P R E L I M I N A R Y

ALTOS ACS 8600 COMPUTER SYSTEM

MONITOR PROGRAM

(Version 1.5)

APRIL 21,1982

DISCLAIMER

THIS IS A PRELIMINARY PUBLICATION OF THE ALTOS ACS 8600 COMPUTER SYSTEM MONITOR PROGRAM DOCUMENTATION AND IS PRESENTED FOR INFORMATION PURPOSES ONLY; IT MAY BE CHANGED WITHOUT NOTICE.

THE INFORMATION CONTAINED HEREIN REPRESENTS THE MOST CURRENT INFORMATION AVAILABLE AT THE TIME OF PRINTING, HOWEVER, THE INFORMATION HAS NOT BEEN THOROUGHLY REVIEWED NOR HAS IT BEEN APPROVED IN ITS PRESENT FORM. ALTOS COMPUTER SYSTEMS, INCORPORATED, ASSUMES NO LIABILITY WHATEVER RELATED TO USE OF INFORMATION PRESENTED IN THIS PUBLICATION.

Altos Computer Systems
World Headquarters
2360 Bering Drive
San Jose, CA 95131
U.S.A.
(408) 946-6700
Telex: 171562-6700

Altos Manual Part Number: 690-12166-001

TABLE OF CONTENTS

--

1.0 Monitor System Overview

The system, or resident, monitor is a collection of programs, remaining permanently in memory (on board PROM), providing the overall coordination and control of the operating system. This collection of programs performs several functions to aid the user. First, it permits several users' programs to be run simultaneously within the system. The monitor makes use of both the time-sharing hardware and software to maximize the systems efficentcy. Second, the monitor will process all data inputs from external programs. The drivers perform the tasks of checking port status, inputting data and handling basic error recovery and error notifacation. The I/O service routine preprocesses data so that all devices appear identical (FIFO) to the users' programs, thus simplifying both high and low level coding. Third, the monitor makes use of the program interrupt system to overlap input/output operations with computation. If a users' program must wait for completion of an input/output operation, the monitor will return control to the system and automatically switch to the next users' program in sequence. The users' program that was passed over will be run during the next pass. Once a program has been completed, the next program,in sequence, will be processed. A users' program may also be dumped on backing storage and discontinued under users' control.

1.1. Physical Location

The monitor is located at fixed locations FF000H to FFFFFH and 400H to C00H (Intel interrupt vectors are located at 000H to 3FFH). The object programs are loaded into the upper portion of the monitor, starting with the highest address, and working backwards (see figure 1). These programs cannot be written into any memory location other than the section specified. The hardware reserved locations are allocated for specific hardware routines. The object and hardware reserved locations cannot be overlayed. The program loader is also located in the upper portion of the monitor. This area can be overlayed by common storage. The monitor remains resident in the lower portion of the memory at all times.

2.0 Monitor Commands

The following index of monitor commands will give the user the format used in constructing the monitor commands.

A (<space>)	ALTER
B (<space>)	BREAKPOINT
D (<space>)	DISPLAY
G (<space>)	GO TO
I <B> (<space>)	BYTE I/O
I <W> (<space>)	WORD I/O
K (<space>)	DISK I/O
L (<space>) (<drive>)	LOAD BOOT
M (<space>)	MOVE
R (<space>)	REGISTER
S (<space>)	SINGLE STEP INSTRUCTION
X (<space>)	READ HEX

2.1 Initialization Commands

The monitor commands are one character command names followed by another command and/or option dependent operands. These operands are user selectable, base/stack relative, address or word values. The word values are limited to four characters and address values are usually two word values. Any byte operand values (e.g. mode, command, etc.) are also limited to a maximum of two characters. All operand numerical inputs must be hexadecimal. All newly inputted data will push (FILO) the old data out of the address. Any inputs, other than hexidecimal, will cause the system to default, "dump" the erroneous code input, sound the console bell and place an asterisk on the screen. This error routine will also initiate, when necessary, during the processing of a command (see Read INTEL Format Hex Data). The prompt will then return to its normal "period prompt" and request further inputs from the user. The user will not see any error indication should there be a hex code input error.

