Commodore 64 memory map

Address (hex, dec)	Description
$0000-$00FF, 0-255 Zero page	
$0000 0	Processor port data direction register. Bits: • Bit #x: 0 = Bit #x in processor port can only be read; 1 = Bit #x in processor port can be read and written. Default: $2F, %00101111.
$0001 1	Processor port. Bits: • Bits #0-#2: Configuration for memory areas $A000-$BFFF, $D000-$DFFF and $E000-$FFFF. Values: ◦ %x00: RAM visible in all three areas. ◦ %x01: RAM visible at $A000-$BFFF and $E000-$FFFF. ◦ %x10: RAM visible at $A000-$BFFF; KERNAL ROM visible at $E000-$FFFF. ◦ %x11: BASIC ROM visible at $A000-$BFFF; KERNAL ROM visible at $E000-$FFFF. ◦ %0xx: Character ROM visible at $D000-$DFFF. (Except for the value %000, see above.) ◦ %1xx: I/O area visible at $D000-$DFFF. (Except for the value %100, see above.) • Bit #3: Datasette output signal level. • Bit #4: Datasette button status; 0 = One or more of PLAY, RECORD, F.FWD or REW pressed; 1 = No button is pressed. • Bit #5: Datasette motor control; 0 = On; 1 = Off. Default: $37, %00110111.
$0002 2	Unused.
$0003-$0004 3-4	Unused. Default: $B1AA, execution address of routine converting floating point to integer.
$0005-$0006 5-6	Unused. Default: $B391, execution address of routine converting integer to floating point.
$0007 7	Byte being search for during various operations. Current digit of number being input. Low byte of first integer operand during AND and OR. Low byte of integer-format FAC during INT().
$0008 8	Byte being search for during various operations. Current byte of BASIC line during tokenization.

	High byte of first integer operand during AND and OR.
$0009 9	Current column number during SPC() and TAB().
$000A 10	LOAD/VERIFY switch. Values: • $00: LOAD. • $01-$FF: VERIFY.
$000B 11	Current token during tokenization. Length of BASIC line during insertion of line. AND/OR switch; $00 = AND; $FF = OR. Number of dimensions during array operations.
$000C 12	Switch for array operations. Values: • $00: Operation was not called by DIM. • $40-$7F: Operation was called by DIM.
$000D 13	Current expression type. Values: • $00: Numerical. • $FF: String.
$000E 14	Current numerical expression type. Bits: • Bit #7: 0 = Floating point; 1 = Integer.
$000F 15	Quotation mode switch during tokenization; Bit #6: 0 = Normal mode; 1 = Quotation mode. Quotation mode switch during LIST; $01 = Normal mode; $FE = Quotation mode. Garbage collection indicator during memory allocation for string variable; $00-$7F = There was no garbage collection yet; $80 = Garbage collection already took place.
$0010 16	Switch during fetch of variable name. Values: • $00: Integer variables are accepted. • $01-$FF: Integer variables are not accepted.
$0011 17	GET/INPUT/READ switch. Values: • $00: INPUT. • $40: GET. • $98: READ.
$0012 18	Sign during SIN() and TAN(). Values: • $00: Positive. • $FF: Negative.

$0013 19	Current I/O device number. Default: $00, keyboard for input and screen for output.
$0014-$0015 20-21	Line number during GOSUB, GOTO and RUN. Second line number during LIST. Memory address during PEEK, POKE, SYS and WAIT.
$0016 22	Pointer to next expression in string stack. Values: $19; $1C; $1F; $22. Default: $19.
$0017-$0018 23-24	Pointer to previous expression in string stack.
$0019-$0021 25-33	String stack, temporary area for processing string expressions (9 bytes, 3 entries).
$0022-$0025 34-37	Temporary area for various operations (4 bytes).
$0026-$0029 38-41	Auxiliary arithmetical register for division and multiplication (4 bytes).
$002A 42	Unused.
$002B-$002C 43-44	Pointer to beginning of BASIC area. Default: $0801, 2049.
$002D-$002E 45-46	Pointer to beginning of variable area. (End of program plus 1.)
$002F-$0030 47-48	Pointer to beginning of array variable area.
$0031-$0032 49-50	Pointer to end of array variable area.
$0033-$0034 51-52	Pointer to beginning of string variable area. (Grows downwards from end of BASIC area.)
$0035-$0036 53-54	Pointer to memory allocated for current string variable.
$0037-$0038 55-56	Pointer to end of BASIC area. Default: $A000, 40960.
$0039-$003A 57-58	Current BASIC line number. Values: • $0000-$F9FF, 0-63999: Line number. • $FF00-$FFFF: Direct mode, no BASIC program is being executed.
$003B-$003C 59-60	Current BASIC line number for CONT.
$003D-$003E 61-62	Pointer to next BASIC instruction for CONT. Values: • $0000-$00FF: CONT'ing is not possible. • $0100-$FFFF: Pointer to next BASIC instruction.
$003F-$0040 63-64	BASIC line number of current DATA item for READ.

$0041-$0042 65-66	Pointer to next DATA item for READ.
$0043-$0044 67-68	Pointer to input result during GET, INPUT and READ.
$0045-$0046 69-70	Name and type of current variable. Bits: • $0045 bits #0-#6: First character of variable name. • $0046 bits #0-#6: Second character of variable name; $00 = Variable name consists of only one character. • $0045 bit #7 and $0046 bit #7: ◦ %00: Floating-point variable. ◦ %01: String variable. ◦ %10: FN function, created with DEF FN. ◦ %11: Integer variable.
$0047-$0048 71-72	Pointer to value of current variable or FN function.
$0049-$004A 73-74	Pointer to value of current variable during LET. Value of second and third parameter during WAIT. Logical number and device number during OPEN. $0049, 73: Logical number of CLOSE. Device number of LOAD, SAVE and VERIFY.
$004B-$004C 75-76	Temporary area for saving original pointer to current BASIC instruction during GET, INPUT and READ.
$004D 77	Comparison operator indicator. Bits: • Bit #1: 1 = ">" (greater than) is present in expression. • Bit #2: 1 = "=" (equal to) is present in expression. • Bit #3: 1 = "<" (less than) is present in expression.
$004E-$004F 78-79	Pointer to current FN function.
$0050-$0051 80-81	Pointer to current string variable during memory allocation.
$0052 82	Unused.
$0053 83	Step size of garbage collection. Values: $03; $07.
$0054-$0056 84-86	JMP ABS machine instruction, jump to current BASIC function. $0055-$0056, 85-86: Execution address of current BASIC function.
$0057-$005B 87-91	Arithmetic register #3 (5 bytes).

