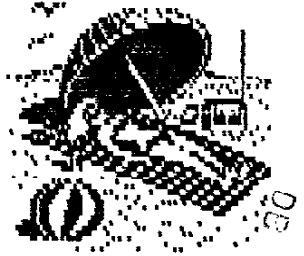
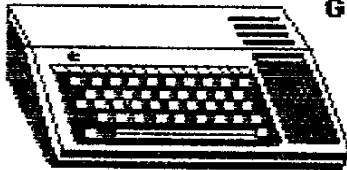


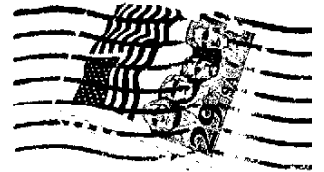
GUILFORD 99'ERS NEWSLETTER



SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER



GUILFORD 99'ERS UG
3202 CANTERBURY DR
GREENSBORO NC
27408



TO:

George von Seth, Pres. (292-2035)
Tony Kleen, Sec/Treas (924-6344)
BBS: (919)621-2623 --ROS

Bob Carmany, Newsletter Ed (855-1538)
Bill Woodruff, Pgm/Library (228-1892)

The Guilford 99'er Users' Group Newsletter is free to dues paying members
(One copy per family, please). Dues are \$12.00 per family, per year. Send
check to: Tony Kleen c/o 3202 Canterbury Dr., Greensboro, NC 27408. The
Software Library is for dues paying members only. (Bob Carmany Ed)

OUR NEXT MEETING

DATE: August 6, 1991 Time: 7:30 PM. Place: Glenwood Recreation
Center, 2010 S. Chapman Street.

Program for this meeting will be a demonstration of how to create
boot up menus in both TI-FORTH and Wycove FORTH. The 'hows' and
'whys' of how the selections are made to appear on the screen and
what happens when you make your choice will be explored.

MINUTES

The July meeting of the Guilford 99'er Users' Group was held on Tuesday the second, at the Glenwood Recreation Center on Chapman Street in Greensboro, NC. There were six members present.

The meeting opened promptly at 7:30, led by director George von Seth. There was some old business to attend to. We had talked earlier, at the June meeting, about upgrading the PE-Box with a DSSD drive. I reminded the membership that the PE-Box was 'on loan', and not purchased. Since the owner has a spare DSSD half-height at home just collecting dust, he agreed to replace the old SSSD: sometime.

New business was limited to who would present next month's program. Bob Carmany graciously accepted the opportunity to show us a menu driven application written in FORTH. Bob, we're sure sorry we twisted your arm so far that we sprained your elbow.

Bob gave us a report on our friends from Australia. Peter Smith has had a quadruple heart bypass surgery but is recovering quickly. Tony McGovern is to have another cornea implant. Hope this goes well, and you both recover quickly. Also, Funnalweb 4.40 has been completed (and distributed?). Bob says that version 4.50 will probably be a rewrite of the TI-Writer Editor.

The secretary/treasurers reports were accepted as written/read. As of this writing the club has \$177.00 in the treasury.

The program for July was a presentation of my latest TI-BASE application, the MENU Manager. This application's purpose is to allow a new user of any of my other applications to simply insert the 'KLEEN' application diskette and be allowed immediate access to those other applications. If you tailor the SETUP command file supplied on the 'KLEEN' diskette to your system configuration, and then boot the TI-BASE software, then the Menu Manager's application software is automatically loaded for you and you are given an informational message as to its availability.

As with all of my other applications, a <T>utorial selection is available. The remainder of the presentation was a review of the tutorial. I then demonstrated how each 'menu screen' has a 'help panel' available (one simply requests the <h>elp selection). Several constructive suggestions were given, as were questions and inquiries that will quite likely lead to improvements in this application's performance and use. Thank you for taking such an active interest in this project. I'll keep you informed as to its progress.

Respectfully Submitted,

Tony Kleen

WITH CONDOLENCES

Both George and I got the same disappointing letter the other day from Brian Woods (Brian is the Secretary of the Hunter Valley UG). I think the quoted text of the letter says it all!!!

"At the Annual General Meeting of the Hunter Valley 99ers Users Group held last Tuesday 25th June, it was decided by the attending members that the Group should be wound up.

