VII INTERFACE

The electrical interface between drive and controller is through the mother board edge connector J3. The description of pin out is as follows:

PIN #	NAME	DESCRIPTION
1	GND	Logic Ground
2	GND	Logic Ground
3	R/WCK	Read/Write Clock
4	GND	Logic Ground
5	RDGT	Read Gate
6	GND	Logic Ground
7	$\overline{\text{WTGT}}$	Write Gate
8	GND	Logic Ground
9	NRZWDATA	NRZ Write Data
10	GND	Logic Ground
11	INDEX	Index Mark
12	GND	Logic Ground
13	Sector	Sector Mark
14	GND	Logic Ground
15	$\overline{\text{NRZRDATA}}$	NRZ Read Data
16	GND	Logic Ground
17	SO	Servo Serial Data Out (into controller)
18	GND	Logic Ground
19	SI	Servo Serial Data In (from controller)
20	GND	Logic Ground

PIN #	NAME	DESCRIPTION
21	SIORDY	Servo SIO Ready-To-Receive
22	SERVORDY	Servo Ready
23	$\overline{\text{WRTSAFE}}$	Write Safe
24	POWEROK	Power OK
25	$\overline{\text{RWI}}$	Reduced Write Current
26	SERVOERR	Servo Error
27	$\overline{\text{PC}}$	Write Pre-Compensation
28	$\overline{\text{SERVORST}}$	Servo Reset
29	$\overline{\text{HSO}}$	Head Select 0 (head 1 + high)
30		reserved
31	-12V	-12V Supply
32	-12V	"
33	+12V	+12V Supply
34	+12V	"
35	+5V	+5V Supply
36	+5V	"
37	+5V	"
38	+5V	"
39	+12VMOTOR	Motor Power
40	GNDMOTOR	Motor Ground Return

apple computer inc.

SIZE A	DRAWING NUMBER 068-6001-A	
SCALE: N/A	SHEET 7	OF 25

The interface signals are TTL compatible and can be grouped into 5 groups:

1. Control Lines
2. Data Transfer Lines
3. Servo Signals
4. Miscellaneous Signal Lines
5. Power Supply Lines

A brief description of the signal lines is as follows:

1. Control Lines

a. READ GATE = Controller enables the drive to read data.

b. $\overline{\text{WRITE GATE}}$ = This is an active low signal to enable the drive to write data onto the disk.

c. POWER OK = Signal is provided by either controller or power supply to signify that power to the drive is OK.

d. $\overline{\text{LOW WRITE CURRENT}}$ = This is an active low line controlled by the controller to reduce the drive write current.

e. $\overline{\text{PC}}$ = This is an active low line controller by the controller to select the write precompensation function.

f. HEAD SELECT = Controller selects heads on the Widget drive. High level selects head and low level selects head 0.

g. $\overline{\text{WRITE SAFE}}$ = This line is controller by the Widget servo board. The line goes low after servo is ready.

2. Data Transfer Line

a. NRZ WRITE DATA = Standard 5 Mbit/s TTL NRZ write data from controller.

b. $\overline{\text{NRZ READ DATA}}$ = Standard 5 Mbit/s TTL NRZ read data to controller.

NOTE: See controller specs for details of read/write timing.

3. Servo Signal Lines

This group of signals consist of:

a. Pin 17 - SO

b. Pin 19 - SI

c. Pin 21 - SIORDY

d. Pin 22 - SERVORDY

e. Pin 26 - SERVOERR

f. Pin 28 - $\overline{\text{SERVORST}}$

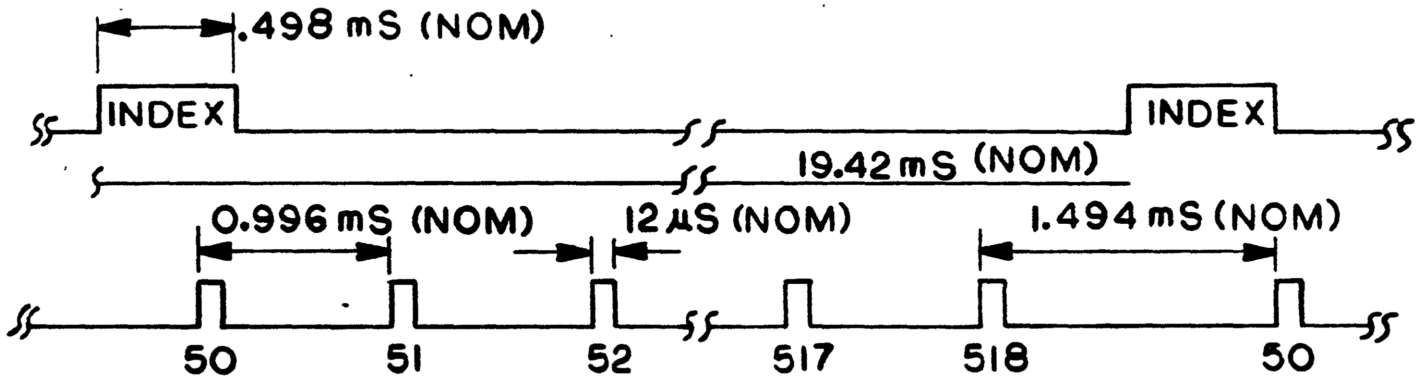
See section IX for details.

