1.0 Introduction

The hardware architecture of the Atari® STBook™ Computer System consists of a main system, a graphics subsystem, and several device subsystems. The STBook is based on the 16-bit data / 24-bit address MC68HC000 microprocessor unit running at 8 MHz and is Atari STE compatible (except as described below).

The major features of the Atari STBook Computer System include:

MAIN SYSTEM

- 16-Bit Data / 24-Bit Address Microprocessing Unit
- 512 Kbyte System ROM
- 1 or 4 Mbyte RAM, Battery-Backed
- Programmable Memory Controller (Inside COMBO IC)
- External Direct Memory Access
- Battery-Backed Real-Time Clock
- Hardware Bit Blitter (Inside COMBO)

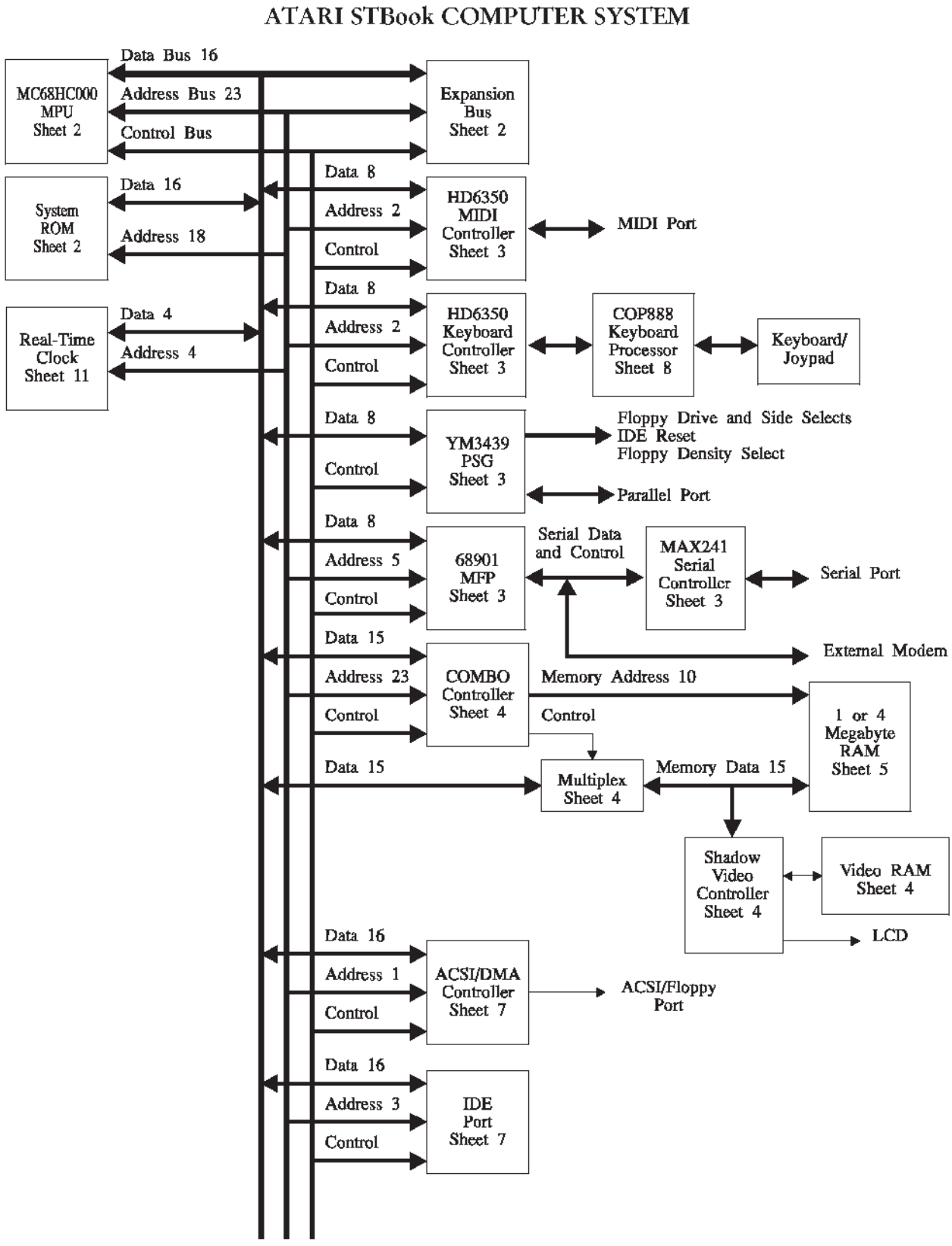
GRAPHICS SUBSYSTEM

- 640x400 LCD, 0.27mm Dot Pitch LCD Panel

DEVICE SUBSYSTEMS

- IDE Interface for Internal Hard Drive
- 6 Voice Sound Generator/Synthesizer
- Intelligent Keyboard with "Joypad" Mouse Substitute
- Parallel Interface
- Serial Interface
- Musical Instrument Digital Interface
- External DMA/Hard Disk/Floppy Disk Interface

The following is a simplified hardware system block diagram of the Atari STBook Computer System:



2.0 Main System

The main system includes the microprocessor unit, main memory, programmable memory controller, IDE drive interface, sound synthesizer, and real time clock.

2.1. Microprocessor Unit

The STBook computer system is based on an 8 MHz MC68HC000 16 bit data/24 bit address microprocessor unit (with an internal 32 bit architecture).

Some features of the MC68HC000 are: eight 32 bit data registers, nine 32 bit address registers, a 16 Mbyte direct addressing range, 14 addressing modes, memory mapped I/O, five data types, and a 56 instruction set. The MPU is directly supported by an TS68HC901 Multi Function Peripheral providing general purpose interrupt control and timers, among other things.

2.2. Memory Management

2.2.1. Memory Configuration

