$02 – Register 02 – Plane A Name Table Location							
7	6	5	4	3	2	1	0
X	**A**	**a**	**a**	**a**	x	x	x
Aaaa	This 17-bit address, %**A aaa**0 0000 0000 0000, must be a multiple of **$2000** and shifted to the right **10** bits. For example, $E000 >>10. "**A**" is only valid in 128K mode.						

$03 – Register 03 – Window Name Table Location							
7	6	5	4	3	2	1	0
x	**A**	**a**	**a**	**a**	**a**	**a**	x
Aaaaaa	This 17-bit address, %**A aaaa a**000 0000 0000, must be a multiple of **$0800** and shifted to the right **10** bits. For example, $F800 >>10. "**A**" is only valid in 128K mode.						

$04 – Register 04 – Plane B Name Table Location							
7	6	5	4	3	2	1	0
x	x	x	x	**A**	**a**	**a**	**a**
Aaaa	This 17-bit address, %**A aaa**0 0000 0000 0000, must be a multiple of **$2000** and shifted to the right **13** bits. For example, $E000 >>13. "**A**" is only valid in 128K mode.						

$05 – Register 05 – Sprite Attribute Table Location							
7	6	5	4	3	2	1	0
A	**a**	**a**	**a**	**a**	**a**	**a**	**a**
Aaaaaaaa	This 17-bit address, %**A aaaa aaa**0 0000 0000, must be a multiple of **$0200** and shifted to the right **9** bits. For example, $FE00 >>9. "**A**" is only valid in 128K mode.						

$06 – Register 06 – Sprite Pattern Base Address							
7	6	5	4	3	2	1	0
x	x	**A**	x	x	x	x	x
A	This 17-bit address, %**A** 0000 0000 0000 0000, must be a multiple of **$10000** and shifted to the right **11** bits. For example, $10000 >>11. "**A**" is only valid in 128K mode.						

$07 – Register 07 – Background Color							
7	6	5	4	3	2	1	0
x	**x**	**P**	**P**	**p**	**p**	**p**	**p**
PPpppp	**PP** is the palette number (0-3), and **pppp** is the palette index (0-15). Note that transparent color indices (0, 16, 32, and 48) can also be used and will display normally.						

Sprite Attribute Table Format															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0-1							*y*	*y*	*y*	*y*	*y*	*y*	*y*	*y*	*y*	*y*
2-3					h	h	v	v		l	l	l	l	l	l	l
4-5	r	P	P	*y*	x	n	n	n	n	n	n	n	n	n	n	n
6-7							x	x	x	x	x	x	x	x	x	x

yyyyyyyyyy	The sprite's Y position values are from 0 through 551, but the displayable values start at 128 and go up to 351 (**V28**) or 367 (**V30**) depending on vertical mode. When interlace mode 2 is enabled (to double vertical resolution) the sprite's Y position range doubles, with the displayable area starting at 256 to 703 (**V28**) or 735 (**V30**) depending on vertical mode.

hhvv

hh ↓ / vv →	%00 = 1	%01 = 2	%10 = 3	%11 = 4
%00 = 1	**0**	**0** **1**	**0** **1** **2**	**0** **1** **2** **3**
%01 = 2	**0** **1**	**0** **2** **1** **3**	**0** **3** **1** **4** **2** **5**	**0** **4** **1** **5** **2** **6** **3** **7**
%10 = 3	**0** **1** **2**	**0** **2** **4** **1** **3** **5**	**0** **3** **6** **1** **4** **7** **2** **5** **8**	**0** **4** **8** **1** **5** **9** **2** **6** **A** **3** **7** **B**
%11 = 4	**0** **1** **2** **3**	**0** **2** **4** **6** **1** **3** **5** **7**	**0** **3** **6** **9** **1** **4** **7** **A** **2** **5** **8** **B**	**0** **4** **8** **C** **1** **5** **9** **D** **2** **6** **A** **E** **3** **7** **B** **F**

