TinyGiant

68000

SINGLE BOARD COMPUTER

Hawthorne Technology

HAWTHORNE TECHNOLOGY

TinyGiant

68000 SINGLE BOARD COMPUTER

Owner's Manual

Hawthorne Technology
8836 S.E. Stark
Portland, OR 97216
(503) 254-2005

TABLE OF CONTENTS

WARRANTY DISCLAIMER

If a defect in workmanship or materials is found within 90 days from the date of purchase Hawthorne Technology will repair or replace the board free of charge if delivered to them. Other than such repair or replacement the are no other warranties expressed or implied.

NOTE: This does not cover -
a. Damage to a board that has been improperly connected to other equipment or damage to other equipment that has not been properly connected to this board.
b. Customer modifications.
c. Repairs required by circumstances beyond the control of Hawthorne Technology including but not limited to fire or other casualty, accident, neglect, abuse or abnormal use.

THERE ARE NO OTHER EXPRESSED OR IMPLIED WARRANTIES. THERE IS NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. HAWTHORNE TECHNOLOGY WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY NATURE RESULTING FROM OR IN CONNECTION WITH THIS PRODUCT.

For service contact Hawthorne Technology, 8836 SE Stark, Portland, OR 97216. Telephone (503) 254-2005. You must first receive a Return Material Authorization Number (RMA) before you send back the board for service. When returning the unit, you must give the serial number, model, place of purchase, date purchased, RMA number and a short description of the problem experienced. Be sure to include your complete name and address.

INTRODUCTION

THE TinyGiant 68000 BOARD

The TinyGiant from Hawthorne Technology is a 68000 single board computer. With the K-OS ONE operating system it provides all of the vital parts of a full computer system. You add a terminal, storage (both physical and data), and a power supply. The power connector to the board is the same style and pin-out as the standard 5 1/4" disk drive power connectors. This makes it possible to use a PC power supply, which is wired for up to four drives, for your board and disk drives.

This board was designed to be low cost while providing the functions necessary for a useful system. It can be used for data collection and manipulation, and is small enough to mount inside other equipment, becoming part of a larger system. The board is the same size and mounting hole pattern as a 5 1/4" disk drive.

The board has a full expansion bus so new projects or additional features can be added to it easily. There is a floppy disk controller and a parallel Centronics port for a printer on the board.

The RAM can be increased to 512k by adding chips to the board. The 68000 allows you to have up to 16 Meg. which you can add on the expansion bus.

The TinyGiant is a fun board to work with. It is open for development of hardware and software and is available at a price that makes it easy to get started.

The board comes with a debug Monitor program in PROM that is useable as is or will boot the operating system from disk as soon as you have the pieces put together.

DESCRIPTION OF THE TinyGiant BOARD

SIZE

The TinyGiant Board is a standard 5.75" by 8.0" size. It can be mounted directly to the side of a 5-1/4" disk drive.

POWER

The power connector used on the TinyGiant board is the same as the standard floppy disk drive power connector. The voltage levels, (+5 and +12) are also the same as those required for a disk drive. The -12 volts for RS-232 is generated on board. A PC power supply with connectors wired for multiple disk drives is an easy way to supply power to your board and drives.

I/O PORTS

There are two RS-232 serial ports on the TinyGiant board. One of these ports is used for the console device for the system. A full terminal and not just a monitor and keyboard is needed. The other port can be used for a modem or other serial device.

A centronics style parallel port is provided for a printer. It is output only.

FLOPPY DISK CONTROLLER

There is a floppy disk controller provided on the TinyGiant board. The connector is a standard 34 pin header.

EXPANSION BUS

A complete expansion bus is provided through two connectors. One of the connectors provides access to all of the 68000 signals and the other one has other signals from the board that could be needed. This is an un-buffered bus. Buffers will need to be included on the first expansion board that is added to the system.