$005C-$0060 92-96	Arithmetic register #4 (5 bytes).
$0061-$0065 97-101	FAC, arithmetic register #1 (5 bytes).
$0066 102	Sign of FAC. Bits: • Bit #7: 0 = Positive; 1 = Negative.
$0067 103	Number of degrees during polynomial evaluation.
$0068 104	Temporary area for various operations.
$0069-$006D 105-109	ARG, arithmetic register #2 (5 bytes).
$006E 110	Sign of ARG. Bits: • Bit #7: 0 = Positive; 1 = Negative.
$006F-$0070 111-112	Pointer to first string expression during string comparison.
$0071-$0072 113-114	Auxiliary pointer during array operations. Temporary area for saving original pointer to current BASIC instruction during VAL(). Pointer to current item of polynomial table during polynomial evaluation.
$0073-$008A 115-138	CHRGET. Machine code routine to read next byte from BASIC program or direct command (24 bytes). $0079, 121: CHRGOT. Read current byte from BASIC program or direct command. $007A-$007B, 122-123: Pointer to current byte in BASIC program or direct command.
$008B-$008F 139-143	Previous result of RND().
$0090 144	Value of ST variable, device status for serial bus and datasette input/output. Serial bus bits: • Bit #0: Transfer direction during which the timeout occured; 0 = Input; 1 = Output. • Bit #1: 1 = Timeout occurred. • Bit #4: 1 = VERIFY error occurred (only during VERIFY), the file read from the device did not match that in the memory. • Bit #6: 1 = End of file has been reached. • Bit #7: 1 = Device is not present. Datasette bits: • Bit #2: 1 = Block is too short (shorter than 192 bytes). • Bit #3: 1 = Block is too long (longer than 192 bytes). • Bit #4: 1 = Not all bytes read with error during pass 1 could be corrected during pass 2, or a VERIFY error occurred, the file read from the device did not match that in the memory.

	• Bit #5: 1 = Checksum error occurred. • Bit #6: 1 = End of file has been reached (only during reading data files).
$0091 145	Stop key indicator. Values: • $7F: Stop key is pressed. • $FF: Stop key is not pressed.
$0092 146	Unknown. (Timing constant during datasette input.)
$0093 147	LOAD/VERIFY switch. Values: • $00: LOAD. • $01-$FF: VERIFY.
$0094 148	Serial bus output cache status. Bits: • Bit #7: 1 = Output cache dirty, must transfer cache contents upon next output to serial bus.
$0095 149	Serial bus output cache, previous byte to be sent to serial bus.
$0096 150	Unknown. (End of tape indicator during datasette input/output.)
$0097 151	Temporary area for saving original value of Y register during input from RS232. Temporary area for saving original value of X register during input from datasette.
$0098 152	Number of files currently open. Values: $00-$0A, 0-10.
$0099 153	Current input device number. Default: $00, keyboard.
$009A 154	Current output device number. Default: $03, screen.
$009B 155	Unknown. (Parity bit during datasette input/output.)
$009C 156	Unknown. (Byte ready indicator during datasette input/output.)
$009D 157	System error display switch. Bits: • Bit #6: 0 = Suppress I/O error messages; 1 = Display them. • Bit #7: 0 = Suppress system messages; 1 = Display them.
$009E 158	Byte to be put into output buffer during RS232 and datasette output. Block header type during datasette input/output. Length of file name during datasette input/output. Error counter during LOAD from datasette. Values: $00-$3E, 0-62.

$009F 159	Auxiliary counter for writing file name into datasette buffer. Auxiliary counter for comparing requested file name with file name read from datasette during datasette input. Error correction counter during LOAD from datasette. Values: $00-$3E, 0-62.
$00A0-$00A2 160-162	Value of TI variable, time of day, increased by 1 every 1/60 second (on PAL machines). Values: $000000-$4F19FF, 0-518399 (on PAL machines).
$00A3 163	EOI switch during serial bus output. Bits: • Bit #7: 0 = Send byte right after handshake; 1 = Do EOI delay first. Bit counter during datasette output.
$00A4 164	Byte buffer during serial bus input. Parity during datasette input/output.
$00A5 165	Bit counter during serial bus input/output. Counter for sync mark during datasette output.
$00A6 166	Offset of current byte in datasette buffer.
$00A7 167	Bit buffer during RS232 input.
$00A8 168	Bit counter during RS232 input.
$00A9 169	Stop bit switch during RS232 input. Values: • $00: Data bit. • $01-$FF: Stop bit.
$00AA 170	Byte buffer during RS232 input.
$00AB 171	Parity during RS232 input. Computed block checksum during datasette input.
$00AC-$00AD 172-173	Start address for SAVE to serial bus. Pointer to current byte during SAVE to serial bus or datasette. Pointer to line in screen memory to be scrolled during scrolling the screen.
$00AE-$00AF 174-175	Load address read from input file and pointer to current byte during LOAD/VERIFY from serial bus. End address after LOAD/VERIFY from serial bus or datasette. End address for SAVE to serial bus or datasette. Pointer to line in Color RAM to be scrolled during scrolling the screen.
$00B0-$00B1 176-177	Unknown.
$00B2-$00B3 178-179	Pointer to datasette buffer. Default: $033C, 828.
$00B4 180	Bit counter and stop bit switch during RS232 output. Bits: • Bits #0-#6: Bit count. • Bit #7: 0 = Data bit; 1 = Stop bit. Bit counter during datasette input/output.

$00B5 181	Bit buffer (in bit #2) during RS232 output.
$00B6 182	Byte buffer during RS232 output.
$00B7 183	Length of file name or disk command; first parameter of LOAD, SAVE and VERIFY or fourth parameter of OPEN. Values: • $00: No parameter. • $01-$FF: Parameter length.
$00B8 184	Logical number of current file.
$00B9 185	Secondary address of current file.
$00BA 186	Device number of current file.
$00BB-$00BC 187-188	Pointer to current file name or disk command; first parameter of LOAD, SAVE and VERIFY or fourth parameter of OPEN.
$00BD 189	Parity during RS232 output. Byte buffer during datasette input/output.
$00BE 190	Block counter during datasette input/output.
$00BF 191	Unknown.
$00C0 192	Datasette motor switch. Values: • $00: No button was pressed, motor has been switched off. If a button is pressed on the datasette, must switch motor on. • $01-$FF: Motor is on.
$00C1-$00C2 193-194	Start address during SAVE to serial bus, LOAD and VERIFY from datasette and SAVE to datasette. Pointer to current byte during memory test.
$00C3-$00C4 195-196	Start address for a secondary address of 0 for LOAD and VERIFY from serial bus or datasette. Pointer to ROM table of default vectors during initialization of I/O vectors.
$00C5 197	Matrix code of key previously pressed. Values: • $00-$3F: Keyboard matrix code. • $40: No key was pressed at the time of previous check.
$00C6 198	Length of keyboard buffer. Values: • $00, 0: Buffer is empty. • $01-$0A, 1-10: Buffer length.