lllllll 0-63 in H32 0-79 in H40	The link field points to the *next* sprite entry to draw; the **last sprite's** link field *must* be set to **0**. The VDP will draw the sprite, then go on to draw the next one indicated by the link. Sprites that come *later* in the linked list have higher "priority" – they will be drawn *above* any other previously drawn sprites. However, setting the priority flag **r** will cause sprite to be drawn above non-prioritized ones regardless.
r	**Set** priority to force appearance above all other non-prioritized sprites, or prioritized sprites at an earlier position in the linked list.
PP	The palette number allows displaying the sprite in one of four different palettes.
y	Set to flip the image vertically.
x	Set to flip the image horizontally.
nnnnnnnnnnn 0-2047	Set the tile index of the sprite's desired pattern in VRAM. If composed of multiple tiles, patterns come from subsequent tiles (see above). Note that 2048 × 32 bytes (for each tile's pattern) = 64K, allowing you to use the entire VRAM for patterns.
xxxxxxxxxx	When a sprite's X position is set to 0, it can be used to mask all other sprites on that same scanline, so long as they have low priority. The range X values is 0 through 551, but the displayable X values begins at 128, up to 383 or 447, depending on horizontal mode.

$00 – Register 00 – Mode Register 1							
7	6	5	4	3	2	1	0
0	0	LBE	HBI	0	1	HV0	DE

LBE	**L**eft **B**order **E**xtend. **Set** to blank leftmost 8-pixel wide tile-column.
HBI	**H**orizontal **B**lank **I**nterrupt. Set to generate a M68K level 4 interrupt every *nth* scanline.
HV0 *aka M3*	**Set** to *stop* modifying the **H/V Counter**, the current location of the electron beam. **Reset** to actively update the counter.
DE	**D**isplay **E**nable. See *Mode Register 2*. This is usually **0**. If **1**, it can be used to overlay video from the CSync pin.

$01 – Register 01 – Mode Register 2							
7	6	5	4	3	2	1	0
64K	DE	VBI	DMA	V30	MD	0	0

64K	**Set** to use an additional 64K of external VRAM.
DE	**D**isplay **E**nable. **Set** to enable, **reset** to disable display. Use to perform quick art loading.
VBI *aka* **IE0**	**V**ertical **B**lank **I**nterrupt. Set to generate an M68K Level 6 interrupt at VBLANK.
DMA	**Set** to allow DMA, **reset** to disable all DMA. This bit also masks **CD5** in the VDP control word if cleared (?).
V30	**Set** for PAL-exclusive 30-tile (240 pixel) mode, **reset** for 28-tile (224 pixel) mode (**V28**).
MD	**Set** for Mega Drive (Mode 5) video mode. **Reset** for Master System (Mode 4) video mode.

$0B – Register 11 – Mode Register 3							
7	6	5	4	3	2	1	0
0	0	0	0	TRI	V	H	H

TRI *aka* **IE2**	**Set** to generate a M68K Level 2 interrupt when any *TH* pin is pulled *high*, such as when a light gun's trigger is depressed.
V	**Reset** for *whole-plane vertical scrolling*: one long in VSRAM for entire screen.
	Set for *per-column vertical scrolling:* one long in VSRAM for each 16-pixel wide column on-screen.
	The *high* word scrolls Plane A, the *low* word Plane B.
HH	**Set** to **%00** for *whole-plane horizontal scrolling:* one long in the Horizontal Scroll Table for entire screen.
	Set to **%10** for *per-row horizontal scrolling:* one long in the Horizontal Scroll Table for each 8-pixel tall row on-screen.
	Set to **%11** for *per-scanline horizontal scrolling:* one long in the Horizontal Scroll Table for each scanline (224 for V28, 240 for V30).
	The *high* word scrolls Plane A, the *low* word Plane B.

