$D000-\$DFFF
	3	OUTPUT	Cassette write line
	4	INPUT	Cassette switch sense (0=play button down)
	5	OUTPUT	Cassette motor control (0=motor spins)

The proper value for the data direction register is as follows:

```

          BITS 5 4 3 2 1 0
          -----
                1 0 1 1 1 1

```

(where 1 is an output, and 0 is an input).

This gives a value of 47 decimal. The Commodore 64 automatically sets the data direction register to this value.

The control lines, in general, perform the function given in their descriptions.

However, a combination of control lines are occasionally used to get a particular memory configuration.

LORAM (bit 0) can generally be thought of as a control line which banks the 8K byte BASIC ROM in and out of the microprocessor address space. Normally, this line is HIGH for BASIC operation. If this line is programmed LOW, the BASIC ROM will disappear from the memory map and be replaced by 8K bytes of RAM from \$A000-\$BFFF.

HIRAM (bit 1) can generally be thought of as a control line which banks the 8K byte KERNAL ROM in and out of the microprocessor address space. Normally, this line is HIGH for BASIC operation. If this line is programmed LOW, the KERNAL ROM will disappear from the memory map and be replaced by 8K bytes of RAM from \$E000-\$FFFF.

CHAREN (bit 2) is used only to bank the 4K byte character generator ROM in or out of the microprocessor address space. From the processor point of view, the character ROM occupies the same address space as the I/O devices (\$D000-\$DFFF). When the CHAREN line is set to 1 (as is normal), the I/O devices appear in the microprocessor address space, and the character ROM is not accessible. When the CHAREN bit is cleared to 0, the character ROM appears in the processor address space, and the I/O devices are not accessible. (The microprocessor only needs to access the

character ROM when downloading the character set from ROM to RAM. Special care is needed for this... see the section on PROGRAMMABLE CHARACTERS in the GRAPHICS chapter). CHAREN can be overridden by other control lines in certain memory configurations. CHAREN will have no effect on any memory configuration without I/O devices. RAM will appear from \$D000-\$DFFF instead.

```
+-----+
| NOTE: In any memory map containing ROM, a WRITE (a POKE) to a ROM |
| location will store data in the RAM "under" the ROM. Writing to a ROM |
| location stores data in the "hidden" RAM. For example, this allows a |
| hi-resolution screen to be kept underneath a ROM, and be changed |
| without having to bank the screen back into the processor address |
| space. Of course a READ of a ROM location will return the contents of |
| the ROM, not the "hidden" RAM. |
+-----+
```

COMMODORE 64 FUNDAMENTAL MEMORY MAP

E000 - FFFF	8K KERNAL ROM OR RAM
D000 - DFFF	4K I/O OR RAM OR CHAR. ROM
C000 - CFFF	4K RAM
A000 - BFFF	8K BASIC ROM OR RAM OR ROM PLUG-IN
8000 - 9FFF	8K RAM OR ROM PLUG-IN
4000 - 7FFF	16 K RAM
0000 - 3FFF	16 K RAM

I/O BREAKDOWN

D000-D3FF	VIC (Video Controller)	1 K Bytes
D400-D7FF	SID (Sound Synthesizer)	1 K Bytes
D800-DBFF	Color RAM	1 K Nybbles
DC00-DCFF	CIA1 (Keyboard)	256 Bytes
DD00-DDFF	CIA2 (Serial Bus, User Port/RS-232)	256 Bytes
DE00-DEFF	Open I/O slot #1 (CP/M Enable)	256 Bytes
DF00-DFFF	Open I/O slot #2 (Disk)	256 Bytes

The two open I/O slots are for general purpose user I/O, special purpose I/O cartridges (such as IEEE), and have been tentatively designated for enabling the Z-80 cartridge (CP/M option) and for interfacing to a low-cost high-speed disk system.

The system provides for "auto-start" of the program in a Commodore 64 Expansion Cartridge. The cartridge program is started if the first nine bytes of the cartridge ROM starting at location 32768 (\$8000) contain specific data. The first two bytes must hold the Cold Start vector to be used by the cartridge program. The next two bytes at 32770 (\$8002) must be the Warm Start vector used by the cartridge program. The next three bytes must be the letters, CBM, with bit 7 set in each letter. The last two bytes must be the digits "80" in PET ASCII.