4. Miscellaneous Signal Lines

a. READ/WRITE CLOCK = The R/W clock is a TTL, 5 MHZ, 50% duty cycle signal. During the write and idle cycle, the clock is synchronized with a crystal clock on the R/W card. During the read cycle, the clock is synchronized with the read data.

b. INDEX = The nominal period is 19.42 ms ± 1%. See drawing below.

c. SECTOR = The nominal period is 0.996 ms between sectors, except between sector 18 and 0, where the period is 1.5X0.996 ms = 1.494 ms.



5. Power Supply Lines

a. -12V ± 5% = Pin 31 and 32
b. +12V ± 5% = Pin 33 and 34
c. +5V ± 5% = Pin 35, 36, 37, and 38
d. Circuit ground = Pin 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20
e. +12V ± 5% for spindle motor = Pin 39
f. Ground for spindle motor = Pin 40

IX SERVO

I. BASIC SERVO FUNCTIONS

Widget servo control functions are handled by a Z8 microprocessor. The Z8 handles all I/O operations, timing operations and communication with a host controller. Control functions to the Z8 Servo Controller are made through the serial I/O.

The following commands for the Widget servo are:

A. HOME - not detented, heads off data zones located at the inner stop.

B. RECAL - detented at one of two positions.

1. FORMAT RECAL: 32,(-0, +3)tracks from HOME. Used only during data formatting.

2. RECAL: 72,(-0, +3)tracks from HOME. Used to initialize into data zone.

C. SEEK - coarse track positioning of data head to any desired track location.

D. TRACK FOLLOWING - heads are detented on a specific track location and the device is ready for another command.

E. OFFSET - controlled microstepping of fine position system during TRACK FOLLOWING (two modes).

1. COMMAND OFFSET - direction and amount of offset is specified to the servo.

2. AUTO OFFSET - command allows the servo to automatically move off track by the amount indicated by the embedded servo signal on the data surface (disk).

F. STATUS - command can read servo status.

G. DIAGNOSTIC - not implemented.

See Table 1 for the actual command description. With the present command structure a SEEK COMMAND can be augmented with an OFFSET COMMAND. Upon completion of a seek, the offset command bit is tested to determine if an offset will occur following a seek (either auto or command offset).

When a SERVO ERROR occurs the Z8 SERVO will attempt to do a short RECAL (ERROR RECAL). Two attempts are made by the system to do the ERROR RECAL function. If either of the two RECAL operations terminate successfully the protocol status will be SERVO READY, SIO READY and SERVO ERROR. Should the ERROR RECAL fail then the system will complete the error recovery by a HOME function.

The two OFFSET commands will be described. First COMMAND OFFSET is a predetermined amount of microstepping of the fine position servo. Included in the OFFSET BYTE (STATREG) bit B6=0 is a COMMAND OFFSET. Bit B7=1 is a forward offset step (toward the spindle); B7=0 is a reverse step.
If bit B6=1, the OFFSET command is AUTO OFFSET.

AUTO OFFSET command normally occurs during a write operation. When the HDA was initially formatted at the factory, special encoded servo data was written on each track "near" the index zone. The reason for this follows:

Normal coarse and fine position information for the position servos is derived from an optical signal relative to the actual data head-track location. Over a period of time, the relative position (optical signal) will be misaligned to the absolute head-track position by some unknown amount (less than 100 uIn). This small change is important for reliability during the write operation. Write/Read reliability can be degraded due to this misalignment. The special disk encoded servo signal is available to the fine position servo. It will correct the difference between the relative position signal of the optics and the absolute head to track position under the data head only at index time. The correction signal can be held indefinitely or updated (if desired at each index time) until a new OFFSET command or move command (SEEK or RECAL) occurs.