MONITOR/DEBUG PROM

The HAWTHORNE TECHNOLOGY Debug/Monitor PROM supplied with your TinyGiant board provides a means for your board to interface with your computer. It gives you access to many of the functions of the processor and surrounding circuitry. It also contains basic Debug features, such as Examine Memory and Change Memory.

OPERATION

On Power-Up or after a hard reset, the system will display a header and send a `:' prompt character to the console device. At this point, enter the command for the function that is to be performed.

CONSOLE COMMANDS

The monitor communicates with the operator through an interactive console device such as a computer or dumb terminal. The monitor will send prompts to the screen. A single character command will initiate the desired action.

The available commands include:

- B   Load binary file from disk.
- C   Compare memory.
- D   Display memory in Hex and ASCII.
- E   Examine and/or change memory (2 bytes).
- F   Fill a block of memory with a value.
- G   Go to an address and execute program.
- K   Peek.
- L   Load program from serial port.
- M   Move memory.
- P   Poke.
- R   Read binary file from disk.
- S   System boot.
- T   Test printer.
- W   Write binary file to disk.
- X   Examine long and/or change memory (4 bytes).

* * * * * * * * * * * *

B Load binary file from disk

The LOAD BINARY command loads a binary file from disk into memory. The format is:

:B filename destaddr <RETURN>

Filename is the name of the file on the disk to be loaded. Destaddr is the address in memory where you want to put the file.

* * * * * * * * * * * *

C Compare memory

The COMPARE MEMORY command compares two sections of memory. The format is:

:C start1addr stop1addr start2addr <RETURN>

The memory between 'start1addr' and 'stop1addr' is compared to the memory beginning at 'start2addr'.

* * * * * * * * * * * *

D Display memory in Hex and ASCII

The DISPLAY command causes the contents of a memory location to be displayed in Hex and ASCII. The format is:

:D startaddr stopaddr <RETURN>

The spaces before the first address and after the second address are optional. There must be one or more spaces between the addresses. An address may be of any length, but only the right four digits are used.

* * * * * * * * * * * *

E Examine and/or change memory (2 bytes)

The EXAMINE/CHANGE command puts the system into an Examine Memory or Change Memory mode. The command format is:

:E addr <RETURN>

A space is required between the \`E' and the address. After you press <RETURN>, the screen will display:

Addr value

Where address is a 4-byte hex number, and value is a 2-byte hex number. To examine, press the <SPACE> key or <RETURN> to call up the next address and value.

To change the value at that address, type in a 2-byte hex number.
To back up while typing, enter a '-'.
To re-enter a value, enter '^X' (control X).
To end the EXAMINE/CHANGE sequence, enter a \`.'.

* * * * * * * * * * * *

F Fill a block of memory with a value

FILL memory block with a given value. The format of the command is:

:F startaddr stopaddr value <RETURN>

The spaces before 'startaddr'and after 'value' are optional. There must be at least one space between numeric fields. Only the right two digits of the value and the right four digits of size and address are used. Numeric fields can be of any length. Fields are filled from left to right, while missing fields are set to zero.

* * * * * * * * * * * *

G Go to an address and execute program

The GO/EXECUTE command causes execution of the code at the specified address, and works like a run command. The command is given in the format:

```
:G1234<RETURN>
```

where '1234' is the address of the code to be executed. Control is turned over to the program in RAM and the program is executed. There may be spaces between the 'G' and the address.

* * * * * * * * * * * *

K Peek

The PEEK command examines one byte. The format is:

```
:K addr
```

* * * * * * * * * * * *

L Load program from serial port

The LOAD command sets up the system to receive an Intel Hex format object file from the auxiliary device through the RS-232 port. The Monitor will read the hex information to determine where in RAM memory to load the data. On receiving an end record of zeros, a 'LOAD COMPLETE' message will be sent to the console device. For example:

```
:L <RETURN>
  :0303000057686FCC
  :00
LOAD COMPLETE
:
```


* * * * * * * * * * * *

M Move memory

MOVE a block of memory. The format of the command is:

```
:M startaddr stopaddr destaddr <RETURN>
```

A space is required before each address. The block of memory between the start address and the stop address is moved to where the destination address becomes the start address.