$00C7 199	Reverse mode switch. Values: • $00: Normal mode. • $12: Reverse mode.
$00C8 200	Length of line minus 1 during screen input. Values: $27, 39; $4F, 79.
$00C9 201	Cursor row during screen input. Values: $00-$18, 0-24.
$00CA 202	Cursor column during screen input. Values: $00-$27, 0-39.
$00CB 203	Matrix code of key currently being pressed. Values: • $00-$3F: Keyboard matrix code. • $40: No key is currently pressed.
$00CC 204	Cursor visibility switch. Values: • $00: Cursor is on. • $01-$FF: Cursor is off.
$00CD 205	Delay counter for changing cursor phase. Values: • $00, 0: Must change cursor phase. • $01-$14, 1-20: Delay.
$00CE 206	Screen code of character under cursor.
$00CF 207	Cursor phase switch. Values: • $00: Cursor off phase, original character visible. • $01: Cursor on phase, reverse character visible.
$00D0 208	End of line switch during screen input. Values: • $00: Return character reached, end of line. • $01-$FF: Still reading characters from line.
$00D1-$00D2 209-210	Pointer to current line in screen memory.
$00D3 211	Current cursor column. Values: $00-$27, 0-39.
$00D4 212	Quotation mode switch. Values: • $00: Normal mode.

	• $01: Quotation mode.
$00D5 213	Length of current screen line minus 1. Values: $27, 39; $4F, 79.
$00D6 214	Current cursor row. Values: $00-$18, 0-24.
$00D7 215	PETSCII code of character during screen input/output. Bit buffer during datasette input. Block checksum during datasette output.
$00D8 216	Number of insertions. Values: • $00: No insertions made, normal mode, control codes change screen layout or behavior. • $01-$FF: Number of insertions, when inputting this many character next, those must be turned into control codes, similarly to quotation mode.
$00D9-$00F1 217-241	High byte of pointers to each line in screen memory (25 bytes). Values: • $00-$7F: Pointer high byte. • $80-$FF: No pointer, line is an extension of previous line on screen.
$00F2 242	Temporary area during scrolling the screen.
$00F3-$00F4 243-244	Pointer to current line in Color RAM.
$00F5-$00F6 245-246	Pointer to current conversion table during conversion from keyboard matrix codes to PETSCII codes.
$00F7-$00F8 247-248	Pointer to RS232 input buffer. Values: • $0000-$00FF: No buffer defined, a new buffer must be allocated upon RS232 input. • $0100-$FFFF: Buffer pointer.
$00F9-$00FA 249-250	Pointer to RS232 output buffer. Values: • $0000-$00FF: No buffer defined, a new buffer must be allocated upon RS232 output. • $0100-$FFFF: Buffer pointer.
$00FB-$00FE 251-254	Unused (4 bytes).
$00FF-$010A 255-266	Buffer for conversion from floating point to string (12 bytes.)
$0100-$01FF, 256-511 Processor stack	
$00FF-$010A 255-266	Buffer for conversion from floating point to string (12 bytes.)
$0100-$013D 256-317	Pointers to bytes read with error during datasette input (62 bytes, 31 entries).

$0100-$01FF 256-511	Processor stack. Also used for storing data related to FOR and GOSUB.
$0200-$02FF, 512-767	
$0200-$0258 512-600	Input buffer, storage area for data read from screen (89 bytes).
$0259-$0262 601-610	Logical numbers assigned to files (10 bytes, 10 entries).
$0263-$026C 611-620	Device numbers assigned to files (10 bytes, 10 entries).
$026D-$0276 621-630	Secondary addresses assigned to files (10 bytes, 10 entries).
$0277-$0280 631-640	Keyboard buffer (10 bytes, 10 entries).
$0281-$0282 641-642	Pointer to beginning of BASIC area after memory test. Default: $0800, 2048.
$0283-$0284 643-644	Pointer to end of BASIC area after memory test. Default: $A000, 40960.
$0285 645	Unused. (Serial bus timeout.)
$0286 646	Current color, cursor color. Values: $00-$0F, 0-15.
$0287 647	Color of character under cursor. Values: $00-$0F, 0-15.
$0288 648	High byte of pointer to screen memory for screen input/output. Default: $04, $0400, 1024.
$0289 649	Maximum length of keyboard buffer. Values: • $00, 0: No buffer. • $01-$0F, 1-15: Buffer size.
$028A 650	Keyboard repeat switch. Bits: • Bits #6-#7: %00 = Only cursor up/down, cursor left/right, Insert/Delete and Space repeat; %01 = No key repeats; %1x = All keys repeat.
$028B 651	Delay counter during repeat sequence, for delaying between successive repeats. Values: • $00, 0: Must repeat key. • $01-$04, 1-4: Delay repetition.
$028C 652	Repeat sequence delay counter, for delaying before first repetition. Values: • $00, 0: Must start repeat sequence. • $01-$10, 1-16: Delay repeat sequence.

$028D 653	Shift key indicator. Bits: • Bit #0: 1 = One or more of left Shift, right Shift or Shift Lock is currently being pressed or locked. • Bit #1: 1 = Commodore is currently being pressed. • Bit #2: 1 = Control is currently being pressed.
$028E 654	Previous value of shift key indicator. Bits: • Bit #0: 1 = One or more of left Shift, right Shift or Shift Lock was pressed or locked at the time of previous check. • Bit #1: 1 = Commodore was pressed at the time of previous check. • Bit #2: 1 = Control was pressed at the time of previous check.
$028F-$0290 655-656	Execution address of routine that, based on the status of shift keys, sets the pointer at memory address $00F5-$00F6 to the appropriate conversion table for converting keyboard matrix codes to PETSCII codes. Default: $EB48.
$0291 657	Commodore-Shift switch. Bits: • Bit #7: 0 = Commodore-Shift is enabled, the key combination will toggle between the uppercase/graphics and lowercase/uppercase character set; 1 = Commodore-Shift is disabled.
$0292 658	Scroll direction switch during scrolling the screen. Values: • $00: Insertion of line before current line, current line and all lines below it must be scrolled 1 line downwards. • $01-$FF: Bottom of screen reached, complete screen must be scrolled 1 line upwards.
$0293 659	RS232 control register. Bits: • Bits #0-#3: Baud rate, transfer speed. Values: ◦ %0000: User specified. ◦ %0001: 50 bit/s. ◦ %0010: 75 bit/s. ◦ %0011: 110 bit/s. ◦ %0100: 150 bit/s. ◦ %0101: 300 bit/s. ◦ %0110: 600 bit/s. ◦ %0111: 1200 bit/s.