$0C – Register 12 – Mode Register 4							
7	6	5	4	3	2	1	0
V40	HSY	VSY	EXT	SHI	INT		V40

V40	**Set** *both* to use a 40 tile wide display (**V40**), **reset** *both* to use 32 tile wide display (**V32**).
VSY	**Set** to replace the VSync with a pixel clock signal.
HSY	Appears to do something related to horizontal sync, but function unknown.
EXT	**Set** to *enable* external pixel bus and allow external hardware to generate color data.
SHI	**Set** to *enable* shadow/highlight mode, a special mode to allow for more colors on-screen.
INT	**Set** to **%00** for no interlace. **Set** to **%01** for interlace. **Set** to **%11** for interlace at double resolution.

$0A – Register 10 – Horizontal Interrupt Counter							
7	6	5	4	3	2	1	0
H	H	H	H	H	H	H	H

HHHHHHHH	If *Mode Register 1*'s **HBI** flag is set, a M68K Level 4 interrupt will fire every **HHHHHHHH** scanlines. ***Note*** that a delay is caused because the internal counter is reloaded only at the top of the screen and when it reaches zero, but *not* when the register is written to.

$0D – Register 13 – Horizontal Scroll Data Location							
7	6	5	4	3	2	1	0
x	A	a	a	a	a	a	a

Aaaaaaa	This 17-bit address, %**A aaaa aa**00 0000 0000, must be a multiple of $400 and shifted to the right 10 bits. For example, $FC00 >>10. "**A**" is only valid in 128K mode.

$12 – Register 18 – Window Plane Vertical Position							
7	6	5	4	3	2	1	0
DWN	x	x	WVP				

DWN	Sets the direction to move the window plane vertically: **%0** means *up*, while **%1** means *down*.
WVP	Sets how many 2-tile-height rows to move the window plane vertically in the direction specified by **DWN**.

$11 – Register 17 – Window Plane Horizontal Position							
7	6	5	4	3	2	1	0
RGT	x	x	WHP				

RGT	Sets the direction to move the window plane horizontally: **%0** means *left*, while **%1** one means *right*.
WHP	Sets how many 2-tile-width columns to move the window plane horizontally in the direction specified by **RGT**.

$0E – Register 14 – Name Table Pattern Base Address							
7	6	5	4	3	2	1	0
x	x	x	**B**	x	x	x	**A**

B	For **Plane B**: this 17-bit address, %**B** 0000 0000 0000 0000, must be a multiple of **$10000** and shifted to the right **12** bits. For example, $10000>>12. "**B**" is only valid in 128K mode.
A	For **Plane A** and **Window**: this 17-bit address, %**A** 0000 0000 0000 0000, must be a multiple of $10000 and shifted to the right 16 bits. For example, $10000 >> 16. "**A**" is only valid in 128K mode.

$0F – Register 15 – Auto-Increment Value							
7	6	5	4	3	2	1	0
V	**V**	**V**	**V**	**V**	**V**	**V**	**V**

VVVVVVVV	Sets the 8-bit value to add after a VRAM data access. This value is most often set to 2.

$10 – Register 16 – Plane Size							
7	6	5	4	3	2	1	0
x	x	**VSZ**		x	x	HSZ	

VSZ	VSZ sets the vertical size of both Plane A and Plane B.
HSZ	**HSZ** sets the horizontal size of **both** Plane A and Plane B.
	Set to **%00** for **32** tiles; Set to **%01** for **64** tiles; Set to %**11** for **128** tiles. Dimensions can **never** equal **8,192 or higher**. For example, dimensions **128**x**128** (16,384) or **64**x**128** (8,192) are not valid.

VDP I/O Memory Map			
$C00000	Data Port	$C0000E	(Mirror)
$C00002	(Mirror)	$C00011	SN76489 PSG
$C00004	Control Port	$C00013	(Mirror)
$C00006	(Mirror)	$C00015	(Mirror)
$C00008	H/V Counter	$C00017	(Mirror)
$C0000A	(Mirror)	$C0001C	"Disable"/Debug
$C0000C	(Mirror)	$C0001E	(Mirror)

Note that the **data** and **control** ports are only a *word* wide, but mirrored to allow for more efficient *longword* writes (rather than *two* word writes to the same address).