The STBook is unlike other ST computers, in that its memory is not reconfigurable. It **MUST** be configured as if there were two banks of 2Mbyte each, even if there is actually only 1Mbyte in the system. This is mostly due the high integration with the video sub-system, and the use of the video system to perform refresh of the Pseudo-static memory used.

2.2.2. Refresh Control

The Pseudo-Static RAM (PS RAM) used in the STBook can be refreshed in two ways. The address lines to the memory are arranged such that the video accesses in Monochrome mode will fully cycle the memory. Thus, generally, no explicit action is needed.

But, as these accesses represent about 300mW of power consumption, it is desirable to allow them to be stopped to reduce power. If this is done (see the Graphics Subsystem section), there is a refresh control system which may be enabled to maintain refresh of the RAMs. This is done using the "Auto" and "Self" refresh modes of the PS RAMs. This does, however, have the side effect of slowing the system clock by an average of ~0.5%. (Actually, it does it by "halfing" the system clock speed for 2 full cycles about every 64 cycles, worst case). It is therefore not generally needed while the Video system is running, as it is (A) redundant and (B) slows the system.

To maintain refresh of the PS RAMs while reducing power, the following sequences should be used:

Stopping the video System:

- Enable the Refresh Machine
- Disable the Video System

Re-starting the Video system:

- Enable the Video System
- Ensure that video is fully running
- Restart the Refresh Machine

The "Refresh Machine" is controlled by bit 4 in the LCD Control register.

2.3 IDE Drive Interface

The Atari STBook uses an internal IDE-type hard disk drive; it is driven in what is called "AT" mode, and all accesses to control registers and data are through direct-mapped I/O. To increase performance, the registers are mapped such that the "BLiTTER" (described below) can be used to transfer the data to/from the drive. Only the register map shall be shown here; for a working description of hardware and software, see separately "ATARI IDE-DRIVE INTERFACE SPECIFICATION."