	◦ %1000: 2400 bit/s. ◦ %1001: 1800 bit/s. ◦ %1010: 2400 bit/s. ◦ %1011: 3600 bit/s. ◦ %1100: 4800 bit/s. ◦ %1101: 7200 bit/s. ◦ %1110: 9600 bit/s. ◦ %1111: 19200 bit/s. • Bits #5-#6: Byte size, number of data bits per byte; %00 = 8; %01 = 7, %10 = 6; %11 = 5. • Bit #7: Number of stop bits; 0 = 1 stop bit; 1 = 2 stop bits.
$0294 660	RS232 command register. Bits: • Bit #0: Synchronization type; 0 = 3 lines; 1 = X lines. • Bit #4: Transmission type; 0 = Duplex; 1 = Half duplex. • Bits #5-#7: Parity mode. Values: ◦ %xx0: No parity check, bit #7 does not exist. ◦ %001: Odd parity. ◦ %011: Even parity. ◦ %101: No parity check, bit #7 is always 1. ◦ %111: No parity check, bit #7 is always 0.
$0295-$0296 661-662	Default value of RS232 output timer, based on baud rate. (Must be filled with actual value before RS232 input/output if baud rate is "user specified" in RS232 control register, memory address $0293.)
$0297 663	Value of ST variable, device status for RS232 input/output. Bits: • Bit #0: 1 = Parity error occurred. • Bit #1: 1 = Frame error, a stop bit with the value of 0, occurred. • Bit #2: 1 = Input buffer underflow occurred, too much data has arrived but it has not been read from the buffer in time. • Bit #3: 1 = Input buffer is empty, nothing to read. • Bit #4: 0 = Sender is Clear To Send; 1 = Sender is not ready to send data to receiver. • Bit #6: 0 = Receiver reports Data Set Ready; 1 = Receiver is not ready to receive data.

	• Bit #7: 1 = Carrier loss, a stop bit and a data byte both with the value of 0, detected.
$0298 664	RS232 byte size, number of data bits per data byte, default value for bit counters.
$0299-$029A 665-666	Default value of RS232 input timer, based on baud rate. (Calculated automatically from default value of RS232 output timer, at memory address $0295-$0296.)
$029B 667	Offset of byte received in RS232 input buffer.
$029C 668	Offset of current byte in RS232 input buffer.
$029D 669	Offset of byte to send in RS232 output buffer.
$029E 670	Offset of current byte in RS232 output buffer.
$029F-$02A0 671-672	Temporary area for saving pointer to original interrupt service routine during datasette input output. Values: • $0000-$00FF: No datasette input/output took place yet or original pointer has been already restored. • $0100-$FFFF: Original pointer, datasette input/output currently in progress.
$02A1 673	Temporary area for saving original value of CIA#2 interrupt control register, at memory address $DD0D, during RS232 input/output.
$02A2 674	Temporary area for saving original value of CIA#1 timer #1 control register, at memory address $DC0E, during datasette input/output.
$02A3 675	Temporary area for saving original value of CIA#1 interrupt control register, at memory address $DC0D, during datasette input/output.
$02A4 676	Temporary area for saving original value of CIA#1 timer #1 control register, at memory address $DC0E, during datasette input/output.
$02A5 677	Number of line currently being scrolled during scrolling the screen.
$02A6 678	PAL/NTSC switch, for selecting RS232 baud rate from the proper table. Values: • $00: NTSC. • $01: PAL.
$02A7-$02FF 679-767	Unused (89 bytes).
$0300-$03FF, 768-1023	
$0300-$0301 768-769	Execution address of warm reset, displaying optional BASIC error message and entering BASIC idle loop. Default: $E38B.
$0302-$0303 770-771	Execution address of BASIC idle loop. Default: $A483.
$0304-$0305 772-773	Execution address of BASIC line tokenizater routine. Default: $A57C.

$0306-$0307 774-775	Execution address of BASIC token decoder routine. Default: $A71A.
$0308-$0309 776-777	Execution address of BASIC instruction executor routine. Default: $A7E4.
$030A-$030B 778-779	Execution address of routine reading next item of BASIC expression. Default: $AE86.
$030C 780	Default value of register A for SYS. Value of register A after SYS.
$030D 781	Default value of register X for SYS. Value of register X after SYS.
$030E 782	Default value of register Y for SYS. Value of register Y after SYS.
$030F 783	Default value of status register for SYS. Value of status register after SYS.
$0310-$0312 784-786	JMP ABS machine instruction, jump to USR() function. $0311-$0312, 785-786: Execution address of USR() function.
$0313 787	Unused.
$0314-$0315 788-789	Execution address of interrupt service routine. Default: $EA31.
$0316-$0317 790-791	Execution address of BRK service routine. Default: $FE66.
$0318-$0319 792-793	Execution address of non-maskable interrupt service routine. Default: $FE47.
$031A-$031B 794-795	Execution address of OPEN, routine opening files. Default: $F34A.
$031C-$031D 796-797	Execution address of CLOSE, routine closing files. Default: $F291.
$031E-$031F 798-799	Execution address of CHKIN, routine defining file as default input. Default: $F20E.
$0320-$0321 800-801	Execution address of CHKOUT, routine defining file as default output. Default: $F250.
$0322-$0323 802-803	Execution address of CLRCHN, routine initializating input/output. Default: $F333.
$0324-$0325 804-805	Execution address of CHRIN, data input routine, except for keyboard and RS232 input. Default: $F157.
$0326-$0327 806-807	Execution address of CHROUT, general purpose data output routine. Default: $F1CA.
$0328-$0329 808-809	Execution address of STOP, routine checking the status of Stop key indicator, at memory address $0091. Default: $F6ED.
$032A-$032B 810-811	Execution address of GETIN, general purpose data input routine. Default: $F13E.
$032C-$032D 812-813	Execution address of CLALL, routine initializing input/output and clearing all file assignment tables. Default: $F32F.