II. COMMUNICATION FUNCTIONS

The servo functions described in the previous section only occur when the servo Z8 microprocessor is in the communication state. Communication states occur immediately after a system reset, upon completing head setting after a recal, seek, offset, read servo status or set servo diagnostic command. A special communication state exists after a servo error has occurred. If + SIO READY is not active, no communication can exist between the external controller and the servo Z8 processor.

Servo commands are serial bits grouped as five separate bytes total. Refer to Table 1 parts I through V for the total communication string. The first byte is the command byte (i.e. seek, read status, recal, etc.). The second byte is the low order difference for a seek (i.e. Byte 2 = $0A is a ten track seek). The third byte is the offset byte (AUTO or COMMAND OFFSET and the magnitude/direction for command offset). The fourth byte is the status and diagnostic byte (use for reading internal servo status or setting diagnostic commands). Byte five is the check sum byte used to check verify that the first four bytes were correctly transmitted (communication error checking).

Part of the communication function requires a specific protocol between the servo Z8 processor and the external controller.

III. Z8 SERVO PROTOCOL

The protocol between the Z8 SERVO microcomputer and the CONTROLLER is based on five I/O lines. Two of the I/O lines are serial input (to Z8 servo from controller) serial output (from Z8 servo to controller). Data stream between the Z8 servo and controller is 8 bit ASCII with no parity bit (the fifth byte of the command string contains check sum byte use for error checking). There are three additional output lines between the Z8 servo used as control lines to the controller. Combining the two serial I/O lines and the three unidirectional port lines generates the bases of the protocol between the Z8 servo and controller. The important operations between the Z8 servo and controller are:

1. Send commands to Z8 servo.

2. Read Z8 servo status.

3. Check validity of all four command bytes.

4. I/O timing signals between the Z8 servo and controller.

5. Z8 servo reset.

Sequencing the Z8 servo controller is an important process following a Power Up (Power On Reset) or if the controller should issue a Z8 Servo Reset at any time. After a Z8 Servo Reset is inhibited, the Z8 I/O ports and internal register are initialized. This takes approximately 75 msec after the Z8 Servo Reset is inhibited. The protocol baud rate is automatically set to 19.2KB and then the system is parked at HOME position and SIO READY is set active. ***IMPORTANT***. If the desired baud rate needs to be increased to 57.6KB; **after a Z8 Servo Reset is the ONLY time this can be done***. Once set to 57.6KB the communication rate remains at 57.6KB until a Z8 Servo Reset occurs. Setting 57.6KB is achieved as follows:

1. Z8 Servo "Power On or Controller" Reset

2. Wait for SIO Ready

3. Send a READ STATUS COMMAND as follows:

   BYTE 1 = $ 00
   BYTE 2 = $ 00
   BYTE 3 = $ 00
   BYTE 4 = $ 87

After the completion of transmitting the bytes, the Z8 Servo Controller changes to 57.6KB and will be waiting for the next transmitted command at 57.6KB.

Before the controller transmits the command byte the controller must pole the SIO READY line from the Z8 servo to determine if it is active (+5 volts). If the line is active then a command can be transmitted to the Z8 servo. The program in the Z8 servo will determine what to do with the command bytes (depending upon the current status of the Z8 servo). After the command (five bytes long) has been transmitted to the Z8 servo, the program in the Z8 servo will determine if the command bytes (first four bytes) are in error by evaluating the check sum byte (fifth byte transmitted). After the controller has transmitted the last serial string it must wait 250 usec then test for SERVO ERROR active (+5 volts). If SERVO ERROR is active the command was rejected (check sum error or invalid command). If SERVO ERROR is set active 600 U sec after the command is sent (and not 250 U sec), this was a command reject. The SERVO ERROR must be cleared by a READ STATUS COMMAND or RECAL COMMAND before transmitting another command.

As long as SIO READY is active the controller can communicate with the Z8 Servo Controller. If SERVO READY is not active the only command that will cause the Widget Servo to set SERVO READY active is a RECAL COMMAND (NORMAL or FORMAT). Read Status will only clear SERVO ERROR, and all other commands will be rejected.

Next, if SERVO READY is active and SERVO ERROR is also active, SERVO ERROR can be cleared by:

1. Any READ STATUS COMMAND.

2. Any RECAL COMMAND.

3. Any other commands will be rejected and maintain SERVO ERROR.

If a SEEK COMMAND is transmitted with both SERVO READY and SERVO ERROR active, the command will be rejected.

It is important to check the status of all three status lines from the Z8 Servo. It is best to avoid sending a SEEK COMMAND with SERVO READY and SERVO ERROR active.