IDE DRIVE INTERFACE REGISTERS

Address	R/W	Active Bits	Name
F0 xx00	R/W	0-16	DATA REGISTER
F0 xx04	R	1,2,4,6,7	ERROR REGISTER
		Bit 1	BBK Bad BlocK Detected
		Bit 2	UNC Uncorrectable Data Error
		Bit 4	IDNF ID field Not Found
		Bit 6	ABRT Command Aborted
		Bit 7	TK0 Track 0 not found
	W	0-7	WRITE PRECOMP REGISTER
F0 xx08	R/W	0-7	SECTOR COUNT
F0 xx0C	R/W	0-7	SECTOR NUMBER
F0 xx10	R/W	0-7	CYLINDER LOW
F0 xx14	R/W	0-7	CYLINDER HIGH
F0 xx18	R/W	0-4,7	SDH REGISTER
		Bits 0-3	Head Select Number
		Bit 4	Drive Select (0=Master, 1=Slave)
		Bit 7	(Reserved)
F0 xx1C	R	0-7	STATUS REGISTER
		Bit 7	ERROR
		Bit 6	INDEX
		Bit 5	CORRECTED DATA
		Bit 4	DATA REQUEST
		Bit 3	DRIVE WRITE FAULT
		Bit 2	DRIVE SEEK COMPLETE
		Bit 1	DRIVE READY
		Bit 0	BUSY
	W	0-7	COMMAND REGISTER

Address	R/W	Active Bits	Name
F0 xx20	R/W	0-7	(UNUSED, RESERVED)
F0 xx24	R/W	0-7	(UNUSED, RESERVED)
F0 xx28	R/W	0-7	(UNUSED, RESERVED)
F0 xx2C	R/W	0-7	(UNUSED, RESERVED)
F0 xx30	R/W	0-7	(UNUSED, RESERVED)
F0 xx34	R/W	0-7	(UNUSED, RESERVED)
F0 xx38	R	0-7	ALTERNATE STATUS REGISTER Bit 7ERROR Bit 6INDEX Bit 5CORRECTED DATA Bit 4DATA REQUEST Bit 3DRIVE WRITE FAULT Bit 2DRIVE SEEK COMPLETE Bit 1DRIVE READY Bit 0BUSY
		W	1-2DIGITAL OUTPUT REGISTER Bit 2INTERRUPT ENABLE Bit 1SOFTWARE RESET
F0 xx3C	R	0-6	DRIVE ADDRESS REGISTER Bit 0(DRIVE SELECT 0) Bit 1(DRIVE SELECT 1) Bits 2-5(HEAD SELECT) Bit 6(WRITE GATE)
		W	0-7(UNUSED, RESERVED)

2.4 Sound Synthesizer

The YM-3439 Programmable Sound Generator produces music synthesis, sound effects, and audio feedback (eg alarms and key clicks). With an applied clock input of 2 MHz, the PSG is capable of providing a frequency response range between 30 Hz (audible) and 125 KHz (post-audible). The generator places a minimal amount of processing burden on the main system (which acts as the sequencer) and has the ability to perform using three independent voice channels. The three sound channel outputs are mixed, along with Audio In, and sent to an internal speaker.

The sound generator's internal registers are accessed via the PSG Register Select Register (write only, reset: registers all zeros). The tone generator registers control a basic square wave while the noise generator register controls a frequency modulated square wave of pseudo random pulse width. Tones and noise can be mixed over individual channels by using the mixer control register.

The amplitude registers allow the specification of a fixed amplitude or of a variable amplitude when used with the envelope generator. The envelope generator registers permit the entry of a skewed attack-decay-sustain-release envelope in the form of a continue-attack-alternate-hold envelope.

2.5 Real Time Clock

The STBook system includes a Ricoh RP5C15 Real Time Clock chip. This provides time of day (down to one second resolution) and date. The RTC is provided with a 32.768 kHz oscillator that is independent of all other system clocks.

The chip is accessed through 32 4-bit registers accessed in two banks. Bank 0 allows reading and setting each digit of the date and time, and also allows access to test and control registers. Bank 1 allows setting the digits of an alarm function, and controlling the mode of operation of the clock chip.