$032E-$032F 814-815	Unused. Default: $FE66.
$0330-$0331 816-817	Execution address of LOAD, routine loading files. Default: $F4A5.
$0332-$0333 818-819	Execution address of SAVE, routine saving files. Default: $F5ED.
$0334-$033B 820-827	Unused (8 bytes).
$033C-$03FB 828-1019	Datasette buffer (192 bytes).
$03FC-$03FF 1020-1023	Unused (4 bytes).
$0400-$07FF, 1024-2047 Default screen memory	
$0400-$07E7 1024-2023	Default area of screen memory (1000 bytes).
$07E8-$07F7 2024-2039	Unused (16 bytes).
$07F8-$07FF 2040-2047	Default area for sprite pointers (8 bytes).
$0800-$9FFF, 2048-40959 BASIC area	
$0800 2048	Unused. (Must contain a value of 0 so that the BASIC program can be RUN.)
$0801-$9FFF 2049-40959	Default BASIC area (38911 bytes).
$8000-$9FFF 32768-40959	Optional cartridge ROM (8192 bytes). $8000-$8001, 32768-32769: Execution address of cold reset. $8002-$8003, 32770-32771: Execution address of non-maskable interrupt service routine. $8004-$8008, 32772-32776: Cartridge signature. If contains the uppercase PETSCII string "CBM80" ($C3,$C2,$CD,$38,$30) then the routine vectors are accepted by the KERNAL.
$A000-$BFFF, 40960-49151 BASIC ROM	
$A000-$BFFF 40960-49151	BASIC ROM or RAM area (8192 bytes); depends on the value of bits #0-#2 of the processor port at memory address $0001: • %x00, %x01 or %x10: RAM area. • %x11: BASIC ROM.
$C000-$CFFF, 49152-53247 Upper RAM area	
$C000-$CFFF 49152-53247	Upper RAM area (4096 bytes).
$D000-$DFFF, 53248-57343 I/O Area	
$D000-$DFFF 53248-57343	I/O Area (memory mapped chip registers), Character ROM or RAM area (4096 bytes); depends on the value of bits #0-#2 of the processor port at memory address $0001:

	• %x00: RAM area. • %0xx: Character ROM. (Except for the value %000, see above.) • %1xx: I/O Area. (Except for the value %100, see above.)
$D000-$DFFF, 53248-57343 Character ROM	
$D000-$DFFF 53248-57343	Character ROM, shape of characters (4096 bytes).
$D000-$D7FF 53248-55295	Shape of characters in uppercase/graphics character set (2048 bytes, 256 entries).
$D800-$DFFF 55295-57343	Shape of characters in lowercase/uppercase character set (2048 bytes, 256 entries).
$D000-$D3FF, 53248-54271 VIC-II; video display	
$D000 53248	Sprite #0 X-coordinate (only bits #0-#7).
$D001 53249	Sprite #0 Y-coordinate.
$D002 53250	Sprite #1 X-coordinate (only bits #0-#7).
$D003 53251	Sprite #1 Y-coordinate.
$D004 53252	Sprite #2 X-coordinate (only bits #0-#7).
$D005 53253	Sprite #2 Y-coordinate.
$D006 53254	Sprite #3 X-coordinate (only bits #0-#7).
$D007 53255	Sprite #3 Y-coordinate.
$D008 53256	Sprite #4 X-coordinate (only bits #0-#7).
$D009 53257	Sprite #4 Y-coordinate.
$D00A 53258	Sprite #5 X-coordinate (only bits #0-#7).
$D00B 53259	Sprite #5 Y-coordinate.
$D00C 53260	Sprite #6 X-coordinate (only bits #0-#7).
$D00D 53261	Sprite #6 Y-coordinate.
$D00E 53262	Sprite #7 X-coordinate (only bits #0-#7).

$D00F 53263	Sprite #7 Y-coordinate.
$D010 53264	Sprite #0-#7 X-coordinates (bit #8). Bits: • Bit #x: Sprite #x X-coordinate bit #8.
$D011 53265	Screen control register #1. Bits: • Bits #0-#2: Vertical raster scroll. • Bit #3: Screen height; 0 = 24 rows; 1 = 25 rows. • Bit #4: 0 = Screen off, complete screen is covered by border; 1 = Screen on, normal screen contents are visible. • Bit #5: 0 = Text mode; 1 = Bitmap mode. • Bit #6: 1 = Extended background mode on. • Bit #7: Read: Current raster line (bit #8). Write: Raster line to generate interrupt at (bit #8). Default: $1B, %00011011.
$D012 53266	Read: Current raster line (bits #0-#7). Write: Raster line to generate interrupt at (bits #0-#7).
$D013 53267	Light pen X-coordinate (bits #1-#8). Read-only.
$D014 53268	Light pen Y-coordinate. Read-only.
$D015 53269	Sprite enable register. Bits: • Bit #x: 1 = Sprite #x is enabled, drawn onto the screen.
$D016 53270	Screen control register #2. Bits: • Bits #0-#2: Horizontal raster scroll. • Bit #3: Screen width; 0 = 38 columns; 1 = 40 columns. • Bit #4: 1 = Multicolor mode on. Default: $C8, %11001000.
$D017 53271	Sprite double height register. Bits: • Bit #x: 1 = Sprite #x is stretched to double height.
$D018 53272	Memory setup register. Bits: • Bits #1-#3: In text mode, pointer to character memory (bits #11-#13), relative to VIC bank, memory address $DD00. Values: ◦ %000, 0: $0000-$07FF, 0-2047.

- %001, 1: $0800-$0FFF, 2048-4095.
- %010, 2: $1000-$17FF, 4096-6143.
- %011, 3: $1800-$1FFF, 6144-8191.
- %100, 4: $2000-$27FF, 8192-10239.
- %101, 5: $2800-$2FFF, 10240-12287.
- %110, 6: $3000-$37FF, 12288-14335.
- %111, 7: $3800-$3FFF, 14336-16383.

Values %010 and %011 in VIC bank #0 and #2 select Character ROM instead.
In bitmap mode, pointer to bitmap memory (bit #13), relative to VIC bank, memory address $DD00. Values:

- %0xx, 0: $0000-$1FFF, 0-8191.
- %1xx, 4: $2000-$3FFF, 8192-16383.

- Bits #4-#7: Pointer to screen memory (bits #10-#13), relative to VIC bank, memory address $DD00. Values:
  - %0000, 0: $0000-$03FF, 0-1023.
  - %0001, 1: $0400-$07FF, 1024-2047.
  - %0010, 2: $0800-$0BFF, 2048-3071.
  - %0011, 3: $0C00-$0FFF, 3072-4095.
  - %0100, 4: $1000-$13FF, 4096-5119.
  - %0101, 5: $1400-$17FF, 5120-6143.
  - %0110, 6: $1800-$1BFF, 6144-7167.
  - %0111, 7: $1C00-$1FFF, 7168-8191.
  - %1000, 8: $2000-$23FF, 8192-9215.
  - %1001, 9: $2400-$27FF, 9216-10239.
  - %1010, 10: $2800-$2BFF, 10240-11263.
  - %1011, 11: $2C00-$2FFF, 11264-12287.
  - %1100, 12: $3000-$33FF, 12288-13311.
  - %1101, 13: $3400-$37FF, 13312-14335.
  - %1110, 14: $3800-$3BFF, 14336-15359.
  - %1111, 15: $3C00-$3FFF, 15360-16383.