Chart V, parts A-I, illustrate some of the serial communication commands and error conditions that can occur between the controller and Z8 SERVO.

IV. ERROR HANDLING

SERVO ERROR will be generated during the following conditions:

1. During Recal mode (velocity control only) access time-out.If a Recal function exceeds 150 msec then an access timeout occurs.

2. During Seek mode (velocity control only) access time-out. If a Seek function exceeds 150 msec then an access time-out occurs.

3. During Settling mode (following a Recal, Seek, or Offset) if there is excessive On Track pulses (3 crossings) indicating excessive head motion, a Settling error check will occur.

4. During a command transmission if a communication error occurs (check sum error).

5. During a command tansmission if a invalid command is sent.

Z8 SERVO COMMAND BYTES
TABLE 1

I. BYTE 1: COMMAND BYTE (DIFCNTH)

		B7	B6	B5	B4	FUNCTIONS
	---	1	0	0	0	access only
	B7	1	0	0	1	access with offset
command	B6	0	1	0	0	normal recal (to trk 72)
bits	B5	0	1	1	1	format recal (to trk 32)
	B4	0	0	0	1	offset-trk following
	---	1	1	0	0	home-send to ID stop
		0	0	1	0	diagnostic command
		0	0	0	0	read status command

	B3 -X- not used
access	B2 -access direction
bits	B1 -hi diff2 (512)
	B0 -hi diff1 (256)
	---

access direction = 1 (FORWARD: toward the spindle)
= 0 (REVERSE: away from the spindle)

hi diff2 (512) = 1 (512 tracks to go)
= 0 (not set)

hi diff1 (256) = 1 (256 tracks to go)
= 0 (not set)

II. BYTE 2: DIFF BYTE (DIFCNTL)

command BYTE 2 contains the LOW ORDER DIFFERENCE COUNT for a seek

---	
B7 -bit7 =	128 tracks
B6 -bit6 =	64 tracks
B5 -bit5 =	32 tracks
B4 -bit4 =	16 tracks
B3 -bit3 =	8 tracks
B2 -bit2 =	4 tracks
B1 -bit1 =	2 tracks
B0 -bit0 =	1 track

III. BYTE 3: OFFSET BYTE (STATREG)

command BYTE 3 contains the INSTRUCTION for an OFFSET COMMAND (seek or during track following)

```
---
B7 -offset direction
B6 -auto offset function
B5 -not used
B4 -offset Bit4 = 16
B3 -offset bit3 =  8
B2 -offset bit2 =  4
B1 -offset bit1 =  2
B0 -offset bit0 =  1
---
```

1. if offset command from BYTE 1 is followed by bit6 set (auto offset); offset direction (bit7) read offset (bit5) and bits 4-0 are ignored but should be set to 0 if not used.

```
2.  OFFSET DIRECTION = 1 (FORWARD OFFSET:  toward the spindle)
                     = 0 (REVERSE OFFSET;  away from the spindle)

3.  AUTO OFFSET      = 1 (normally used preceeding a write operation)
                     = 0 (manual offset:  MUST send direction and
                         magnitude of offset)
```

IV. BYTE 4: STATUS BYTE (CNTREG)

```
---
B7 -communication rate
B6 -power on reset
B5 -not used
B4 -not used
B3 -status or diagnostic bits
B2 -status or diagnostic bits
B1 -status or diagnostic bits
B0 -status or diagnostic bits
---
```

```
B7 = 0; communication rate is 19.2 KBAUD
   = 1; communication rate is 57.6 KBAUD

B6 = 0; power on reset bit is not active
   = 1; power on reset bit is active
```

V. BYTE 5: CHECKSUM BYTE (CKSUM)

[B7 B6 B5 B4 B3 B2 B1 B0]

results of the transmitted CHECKSUM BYTE are derived as:

$\overline{(\text{BYTE 1 + BYTE 2 + BYTE 3 + BYTE 4})}$ = CHECKSUM BYTE

(+) is defined as the addition of each BYTE

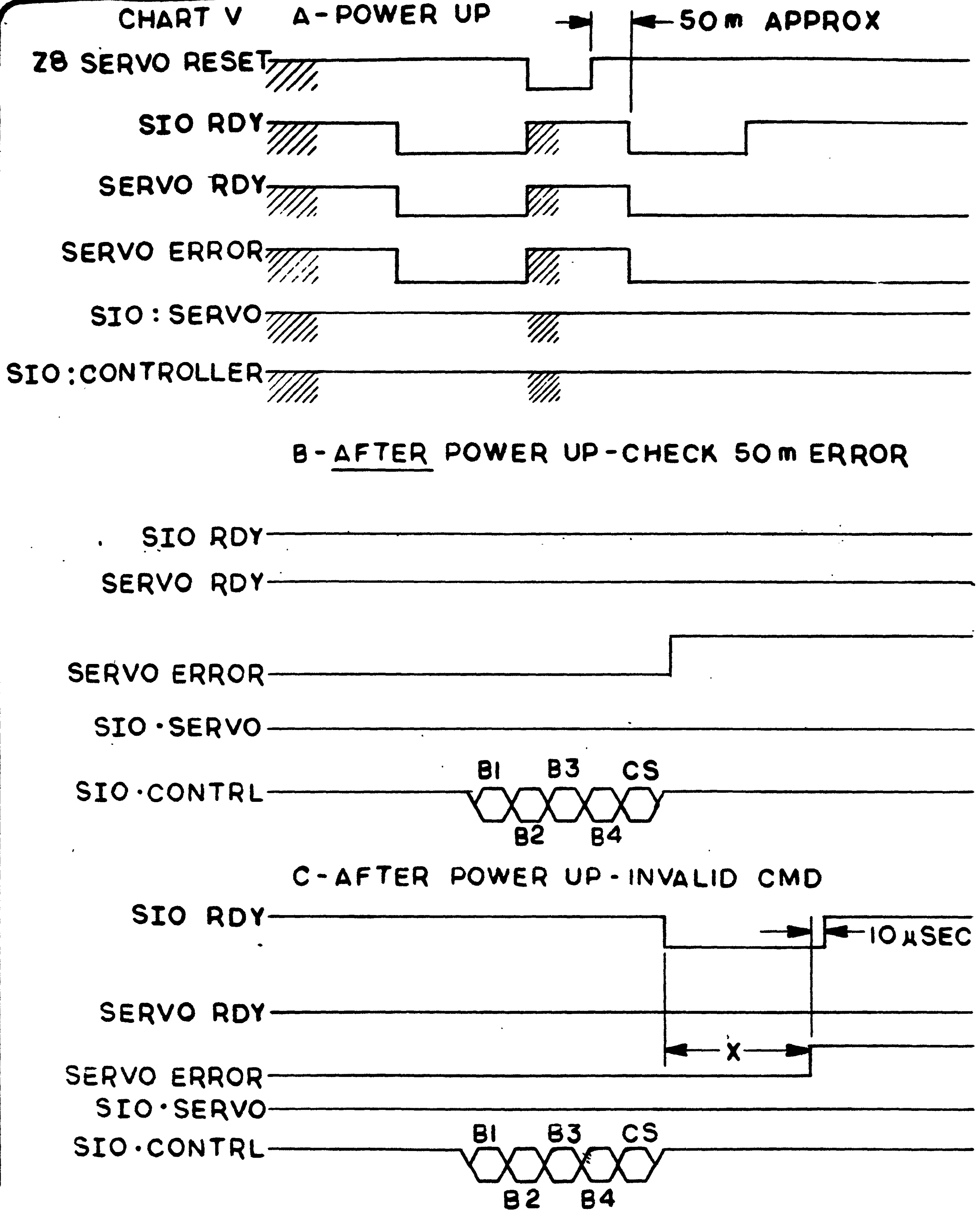
$\overline{(\text{BYTE})}$ is defined as the compliment of the BYTES (1-4)

VI. The SERVO STATUS lines (SIO RDY, SERVO RDY, SERVO ERROR) must have the following conditions in order to send the listed Z8 COMMANDS:

		SERVO STATUS		
		SIO RDY	SRV RDY	SRV ERR
Z8 SERVO CMD	HEX			
access (only)	8X	1	1	0
access (offset)	9X	1	1	0
recal (data)	40	1	X	X
recal (format)	70	1	X	X
park	C0	1	X	X
offset (detent)	10	1	1	0
status	00	1	X	X
diagnostic	20	———————————— not implemented		