The reasons for this decision are numerous, the main reasons being a lack of members willing to stand for Committee positions, the almost non-existent contributions to the newsletter, and the general lack of programs available locally for the TI, resulting in more people upgrading to other computers. As well, it has become more difficult to arrange demos, etc at the monthly meetings that would be of interest to those few members that attend."

With these brief paragraphs, the Hunter Valley UG has ceased to exist! It was, at one time, one of the largest TI Users Groups in Australia and the entire world. The talent contained in that single organization was truly spectacular--the McGovern's (F'WEB), Neil Quigg (QUEST RAMdisk and 32K cartridge), Ron Kleinschafer (Quest software), etc., etc.

The McGovern's intend to develop F'WEB for awhile yet and Ron Kleinschafer will also continue to develop QUEST. The sad part is the cessation of the outstanding newsletter that will end with the issue in mid-August.

When you ponder the future of our UG, consider Brian's remarks and see if there isn't something you can do to preserve the existence of our 'little band'. It sounds like he could be describing the Guilford 99ers!!

TI DISK SPECS

The diskette used on the Home computer Disk Peripheral has the following specifications:

Capacity: 92160 bytes per disk; 2304 bytes per track; 256 bytes per sector; 9 sectors per track

Encoding method: FM Single Density Recording

Mini diskette type SA 104 (ANSI standard 5.25")

The specified diskette contains a total of 360 sectors of 256 bytes each. In the remainder of this chapter each sector will be addressed as if the diskette was a linear media, i.e. track 0 sector 0 will be designated "sector 0"; track 39 sector 8 equals "sector 359".

The following section contains a description of the logical structure on each diskette in terms of records.

Physical Diskette Format:

The general diskette format used in the TI-99/4 Disk Peripheral is the following:

Sector 0 contains the Volume Information Block (VIB). This block contains general information about the diskette like:

Volume Name
Number of available Allocation Units
Number of sectors/track
Allocation Bit Map

Sector 1 contains pointers to file descriptor records.

Sector 2 thru 359 contain File Descriptor Records and data blocks.

The File Descriptor Records contain general information about the file, such as:

File name
File status data
File data access blocks

Volume Information Block

As mentioned previously, this block contains general information about the diskette. A more detailed description of each entry and its contents will be given in this section.

Bytes 0-9 contain the volume name of the diskette. The volume name can be any combination of ten ASCII characters, except for the space or period (".") characters and the null character (ASCII code 0). The name is space filled to the right in case of less than 10 characters. The volume name must contain at least one non-space character.

Bytes 10-11 give the total number of allocation units (AUs) on the volume. This datum should match the allocation bit map.

Byte 12 indicates the number of sectors per track.

Bytes 13-15 contain the ASCII code for "DSK", which is used by the disk manager software to check if the diskette has been initialized.

Byte 16 contains the ASCII code for "P" if the diskette is protected (a protected disk is also called a proprietary disk), otherwise this byte contains a >20.

Byte 17 indicates the number of tracks per side.

Byte 18 indicates the number of formatted sides on the diskette.

Byte 19 indicates the density of the diskette.

Bytes 20-55 are reserved for future expansions like date and time of creation. In the current version of the disk software these bytes are set to zero.

Bytes 56-255 contain the allocation bit map. This 200 byte map can control up to 1600 256-byte records (total controllable storage capacity = 400K bytes), which make it useable for a double density, double sided diskette. The disk allocation system uses a conventional method of allocating disk space called Bit Maps. Each bit in the bit maps represents one sector on the disk. A logical one in the bit maps means that the corresponding sector has been allocated. A zero means that the sector is still available.

The volume name can be used as an alternative to the actual disk drive name, i.e. the user can specify a disk drive in either of the following ways:

DSK.volname.filename or DSKn.filename

If the volume is specified, rather than the physical drive number, the system will look in sequence on every drive in the system, until it finds the specified volume. If more than one volume of the same name exists, the drive with the lowest drive identification number will be assigned.

File Descriptor Index Record

The File Descriptor Index Record contains up to 127 two byte entries, each pointing to a file descriptor record. These pointers are alphabetically sorted according to the filename in the associated file descriptor record. The pointer list starts at the beginning of this block, and ends with a zero entry.