$D019 53273	Interrupt status register. Read bits: • Bit #0: 1 = Current raster line is equal to the raster line to generate interrupt at. • Bit #1: 1 = Sprite-background collision occurred. • Bit #2: 1 = Sprite-sprite collision occurred. • Bit #3: 1 = Light pen signal arrived. • Bit #7: 1 = An event (or more events), that may generate an interrupt, occurred and it has not been (not all of them have been) acknowledged yet. Write bits: • Bit #0: 1 = Acknowledge raster interrupt. • Bit #1: 1 = Acknowledge sprite-background collision interrupt. • Bit #2: 1 = Acknowledge sprite-sprite collision interrupt. • Bit #3: 1 = Acknowledge light pen interrupt.
$D01A 53274	Interrupt control register. Bits: • Bit #0: 1 = Raster interrupt enabled. • Bit #1: 1 = Sprite-background collision interrupt enabled. • Bit #2: 1 = Sprite-sprite collision interrupt enabled. • Bit #3: 1 = Light pen interrupt enabled.
$D01B 53275	Sprite priority register. Bits: • Bit #x: 0 = Sprite #x is drawn in front of screen contents; 1 = Sprite #x is behind screen contents.
$D01C 53276	Sprite multicolor mode register. Bits: • Bit #x: 0 = Sprite #x is single color; 1 = Sprite #x is multicolor.
$D01D 53277	Sprite double width register. Bits: • Bit #x: 1 = Sprite #x is stretched to double width.
$D01E 53278	Sprite-sprite collision register. Read bits: • Bit #x: 1 = Sprite #x collided with another sprite. Write: Enable further detection of sprite-sprite collisions.
$D01F 53279	Sprite-background collision register. Read bits: • Bit #x: 1 = Sprite #x collided with background.

	Write: Enable further detection of sprite-background collisions.
$D020 53280	Border color (only bits #0-#3).
$D021 53281	Background color (only bits #0-#3).
$D022 53282	Extra background color #1 (only bits #0-#3).
$D023 53283	Extra background color #2 (only bits #0-#3).
$D024 53284	Extra background color #3 (only bits #0-#3).
$D025 53285	Sprite extra color #1 (only bits #0-#3).
$D026 53286	Sprite extra color #2 (only bits #0-#3).
$D027 53287	Sprite #0 color (only bits #0-#3).
$D028 53288	Sprite #1 color (only bits #0-#3).
$D029 53289	Sprite #2 color (only bits #0-#3).
$D02A 53290	Sprite #3 color (only bits #0-#3).
$D02B 53291	Sprite #4 color (only bits #0-#3).
$D02C 53292	Sprite #5 color (only bits #0-#3).
$D02D 53293	Sprite #6 color (only bits #0-#3).
$D02E 53294	Sprite #7 color (only bits #0-#3).
$D02F-$D03F 53295-53311	Unusable (17 bytes).
$D040-$D3FF 53312-54271	VIC-II register images (repeated every $40, 64 bytes).
$D400-$D7FF, 54272-55295 SID; audio	
$D400-$D401 54272-54273	Voice #1 frequency. Write-only.
$D402-$D403 54274-54275	Voice #1 pulse width. Write-only.
$D404 54276	Voice #1 control register. Bits: • Bit #0: 0 = Voice off, Release cycle; 1 = Voice on, Attack-Decay-Sustain cycle. • Bit #1: 1 = Synchronization enabled.

	• Bit #2: 1 = Ring modulation enabled. • Bit #3: 1 = Disable voice, reset noise generator. • Bit #4: 1 = Triangle waveform enabled. • Bit #5: 1 = Saw waveform enabled. • Bit #6: 1 = Rectangle waveform enabled. • Bit #7: 1 = Noise enabled. Write-only.
$D405 54277	Voice #1 Attack and Decay length. Bits: • Bits #0-#3: Decay length. Values: ◦ %0000, 0: 6 ms. ◦ %0001, 1: 24 ms. ◦ %0010, 2: 48 ms. ◦ %0011, 3: 72 ms. ◦ %0100, 4: 114 ms. ◦ %0101, 5: 168 ms. ◦ %0110, 6: 204 ms. ◦ %0111, 7: 240 ms. ◦ %1000, 8: 300 ms. ◦ %1001, 9: 750 ms. ◦ %1010, 10: 1.5 s. ◦ %1011, 11: 2.4 s. ◦ %1100, 12: 3 s. ◦ %1101, 13: 9 s. ◦ %1110, 14: 15 s. ◦ %1111, 15: 24 s. • Bits #4-#7: Attack length. Values: ◦ %0000, 0: 2 ms. ◦ %0001, 1: 8 ms. ◦ %0010, 2: 16 ms.

	◦ %0011, 3: 24 ms. ◦ %0100, 4: 38 ms. ◦ %0101, 5: 56 ms. ◦ %0110, 6: 68 ms. ◦ %0111, 7: 80 ms. ◦ %1000, 8: 100 ms. ◦ %1001, 9: 250 ms. ◦ %1010, 10: 500 ms. ◦ %1011, 11: 800 ms. ◦ %1100, 12: 1 s. ◦ %1101, 13: 3 s. ◦ %1110, 14: 5 s. ◦ %1111, 15: 8 s. Write-only.
$D406 54278	Voice #1 Sustain volume and Release length. Bits: • Bits #0-#3: Release length. Values: ◦ %0000, 0: 6 ms. ◦ %0001, 1: 24 ms. ◦ %0010, 2: 48 ms. ◦ %0011, 3: 72 ms. ◦ %0100, 4: 114 ms. ◦ %0101, 5: 168 ms. ◦ %0110, 6: 204 ms. ◦ %0111, 7: 240 ms. ◦ %1000, 8: 300 ms. ◦ %1001, 9: 750 ms. ◦ %1010, 10: 1.5 s. ◦ %1011, 11: 2.4 s. ◦ %1100, 12: 3 s.

	◦ %1101, 13: 9 s. ◦ %1110, 14: 15 s. ◦ %1111, 15: 24 s. • Bits #4-#7: Sustain volume. Write-only.
$D407-$D408 54279-54280	Voice #2 frequency. Write-only.
$D409-$D40A 54281-54282	Voice #2 pulse width. Write-only.
$D40B 54283	Voice #2 control register. Write-only.
$D40C 54284	Voice #2 Attack and Decay length. Write-only.
$D40D 54285	Voice #2 Sustain volume and Release length. Write-only.
$D40E-$D40F 54286-54287	Voice #3 frequency. Write-only.
$D410-$D411 54288-54289	Voice #3 pulse width. Write-only.
$D412 54290	Voice #3 control register. Write-only.
$D413 54291	Voice #3 Attack and Decay length. Write-only.
$D414 54292	Voice #3 Sustain volume and Release length. Write-only.
$D415 54293	Filter cut off frequency (bits #0-#2). Write-only.
$D416 54294	Filter cut off frequency (bits #3-#10). Write-only.
$D417 54295	Filter control. Bits: • Bit #0: 1 = Voice #1 filtered. • Bit #1: 1 = Voice #2 filtered. • Bit #2: 1 = Voice #3 filtered. • Bit #3: 1 = External voice filtered. • Bits #4-#7: Filter resonance. Write-only.
$D418 54296	Volume and filter modes. Bits: • Bits #0-#3: Volume. • Bit #4: 1 = Low pass filter enabled.

