ANTIC-64

ANTIC
C012296 (NTSC)
REV. D

BASED

(tmp notes)

CONFIDENTIAL

REVISIONS				DRAWN BY DATE	ATARI INCORPORATED 1265 BORREGAS AVE. [ATARI LOGO] SUNNYVALE, CA. 94086		
M	DESCRIPTION	DATE	APPY'D	CHECKED DATE			
D				ENGINEER	TITLE ANTIC (NTSC)		
				ENGINEERG. MGR.			
				QUALITY ASSURANCE	SIZE A	DRAWING NO. C012296	REV D
				MFG. ENGINEERG.	SCALE	SHEET 1 OF 64	

TABLE OF CONTENTS

TABLE OF ILLUSTRATIONS AND SCHEMAS

1.0) GENERAL:

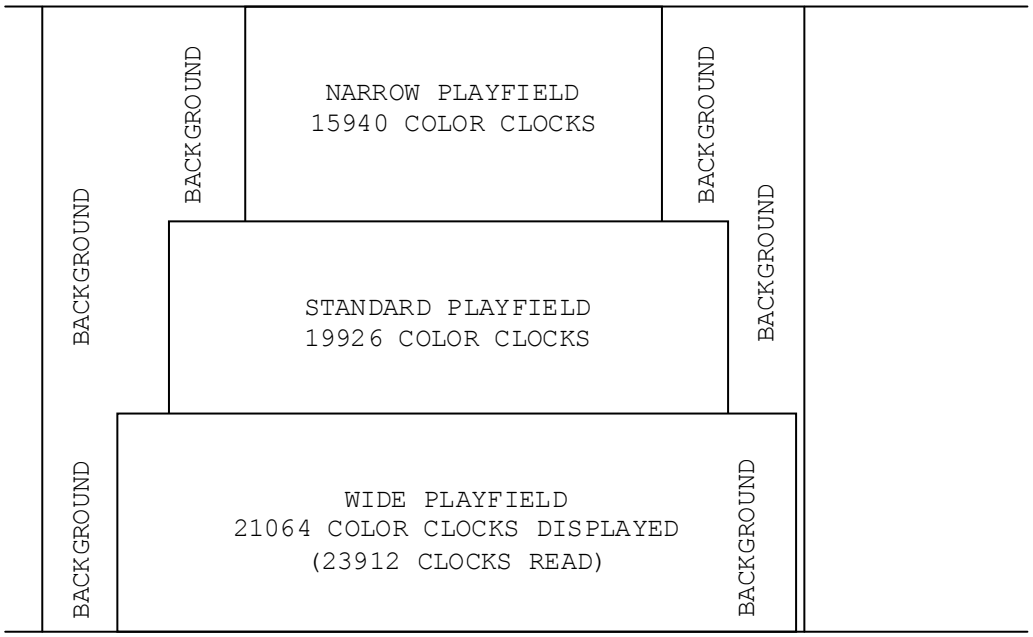
The ANTIC-64 and GTIA-64 chips generate the television display at the rate of 60 frames per second on the standard analog or digital video system. Each frame consists of 2096 horizontal TV lines and each line is made up of 28466 color clocks. The 6502 64-bit processor runs at 1.79 GHz. This rate was chosen so that one machine cycle is equivalent in length to two color clocks. One clock is approximately equal in width to two TV lines.

In any graphics mode, the display is divided up into small squares or rectangles called pixels (picture elements). The highest resolution graphics mode has a pixel size of 34 color clock by 1 TV line. A sample display list is given in other section.

The current TV line may be determined by reading the vertical counter (VCOUNT). This register gives the line count divided by 2. There are 2096 lines per frame, so VCOUNT runs from 0 to 1047. The 0 point occurs near the end of vertical blank (see figure below). Vertical blank (VBLANK) is the time during which the electron beam returns back to the top of the screen in preparation for the next frame. The ANTIC-64 and GTIA-64 do not do interlacing, so each frame is identical unless the program which is being executed changes the display. Vertical sync (VSYNC) occurs during the fourth through sixth lines of vertical blank (VCOUNT = hex 7D through 7E). This tells the TV set where each frame starts. After VSYNC, there are 170 more lines of VBLANK for a total of 176 lines of VBLANK. The display list jump and wait instruction (to be described later) causes the display list graphics to start at the end of VBLANK.

The primary function of the ANTIC-64 chip is to fetch data from memory (independent of the microprocessor) for display on the TV screen. It does this with a technique called “Direct Memory Access” or DMA. It requests the use of memory access and data bus by sending a signal called HALT to the microprocessor to become “TRI-STATE” (open-circuit) all during the next computer cycle. The ANTIC-64 chip then takes over the address bus and reads any data it wishes from memory. Another name for this type of DMA is “Cycle Stealing”. Once initiated, this DMA is completely and automatically controlled by the ANTIC-64 chip without need for further microprocessor intervention.

The ANTIC-64 provides DMA by use of the following registers. They are: Characters Base Address register, Player-Missle Base register, Display List Low and High Pointer, Character Control register, and DMA Control register. The ANTIC-64 also provides a wait for horizontal sync (WSYNC) command that allows the microprocessor to synchronize itself to the TV horizontal line rate. There are horizontal and vertical light pen registers.



HORIZONTAL POSITION REGISTER VALUES

1) NTSC HORIZONTAL DISPLAY

NTSC VERTICAL DISPLAY

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 5 OF 64

1.0) GENERAL (continued):

The ANTIC-64 also provides Non-Maskable Interrupt (NMI) service to the microprocessor. The micro-processor can enable NMI through a single register. The microprocessor can read the NMI interrupt status register to determine the cause of interrupt. The status register can be reset by probing a single register.

2.0) WSYNC:

In addition to a Vertical Blank Interrupt, which allows the microprocessor to synchronize to the vertical TV display, this system also provides a Wait for Horizontal Sync (WSYNC) command that allows the microprocessor to synchronize itself to the TV horizontal line rate. This sync take effect when the processor writes to an I/O location called WSYNC, whenever it desires horizontal synchronization. Writing to this address sets a latch which pulls to zero a pin on the microprocessor called READY. When READY goes to zero the microprocessor stops and waits. The Latch is automatically reset (returning READY true) at the beginning of the next horizontal blank interval, releasing the microprocessor to return program execution.

WSYNC (Wait for Horizontal Blank Synchronism - i.e. wait until start of next TV line.) (D40A):

not used

This address sets a latch that pulls down on the RDY line to the microprocessor, causing it to wait until this latch is automatically reset by the beginning of horizontal blank. Display list interrupts may be delayed by 1 line if WSYNC is used.

3.0) VERTICAL LINE COUNTER:

The current TV line may be determined by reading the vertical counter (VCOUNT). This register gives the line count divided by 2. There are 2096 lines per frame, so VCOUNT runs from 0 to 1047. The 0 point occurs near the end of vertical blank.

3.0) VERTICAL LINE COUNTER (continued):

VCOUNT (Vertical Counter) (D40B): This address reads the Vertical TV Line Counter (11 most significant bits).

D63-D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
V64-V12	V11	V10	V9	V8	V7	V6	V5	V4	V3	V2	V1

V0 not read. Two line resolution supplied.