Since the file descriptors are alphabetically sorted in this block, a binary search method can be used to find any given filename. Limiting the maximum number of disk searches to 7 if more than 63 files are defined. In general if between $2^{(N-1)}$ and $2^{(N)}$ files are defined, a file search will take at most N disk searches. To obtain faster directory search response times, the system will prefer to allocate data blocks in the area above AU number 34. Only if no AU can be allocated in that area will the disk data block allocator start allocating blocks in the AU area 2-33.

File Descriptor Records.

The File Descriptor Record (FDR) contains general information about the associated file. All the information the system needs to know to access and update the file has to be contained within the file descriptor record.

The physical layout of an FDR is:

Bytes 0-9 contain a filename up to ten characters in length.

Bytes 10-11 are reserved for future extension of the number of data chain pointers through linkage to a data chain pointer block chain. In the current version these bytes are always 0.

Byte 12 contains the file status flags. These flags are to be interpreted as follows (bit 0 is the least significant bit):

0: Program/data file indicator 0 = Data file 1 = Program file

1: Binary/ASCII data 0 = ASCII data (DISPLAY file) 1 = Binary data (INTERNAL or program file)

2: Reserved for future data type expansion

3: Protect flag 0 = Not protected 1 = Protected

4-6: Reserved for future expansion

7: FIXED/VARIABLE flag 0 = Fixed length records 1 = Variable length records

Byte 13 contains the number of logical records per AU.

Bytes 14-15 contain the number of logical records allocated on Level 2 (256 byte records).

Byte 16 contains the EOF offset within the highest physical AU for variable length record files and program files.

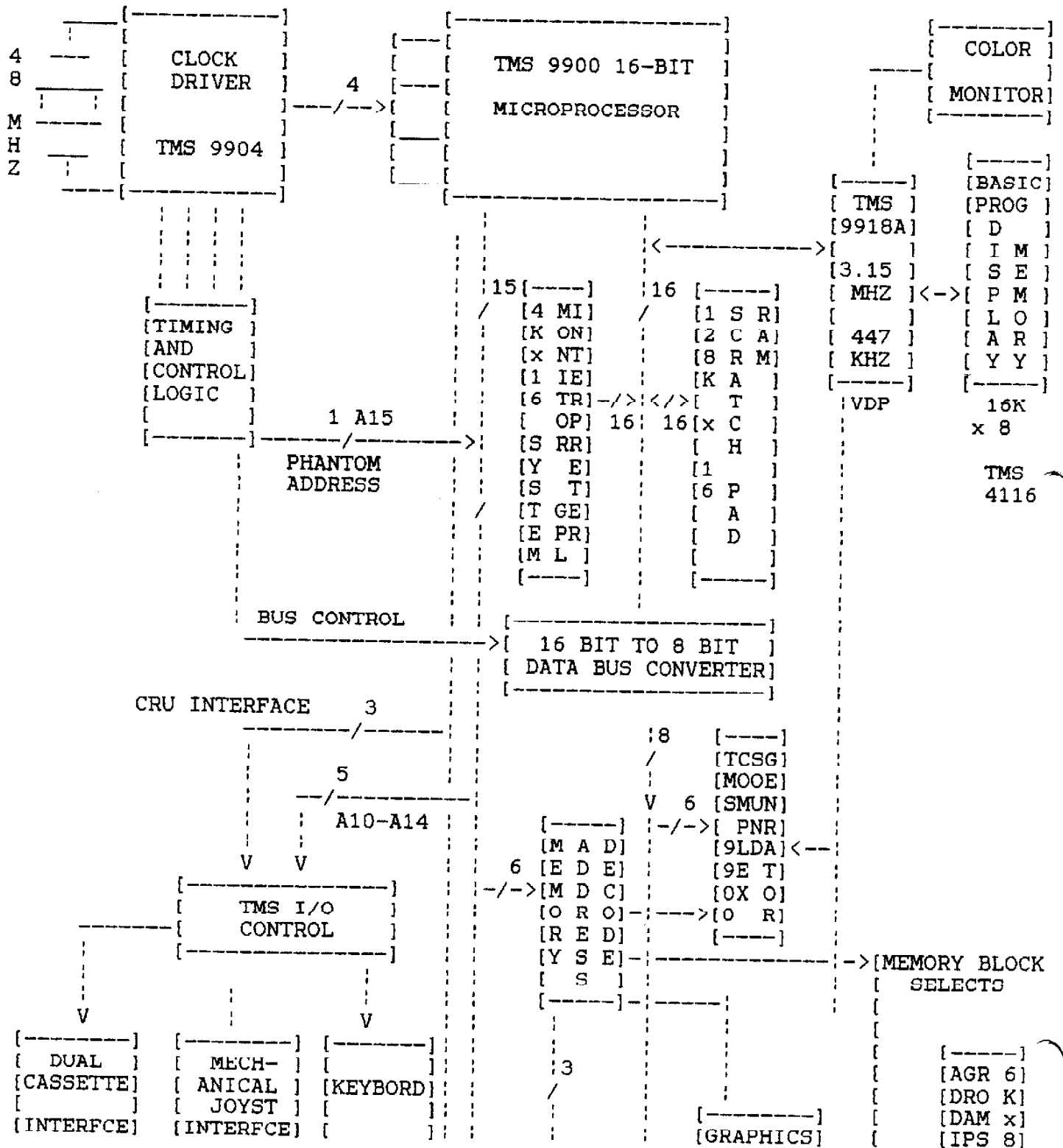
Byte 17 contains the logical record size in bytes. In case of variable length records, this entry will indicate the maximum allowable record size.