The user specified command port is read continiously for INTEL format hexadecimal data. The data is then placed in memory based on the control information within the data stream. If a checksum failure is dected during an input operation, the console bell will sound, an error indicator (asterisk) will be displayed, and the command will be terminated. If no errors are detected, the data stream will be read until the ending record is read from the port. This, then, reverts control to the user console. If the data stream contained a start address record, the CS and IP register save areas will be updated with that address. This will cause control to be passed to the loaded program from any user specified, subsequent Go command.

Load Bootstrap Command

L(<space>)(<drive>)<cr>

The first sector, of the user specified diskette, is read into memory. (This sector is expected to be a CP/M formatted command header record.) The header record, in the first sector, normally contains the base paragraph address for the command to be loaded. The base address is then extracted from the header record. The base address, and the remainder of the first two tracks of the diskette, are then read into that location. System control, of the Load Bootstrap Command, is then passed to the first byte of the relocated command. If the drive is not ready, the monitor is reentered and an error code is displayed representing the return code from the floppy controller.
If the <drive> specified is the hard disk, the track zero is read into memory starting at location 00C00H. System control is then passed to location 00C0:0000.

Byte or Word I/O Command

I<B>or<W>(<space>)<portno><space>or<,><data><cr>

This commandreads the word or byte of data from the specified port and displays the result to the user.

O<B>or<W>(<space>)<portno>space>or<,><data><cr>

This command writes the word or byte of data, specified by the user, to the selected I/O port.

Go to Address Command

G(<space>)(<starting address>)<cr>

The command transfers program control by setting the CS and IP registers to a newly specified value. No other specification will cause control to be transfered to the current settings of the CS and IP registers as defined in the register save area.

Step a Single Instruction

S(<space>)(<starting address>)<cr>

This command transfers control to the target address, if any, and executes a single instruction at that location. If no address is supplied, the system executes the next instruction pointed to be the CS:IP register pair. The monitor receives control following execution of the instruction, and displays the CS:IP register pair.

Set Breakpoint Command

B(<space>)<cr>
or
B(<space>)(<address>)<cr>
or
B(<space>)(<- breakpoint number>)<cr>

The user has three options available to him with this command:

(1) If no operands are specified in this command, the contents of the Breakpoint Table will be displayed on the terminal. This table consists of a breakpoint number and the address of the location to be breakpointed during the execution of a Go command.

(2) If an address is specified, it is set into the Breakpoint Table in the first available slot (assuming there is space available in the table). (The Breakpoint Table, has a maximum of eight slots/breakpoints pending at any one time.) The targeted memory locations will not be altered until the user specifies a Go instruction. The Breakpoint instruction (INT 3) will then be inserted into the program. This states that any subsequent addressing of that location will cause a program interrupt and reentry to the Monitor.

(3) The third option available with this command is to remove a breakpoint instruction from the table. This may be done to free up a slot in the breakpoint table, or just because the particular breakpoint has no further applicability. This is accomplished by entering a minus sign followed immediately by the breakpoint location to be removed from the table. The number may be obtained from the display (see option (1) of this section).

NOTE: Setting a TRAP flag in the register save area will also cause Breakpoints to occur. In this case, a Breakpoint will occur following the execution of each instruction.

Display Memory Command

D(<space>)(<starting address>)(<space>)<cr>
or
(<,><number of bytes>)<cr>

starting address - hex address of first byte to be displayed
number of bytes - hex number of bytes to be displayed

The monitor will automatically format the data into groups of sixteen bytes. Each line displayed will contain: (1) the address of the first byte displayed on the line and; (2) the number of bytes requested within the group of sixteen. Each line is broken up into groups of four bytes with the ASCII data displayed tothe right of each group. If the user wishes to interrupt a long display operation, simply press the escape (ESC) key and the display will terminate.

4.1. Functional Descriptions

4.1.1. MONITOR CALL (Monitor Code 00)

The monitor will reload all monitor segment registers and issue the dot prompt for Monitor commandsdiscussed earlier in this document.