2.6 Configuration Switch Register

The STBook implements an 8-bit configuration switch register to indicate the presence or absence of options. Depending on printed circuit board layout, the register may be implemented using an 8-bit DIP switch, solder pads, or double "row of stakes" jumpers. A bit will read as a "1" if the circuit is open As of this writing, the following bits have been assigned meanings:

Bit	Meaning
7	0 = No DMA sound hardware is installed. 1 = DMA Sound hardware is available.
6	0 = High speed (16 MHz) 1772 Floppy Disk controller is installed. 1 = Only low speed (8 MHz) 1772 Floppy Disk controller is installed.
5	0 = Bypass Self Test 1 = Self Test
4-0	Undefined, reserved.

3.0 Graphics Subsystem

The basic components of the graphics subsystem are video display memory, video controller (Internal to COMBO IC), SHADOW LCD Controller, and a Bit-level Transfer controller (BLiTTER inside COMBO IC).

3.1 Video Display Memory

Video display memory is configured as 1 logical plane in one 32Kbyte (actually 0x7d00) physical plane starting at any 256 byte half page boundary (in RAM only). The starting address of display memory is placed in the Video Base Address Register (read/write, reset: all zeros) which is then loaded into the Video Address Counter Register (read only, reset: all zeros) and incremented.

The STBook possesses only one of the three ST modes of video configuration: 640 x 400 resolution with 1 plane. The mode is set through the Shift Mode Register (read/write, reset: all zeros). An inverter is provided for inverse video, controlled by bit 0 of palette color 0 (normal video is black 0, white 1). In monochrome mode the border color is always black.

3.2 Video Controller

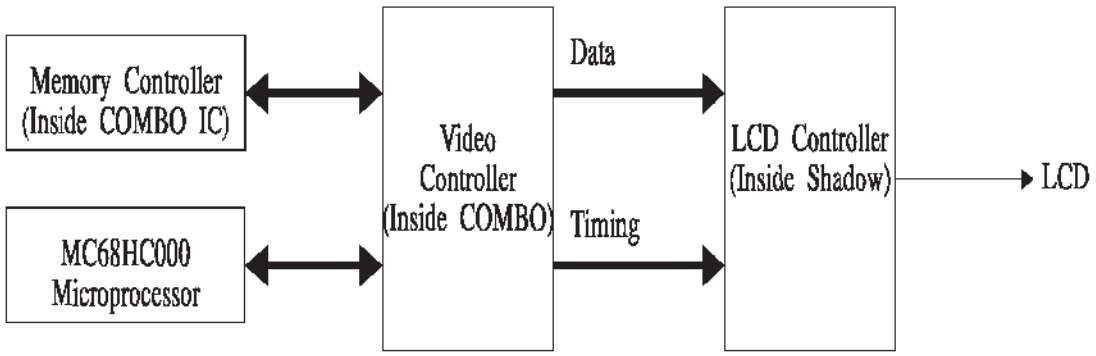
The video controller (a sub-section of the COMBO IC) controls the timing and memory transfers of the video system, including V/H Blank/Sync (which, in this LCD system, are relevant only as timing information).

The general flow of the video controller is as follows:

Bitmap data is taken from main memory one word at a time and presented to the SHADOW LCD Controller, along with synchronization information (i.e. Display Enable). It also presents enough data such that Horizontal Scrolling can be performed. The accesses to main memory are interleaved with the CPU accesses, such that the CPU can operate at virtually full speed.

There is (intentionally) no source of External Sync in the STBook; if it is selected, then the video controller will stop passing data from main memory to the SHADOW LCD controller. The SHADOW LCD controller is independent enough to maintain the LCD image without these updates; see below.