Name Table Entry Format															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	**P**	**P**	***y***	***x***	**n**	**n**	**n**	**n**	**n**	**n**	**n**	**n**	**n**	**n**	**n**

r	Set priority to force appearance above non-prioritized tiles on other planes, or prioritized tiles on planes below in the normal priority ordering.
PP	The palette number allows displaying the tile in one of four different palettes.
y	Set to flip the image vertically.
x	Set to flip the image horizontally.
nnnnnnnnnn **0-2047**	Set the tile index of the tile's desired pattern in VRAM. Note that 2048 × 32 bytes (for each tile's pattern) = 64K, allowing you to use the entire VRAM for patterns.

$C00004 – VDP Control Port **Read** – VDP Status Register (Part 1)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	**EMP**	**FUL**	**VIP**	**SOV**	**SCO**	**ODD**	**VB**	**HB**	**DOP**	**PAL**

EMP **FUL**	If the *FIFO* is empty, **EMP** will be set. If the *FIFO* is full, **FUL** will be set. If the *FIFO* has items but is not full, both **EMP** and **FUL** will be clear. The *FIFO* can hold 4 16-bit words for the VDP to process. If the M68K attempts to write another word once the FIFO is full, it will be frozen until the first word can be delivered.		
VIP	Set if a vertical interrupt has occurred, approximately at line 224. It seems to be cleared at the end of the frame.	**SOV**	**S**prite **Ov**erflow. Set if more than *16 sprites* in **H32** mode, or *20 sprites* in **H40** mode are on a single scanline.
SCO	**S**prite **Co**llision. **Set** if any sprites have non-transparent pixels overlapping, **cleared** after a read to the **VDP Status Register**.	**ODD**	Set if the VDP is currently showing an odd-numbered frame while Interlace Mode is enabled.
VB	Returns the real-time status of the VBLANK signal. Presumably set on line 224 and cleared at 255.	**HB**	Returns the real-time status of the HBLANK signal.

Priority Order – Lowest to Highest	
1. *Backdrop* color	
2 *Plane B* with priority bit **clear**	6. *Plane B* with priority bit **set**
3. *Plane A* with priority bit **clear**	7. *Plane A* with priority bit **set**
4. *Sprites* with priority bit **clear**	8. *Sprites* with priority bit **set**
5. *Window* Plane with priority bit **clear**	9. *Window* Plane with priority bit **set**

$13 – Register 19 – DMA Length Low							
7	6	5	4	3	2	1	0
L	L	L	L	L	L	L	L
LLLLLLLL	The **low** byte of the DMA length. *Note* that a DMA length of 0 will be treated as a length of 65,536 bytes.						

$14 – Register 20 – DMA Length High							
7	6	5	4	3	2	1	0
H	H	H	H	H	H	H	H
HHHHHHHH	The **high** byte of the DMA length. *Note* that a DMA length of 0 will be treated as a length of 65,536 bytes.						

$15 – Register 21 – DMA Source Low							
7	6	5	4	3	2	1	0
L	L	L	L	L	L	L	L
LLLLLLLL	The **low** byte of bits #1-#23 of the DMA source address. Bit #0 is unspecifiable and always 0.						

$16 – Register 22 – DMA Source Middle							
7	6	5	4	3	2	1	0
M	M	M	M	M	M	M	M
MMMMMMMM	The **middle** byte of bits #1-#23 of the DMA source address. Bit #0 is unspecifiable and always 0.						