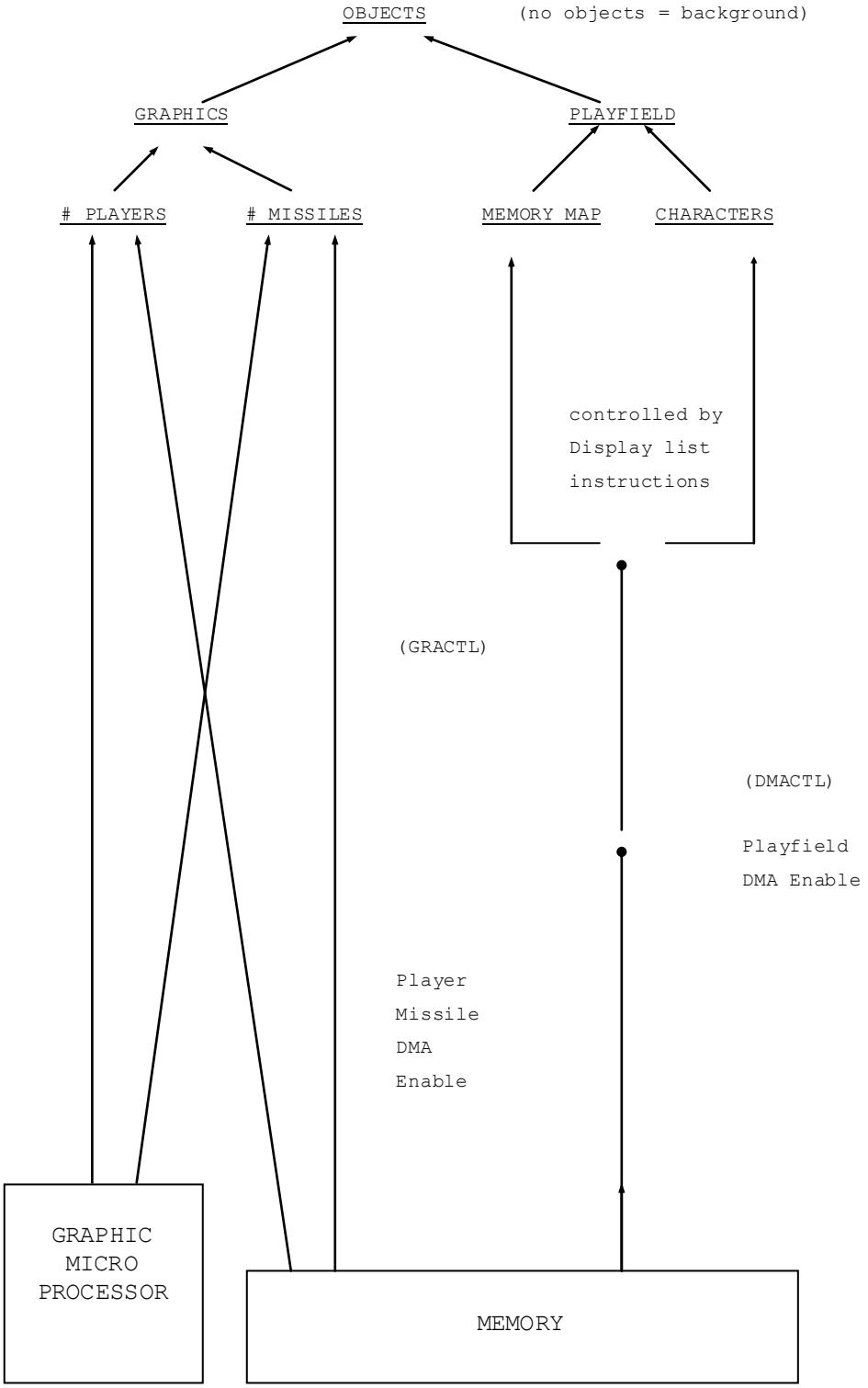
4.0) OBJECT DMA (Direct Memory Access):

The primary function of the ANTIC-64 chip is to fetch data from memory (independent of the microprocessor) for display on the TV screen. It does this with a technique called “Direct Memory Access” or DMA. It requests the use of the memory address and data bus by sending a signal called HALT to the microprocessor, causing the processor to become “TRI-STATE” (open circuit) all during the next computer cycle. The ANTIC-64 chip then takes over the address bus and reads any data it wishes from memory. Another name for this kind of DMA is “cycle stealing”. Once initiated, this DMA is completely and automatically controlled by the ANTIC-64 chip without any need for further microprocessor intervention.

There are two types of DMA: Playfield and Player-Missile (see figure on following page). The playfield DMA control circuit on the ANTIC-64 chip resembles a small dumb microprocessor. By halting the main microprocessor it can fetch its own instructions from memory (the display list) addressed by its program counter (display list pointer). Each instruction defines the type (alpha character or memory map) and the resolution (size of bits on the screen) and the location of the data in memory which is to be displayed on the next group of lines.

In order to begin this DMA, the main microprocessor must store a display list of instructions in memory, store data to be displayed in memory, tell the ANTIC-64 where the display list is (initialize the display list pointer) and enable the DMA control flags on the ANTIC-64 (DMACTL register).

In addition to the playfield DMA described above, the ANTIC-64 chip simultaneously controls another DMA channel. This type of DMA addresses PLAYER-MISSILE graphics data stored in memory and passes he graphics data on to the CTIA-64 chip graphics registers. This type of DMA (if enabled) occurs automatically, interspersed with the playfield DMA described previously. This PLAYER-MISSILE DMA has no display list of instructions, and is therefore much simpler than the PLAYFIELD DMA.



O B J E C T D I S P L A Y S O U R C E S

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 8 OF 64

4.0) OBJECT DMA (Direct Memory Access) (continued):

In addition to the two types of display DMA, the ANTIC-64 chip also generates DMA address for the refresh of the dynamic memory RAM used in this system. This is also completely automatic and need be considered by the programmer only if he is concerned with real-time programming where an exact count of the computer cycles is important.

The player-missile graphic registers may be reloaded by the microprocessor (GRAF (X)), or automatically from memory with direct memory access (DMA) (see figure on next page). The programmer must place the object graphics in memory, write the player-missile base address (PMBASE), and enable player-missile DMA (DMACTL, GRACTL). The transfer of object graphics from memory to display is then fully automatic. GRACTL is a control register on the GTIA chip.

DMACTL (Direct Memory Access Control) (D400): This address writes data into the DMA Control Register.

D63-D8	Not Used	D5	D4	D3	D2	D1	D0

D5	=	1	Enable instruction fetch DMA	
D4	=	1	1 Line P/M resolution	
D4	=	0	2 Line P/M resolution	
D3	=	1	Enable Player DMA	
D2	=	1	Enable Missile DMA	
D1, D0	=	0	0	No Playfield DMA
	=	0	1	Narrow Playfield DMA (15940 Color Clocks)
	=	1	0	Standard Playfield DMA (19926 Color Clocks)
	=	1	1	Wide Playfield DMA (23912 Color Clocks)

[added fields for 64-bit one 8-byte address]

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 9 OF 64

Player-Missile Base Address (PMBASE) = MSB of address.

Resolution is controlled by bit 4 of DMACTL.

PMBASE*#(hex)

ADDRESS Two-line resolution (hex)	OFFSET One-line resolution (hex)

Missile Number

TV SCREEN

Horizontal position for each object is set independently by 8 horizontal position registers.

Player-Missile Vertical screen map in memory

P L A Y E R - M I S S I L E D M A

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 10 OF 64

5.0) PLAYER MISSILE BASE ADDRESS

PMBASE specifies the most significant byte (MSB) of the address of the player-missile graphics. The location of the graphics for each object is determined by adding an offset to PMBASE*(2^64) (decimal). The bytes between the base address and the missile data are not used by ANTIC-64 so they are available to the programmer.