* * * * * * * * * * * *

P Poke

POKE stores one byte to the specified address. The format is:

```
:P addr value
```

* * * * * * * * * * * *

R Read binary file from disk

READ one record from disk. The command format is:

```
:R record destaddr <RETURN>
```

* * * * * * * * * * * *

S System boot

SYSTEM boot. The format of the command is:

```
:S <RETURN>
```

This command causes the operating system to be loaded from a disk in drive A and control is passed to the command processor.

* * * * * * * * * * * *

T Test printer

The TEST command sends 95 printable characters out to the printer. (All ASCII chars from $20 - $7E)

* * * * * * * * * * * *

W Write a binary file to disk

WRITE a record to the disk. The format is:

:W record sourceaddr

Writes 512 bytes beginning at the 'sourceaddr' to the disk 'record' number specified.

* * * * * * * * * * * *

X Examine long and/or change memory (4 bytes)

The EXAMINE/CHANGE LONG command puts the system into an Examine Memory or Change Memory mode. The command format is:

:X addr <RETURN>

A space is required between the `X' and the address. After you press <RETURN>, the screen will display:

Addr value

Where address is a 4-byte hex number, and value is a 4-byte hex number. Pressing the <SPACE> key will call up the next address and value.

To change the value at that address, type in a 4-byte hex number. To end the EXAMINE/CHANGE sequence, enter a `.'.

When you have made a change to a memory location, re-initiating the CHANGE/EXAMINE sequence allows you to check that the changes have been made correctly.

BRINGING UP THE TinyGiant BOARD

The procedure for getting your TinyGiant 68000 system running is quite easy.

Some of the components on the TinyGiant board are sensitive to static discharge. Ground yourself and the board while you are working with it to prevent damaging these components.

CAUTION: When mounting the TinyGiant board, take care to insulate the board from mounting screws, disk drive or chassis at any point where +5 volts would come into contact with any conductive material. Shorting the +5 volt line to ground can damage the TinyGiant board and possibly your power supply and disk drive.

POWER SUPPLY

First you must provide power to the board. It requires +5 volts and +12 volts. The size of the supply needed will be determined by the type and number of devices you attach. For the board alone, you should have about 2 amps on the 5v line and 75 ma on the 12v line. Be careful to check that you have the correct voltages on the proper pins of the connector.

CAUTION: Applying improper voltages to the board can distroy components. Double check connections prior to applying power.

CONSOLE TERMINAL

A console device is needed for the TinyGiant system. This should be an Asynchronus, Serial, ASCII termial. Connect the terminal via the RS-232 serial port at connector P4. The pin assignments are listed later in this manual.

In most cases a three wire cable is sufficient for the console terminal. Connect Transmit, Receive, and Ground across the cable. Tie the Clear-To-Send to the Request-To-Send at the connector, and you might need to tie the signal ground (pin 7) to the chassis ground (pin 1).

The TinyGiant board comes up with the console port set at 9600 baud.

APPLYING POWER

First turn on the terminal. Then turn on the power to the TinyGiant board. The board will first look for a disk to boot the K-OS ONE operating system. When the boot disk is not found, it will go into the MONITOR program. You should receive a sign-on message:

HAWTHORNE TECHNOLOGY 68000 MONITOR C 1986

* If you don't get any message on the screen, check your voltage levels and all connections.

** If you get characters on the screen, but the message is not readable, check to see that your terminal is set for the proper baud rate.

You are now in the MONITOR program and can use the commands described in that section of this manual, with the exception of the commands that require access to a disk drive.

DISK DRIVE

Hooking a standard 5 1/4" floppy disk drive to the system requires two cables. One for power and one for signals. Most drives also have a ground lug. This is to connect this to your chassis ground.