X = either 0, 1

CHART V



A-POWER UP

B-AFTER POWER UP-CHECK 50m ERROR

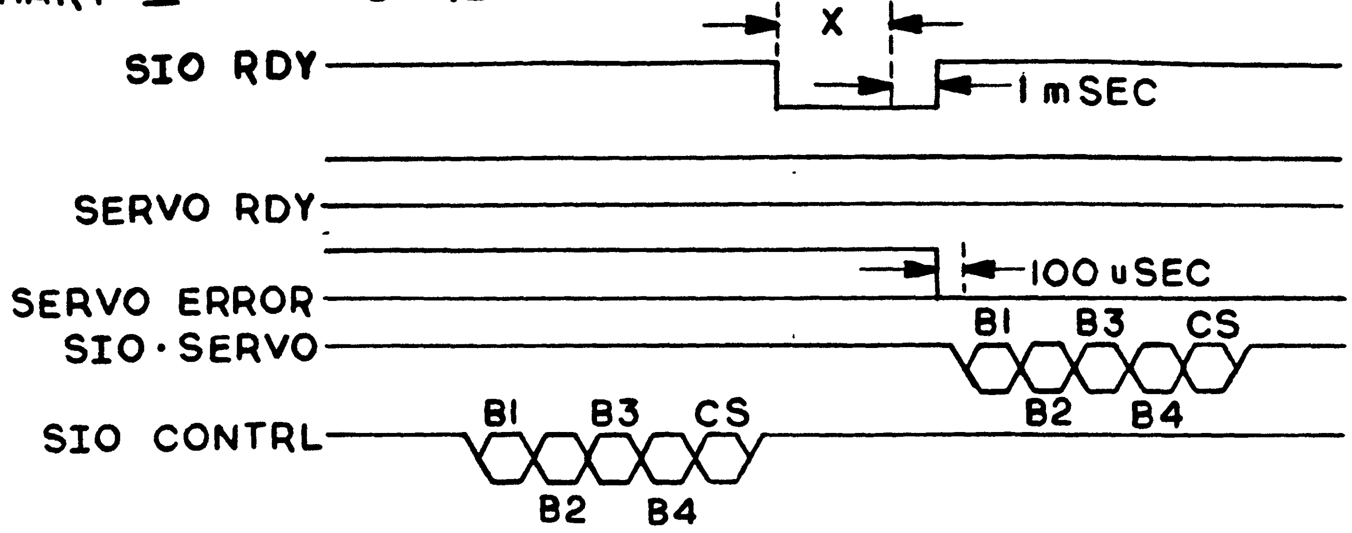
C-AFTER POWER UP-INVALID CMD

apple computer inc.

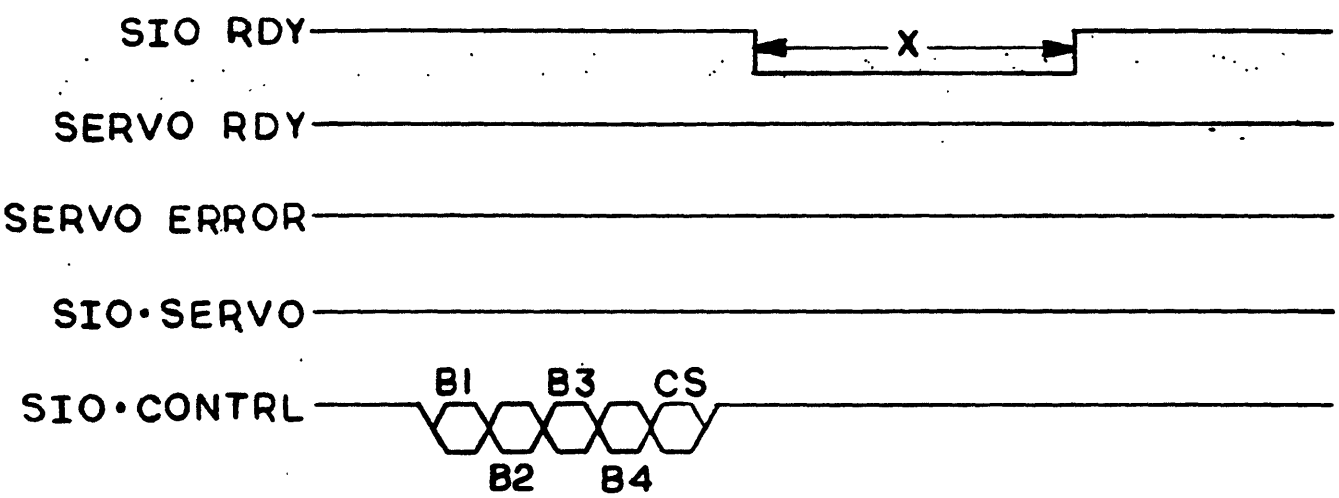
SIZE	DRAWING NUMBER
A	068-6001-A
SCALE: N/A	SHEET 20 OF 25