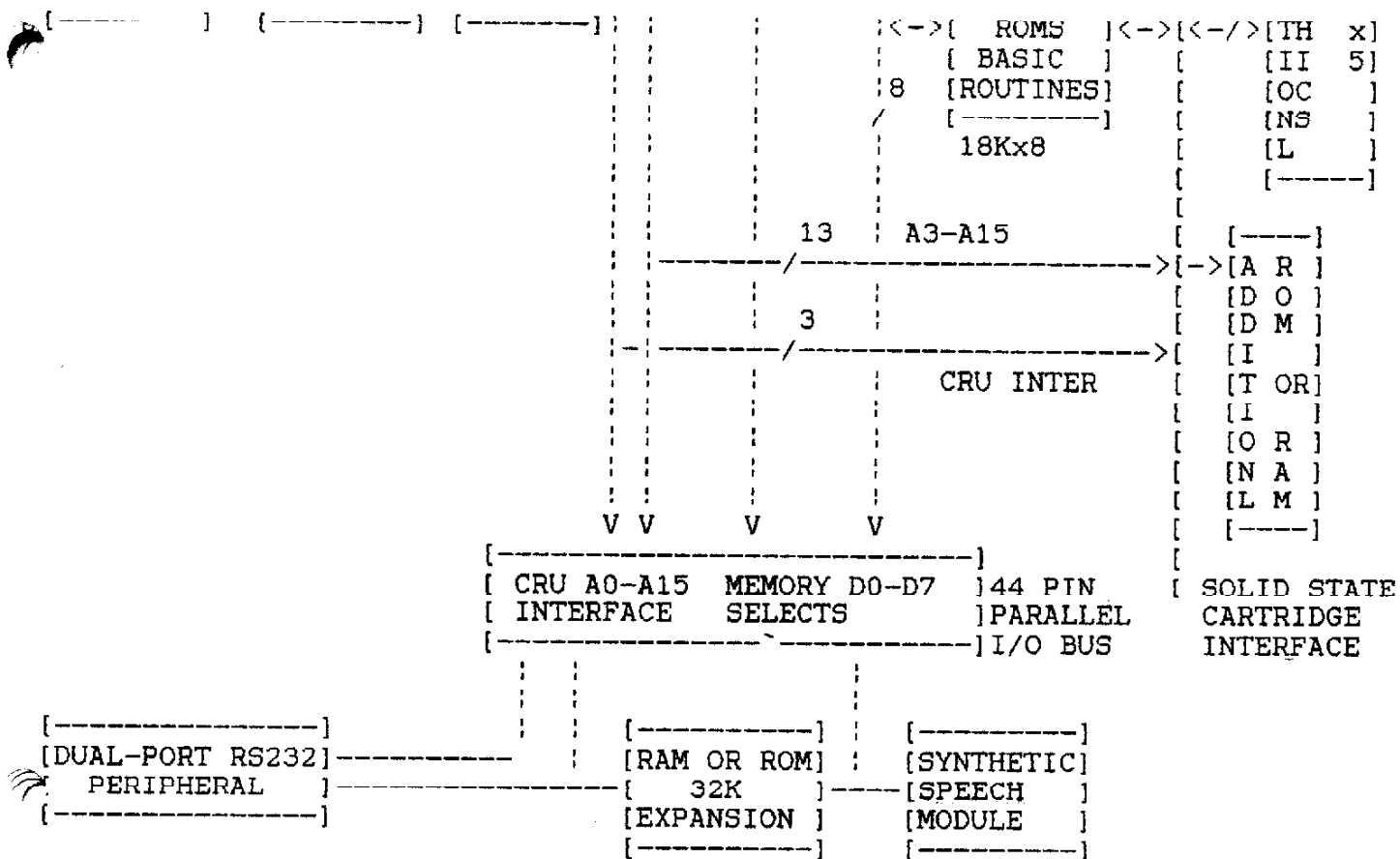
Bytes 18-19 contain the number of records allocated on Level 3. For variable length records, this entry is replaced with the number of Level 2 records actually used. (NOTE: The bytes in this entry are in reverse order.)