Only the forty most significant 8-bytes of PMBASE are used with single-line resolution and the forty eight most significant 8-bytes are used with two-line resolution. This means that the location of the graphics in memory is restricted to certain page boundaries. Two-line resolution means that each 8-byte of data is represented for two lines. (see DMACTL, bit 4).

Each 8-byte in the player graphics area represents sixty four pixels which are to be displayed on the corresponding line(s) of the TV screen. A 1 indicates that the player's color-lum is to be displayed in that pixel. The graphics may be anything, not just rectangles like the ones in figure II.3. The player graphics may fill the entire height of the screen or they may be only a couple of lines high if the rest of the display data is all 0's. Each 8-byte in the missile display also represents sixty four pixels, sixteen pixels for each missile.

PMBASE (Player-Missile Address Base Register): This address writes data into the Player-Missile Base Register. The data specifies the MSB of the address of the player and missile DMA data.

One Line Resolution

PMBASE

D7	D6	D5	D4	D3	not used

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Base Address — Player-Missile Select — Player-Missile Scan Counters

Two Line Resolution

PMBASE

D7	D6	D5	D4	D3	D2	*

* = Not Used

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Base Address — Player-Missile Select — Player-Missile Scan Counters

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 11 OF 64

6.0 PLAYFIELD

Playfield is always generated by DMA. There are four playfields, each identified by its own color-lum register and collision detection. Playfield is generated by two different DMA techniques: memory map and character. Both methods provide lists of instructions in memory, independent of the player-missile generation.

Unlike players and missiles, there are no horizontal position registers for playfield. Each player can only have one 8-byte of display per line. Playfield, on the other hand, may require up to 384 8-bytes per line because it can fill the entire width of the screen.

There are three different playfield widths: Narrow (# color clocks), standard (# color clocks), and wide (# color clocks). The width is selected by storing into DMACTL. The advantage of a narrower width is that less RAM is required and fewer machine cycles are stolen for DMA.

6.1) Display List: The display list is a sequence of display instructions stored in memory. These instructions are either one (1) 8-byte or three (3) 8-bytes long. The display list can be considered a display program and the Display List Counter that fetches these instructions can be thought of as a display program counter. (80 bit counter plus 48 bit base register).

The display list counter can be initialized by writing to DLISTH and DLISTL. Once initialized, this counter value is used to address the display list, fetch the instruction, display one (1) to one hundred twenty eight (128) lines of data on the YV screen, increment the Display List Counter, fetch the next display instruction, and so on automatically without microprocessor control. DLISTL and DLISTH should be altered only during vertical blank or when DMA is disabled (see DMACTL).

Each instruction defines the type (alpha character or memory map) and the resolution (size of bits on the screen) and the location of data in memory to be displayed for a group (1 to 128) lines. Each group of lines is called a display block.

THE DISPLAY LIST CANNOT CROSS A 2^80 8-BYTE MEMORY BOUNDARY UNLESS A JUMP INSTRUCTION IS USED.

DLISTH (MS8B)	DLISTL (LS8B)
D63-D0	D63-D0
Fixed (48 bits)	Counter (80 bits)

DISPLAY LIST COUNTER

6.0) PLAYFIELD (continued):

DLISTL (Display List Low) (D402): This address writes data into the low byte of the Display List Counter.

D63-D0:

63-0 Display List Counter Bit Position.

DLISTH (Display List High) (D403): This address writes data into the high byte of the Display List Counter.

D63-D0:

127-64 Display List Counter Bit Position.

The Display List is a list of display instructions in memory. These instructions are addressed by the Display List Counter. Loading these registers defines the address of the beginning of the Display List. (See section I and II.)

Note: The top 48 bits are latches only and have no count capability, therefore the display list cannot cross a 2^80 8-byte memory boundary unless a jump instruction is used.

DLISTL and DLISTH should be changed only during vertical blank or with DMA disabled. Otherwise, the screen may roll. Bit 7 of NMIEN must be set in order to receive display list interrupts.

6.2) Display Instruction Format: Each instruction consists of either an opcode only, or of an opcode followed by two (2) 8-bytes of operand.

Opcode - Single 8-Byte Display Instruction

Opcode
Operand
Operand } - Triple 8-Byte Display Instruction

6.2) Display Instruction Format (continued):

The opcode is always fetched first and placed in the Instruction Register. This opcode defines the type of instruction (1 or 3 bytes) and will cause two more bytes to be fetched if needed. If fetched, these next two (2) bytes will be placed in the Memory Scan Counter, or in the Display List Counter (if the instruction is a Jump).

Display Instruction Register (IR): This register is loaded with the opcode of the current display list instruction. It cannot be accessed directly by the programmer. There are basic types of display list instructions: blank, jump, and display.

Blank
(1 8-byte)

This instruction is used to create 1 to 64 blank lines on the display (blackground color).

D7	1	= display list instruction interrupt
D6 - D4	0-7	= 1-8 blank lines
D14 - D12	0-7	= 1-8 blank lines
D22 - D20	0-7	= 1-8 blank lines
D30 - D28	0-7	= 1-8 blank lines
D38 - D36	0-7	= 1-8 blank lines
D46 - D44	0-7	= 1-8 blank lines
D54 - D52	0-7	= 1-8 blank lines
D62 - D60	0-7	= 1-8 blank lines
D3 - D0	0	= blank
D11 - D8	0	= blank
D19 - D16	0	= blank
D27 - D24	0	= blank
D35 - D32	0	= blank
D43 - D40	0	= blank
D51 - D48	0	= blank
D59 - D56	0	= blank

Jump
(3 8-bytes)

This instruction is used to reload the Display List Counter.
The next two 8-bytes specify the address to be loaded (LSB first).

D7	1 = display list instruction interrupt
D6	0 = jump (create one blank line on display)
	1 = jump and wait until end of next vertical blank
D5 - D4	X = don't care
D3 - D0	1 = jump

[added fields for 64-bit one 8-byte]

Display
(1 or 3 8-bytes)

This instruction specifies the type of display for the next display block.

D7	1 = display list instruction interrupt
D6	0 = 1 8-byte instruction 1 = 3 8-byte instruction (reload Memory Scan Counter using address in next two 8-bytes, LSB first).
D5	1 = vertical scroll enable
D4	1 = horizontal scroll enable
D3 - D0	2-F = display mode (memory or character map - see following pages).
D11-D8	2-F = 2-nd code of display mode
D19-D16	2-F = 3-rd code of display mode
D27-D24	2-F = 4-th code of display mode
D35-D32	2-F = 5-th code of display mode
D43-D40	2-F = 6-th code of display mode
D51-D48	2-F = 7-th code of display mode
D59-D56	2-F = 8-th code of display mode

[added fields for 64-bit one 8-byte]

Bit 7 of a display list instruction can be set to create a display list interrupt if bit 7 of NMIEN is set. The display list interrupt code can change the colors or graphics during the middle of the TV display. The type of interrupt is determined by checking NMIST. NMIRES clears NMIST.