	• Bit #5: 1 = Band pass filter enabled. • Bit #6: 1 = High pass filter enabled. • Bit #7: 1 = Voice #3 disabled. Write-only.
$D419 54297	X value of paddle selected at memory address $DC00. (Updates at every 512 system cycles.) Read-only.
$D41A 54298	Y value of paddle selected at memory address $DC00. (Updates at every 512 system cycles.) Read-only.
$D41B 54299	Voice #3 waveform output. Read-only.
$D41C 54300	Voice #3 ADSR output. Read-only.
$D41D-$D41F 54301-54303	Unusable (3 bytes).
$D420-$D7FF 54304-55295	SID register images (repeated every $20, 32 bytes).
$D800-$DBFF, 55296-56319 Color RAM	
$D800-$DBE7 55296-56295	Color RAM (1000 bytes, only bits #0-#3).
$DBE8-$DBFF 56296-56319	Unused (24 bytes, only bits #0-#3).
$DC00-$DCFF, 56320-56575 CIA#1; inputs (keyboard, joystick, mouse), datasette, IRQ control	
$DC00 56320	Port A, keyboard matrix columns and joystick #2. Read bits: • Bit #0: 0 = Port 2 joystick up pressed. • Bit #1: 0 = Port 2 joystick down pressed. • Bit #2: 0 = Port 2 joystick left pressed. • Bit #3: 0 = Port 2 joystick right pressed. • Bit #4: 0 = Port 2 joystick fire pressed. Write bits: • Bit #x: 0 = Select keyboard matrix column #x. • Bits #6-#7: Paddle selection; %01 = Paddle #1; %10 = Paddle #2.
$DC01 56321	Port B, keyboard matrix rows and joystick #1. Bits: • Bit #x: 0 = A key is currently being pressed in keyboard matrix row #x, in the column selected at memory address $DC00. • Bit #0: 0 = Port 1 joystick up pressed.

	• Bit #1: 0 = Port 1 joystick down pressed. • Bit #2: 0 = Port 1 joystick left pressed. • Bit #3: 0 = Port 1 joystick right pressed. • Bit #4: 0 = Port 1 joystick fire pressed.
$DC02 56322	Port A data direction register. • Bit #x: 0 = Bit #x in port A can only be read; 1 = Bit #x in port A can be read and written.
$DC03 56323	Port B data direction register. • Bit #x: 0 = Bit #x in port B can only be read; 1 = Bit #x in port B can be read and written.
$DC04-$DC05 56324-56325	Timer A. Read: Current timer value. Write: Set timer start value.
$DC06-$DC07 56326-56327	Timer B. Read: Current timer value. Write: Set timer start value.
$DC08 56328	Time of Day, tenth seconds (in BCD). Values: $00-$09. Read: Current TOD value. Write: Set TOD or alarm time.
$DC09 56329	Time of Day, seconds (in BCD). Values: $00-$59. Read: Current TOD value. Write: Set TOD or alarm time.
$DC0A 56330	Time of Day, minutes (in BCD). Values: $00-$59. Read: Current TOD value. Write: Set TOD or alarm time.
$DC0B 56331	Time of Day, hours (in BCD). Read bits: • Bits #0-#5: Hours. • Bit #7: 0 = AM; 1 = PM. Write: Set TOD or alarm time.
$DC0C 56332	Serial shift register. (Bits are read and written upon every positive edge of the CNT pin.)
$DC0D 56333	Interrupt control and status register. Read bits: • Bit #0: 1 = Timer A underflow occurred. • Bit #1: 1 = Timer B underflow occurred. • Bit #2: 1 = TOD is equal to alarm time. • Bit #3: 1 = A complete byte has been received into or sent from serial shift register. • Bit #4: Signal level on FLAG pin, datasette input. • Bit #7: An interrupt has been generated. Write bits:

	• Bit #0: 1 = Enable interrupts generated by timer A underflow. • Bit #1: 1 = Enable interrupts generated by timer B underflow. • Bit #2: 1 = Enable TOD alarm interrupt. • Bit #3: 1 = Enable interrupts generated by a byte having been received/sent via serial shift register. • Bit #4: 1 = Enable interrupts generated by positive edge on FLAG pin. • Bit #7: Fill bit; bits #0-#6, that are set to 1, get their values from this bit; bits #0-#6, that are set to 0, are left unchanged.
$DC0E 56334	Timer A control register. Bits: • Bit #0: 0 = Stop timer; 1 = Start timer. • Bit #1: 1 = Indicate timer underflow on port B bit #6. • Bit #2: 0 = Upon timer underflow, invert port B bit #6; 1 = upon timer underflow, generate a positive edge on port B bit #6 for 1 system cycle. • Bit #3: 0 = Timer restarts upon underflow; 1 = Timer stops upon underflow. • Bit #4: 1 = Load start value into timer. • Bit #5: 0 = Timer counts system cycles; 1 = Timer counts positive edges on CNT pin. • Bit #6: Serial shift register direction; 0 = Input, read; 1 = Output, write. • Bit #7: TOD speed; 0 = 60 Hz; 1 = 50 Hz.
$DC0F 56335	Timer B control register. Bits: • Bit #0: 0 = Stop timer; 1 = Start timer. • Bit #1: 1 = Indicate timer underflow on port B bit #7. • Bit #2: 0 = Upon timer underflow, invert port B bit #7; 1 = upon timer underflow, generate a positive edge on port B bit #7 for 1 system cycle. • Bit #3: 0 = Timer restarts upon underflow; 1 = Timer stops upon underflow. • Bit #4: 1 = Load start value into timer. • Bits #5-#6: %00 = Timer counts system cycles; %01 = Timer counts positive edges on CNT pin; %10 = Timer counts underflows of timer A; %11 = Timer counts underflows of timer A occurring along with a positive edge on CNT pin. • Bit #7: 0 = Writing into TOD registers sets TOD; 1 = Writing into TOD registers sets alarm time.
$DC10-$DCFF 56336-56575	CIA#1 register images (repeated every $10, 16 bytes).