4.1.2. CONSTAT (Monitor Code 01)

The Monitor will select the register CL indexed console port and test to see if it has a character waiting to be read. If there is a read pending, 0FFH will be returned in register AL; if no character read is pending, 000H is returned. No validity checks are performed on the console index, thus an invalid index will cause unpredictable results.

4.1.3. CONIN (Monitor Code 02)

The Monitor will select the register CL indexed console port and read a character from that location. If no character is available, the Monitor will wait until a character is available before returning control to the caller.

4.1.4. CONOUT (Monitor Code 03)

The Monitor will select the register CL indexed console and check the availability of the output channel. If the channel is available (DTR and TXE both true) then the character supplied in register DL will be written to the console. If the channel is unavailable, the Monitor will wait until it is able to output the character.

4.1.5. GETATTRIB (Monitor Code 04)

Return console attributes* in AX register for console index in CL register.

4.1.6. SETATTRIB (Monitor Code 05)

Set console attributes* from DX register for console index in CL.

*NOTE Console Attributes: The MSB of the attributes are the intelligent serial channel parameter register bits 0-7. The LSB of the attributesare the parameter register bits 8-15 (see intelligent serial channel specification, draft 9).

4.1.7. CRLF (Monitor Code 06)

The Monitor will select the register CL indexed console and write a carriage return and a line feed to that device. If the console is busy, the Monitor will wait until it is available, write the sequence and then return to the caller.

NOTE:If bit 15 of the track number is 1, the disk seek is initiated and the Monitor will return immediately to the caller. No data transfer will occur.