The following is a block diagram of the video controller:



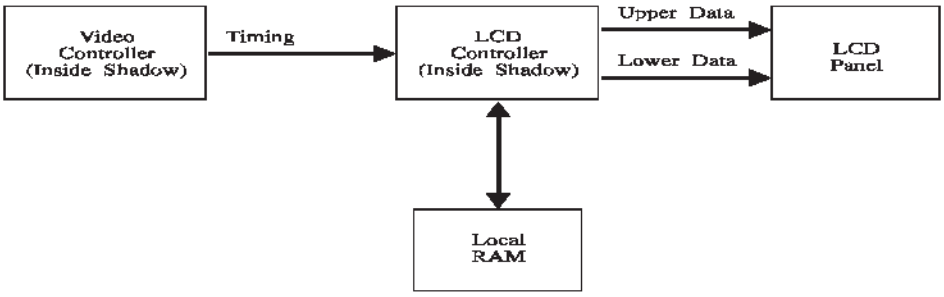
Video Controller Block Diagram

3.3 SHADOW LCD Controller

The LCD Controller acts as a buffer and multiplexer between the Video Controller and the LCD Panel. One reason this is necessary is that the LCD Panel is implemented (like most large-scale panels) as an upper and lower panel, driven in parallel. As such, it is scanned/loaded from Upper Left to Center Right, AND Center Left to Lower Right, simultaneously. As the Video controller transfers data corresponding to Upper Left to Lower Right, the LCD Controller must buffer the data so that it can be presented properly to the LCD Panel. It maintains a Local Static RAM to accomplish this.

The timing of the transfer from main memory and transfer to LCD Panel are independent. If the transfers from main memory stop (if, for example, External Sync is selected), the LCD Controller will continue to send data from its local RAM to the LCD Panel, maintaining the image. This feature is what allows us to stop video transfers (to save power) invisibly to the user. Video "updates" need only be performed when the image changes.

The following is a block diagram of the LCD Controller:



LCD Controller Block Diagram

3.5. Bit-Block Transfers

The Atari STBook Bit-Block Transfer Processor (BLiTTER) is a hardware implementation of the bit-block transfer (BitBlt aka blit) algorithm. Bit Blt can be simply described as a procedure that moves bit-aligned data from a source location to a destination location through a given logic operation. The BitBlt primitive can be used to perform such operations as:

- Area seed filling
- Rotation by recursive subdivision
- Slice and smear magnification
- Brush line drawing using Bresenham DDA
- Text transformations eg bold, italic, outline
- Text scrolling
- Window updating
- Pattern filling
- General memory-to-memory block copying

There are sixteen logic combination rules associated with the merging of source and destination data. Note that this set contains all possible combinations between source and destination. The following table contains the valid BitBlt combination rules:

3.5.1 Logic Operations

OP	COMBINATION RULE
0	All zeros
1	Source AND destination
2	Source AND NOT destination
3	Source
4	NOT source AND destination
5	Destination
6	Source XOR destination
7	Source OR destination
8	NOT source AND NOT destination
9	NOT source XOR destination
A	NOT destination
B	Source OR NOT destination
C	NOT source
D	NOT source OR destination
E	NOT source OR NOT destination
F	All ones

Adjustments, block extents, and several other transfer parameters are determined prior to the invocation of the actual block transfer. These adjustments and parameters include clipping, skew, end masks, and overlap.

Clipping. The source and destination block extents are adjusted to conform with a specified clipping rectangle. Since both source and destination blocks are of equal dimension, the destination block extent is clipped to the extent of the source block (or vice versa). Note that the block transfer need not be performed if the resultant extent is zero.

Skew. The source-to-destination horizontal bit skew is calculated.

End Masks. The left and right partial word masks are determined. The masks are merged if the destination is one word in width.

Overlap. The block locations are checked for possible overlap in order to avoid the destruction of source data before it is transferred. In non-overlapping transfers the source block scanning direction is inconsequential and can by default be from upper left to lower right. In overlapping transfers the source scanning direction is also from upper left to lower right if the source-to-destination transfer direction is up and/or to the left (ie source address is greater than or equal to destination address). However, if the overlapping source-to-destination transfer direction is down and/or to the right (ie source address is less than destination address), then the source data is scanned from lower right to upper left.

After the transfer parameters are determined the bit-block transfer operation can be invoked, transferring source to destination through the logic operation:

BIT-BLOCK TRANSFER