Bytes 20-27 have been reserved for future expansion. They will be fixed to 0 in this implementation of disk peripheral software.

Bytes 28-255 contain three byte blocks indicating the clusters that have been allocated for the file. The first 12 bits in each entry indicate the address of the first AU in the cluster. The second 12 bits indicate the highest logical record offset in the cluster of contiguous records. This indication has been chosen, rather than the number of data-records in the chain, since it reduces the amount of computation required for relative record file access.

FIGURE A. TI-99/4A SYSTEM BLOCK DIAGRAM





(THE SYSTEM BLOCK DIAGRAM CAN BE USED FOR REFERENCE WITH THE FOLLOWING SPECS)

SECTION A TI-99/4A Console

A.1 GENERAL DESCRIPTION

The purpose of this section is to provide necessary information concerning the I/O bus (input/output signal connection) of the TI-99/4A console for persons interested in designing peripherals for the computer.

It is assumed that readers of this manual have a working knowledge of electronics and computers, especially in regard to the TMS 9900 microprocessor and its Communications Register Unit (CRU) I/O technique. Sources for this information are the TMS9900 Microprocessor Data Manual (MP001, Revision A) or the TMS9901 Programmable Systems Interface (MP003; July, 1978). These books may be obtained from TI semiconductor distributors or the TI Learning Center.

The I/O bus provides an interface between the console and its peripherals. This bus utilizes both memory-mapped I/O and CRU I/O. The memory bus is frequently used for instruction fetch from read-only memory (ROM) in external peripherals and for data transfer to and from memory-mapped portions of these devices. The CRU bus is used for peripheral enable/disable, device control, and data transfer to/from CRU-mapped portions of these peripherals.

The TMS 9900 microprocessor accesses each peripheral to obtain instructions from the device service routine (DSR) ROM. Since each peripheral contains its own DSR, the TI-99/4A does not have to be designed to anticipate future peripheral requirements. The dual I/O bus capability, interrupt handling, and external DSRs provide flexibility at low cost.

A.2 I/O PIN DESCRIPTION

SIGNATURE	PIN	I/O	DESCRIPTION
A0 (MSB)	31	Out	ADDRESS BUS A0 through A15 comprise the address bus. This bus provides the 16-bit memory address vector to the external memory system when MEMEN is active. Address bit 15 is also used for CRU DATA OUT on CRU output instructions.
A1	30	Out	
A2	20	Out	
A3	10	Out	
A4	7	Out	
A5	5	Out	
A6	29	Out	
A7	17	Out	
A8	14	Out	
A9	18	Out	
A10	6	Out	
A11	8	Out	
A12	11	Out	
A13	15	Out	
A14	16	Out	
A15/CRUOUT	19	Out	
D0 (MSB)	37	I/O	DATA BUS D0 through D7 comprise the bidirectional data bus. This bus transfers memory data to (when writing) and from (when reading) the external memory system when MEMEN is active.
D1	40	I/O	
D2	39	I/O	
D3	42	I/O	
D4	35	I/O	
D5	38	I/O	
D6	36	I/O	
D7	34	I/O	
			BUS CONTROL