CHART V

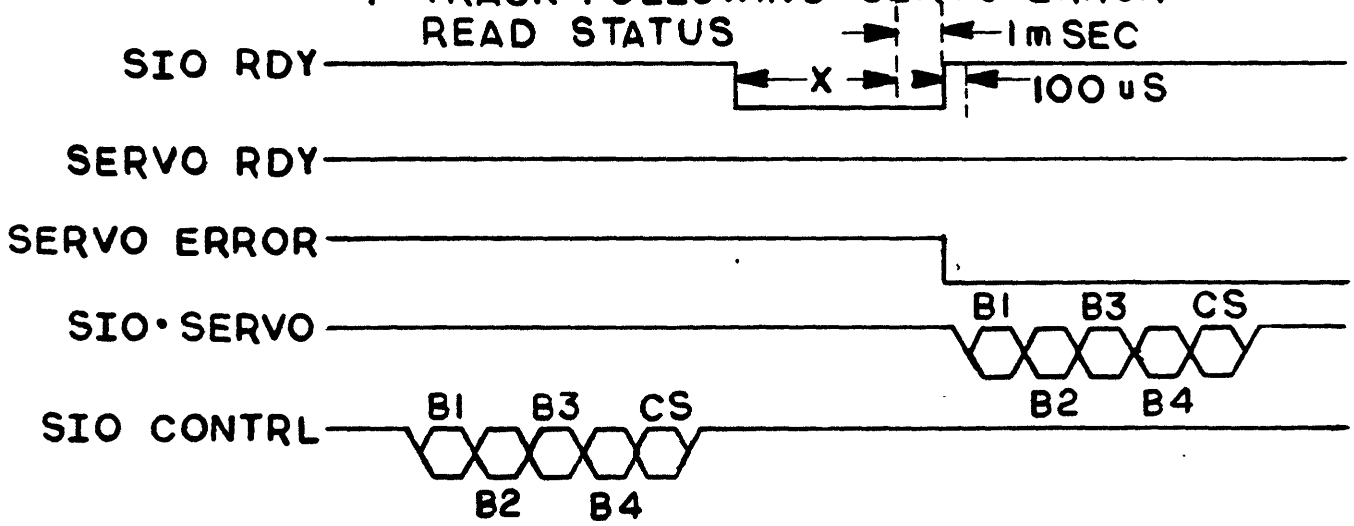
D - READ STATUS COMMAND



E - TRACK FOLLOWING SERVO ERROR - INVALID COMMAND



F - TRACK FOLLOWING SERVO ERROR - READ STATUS

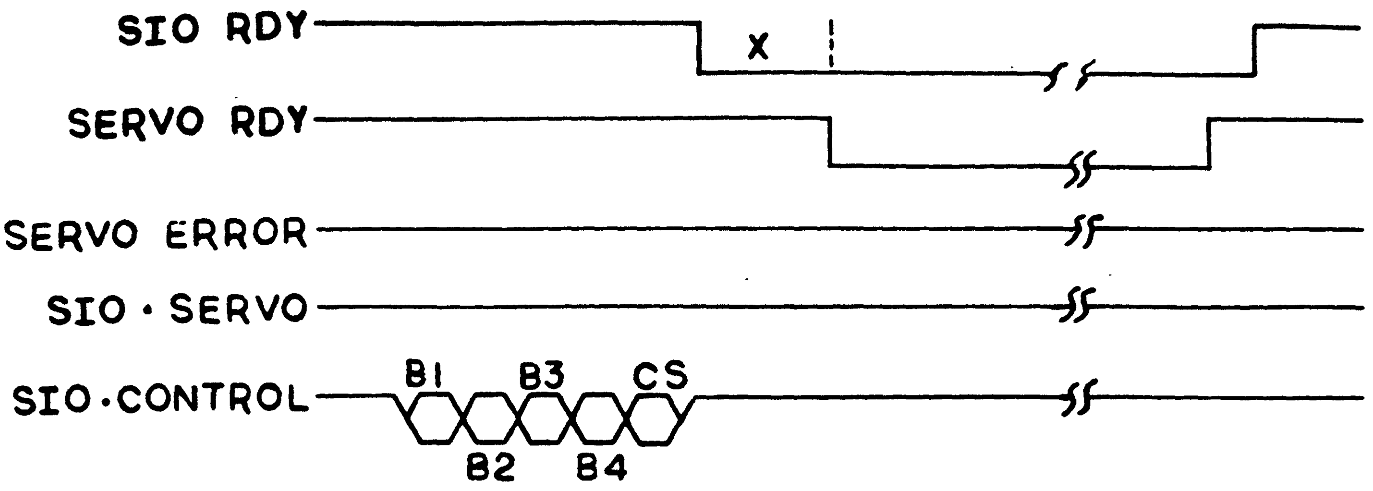


apple computer inc.