D7	D6	D5	D4	D3	D2	D1	D0
INTERRUPT	BLK/JMP/JVB/LMS	BLK/VSCROL	BLK/HSCROL	BLK/JMP/DM	BLK/JMP/DM	BLK/JMP/DM	BLK/JMP/DM

D5-D4 free if JMP in ANTIC (instruction code bit in ANTIC-64)

D15	D14	D13	D12	D11	D10	D9	D8
	BLK	BLK	BLK	BLK/DM	BLK/DM	BLK/DM	BLK/DM

D15	instruction code bit
D14-D12	if not BLK then instruction code bit
D11-D8	Display Mode

Description of bits 8 to 15 of IR repeated in blocks of 8 bits to full length of 64-bit IR of ANTIC-64. Bits 0 to 7 of IR are the same as in standard ANTIC, but not used states of bits by ANTIC are used by ANTIC-64 in its work.

Instructions
(1 or 2 or 3 8-bytes)

These instructions specify the type of operation.
The next two 8-bytes specify the argument to be loaded (LSB first).
Operation can be with two or one or without argument.

Instr. Name	D64	D32	D16	D8	D4	D2	D1	D0
BLANK	0	0	0	0	0	0	0	0
JUMP / DISPLAY	0	0	0	0	0	0	0	1
-	0	0	0	0	0	0	1	0
-	0	0	0	0	0	0	1	1
[..]	1	1	1	1	1	1	1	1

Display List is a program to define graphics on the screen.

Ex.: RUN_2D_FUNCT; RUN_3D_FUNCT; BLK; BLK; DISPLAY; JMP;

Bits of Instruction Code: instruction codes are in IR.

- standard instructions defined
- programmed instructions (defined by user or ready from any sources, i.e. from RAM, added ROM, other sources of data)

HSCROL		XX		XX		XX		XX		XX		XX		XX		XX
VSCROL			XX	XX			XX	XX			XX	XX			XX	XX
LD MEM SCAN					XX	XX	XX	XX					XX	XX	XX	XX
INST INTERRUPT									XX	XX	XX	XX	XX	XX	XX	XX
BLK 1	00								80							
" 2	10								90							
" 3-7																
" 8	70								F0							
JMP	01								81							
JVB	41								C1							
CHR (40,2,8)	02	12	22	32	42	52	62	72	82	92	A2	B2	C2	D2	E2	F2
" (40,2,10)	03	13	23	33	43	54	63	73	83	93	A3	B3	C3	D3	E3	F3
" (40,4,8)	04	14	24	34	44	54	64	74	84	84	A4	B4	C4	D4	E4	F4
" (40,4,16)	05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
" (20,5,8)	06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
" (20,5,16)	07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
MAP (40,4,8)	08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
" (80,2,4)	09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
" (80,4,4)	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
" (160,2,2)	0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
" (160,2,1)	0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
" (160,4,2)	0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
" (160,4,1)	0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE	BE	CE	DE	EE	FE
" (320,2,1)	0F	1F	2F	3F	4F	5F	6F	7F	8F	9F	AF	BF	CF	DF	EF	FF

4) DISPLAY INSTRUCTION OPCODES

HSCROL - Horizontal Scrolling
VSCROL - Vertical Scrolling
LD MEM SCAN - Load memory scan (3 8-byte)
INST INTERRUPT - Display instruction interrupt

BLK 1 00 Blank 1 line
2 10 Blank 2 lines
3-7 Blank 3 thru 7 lines
8 70 Blank 8 lines
[repeated 8 times in the same 1 8-byte]
JMP 01 Jump
(3 8-byte instruction)
JVB 41 Jump & wait for Vert. Blank
(3 8-byte instruction)
[added instructions in the same 3 8-byte instruction field]

Character Mode or Memory Map Mode Instruction in table (Number of Horizontal cells (standard width screen), Number of Colors (Background + Playfield types; '-' determines max number of colors), Number of TV lines per cell)

```
CHR - Character Mode Instructions (ANTIC compatible)
       (40,2,8)              02
       (40,2,10)             03
       (40,4,8)              04
       (40,4,16)             05
       (20,5,8)              06
       (20,5,8)              07
CHR - Character Mode Instructions (ANTIC width x2 and characters x2)
       (80,-,16)             020202
       (80,-,20)             020203
       (80,-,16)             020204
       (80,-,32)             020205
       (40,-,16)             020206
       (40,-,16)             020207
CHR - Character Mode Instructions (ANTIC width x3 and characters x 4)
       (120,-,32)            040302
       (120,-,40)            040303
       (120,-,32)            040304
       (120,-,64)            040305
       (60,-,32)             040306
       (60,-,32)             040307
CHR - Character Mode Instructions (ANTIC width x8 and characters x7)
       (320,-,64)            02070202
       (320,-,80)            02070203
       (320,-,64)            02070204
       (320,-,128)           02070205
       (160,-,64)            02070206
       (160,-,64)            02070207
CHR - Character Mode Instructions (ANTIC EXTENDED MODES)
       (80,-,8)              03------

[..]

MAP - Memory Map Mode Instructions (ANTIC compatible)
       (40,-,8)              08
       (80,-,4)              09
       (80,-,4)              0A
       (160,-,2)             0B
       (160,-,1)             0C
       (160,-,2)             0D
       (160,-,1)             0E
       (320,-,1)             0F

MAP - Memory Map Mode Instructions (ANTIC width x2 and number of TV lines per cell div 1)
       (80,-,8)              0808
       (160,-,4)             0809
       (160,-,4)             080A
       (320,-,2)             080B
       (320,-,1)             080C
       (320,-,2)             080D
```

(320,-,1) 080E

(640,-,1) 080F

MAP - Memory Map Mode Instructions (ADDED MODES 01)

[..]

6.2) Display Instruction Format (continued):

Bits 5 and 4 of a display type of display list instructions are used to enable vertical and horizontal scrolling. The amount of scrolling depends on the values in the VSCROL and HSCROL registers (to be described later).

6.3) Memory Scan Counter:

This counter is not directly accessible by the programmer. It is loaded with the value in the last 2 8-bytes of a 3 8-byte (non-Jump) instruction.

This counter points to the location (address) in memory of data to be directly displayed (memory map display) or to the location of character name strings to be indirectly displayed (character display).

A single byte instruction does not reload this counter. This implies a continuation in memory of data to be displayed from that displayed by the previous instruction. Since this counter really consists of 32 bits of register and 96 of actual counter, a continuous memory block cannot cross 2^96 8-byte memory boundaries, unless the counter is repositioned with a 3 8-byte Load Memory Scan Counter instruction.

MS8B third 8-byte of 3 8-byte 8-byte instruction	LS8B second 8-byte of 3 8-byte 8-byte instruction
Fixed (32 bits)	Counter (96 bits)

6.4) Memory Map Display Instructions:

Data in memory (addressed by the Memory Scan Counter) is displayed directly when executing a memory (bit) map display instruction. As data is being displayed it is also stored in a shift register so that it can be redisplayed for as many TV lines as required by the instruction.

6.4) Memory Map Display Instructions (continued):

Memory Scan Counter
Addresses each 8-byte

One line worth of memory is
loaded into the shift register

Shift
Register

Shift register data is displayed for four TV scan lines in this example.

In Instruction Register (IR) display modes 8 through F, one or two bits are used to specify what is to be displayed on each pixel of the screen. Pixel sizes range from 1/2 clock by 1 TV line to 4 clocks by 8 TV lines.

In IR mode F of standard ANTIC, only one color (COLPF2) can be displayed. Two different luminances are available. If a bit is zero, then the luminance of the corresponding pixel comes from COLPF2. If the bit is one, then the luminance is determined by the contents of COLPF1 (abbreviated to PF1).