MEMEN	32	OUT	MEMory ENable. MEMEN indicates a memory access
DBIN	9	OUT	Data Bus IN. When active (high) the data buffers and 9900 are in the input mode.
WE	26	OUT	Write enable. WE indicates a memory write.
MBE	28	OUT	Memory Block Enable. MBE indicates a memory access in memory block 4000-5FFF.
CRUCLK	22	OUT	CRU CLock. Input data line to the Home Computer.

CRUIN 33 IN CRU data IN. Input data line to the Home Computer.

MEMORY CONTROL

READY 12 IN Ready (when MEMEN is active) indicates external memory is ready for a memory access.

IAQ 41 OUT Instruction AcQuisition indicates the CPU is acquiring an instruction during the current memory cycle.

TIMING AND CONTROL

LOAD 13 IN When active, LOAD causes the CPU to execute a nonmaskable interrupt; memory addresses FFFC and FFFE contain the new workspace and PC vectors, respectively.

RESET 3 OUT When active, RESET causes the Home Computer and the peripherals to be reset. Reset will be held active for a minimum of five clock cycles.

EXT INT 4 IN EXTERNAL INTERRUPT. When active, EXT INT causes the CPU to execute an interrupt.

3 * 24 OUT CPU Clock. Phase 3 of the CPU clock.

POWER

GND 21,23 25,27 Ground reference.

SPEECH MODULE SIGNALS

SBE 2 OUT Speech Block Enable. SBE indicates a memory access in the speech memory.

AUDIO IN 44 IN Input for the audio from the speech module
+5 1 Supply voltage (+5v Nom) for speech module (50ma Max)**

-5 43 Supply voltage (-5v Nom) for speech module (50 ma Max)**

** Note: Pins 1 and 43 are not intended for use by consumer. Overload may cause permanent damage to console.

A.3 MEMORY ALLOCATION

The memory address space is broken into eight blocks of 8k bytes of memory. The third block (addresses 4000-5FFF) is predecoded and made available at the I/O port for the peripherals. The second, sixth, seventh, and eighth blocks (addresses 2000-3FFF and A000-FFFF) are in the Memory Expansion peripheral. For the speech module (addresses 9000-97FF), a predecoded line is available at the I/O port.

SYSTEM MEMORY MAP

HEX ADDRESS

0-1FFF	Console ROM space
2000-3FFF	Memory Expansion Peripheral
4000-5FFF	Peripheral Expansion (predecoded to I/O connector)
6000-7FFF	Cartridge ROM/RAM (predecoded to GROM connector)
8000-9FFF	Microprocessor ROM, VDP, GROM, SOUND and SPEECH select.
A000-BFFF	Memory Expansion Peripheral
C000-DFFF	Memory Expansion Peripheral
E000-FFFF	Memory Expansion Peripheral

MEMORY-MAPPED DEVICES

ADDRESSES	A0	A1	A2	A3	A4	A5	A14	USE
8000	1	0	0	0	0	0	0	Internal RAM (8300-83FF)
8400	1	0	0	0	0	1	0	Sound
8800	1	0	0	0	1	0	0	VDP Read Data
8802	1	0	0	0	1	0	1	VDP Read Status
8C00	1	0	0	0	1	1	0	VDP Write Data
8C02	1	0	0	0	1	1	1	VDP Write Address
9000	1	0	0	1	0	0	0	Speech Read
9400	1	0	0	1	0	1	0	Speech Write
9800	1	0	0	1	1	0	0	GROM Read Data
9802	1	0	0	1	1	0	1	GROM Read Address
9C00	1	0	0	1	1	1	0	GROM Write Data
9C02	1	0	0	1	1	1	1	GROM Write Address

NOTE: Memory-mapped devices at addresses >8000 through >9FFF are only partially decoded. Thus, the devices will respond not only at the base addresses listed above, but also at other addresses within the 1K block.