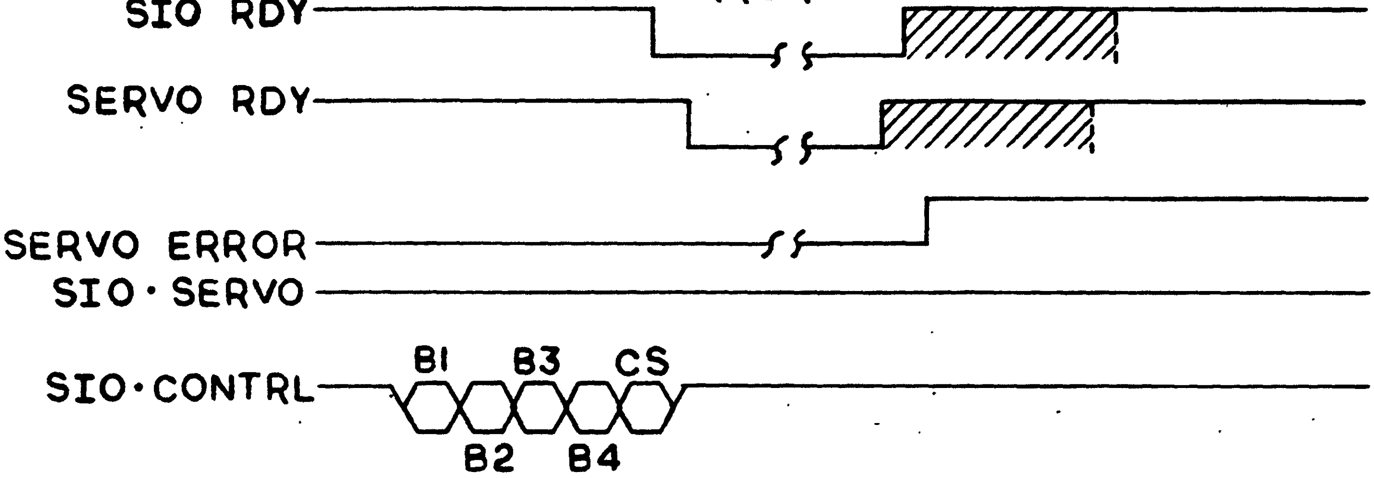
SIZE A	DRAWING NUMBER 068-6001-A	
SCALE: N/A	SHEET 21 OF 25	

CHART V

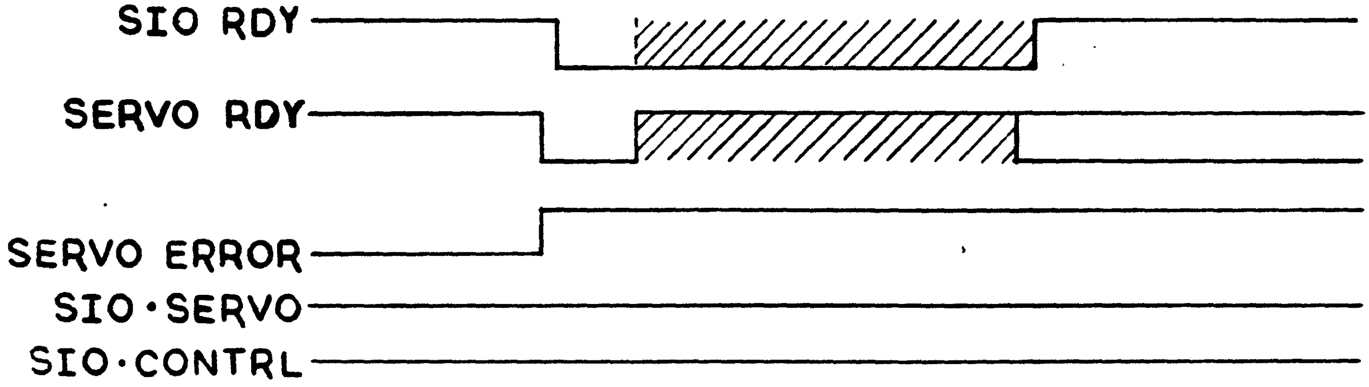
G- TRACK FOLLOWING VALID COMMAND (MOVE)



H - TRACK FOLLOWING (MOVE CMD) FOLLOWED BY SERVO ERROR



I - TRACK FOLLOWING (NO CMD) SERVO ERROR



apple computer inc.

SIZE	DRAWING NUMBER
A	068-6001-A

SCALE: N/A

SHEET 22 OF 25