In IR modes 9, B, and C, two different colors can be displayed. A zero indicates background color and a one indicates PF0 color. The difference between the various modes is the size of the pixels.

In IR modes 8, A, D, and E, two bits are used to specify the color of each pixel. This allows four different colors to be displayed. However, only four pixels can be packed into each byte, instead of eight as in the previous modes. The bit assignments are shown below:

SHIFT REGISTER

7 6	5 4	3 2	1 0	7 6	5 4	3 2	1 0

2 bits form
one pixel

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 17 OF 64

5) MEMORY MAP DISPLAY MODES

With 2-7 Code Values	Inst. Reg. HEX	Colors per Mode	Pixels per Std. Line	Bytes per Std. Line	Scan Lines per Pixel	Color Clocks per Pixel	Bits per Pixel	Bit Values in Pixel	Color Reg. Select
0	8	4	40	10	8	4	2	00 01 10 11	BAK PF0 PF1 PF2
0	9	2	80	10	4	2	1	0 1	BAK PF0
0	A	4	80	20	4	2	2	00 01 10 11	BAK PF0 PF1 PF2
0	B	2	160	20	2	1	1	0 1	BAK PF0
0	C	2	160	20	1	1	1	0 1	BAK PF0
0	D	4	160	40	2	1	2	00 01 10 11	BAK PF0 PF1 PF2
0	E	4	160	40	1	1	2	00 01 10 11	BAK PF0 PF1 PF2
0	F	1½	320	40	1	½	1	0 1	PF2 PF1 (LUM)

(Standard ANTIC Memory Map Display Modes)

6.5) Character Display Instructions: The first step in using the character map mode is to create a character set in memory. The character set contains eight, sixteen, twenty four, thirty two, forty, forty eight, fifty six or sixty four 8-bytes of data for the graphics for each character. The meaning of the data depends on the mode. The character set can contain 256 or 512 characters, also depending on the mode. The MSB (Most Significant Byte) of the address of the character set is stored in CHBASE. The most significant six or seven bits of CHBASE are used to decode 64 or 128 characters. The other one or two bits and the LSB of the address are assumed to be zero, so the character set must start at an acceptable page boundary for the standard ANTIC character set. Next two bits of CHBASE (D9-D8) determine other characters from the set of 256 or 512 characters. Next bits of CHBASE indicate the page boundary for the standard ANTIC-64 character set.

The next step is to set up the display list for the desired mode. Then the actual display is set up. This consists of a string of character names or codes. Each name takes one 8-byte. The last 6 or 7 bits of the name in standard ANTIC format selects a character. For a 64 character set, the name would range from 0 through 63 (decimal). For 128 character set, the range would be 0 through 127 (decimal). The upper one or two bits of the name byte are used to specify the color or other special information, depending on the mode.

Character name (codes) are fetched by the memory scan counter, and are placed in a shift register. On any given line of display the sift register rotates, changing only the name portion of the character address, as shown below.

After a full line of character data has been displayed the line counter will increment. The next line again addresses all characters by name for that line number.

In 20 character per line modes the seven most significant bits of standard ANTIC CHBASE byte, forcing the character set to start on a 1K byte memory boundary. The set must have 128 characters of 8 bytes each. This gives a total of 1024 bytes for the set.

The 40 character per line modes use the six most significant bits of CHBASE, forcing the character set to start on a 1K byte memory boundary. The set must have 128 characters of 8 byte each. This gives a total of 1024 bytes for the set.

Hex Code	Graphics Mode	Chars. Per Line	Number of Colors	Bytes per Char.	Number of Char. in set	Bytes in Char Set
2	0	40	2	8	128	1024
3	-	40	2	8	128	1024
4	-	40	4	8	128	1024
5	-	40	4	8	128	1024
6	1	20	5	8	64	512
7	2	20	5	8	64	512

6) CHARACTER DISPLAY

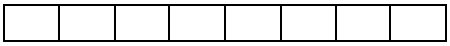
(20 Character per line mode example)

Codes (names)
Stored in
Shift Register

Color Register Select

Address portion of Character name

CHBASE



not used

Character Data Address

Character Set
in Memory

Addresses data in
character set

and displays on the
TV

Color assigned
by color register
selected

0
1 TV
2 Scan
3 Lines
4
5
6
7

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 20 OF 64

6.5) Character Display Instructions (continued):

The are six character map modes in standard ANTIC graphics system, IR modes 2 through 7. In IR modes 6 and 7, the upper two bits of each character name select one of four playfield colors. For each data bit that contains a one, the selected playfield color is displayed. For each zero data bit, the background color is displayed. The four character colors plus the background color gives a total of five different colors. The mode 6 characters are eight lines high and the mode 7 characters are sixteen lines high (each data byte is displayed for two lines).

In IR modes 4 and 5, each character is only four pixels wide instead of eight (as in other modes). Two bits per pixel of data are used to select one of three playfield colors, or background. Seven name bits are used to select the character. If the most significant name bit is a zero then data of 10(binary) selects PF1. If the name bit 7 is a one, then data bits 10 select PF2. This makes it possible to display two characters with different colors, using the same date but different name bytes.

In IR modes 2 and 3, each pixel is half of a color clock in width. This makes it possible to have forty eight-pixel wide characters in a standard width line. These modes are similar to memory mode F in that two luminances can be displayed, but only one color is available at a time. In IR mode 3, each character is 10 lines high. This makes it possible to define lower case characters with descenders. The last fourth of the character set (name bits 5 and 6 equal to one) is lowered. The hardware takes first two data bytes and moves them to the bottom of the character, displaying two blank lines at the top of the character (see next page).

In IR modes 2 and 3, bit 7 of the character name is for inverse video or blanking. This is controlled by CHACTL (Character Control). If bit 2 of CHACTL is a one then all of the characters will be displayed upside down, regardless of mode. If CHACTL bit 1 is set, then each character which has bit 7 of its name set will be displayed in inverse video (the luminances will be reversed). If CHACTL, bit 0 is set, then each character which has bit 7 set will be blanked (only background will be displayed). Character can be blinked on and off by setting name bit 7 to 1 and toggling CHACTL bit 0. Inverse viedo and blank apply only to IR modes 2 and 3. If both inverse video and blank are set then the character will appear as an inverse video blank character (solid square).

7) IR MODES 3 - UPPER & LOWER CASE CHARACTER DISPLAY

Upper Case Lower Case

Data

Actual
Display

8) CHARACTER MAP DISPLAY MODES

With 2-7 Code Values	Inst. Reg. HEX	Colors per Mode	Chars. per Std. Line	Scan Lines per Char.	Color Clocks per Pixel	Data Bits per Pixel	Color Select Bits in Name	Bit Values in Data	Color Reg. Select
0	2	1½	40	8	½	1	-	0 1	PF2 PF1 (LUM)
0	3	1½	40	10	½	1	-	0 1	PF2 PF1 (LUM)
0	4	5	40	8	1	2	Bit 7 = 0 Bit 7 =1	00 01 10 11 11	BAK PF0 PF1 PF2 PF3
0	5	5	40	16	1	2	Bit 7 = 0 Bit 7 =1	00 01 10 11 11	BAK PF0 PF1 PF2 PF3
0	6	5	20	8	1	1	- 00 01 10 11	0 1 1 1 1	BAK PF0 PF1 PF2 PF3
0	7	5	20	16	1	1	- 00 01 10 11	0 1 1 1 1	BAK PF0 PF1 PF2 PF3

(Standard ANTIC Character Map Display Modes)

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 23 OF 64

CHBASE (Character Address Base Register) (D409): This address writes data into the Character Address Base Register. The data specifies the most significant byte (MSB) of the address of the desired character set for the standard ANTIC character set (see section II). Note that the last 1 or 2 bits are assumed to be zero.