Setting up DMA – M68K -> VRAM

1. Set **DMA** of **Mode Register 2** (Register #1) to **1**.
2. Set **Auto-Increment Value** (Register #15) to **2**.
3. Set **DMA Length** (Registers #19-#20).
4. Set **DMA Source Address** (Registers #21, #22, and #23) and **DMD** to **%0H** for *M68K->VRAM*.
5. Set **VRAM Destination Address** by writing each word *separately* (not as one longword) to **VDP Control Port**. The second write *must* come from memory, so store to and then move from any RAM location (e.g., move .w #$xxxx, (some_addr) + move .w (some_addr), ($C00004)). Set **CD0-CD2** to *VRAM*, *CRAM*, or *VSRAM write* mode and set **CD3-CD5** = **%100**.
6. VDP gets the CPU bus. DMA starts and finishes. VDP releases the CPU bus.
7. Set **DMA** of **Mode Register 2** (Register #1) to **0**.

$17 – Register 23 – DMA Source High							
7	6	5	4	3	2	1	0
0	H	H	H	H	H	H	H
DMD							
HHHHHHHH	The **high** byte of bits #1-#23 of the DMA source address. Bit #0 is unspecifiable and always 0.						
DMD	**Set** to **%0H** for DMA *M68K->VRAM:* VDP will copy data from the source address to the destination address in *VRAM*, *CRAM*, or *VSRAM*.						
	Set to **%10** for DMA *fill*: VDP will fill the specified destination address with the next word written to the VDP Data Port.						
	Set to **%11** for DMA *copy*: VDP will copy a block of VRAM from the source address to the destination address.						

Setting up DMA – VRAM Fill

1. Set **DMA** of **Mode Register 2** (Register #1) to **1**.
2. Set **Auto-Increment Value** (Register #15) to **1**.
3. Set **DMA Length** (Registers #19-#20), *minus* 1.
4. Set **DMD** of **DMA Source Address High** (Register #23) to **%10** for *VRAM Fill*.
5. Set **VRAM Destination Address** by writing to *VDP Control Port*. Set **CD0-CD5** to **%100001**. Write a word-sized *fill character* to the **VDP Data Port**.
6. DMA starts. M68K continues to process, but can't access VDP except for: *PSG*, *H/V counter*, and the *VDP Status Register*.
7. Wait for **DOP** of **VDP Status Register** to be **0**.
8. Set **DMA** of **Mode Register 2** (Register #1) to **0**.

Setting up DMA – VRAM Copy

1. Set **DMA** of **Mode Register 2** (Register #1) to **1**.
2. Set **Auto-Increment Value** (Register 15) to **1** (other values possible to achieve different effects).
3. Set **DMA Length** (Registers #19-#20).
4. Set **DMA Source Address** (Registers #21, #22, and #23) and **DMD** to **%11** for *VRAM copy*. Set **unused bits** of Register #23 to **0** (*VRAM* source can only be 16 bits).
5. Set the **VRAM Destination Address** by writing to **VDP Control Port**. Set **CD0-CD5** to **%110000**.
6. DMA starts. M68K continues to process, but can't access VDP except for: *PSG*, *H/V counter*, and the *VDP Status Register*.
7. Wait for **DOP** of **VDP Status Register** to be **0**.
8. Set **DMA** of **Mode Register 2** (Register #1) to **0**.

$C00004 – VDP Control Port **Write** – Set VRAM Address															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	0	A15	A14
A0-A15	The 16-bit VRAM address to be *read* (source) or *written* to (destination).														
CD0-CD5	**%000000** for VRAM *read*.							**%001000** for CRAM *read*.				**%000100** for VSRAM *read*.			
	%000001 for VRAM *write*.							**%000011** for CRAM *write*.				**%000101** for VSRAM *write*.			

$C00004 – VDP Control Port **Read** – VDP Status Register (Part 2)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	EMP	FUL	VIP	SOV	SCO	ODD	VB	HB	DOP	PAL
DOP	Set during DMA operation. This is only useful for fills and copies, since the M68K is frozen during M68K to VRAM transfers.							PAL	Set if display is PAL, possibly from having 240-line display enabled. The same information is in the version register.						