COMMODORE 64 MEMORY MAPS

The following table lists the various memory configurations available on the COMMODORE 64, the states of the control lines which select each memory map, and the intended use of each map.

The leftmost column of the table contains addresses in hexadecimal notation. The columns aside it introduce all possible memory configurations. The default mode is on the left, and the absolutely most rarely used Ultimex game console configuration is on the right. Each memory configuration column has one or more four-digit binary numbers as a title. The bits, from left to right, represent the state of the /LORAM, /HIRAM, /GAME and /EXROM lines, respectively. The bits whose state does not matter are marked with "X". For instance, when the Ultimex video game configuration is active (the /GAME line is shorted to ground, /EXROM kept high), the /LORAM and /HIRAM lines have no effect.

	LHGE	LHGE	LHGE	LHGE	LHGE	LHGE	LHGE	LHGE	LHGE
10000	1111 default	101X	1000	011X 00X0	001X	1110	0100	1100	XX01 Ultimax
F000	Kernal	RAM	RAM	Kernal	RAM	Kernal	Kernal	Kernal	ROMH(*)
E000									
D000	IO/C	IO/C	IO/RAM	IO/C	RAM	IO/C	IO/C	IO/C	I/O
C000	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	-
B000	BASIC	RAM	RAM	RAM	RAM	BASIC	ROMH	ROMH	-
A000									
9000	RAM	RAM	RAM	RAM	RAM	ROML	RAM	ROML	ROML(*)
8000									
7000									
6000	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	-
5000									
4000									
3000									
2000	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	-
1000									
0000	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM

NOTE: (1) (2) (3) (4) (5) (6) (7) (8) (9)

*) Internal memory does not respond to write accesses to these areas.

264 BASIC TO MACHINE LANGUAGE

~

Legend: Kernal E000-FFFF Kernal ROM.
IO/C D000-DFFF I/O address space or Character

		generator ROM, selected by -CHAREN. If the CHAREN bit is clear, the character generator ROM is chosen. If it is set, the I/O chips are accessible.
I/O/RAM	D000-DFFF	I/O address space or RAM, selected by -CHAREN. If the CHAREN bit is clear, the character generator ROM is chosen. If it is set, the internal RAM is accessible.
I/O	D000-DFFF	I/O address space. The -CHAREN line has no effect.
BASIC	A000-BFFF	BASIC ROM.
ROMH	A000-BFFF or E000-FFFF	External ROM with the -ROMH line connected to its -CS line.
ROML	8000-9FFF	External ROM with the -ROML line connected to its -CS line.
RAM	various ranges	Commodore 64's internal RAM.
-	1000-7FFF and A000-CFFF	Open address space. The Commodore 64's memory chips do not detect any memory accesses to this area except the VIC-II's DMA and memory refreshes.

- (1) This is the default BASIC memory map which provides BASIC 2.0 and 38K contiguous bytes of user RAM.
- (2) This map provides 60K bytes of RAM and I/O devices. The user must write his own I/O driver routines.
- (3) The same as 2, but the character ROM is not

accessible by the CPU in this map.

- (4) This map is intended for use with softload languages (including CP/M), providing 52K contiguous bytes of user RAM, I/O devices, and I/O driver routines.
- (5) This map gives access to all 64K bytes of RAM. The I/O devices must be banked back into the processor's address space for any I/O operation.
- (6) This is the standard configuration for a BASIC system with a BASIC expansion ROM. This map provides 32K contiguous bytes of user RAM and up to 8K bytes of BASIC "enhancement".
- (7) This map provides 40K contiguous bytes of user RAM and up to 8K bytes of plug-in ROM for special ROM-based applications which don't require BASIC.
- (8) This map provides 32K contiguous bytes of user RAM and up to 16K bytes of plug-in ROM for special applications which don't require BASIC (word processors, other languages, etc.).
- (9) This is the ULTIMAX video game memory map. Note that the 2K byte "expansion RAM" for the ULTIMAX, if required, is accessed out of the COMMODORE 64 and any RAM in the cartridge is ignored.

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

https://codebase64.org/doku.php?id=base:memory_management&rev=1527574720

Last update: **2018-05-29 08:18**