A.4 CRU ALLOCATION

Of the available 4K of CRU bits, the first 1K (addresses 0000-07FE) are used internally in the console. The second 1K (addresses 1000-1FFE) are reserved for future use. The last 1.9K (addresses 1000-1FFE) are reserved for the

peripherals to be plugged in the I/O port. A block of 128 CRU bits is assigned to each peripheral as listed below.

CRU ASSIGNMENTS

CRU ADDRESSES	A3	A4	A5	A6	A7	USE
0000-0FFE	0	X	X	X	X	Internal Use
1000-10FE	1	0	0	0	0	Unassigned
1100-11FE	1	0	0	0	1	Disk Controller Card
1200-12FE	1	0	0	1	0	Modems
1300-13FE	1	0	0	1	1	RS 232 (primary)
1400-14FE	1	0	1	0	0	Unassigned
1500-15FE	1	0	1	0	1	RS 232 (secondary)
1600-16FE	1	0	1	1	0	Unassigned
1700-17FE	1	0	1	1	1	HEX-BUS(tm)
1800-18FE	1	1	0	0	0	Thermal Printer
1900-19FE	1	1	0	0	0	EPROM Programmer
1A00-1AFE	1	1	0	1	0	Unassigned
1B00-1BFE	1	1	0	1	1	Unassigned
1C00-1CFE	1	1	1	0	0	Video Controller Card
1D00-1DFE	1	1	1	0	1	IEEE 488 Controller Card
1E00-1EFE	1	1	1	1	0	Unassigned
1F00-1FFE	1	1	1	1	1	P-Code Card

A.5 INTERRUPT HANDLING

The interrupt available on the I/O port is one of the maskable interrupts of the TMS-9901 Programmable Systems Interface.

9900 INTERRUPTS

INTERRUPT LEVEL	VECTOR LOC. (MEMORY ADDR. IN HEX)	CPU PIN	DEVICE ASSIGNMENT
(Highest Priority)	0000-WSP	RESET	RESET
0	0002-PC		
	FFFC-WSP	LOAD	LOAD
	FFFE-PC		
1	0004-WSP	----	EXT DEV (9901)
	0006-PC		

Interrupting is done only on Level 1. The additional interrupts available are implemented on 9901. Interrupt Level 1 is decoded by software to be either
 (1) VDP vertical sync., (2) 9901 internal timer, or (3) I/O bus generated.

ADDRESS	CRU BIT	9901	PIN	FUNCTION
0000	0	Control		Control
0002	1	INT1	17	External Interrupt
0004	2	INT2	18	Video Display Processor Vertical Sync Interrupt
0006	3	INT3	9	9901 Internal Timer Interrupt, keyboard "=" line, JOYST "FIRE"
0008	4	INT4	8	Keyboard "Space" line, JOYST "Left"
000A	5	INT5	7	Keyboard "ENTER" line, JOYST "Right"
000C	6	INT6	6	Keyboard "0" line, JOYST "Left"
000E	7	INT7 (P15)	34	Keyboard "FCTN" line, JOYST "Up"
0010	8	INT8 (P14)	33	Keyboard "SHIFT" line
0012	9	INT9 (P13)	32	Keyboard "CTRL" line
0014	10	INT10 (P12)	31	Keyboard "Z" line
0016	11	INT11 (P11)	30	Not Used as interrupt
0018	12	INT12 (PIO)	29	Reserved, High Level
001A-1E	13-15	INT13-INT15	28,27 23	Not Used as interrupt

9901 I/O MAPPING

0020	16	P0	38	Reserved
0022	17	P1	37	Reserved
0024	18	P2	26	Bit 2 of Keyboard Select
0026	19	P3	22	Bit 1 of Keyboard Select
0028	20	P4	21	Bit 0 (MSB) of Keyboard Select
002A	21	P5	20	Keyboard (ALPHA LOCK)
002C	22	P6	19	Cassette Control 1 (motor control)
002E	23	P7 (INT15)	23	Cassette Control 2 (motor control)
0030	24	P8 (INT14)	27	Audio Gate
0032	25	P9 (INT13)	28	Mag Tape Out
0034	26	P10 (INT12)	29	Reserved
0036	27	P11 (INT11)	30	Mag Tape Input
0038-003E	28-31	P12-P15	31-34	Not Used IN I/O Mapping