*** Turn off the power prior to connecting the disk drive, or changing any connections.

The signal cable for the floppy disk drive is a standard 34 pin ribbon cable. It should have a female header connector on the board end, and an edge connector on the drive end. Match the pin one indicators on the cable to the indicators on the board and on the disk drive.

**** If you have access to an MS-DOS compatible machine, you should back up the system disk that came with your TinyGiant board before using it.

With the power turned off, hook up your disk drive. Turn on the power, then insert the system disk. The TinyGiant will be looking for the system disk, and when you insert it, it will load in the K-OS ONE operating system. When the operating system is booted, the sign on message will be displayed:

K-OS ONE COMMAND PROCESSOR
COPYRIGHT 1986 HAWTHORNE TECHNOLOGY

The K-OS ONE operating system is now loaded onto your TinyGiant board. Refer to the operating system manual for it's operation.

***** If the TinyGiant stops looking for the boot disk and enters the MONITOR, you can load the system through the MONITOR program. Use the 'S' command.

Syntax: S <return>

THE 68000 CHIP

The Motorola M68000 16/32 Microprocessor is the CPU for the TinyGiant Board. It runs at 8 MHz. The 68000 has 16 bit address and data buses, and 32 bit registers. Other members of the 68000 family are software compatible but the size and shape of the packages prevent them from being used on the TinyGiant board.

THE 68681 CHIP

The MC68681 chip is a Motorola Dual Asynchronous Receiver / Transmitter (DUART). The main function of the MC68681 on the TinyGiant board is to provide two RS-232 serial ports. The chip itself has 8 major sections: internal control logic, timing logic, interrupt control logic, a bi-directional 8-bit data bus buffer, two independent communications channels (A and B), a 6 bit input port, and an 7 bit output port.

In operating the TinyGiant board, one of the communications channels will be used regularly for the console device. The other channel is available for connecting another serial device the your system. The two channels can be set independently for baud rates between 50 and 38.4K.

The following is a description of how each part of the 68681 is used and how it has been configured.

The serial channels are given the names CON (console) and AUX (auxiliary) in both the BIOS and operating system. Each channel has a receiver and a transmitter which must be programmed. The console is programmed to run at 9600 baud and is the primary I/O device for the system. The auxiliary channel is programmed to run at 1200 baud and may be connected to a modem or other serial device. The console and auxiliary channels are wired to connectors P4 and P5 respectively. Both channels use only three of the RS-232C lines: RXD, TXD and GND.

The 68681's counter/timer is used to update the real-time clock. It is programmed in timer mode, which causes it to load a value, count down to zero, and then interrupt the processor. The timer will interrupt the CPU every 10 ms except when a disk read or write is in progress. Disk activity will cause the real-time clock to run a little bit slow.

The 68681 input port is not used by the BIOS.

The output port is used in the floppy disk interface. Four output lines are used to select the active disk drive. One line is used to select the side of the disk to read or write.

Once the 68681 is configured, the interrupts from the two receivers and the timer are unmasked by writing to the DUART's Interrupt Mask Register. This generates an interrupt whenever a character is received from the console or the auxiliary device, or when the timer reaches zero. If a transmitter interrupt is unmasked an interrupt will be generated whenever the transmitter is ready to send a character. For that reason, interrupts for the transmitters are unmasked only when there are characters waiting to be sent. The interrupt service routine for the DUART must determine which of the five possible interrupt sources needs attention and take appropriate action.

Characters received from the serial devices are stored in input queues. The BIOS routines GETCON and GETAUX remove a character from the appropriate input queue. The BIOS routines PUTCON and PUTAUX insert a character in an output queue and unmask the transmitter interrupt for the appropriate channel. When that transmitter is ready, an interrupt is generated and the interrupt service routine removes a character from the queue and writes it to the transmitter. When the output queue has been emptied, that transmitter's interrupt is again masked.