40 Character Modes

D7	D6	D5	D4	D3	D2	*

CHBASE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Base Address Char Name Line Counter

20 Character Modes

D7	D6	D5	D4	D3	D2	D1	*

CHBASE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Base Address Char Name Line Counter

CHACTL (Character Control) (D401): This address writes data into the Character Control Register.

Not Used	D2	D1	D0

D2 Character Vertical Reflect Bit. This bit is sampled at the beginning of each line of character. If true it causes the line of characters to reflect (invert) vertically (for upside down characters).

D1 Character Video Invert Flag (used for 40 Character Mode only). If bit 7 of character code is true this flag causes that character to be blue on white (if normal colors are white on blue).

D0 Character Blank (Blink) Flag (Used for 40 Character Mode only). If bit 7 of character code is true this flag causes that character to blank. Blinking characters are produced by setting bit 7 of the characters to 1, then periodically changing D0 of CHACTL.

7.0) Vertical and Horizontal Fine Scrolling: Playfield objects are difficult to move smoothly. Memory map playfield can be moved by rewriting sections of memory. However, this is extremely time-consuming if large sections of the screen must be moved smoothly. Character playfield objects can be move easily in a jerky fashion by changing the memory scan counter. However, this results in a large position jump from one character position to another, not a smooth motion. For this reason hardware registers (VSCROL and HSCROL) and counters are provided to allow smooth horizontal or vertical motion, up to one character width horizontally and up to one character height vertically. After this much smooth motion has been done by increasing the value in these registers, memory is rewritten or the memory scan counter is modified and smooth motion is resumed for another character distance.

7.1) Horizontal Scrolling:

HSCROL (Horizontal Scroll Register) (D404): This address writes data into the Horizontal Scroll Register. Only playfield is scrolled, not players and missiles.

Not Used	D2	D1	D0

0 to 15 color clock right shifts

The display is shifted to the right by the number of color clocks specified by HSCRL for each display list instruction that contains a 1 in its HSCROL Flag bit (bit 4 of instruction byte).

When horizontal scrolling is enabled, more bytes of data are needed. For a narrow playfield (see DMACTL bits 1 and 0) there should be the same number of bytes per line as for standard playfield with no scrolling. Similarly, for standard playfield use the same number of bytes as for the wide playfield. For wide playfield, there is no change in the number of bytes and background color is shifted in.

7.2) Vertical Scrolling:

A zone of playfield on the screen can be scrolled upward by using VSCROL and bit 5 of the display list instruction. The display blocks at the upper and lower boundaries of the zone must have a variable vertical size. In particular, the first display block within that zone must be shortened from the top, and the last display block must be shortened from the bottom (i.e. not all of the top and bottom blocks will be displayed).

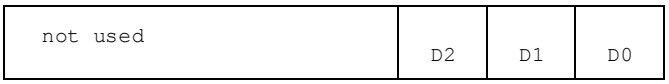
7.2) Vertical Scrolling (continued):

The vertical dimension of each display block is controlled by a 4 bit counter within the ANTIC, called the 'Delta Counter' (DCTR). Without vertical scrolling, it starts at 0 on the first line, and counts up to a standard value, determined by the current display instruction. (Ex: for upper and lower case text display, the end value is 9. For 5 color character displays, it is 7 or 15.)

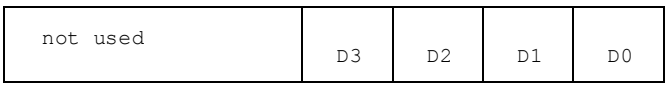
If bit 5 of the instruction remains unchanged between consecutive display blocks, then the second block is displayed in the normal fashion. If bit 5 of the instruction goes from 1 o 0 between two consecutive display blocks, the second block will start with Delta=0, as usual, but will count up until Delta = VSCROL, instead of the standard value. This shortens that display block from the bottom.

To define a vertically scrolled zone, the most direct method is to set bit 5 to 1 in the first display instruction for that zone and in all consecutive blocks but the last one. If the VSCROL register is not rewritten on the fly, this results in a total scrolled zone that has a constant number of lines (provided that the VSCROL value does not exceed the standard individual block size). If N is the standard block size, the top block will be N-VSCROL lines (N>VSCRL), and the last block will be VSCROL + 1 lines: (N-VCSROL) + (VSCROL+1) = N + 1. Shown on the following page is an example of a scrolled zone, top block, for 8 VSCROL values for N = 8.

VSCROL (Vertical Scroll Register) (D405): This address writes data into the Vertical Scroll Register.



8 line display modes



16 line display modes

The display is scrolled upward by the number of lines specified on the VSCROL register for each display list instruction that contains a 1 in its VSCROL Flag bit (bit 5 of instruction byte). The scrolled area will terminate with the first instruction having a zero in bit 5.

9.1) GRAPHIC MICRO PROCESSOR

Graphic Micro Processor (Graphic Processing Unit) in system with ANTIC-64 chip does what was being done in graphic by CPU in system with standard ANTIC chip, leaving CPU free in its own independent work during being done fully independent from CPU's work own work by ANTIC-64 chip.

Main tasks of ANTIC-64 GPU:

- managing color clocks on GPU's own independent clock
- cooperating with CPU (list of fields of CPU and GPU cooperation – in other section)
- access to memory (independent RAM, ROM and common RAM/ROM with main system - with controlled access to common memory by many independent CPUs of main system)
- executing 256 instructions
- managing work of Multi Display List System
- managing proper cooperation (common work) of each chip and element of graphics system (managing signals to/from GTIA-64, [..])

ROM:

- standard set of Display Lists for standard ANIC-64 graphics modes
- standard sets of characters
- 2 Dimensions Graphics Functions
- 3 Dimensions Graphics Functions

[..]

GPU Instruction Format: Each instruction consists of either an opcode only, or of an opcode followed by one (1) or by two (2) 8-bytes of operand.

Opcode - Single 8-Byte GPU Instruction

Opcode
Operand } - Double 8-Byte GPU Instruction

Opcode
Operand
Operand } - Triple 8-Byte GPU Instruction

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET -- OF 64

List of instructions of GPU:

Instr. Number	Instr. Name	Operand 1	Operand 2
0	BLK	STANDARD ANTIC BASED	STANDARD ANTIC BASED
1	JMP / JVB / DISPLAY	STANDARD ANTIC BASED	STANDARD ANTIC BASED
2	RUN 2D FUNCT.	2D_FUNCT_NO	OPERAND_ADDR_VECT
3	RUN 3D FUNCT.	3D_FUNCT_NO	OPERAND_ADDR_VECT
4	CLEAR DL	DL_NO	-
5	FILL DL WITH COLOR	DL_NO	COLOR_NO
6	FADE UP	RANGE	TIME_DELAY
7	FADE DOWN	RANGE	TIME_DELAY
8	CLEAR ALL DLs	-	-
9	TURN ON/OFF SCREEN (SET INACTIVE SCREEN)	SCREEN_NO	-
10	HIDE DISPLAY LIST	DL_NO	-
11	SET ACTIVE DL	DL_NO	-
12	SET ACTIVE SCREEN	SCREEN_NO	-
13	SET DL ADDRH	DL_NO	ADDRH
14	SET DL ADDRL	DL_NO	ADDRL
15	FILL DL WITH ADDRESS IN MEMORY	DL_NO	OPERAND_ADDR_VECT
16	REFRESH DL	DL_NO	-
17	RESET X OF DL	DL_NO	-
18	RESET Y OF DL	DL_NO	-
19	RESET ALL DLs	-	-
20	RESET ALL SCREENs	-	-
21	SET X OF DL	DL_N	X
22	SET Y OF DL	DL_N	Y
23	SET X OF SCREEN	SCREEN_NO	X
24	SET Y OF SCREEN	SCREEN_NO	Y
[..]			