$DD00-$DDFF, 56576-56831 CIA#2; serial bus, RS232, NMI control	

$DD00
56576
	Port A, serial bus access. Bits:
	• Bits #0-#1: VIC bank. Values:
	  ◦ %00, 0: Bank #3, $C000-$FFFF, 49152-65535.
	  ◦ %01, 1: Bank #2, $8000-$BFFF, 32768-49151.
	  ◦ %10, 2: Bank #1, $4000-$7FFF, 16384-32767.
	  ◦ %11, 3: Bank #0, $0000-$3FFF, 0-16383.
	• Bit #2: RS232 TXD line, output bit.
	• Bit #3: Serial bus ATN OUT; 0 = High; 1 = Low.
	• Bit #4: Serial bus CLOCK OUT; 0 = High; 1 = Low.
	• Bit #5: Serial bus DATA OUT; 0 = High; 1 = Low.
	• Bit #6: Serial bus CLOCK IN; 0 = Low; 1 = High.
	• Bit #7: Serial bus DATA IN; 0 = Low; 1 = High.

$DD01 56577	Port B, RS232 access. Read bits: • Bit #0: RS232 RXD line, input bit. • Bit #3: RS232 RI line. • Bit #4: RS232 DCD line. • Bit #5: User port H pin. • Bit #6: RS232 CTS line; 1 = Sender is ready to send. • Bit #7: RS232 DSR line; 1 = Receiver is ready to receive. Write bits: • Bit #1: RS232 RTS line. 1 = Sender is ready to send. • Bit #2: RS232 DTR line. 1 = Receiver is ready to receive. • Bit #3: RS232 RI line. • Bit #4: RS232 DCD line. • Bit #5: User port H pin.

$DD02 56578	Port A data direction register. • Bit #x: 0 = Bit #x in port A can only be read; 1 = Bit #x in port A can be read and written.
$DD03 56579	Port B data direction register. • Bit #x: 0 = Bit #x in port B can only be read; 1 = Bit #x in port B can be read and written.
$DD04-$DD05 56580-56581	Timer A. Read: Current timer value. Write: Set timer start value.
$DD06-$DD07 56582-56583	Timer B. Read: Current timer value. Write: Set timer start value.
$DD08 56584	Time of Day, tenth seconds (in BCD). Values: $00-$09. Read: Current TOD value. Write: Set TOD or alarm time.
$DD09 56585	Time of Day, seconds (in BCD). Values: $00-$59. Read: Current TOD value. Write: Set TOD or alarm time.
$DD0A 56586	Time of Day, minutes (in BCD). Values: $00-$59. Read: Current TOD value. Write: Set TOD or alarm time.
$DD0B 56587	Time of Day, hours (in BCD). Read bits: • Bits #0-#5: Hours. • Bit #7: 0 = AM; 1 = PM. Write: Set TOD or alarm time.
$DD0C 56588	Serial shift register. (Bits are read and written upon every positive edge of the CNT pin.)
$DD0D 56589	Interrupt control and status register. Read bits: • Bit #0: 1 = Timer A underflow occurred. • Bit #1: 1 = Timer B underflow occurred. • Bit #2: 1 = TOD is equal to alarm time. • Bit #3: 1 = A complete byte has been received into or sent from serial shift register. • Bit #4: Signal level on FLAG pin. • Bit #7: A non-maskable interrupt has been generated. Write bits: • Bit #0: 1 = Enable non-maskable interrupts generated by timer A underflow. • Bit #1: 1 = Enable non-maskable interrupts generated by timer B underflow. • Bit #2: 1 = Enable TOD alarm non-maskable interrupt. • Bit #3: 1 = Enable non-maskable interrupts generated by a byte having been received/sent via serial shift register.

	• Bit #4: 1 = Enable non-maskable interrupts generated by positive edge on FLAG pin. • Bit #7: Fill bit; bits #0-#6, that are set to 1, get their values from this bit; bits #0-#6, that are set to 0, are left unchanged.
$DD0E 56590	Timer A control register. Bits: • Bit #0: 0 = Stop timer; 1 = Start timer. • Bit #1: 1 = Indicate timer underflow on port B bit #6. • Bit #2: 0 = Upon timer underflow, invert port B bit #6; 1 = upon timer underflow, generate a positive edge on port B bit #6 for 1 system cycle. • Bit #3: 0 = Timer restarts upon underflow; 1 = Timer stops upon underflow. • Bit #4: 1 = Load start value into timer. • Bit #5: 0 = Timer counts system cycles; 1 = Timer counts positive edges on CNT pin. • Bit #6: Serial shift register direction; 0 = Input, read; 1 = Output, write. • Bit #7: TOD speed; 0 = 60 Hz; 1 = 50 Hz.
$DD0F 56591	Timer B control register. Bits: • Bit #0: 0 = Stop timer; 1 = Start timer. • Bit #1: 1 = Indicate timer underflow on port B bit #7. • Bit #2: 0 = Upon timer underflow, invert port B bit #7; 1 = upon timer underflow, generate a positive edge on port B bit #7 for 1 system cycle. • Bit #3: 0 = Timer restarts upon underflow; 1 = Timer stops upon underflow. • Bit #4: 1 = Load start value into timer. • Bits #5-#6: %00 = Timer counts system cycles; %01 = Timer counts positive edges on CNT pin; %10 = Timer counts underflows of timer A; %11 = Timer counts underflows of timer A occurring along with a positive edge on CNT pin. • Bit #7: 0 = Writing into TOD registers sets TOD; 1 = Writing into TOD registers sets alarm time.
$DD10-$DDFF 56592-56831	CIA#2 register images (repeated every $10, 16 bytes).
$DE00-$DEFF, 56832-57087 I/O Area #1	
$DE00-$DEFF 56832-57087	I/O Area #1, memory mapped registers or machine code routines of optional external devices (256 bytes). Layout and contents depend on the actual device.
$DF00-$DFFF, 57088-57343 I/O Area #2	

$DF00-$DFFF 57088-57343	I/O Area #2, memory mapped registers or machine code routines of optional external devices (256 bytes). Layout and contents depend on the actual device.
$E000-$FFFF, 57344-65535 KERNAL ROM	
$E000-$FFFF 57344-65535	KERNAL ROM or RAM area (8192 bytes); depends on the value of bits #0-#2 of the processor port at memory address $0001: • %x0x: RAM area. • %x1x: KERNAL ROM.
$FFFA-$FFFF, 65530-65535 Hardware vectors	
$FFFA-$FFFB 65530-65531	Execution address of non-maskable interrupt service routine. Default: $FE43.
$FFFC-$FFFD 65532-65533	Execution address of cold reset. Default: $FCE2.
$FFFE-$FFFF 65534-65535	Execution address of interrupt service routine. Default: $FF48.