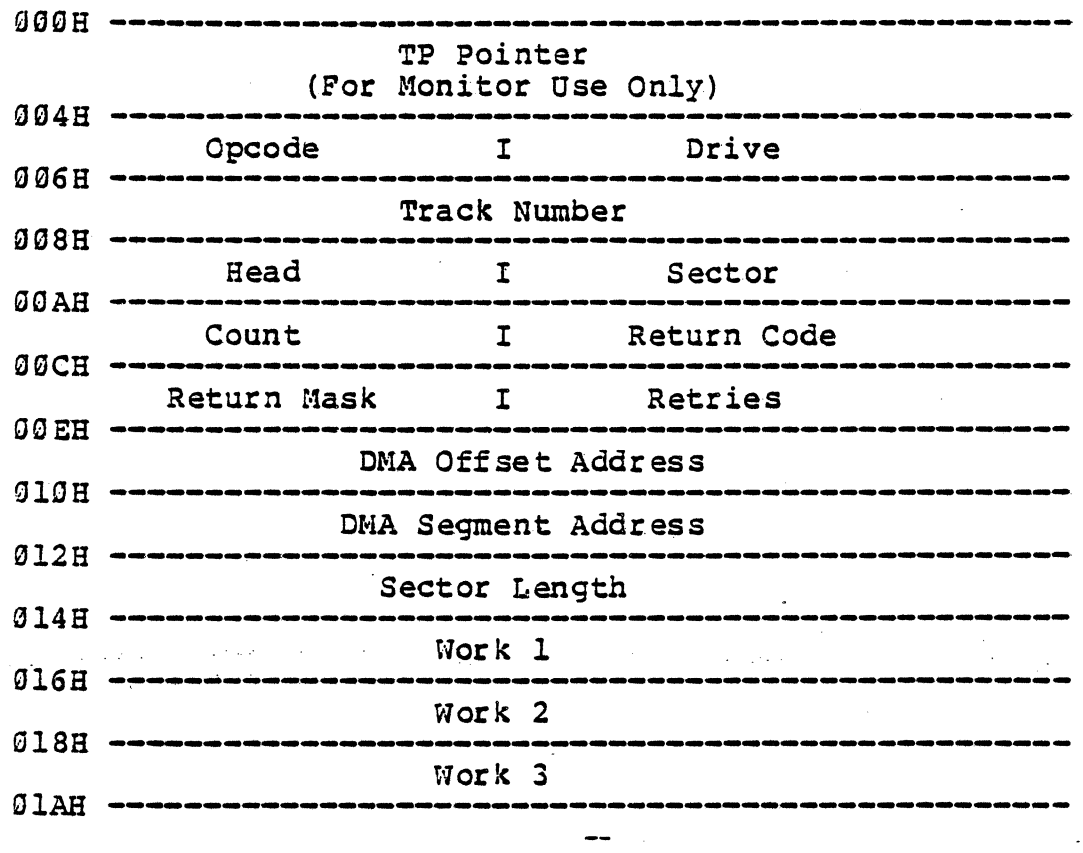

Figure 4-1 IOPB Mapping for Altos ACS8600

NOTE: Work 1, Work 2, and Work 3 are for 8089 use only

4.1.10. DISKCHECK (Monitor Code 09)

This checks the completion status of a disk seek in progress and returns it in the AL register (0 = not complete). Upon entry, CX must point to an IOPB previously used in a DISKIO code with bit 15 of the track number = 1.

4.1.11. CONDEF (Monitor Code 10)

Returns default console index being used by the Monitor in AL and top of memory in ES:DX.

4.1.12. BOOTCODE (Monitor Code 11)

Returns the Monitor boot code in AL (01H = Hard Disk, 02H = Floppy Disk).

5.0 Low Memory Utilization

The monitor needs low memory area because the monitor : (1) resides in EPROM and ; (2) needs RAM resident fields for temporary storage of variables (e.g. current disk drive number, time of day, etc.).

```
dbgptr equ   this dword ;400H pointer todebuggerstart location
dbgptr_off dw  0        ;400H debugger start offset
dbgptr_seg dw  0        ;402H debugger start segment

dbgptr2 equ    this dword ;404H pointer to debug monitorcall loc
dbgptr2_off dw 0        ;404H debugger start offset
dbgptr2_seg dw 0        ;406H debugger start segment

init_low epu   this word
scb      label  word     ;408H System Config. Block for 8089
soc      db     00000001B ;408H System Operation Command
         db     00000000B ;409H reserved
         dd     ccb       ;40AH pointer to Channel Control Block
;end of scb

ccb      label  word      ;40EH channel Control Block
ccw1     db     00000000B ;40EH CCW for Channel 1
busy1    db     00H       ;40FH busy flag for channel 1
cptr1    dd     cpb1      ;410H pointer to Command Parameter Blk.
         dw     00H       ;414H reserved
ccw2     db     00000000B ;416H CCW for channel 2
busy2    db     00H       ;417H busy flag for channel 2
cptr2    dd     cpb2      ;418H pointer to Command Parameter Blk.
         dw     00H       ;41CH reserved
;end of ccb

;Channel IOPB's
cpb1     dd     0         ;41EH space for IOPB
         dd     0         ;rest of IOPB
         dd     0         ;
         dd     0         ;
         dd     0         ;
         dd     0         ;
         dw     0         ;
iopblen equ     $-cpb1
cpb2     dd     0         ;438H space for IOPB
         dd     0         ;rest of IOPB
         dd     0         ;
         dd     0         ;
         dd     0         ;

         dd     0         ;
         dw     0         ;

;end of permanent offset area
```

7.0 INTEL Reserved Prom Locations

```
;-------------------------------------------------;
;                                                 ;
;    STRTSEG                                      ;
;                                                 ;
;    This segment provides the entry point to     ;
;    the system from the hardware originated      ;
;    jump.  It is also followed by a control      ;
;    block for the 8089 I/O processor which       ;
;    must reside at a known location in memory.   ;
;                                                 ;
;                                                 ;
;-------------------------------------------------;

;This segment MUST be loaded ar 0FFFF0H!!!!!!!!!!!!!!

strtss segment para public 'CODE'

       assume  cs:strtss

      jmp      far ptr reset  ;far jump to start system
      db       0              ;dummy filler to 0FFFF6H

;-------------------------------------------------;
;                                                 ;
;    The following block is the System Config-    ;
;    uration Pointer required by the 8089         ;
;    device.  The 8089 looks at this location,    ;
;    following the first CA after reset, to       ;
;    determine the start of the channel control   ;
;    block chain.                                 ;
;                                                 ;
;-------------------------------------------------;

scp   db        00000001B      ;indicate 16 bit data bus
      db        11111111B      ;reserved(must be 0FFHper INTEL)
      dd        scb            ;far add of Sys Config Block

strtsg ends
```