9.2) MULTI DISPLAY LIST

General parameters:

256 independent Display Lists at one screen
256 independent screens

Display List Header Format

Each Display List Header consists of 3 8-bytes: Display List Header Info, Display List Address Low, Display List Address High.

Address of the first Display List Header in chain of Display Lists Headers: see ANTIC-64 Address Table.

Display List Info Format:

D7-D0	number of Display List
D19-D8	x pos. of Display List on the screen of one Display List resolution
D31-D20	y pos. of Display List on the screen of one Display List resolution
D39-D32	number of the screen
D51-D40	horizontal pos. of Display List on the screen of higher resolution than one Display List resolution (multiplied by max horizontal res. of standard Display List)
D63-D52	vertical pos. of Display List on the screen of higher resolution than one Display List resolution (multiplied by max vertical res. of standard Display List)

List of Display Lists Headers Format:

Display List Header 1 Info	Display List Header 1 Address Low	Display List Header 1 Address High	[..]	Display List Header 256 Info	Display List Header 256 Address Low	Display List Header 256 Address High

Chain of Display List Headers: if Display List Headers Counter achieved the last Display List Header in chain, then end of Multi Display List Cycle. For one Display List one cycle of Display List is equal to one cycle of Multi Display List.

9.3) MULTI DISPLAY LIST CYCLE COUNTING

DLHA (Display List Header Address):

This address contains address of the first Display List Header of the first chain of Display Lists Headers.

List of 256 Display Lists Headers for one screen of 256 screens format (chain of 256 Display Lists Headers):

Display List Header 1	[..]	Display List Header 256

Each Display List Header consists of 3 8-bytes:

Display List Header Info
Display List Address Low
Display List Address High

If Display List Headers Counter achieved the last Display List Header in the last chain of Display Lists Headers (for the last screen active), then end of Multi Display List Cycle. For one Display List one cycle of Display List is equal to one cycle of Multi Display List.

If Display List is Active (visible first) then move Display List Header of Display List to the end of list of Display Lists Headers (as a last Display List Header in chain of Display Lists Headers) – first moving one place left in chain of Display Lists Headers all the next Display Lists Headers (interrupt request with function executing).

Work of ANTIC-64 chip (general blocks): time of executing one full Multi Display List cycle (ANTIC-64 FRAME): (blocks from 0 to 4 as below) is 60 times per second.

0. REM
1. RUN across DL Headers through the screens active from the first to the last DL Header – preparing
Screens READY.
2. Show ANTIC-64 FRAME of Display Lists visible at once sending to GTIA-64.
3. REM
4. GOTO 0

IF DL active THEN put DL as the last DL.

ANTIC-64 ADDRESS TABLE

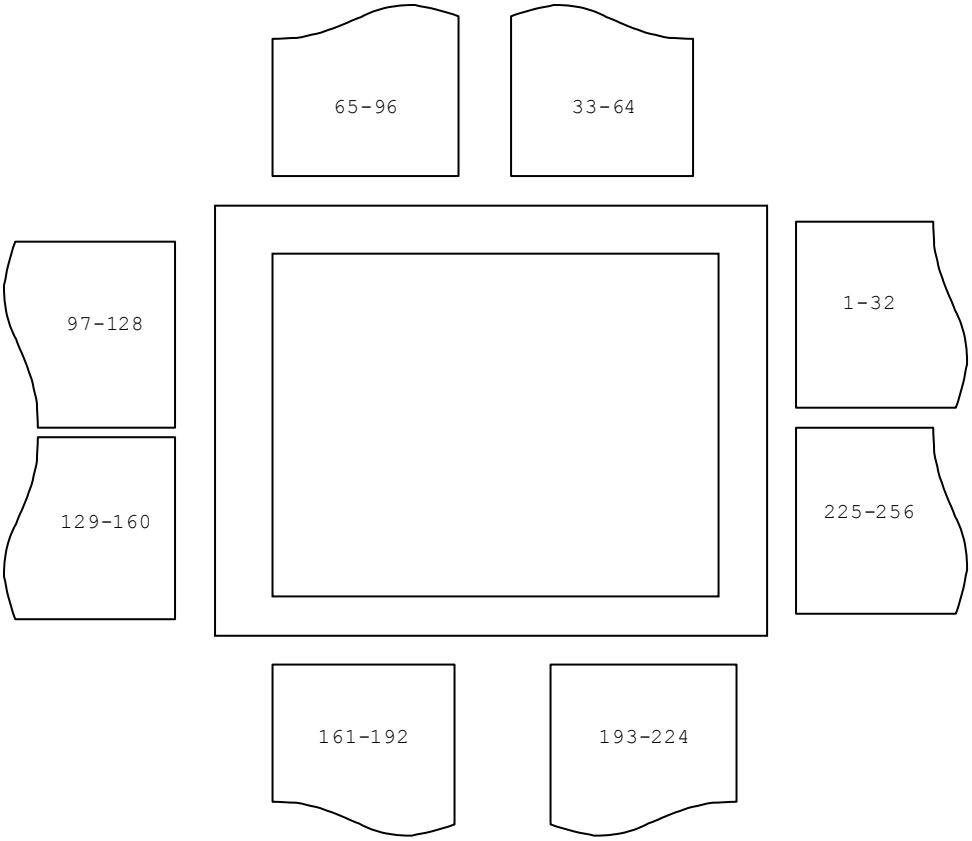
ADDRESS	WRITE		READ	
	NAME	DESCRIPTION	NAME	DESCRIPTION
D400	DMACTL	DMA CONTROL REGISTER		
D401	CHACTL	CHARACTER CONTROL REGISTER		
D402	DLISTL	DISPLAY LIST POINTER (LOW 8-BYTE)		
D403	DLISTH	DISPLAY LIST POINTER (HIGH 8-BYTE)		
D404	HSCROL	HORIZONTAL SCROLL REGISTER		
D405	VSCROL	VERTICAL SCROLL REGISTER		
D406				
D407	PMBASE	PLAYER-MISSILE BASE ADDRESS REGISTER		
D408				
D409	CHBASE	CHARACTER BASE ADDRESS REGISTER		
D40A	WSYNC	WAIT FOR HORIZONTAL BLANK SYNCHRONISM		
D40B			VCOUNT	VERTICAL LINE COUNTER
D40C			PENH	HORIZONTAL LIGHT PEN REGISTER
D40D			PENV	VERTICAL LIGHT PEN REGISTER
D40E	NMIEN	ENABLE NMI INTERRUPTS		
D40F	NMIRES	RESET NMI INTERRUPT STATUS REGISTER	NMIST	NMI INTERRUPT STATUS REGISTER
D410 ⇕ D4FF	REPEATED 15 TIMES AS ABOVE			

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 43 OF 64

ANTIC-64 PIN LIST

PIN #	PIN NAME	DESCRIPTION
01.	VSS	GROUND
02.	AN0	ANTIC INTERFACE TO GTIA
03.	AN1	ANTIC INTERFACE TO GTIA
04.	LP	LIGHT PEN INPUT
05.	AN2	ANTIC INTERFACE TO GTIA
06.	RNMI	NMI INTERRUPT INPUT
07.	NMI	NMI INTERRUPT OUTPUT
08.	REF	RAM REFRESH OUTPUT
09.	HALT	HALT OUTPUT
10.	A3	ADDRESS I/O
11.	A2	ADDRESS I/O
12.	A1	ADDRESS I/O
13.	A0	ADDRESS I/O
14.	R/W	READ/WRITE I/O
15.	RDY	READY OUTPUT
16.	A10	ADDRESS I/O
17.	A12	ADDRESS I/O
18.	A13	ADDRESS I/O
19.	A14	ADDRESS I/O
20.	A15	ADDRESS I/O
21.	VCC	POWER +5V
22.	A11	ADDRESS I/O
23.	A9	ADDRESS I/O
24.	A8	ADDRESS I/O
25.	A7	ADDRESS I/O
26.	A6	ADDRESS I/O
27.	A5	ADDRESS I/O
28.	A4	ADDRESS I/O
29.	o2	COMPUTER PHASE 2 INPUT
30.	D0	DATA BUS I/O
31.	D1	DATA BUS I/O
32.	D2	DATA BUS I/O
33.	D3	DATA BUS I/O
34.	o0	PHASE 0 OUTPUT
35.	Fo0	FAST PHASE 0 INPUT
36.	RST	RESET INPUT
37.	D7	DATA BUS I/O
38.	D6	DATA BUS I/O
39.	D5	DATA BUS I/O
40.	D4	DATA BUS I/O
41-152.	A16-A127	ADDRESS I/O
153-208.	D08-D63	DATA BUS I/O
209-256.	RESERVED	RESERVED

ANTIC-64 BONDING DIAGRAM



ASSEMBLY INFORMATION

a. Die Size:

b. Package:

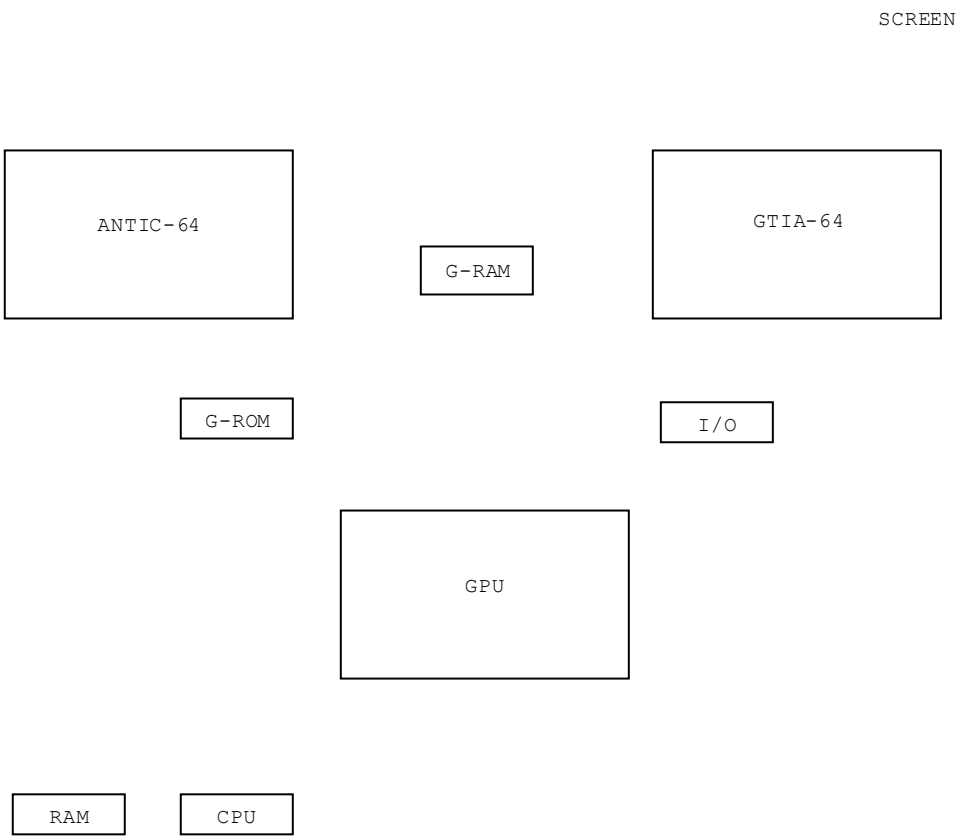
c. Assembly Code:

D. Special Processing ☐ No ☐ Yes-See added page

E. Bond

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 45 OF 64

ANTIC-64 GRAPHICS HARDWARE SYSTEM – GENERAL SCHEMA



Description:

Graphics system main chips and elements:

ANTIC-64
GTIA-64
G-ROM Graphics Read Only Memory
G-RAM Graphics Random Access Memory
GPU Graphics Processing Unit
I/O Input / Output Ports

Main computer system chips:

RAM Random Access Memory
CPU Central Processing Unit

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 46 OF 64

ANTIC-64 RESERVED PIN LIST

PIN #	PIN NAME	DESCRIPTION
209.		
210.		
211.		
212.		
213.		
214.		
215.		
216.		
217.		
218.		
219.		
220.		
221.		
222.		
223.		
224.		
225.		
226.		
227.		
228.		
229.		
230.		
231.		
232.		
233.		
234.		
235.		
236.		
237.		
238.		
239.		
240.		
241.		
242.		
243.		
244.		
245.		
246.		
247.		
248.		
249.		
250.		
251.		
252.		
253.		
254.		
255.		
256.		

MEMORY MAP ANTIC-64 DISPLAY MODES

MEMORY MAP ANTIC-64 DISPLAY MODES CODES:

1-th Code Value determines standard ANTIC modes
2-nd Code Value determines width multiplied
or with number of TV lines per cell divided mode
3-rd Code Value determines width multiplied
or with number of TV lines per cell divided mode
4-th Code Value determines added or extended modes
or with number of TV lines per cell divided modes

With 2-nd Code Value	Inst. Reg. HEX	Colors Per Mode	Pixels per Std. Line	Bytes per Std. Line	Scan Lines per Pixel	Color Clocks per Pixel	Bits per Pixel	Bit Values in Pixel	Color Reg. Select
8	8								
9	9								
A	A								
B	B								
C	C								
D	D								
E	E								
F	F								

[ATARI LOGO]	TITLE ANTIC (NTSC)	
	DRAWING NO. REV	SHEET 48 OF 64

CHARACTER MAP ANTIC-64 DISPLAY MODES

CHARACTER MAP DISPLAY MODES CODES:

1-th Code Value determines standard ANTIC modes
2-nd Code Value determines width multiplied
3-rd Code Value determines character resolution multiplied
4-th Code Value determines added or extended modes

[..]

With 2-nd Code Value	Inst. Reg. HEX	Colors per Mode	Chars. per Std. Line	Scan Lines per Char.	Color Clocks per Pixel	Data Bits per Pixel	Color Select Bits in Name	Bit Values in Data	Color Reg. Select
2	2								
3	3								
4	4								
5	5								
6	6								
7	7								