THE 1770 CHIP

The Western Digital WD1770 is a Floppy Disk Controller/Formatter. It provides the TinyGiant with the ability to control up to four floppy disk drives. The 1770 was designed for controlling 5 1/4" drives. It can handle either single or double density operation and has a programmable Motor On signal. This chip's instruction set is nearly identical to that of the WD179X and WD279X controllers.

The BIOS routines DSKREAD1 and DSKWRITE1 are part of the device driver for disk drive 1. A device driver is merely a collection of routines which provide some uniform method of accessing a device. In this case, the 'uniform method' is the way that K-OS ONE expects all disk read and write routines to behave. The driver accepts the command from the operating system and translates that into the proper instructions for the disk controller chip.

The BIOS has drivers for four 5 1/4" DSDD floppy-disk drives which the operating system labels A through D. The operating system itself contains a driver for drive E, the RAMdisk. The 256K RAMdisk for the TinyGiant resides at $40000 thru $7FFFF. If you expand memory to 512K, you may want to re-compile the operating system after un-commenting the invokation of INITD0, which initializes the fat and directory for the RAMdisk and allows it to be used.

Only four of the 1770's fifteen instructions are used by the driver: restore, seek, read sector, and write sector. Restore sets the head on track 0. Seek moves the head to a specific track. Read and write access a specific sector in the current track. With these instructions it is possible to access any sector on one side of one disk. In order to access the other side of the disk or other disks it is necessary to change the output port of the 68681 DUART.

There are four drive select lines on the disk ribbon-cable. When one of these lines is low, a disk drive is selected. There is also a side select line which determines which side of the disk drive is used. These five signals are provided by the output port of the DUART.

1770 COMMAND SUMMARY:

TYPE	COMMAND	7	6	5	4	3	2	1	0
I	RESTORE	0	0	0	0	h	V	r1	r0
I	SEEK	0	0	0	1	h	V	r1	r0
I	STEP	0	0	1	u	h	V	r1	r0
I	STEP-IN	0	1	0	u	h	V	r1	r0
I	STEP-OUT	0	1	1	u	h	V	r1	r0
II	READ SECTOR	1	0	0	m	h	E	0	0
II	WRITE SECTOR	1	0	1	m	h	E	P	a0
III	READ ADDRESS	1	1	0	0	h	E	0	0
III	READ TRACK	1	1	1	0	h	E	0	0
III	WRITE TRACK	1	1	1	1	h	E	P	0
IV	FORCE INTERRUPT	1	1	0	1	I3	I2	I1	I0

(BITS)

FLAG SUMMARY, TYPE I COMMANDS:

h = MOTOR ON FLAG (Bit 3)

h = 0, ENABLE SPIN-UP SEQUENCE
h = 1, DISABLE SPIN-UP SEQUENCE

V = VARIFY FLAG (Bit 2)

V = 0, NO VARIFY
V = 1, VARIFY ON DESTINATION TRACK

r1, r0 = STEPPING RATE (Bits 1, 0)

r1	r0	WD1770
0	0	6 ms
0	1	12 ms
1	0	20 ms
1	1	30 ms

u = UPDATE FLAG (Bit 4)

u = 0, NO UPDATE
u = 1, UPDATE TRACK REGISTER

FLAG SUMMARY, TYPE II & III COMMANDS:

m = MULTIPLE SECTOR FLAG (Bit 4)

m = 0, SINGLE SECTOR
m = 1, MULTIPLE SECTOR

a0 = DATA ADDRESS MARK (Bit 0)

a0 = 0, WRITE NORMAL DATA MARK
a0 = 1, WRITE DELETED DATA MARK

E = 30MS SETTLING DELAY (Bit 2)

E = 0, NO DELAY
E = 1, ADD 30MS DELAY

P = WRITE PRECOMPENSATION (Bit 1)

P = 0, ENABLE WRITE PRECOMP
P = 1, DISABLE WRITE PRECOMP

FLAG SUMMARY, TYPE IV COMMANDS:

I3 - I0 INTERRUPT CONDITION (Bits 3-0)

```
   I0 = 1, DON'T CARE
   I1 = 1, DON'T CARE
   I2 = 1, DON'T CARE
   I3 = 1, INTERRUPT ON INDEX PULSE
I3-I0 = 0, TERMINATE WITHOUT INTERRUPT
```


STATUS REGISTER DESCRIPTION

BIT NAME	MEANING
S7 MOTOR ON	THIS BIT REFLECTS THE STATUS OF THE MOTOR ON OUTPUT.
S6 WRITE PROTECT	ON READ RECORD: NOT USED. ON READ TRACK: NOT USED. ON ANY WRITE: WRITE PROTECTED BIT IS RESET WHEN UPDATED.
S5 RECORD TYPE / SPIN-UP	TYPE 1 COMMANDS, SET INDICATES MOTOR SPIN-UP SEQUENCE COMPLETE (6 REVOLUTIONS). TYPE 2 & 3 CMDS, BIT INDICATES RECORD TYPE. 0=DATA MARK 1=DELETED DATA MARK
S4 RECORD NOT FOUND (RNF)	SET INDICATES THAT THE DESIRED TRACK, SECTOR, OR SIDE WERE NOT FOUND. BIT IS RESET WHEN UPDATED.
S3 CRC ERROR	IF S4 IS SET, AN ERROR IS FOUND IN ONE OR MORE ID FIELDS; OTHERWISE IT INDICATES ERROR IN DATA FIELD. BIT IS RESET WHEN UPDATED.
S2 LOST DATA / TRACK 00	SET INDICATES THE COMPUTER DID NOT RESPOND TO DRQ IN ONE BYTE TIME. BIT IS RESET TO 0 WHEN UPDATED. ON TYPE 1 COMMANDS, THIS BIT REFLECTS THE STATUS OF THE 'TRACK 00' PIN.
S1 DATA REQUEST / INDEX	THIS BIT IS A COPY OF THE DRQ OUTPUT. SET INDICATES THE 'DR' IS FULL ON A READ OPERATION OR THE 'DR' IS EMPTY ON A WRITE. THIS BIT IS RESET TO 0 WHEN UPDATED. ON TYPE 1 COMMANDS, THIS BIT INDICATES THE STATUS OF THE INDEX PIN.
S0 BUSY	WHEN SET, COMMAND IS UNDER EXECUTION. WHEN RESET, NO COMMAND IS UNDER EXECUTION.

THE MEMORIES

RAM

The RAM chips used on the TinyGiant board are 64K x 4, Dynamic RAM. This is the same RAM as is used in the Commodore Amiga (tm) computer. The chip number is 41464-12 (120 ns). The board is supplied with 128K of RAM and by adding 12 more chips, you can expand it to 512K.

The row of 4 chips next to the edge of the board are the ones that are installed on your TinyGiant when you receive it. It provides 128K of RAM. Sockets are installed on the board so you can expand this up to 512k. This can be done all at once or in 128k steps.

ROM

There are two EPROM sockets on the board. Each socket accepts a 2764, 27128 or 27256 EPROM. (Jumper J2 must be installed to use 27256s.) The address space of the EPROMs is split into even and odd bytes. The EPROM in socket U4 contains the even-addressed bytes and the EPROM in socket U3 contains the odd-addressed bytes. The TinyGiant comes with a boot program, Basic Input/Output System (BIOS), and a monitor in EPROMs.

The boot program performs two tasks. The first task is to initialize the interrupt vector table and BIOS variables and to configure the hardware. The second task is to attempt to load the operating system from the boot disk. If the operating system is successfully loaded, control is transferred to the system, otherwise control is transferred to the ROM monitor.

The BIOS is comprised of several subroutines which are called by the operating system to deal with the hardware. Among these are procedures which transfer data to and from the serial ports, the printer port, and the disk drives.

PRINTER PORT

The Centronics style printer port is an 8-bit parallel output port using two hand-shake lines, /STROBE and /ACK. The /STROBE output and /ACK input are stored in flip-flops.

Characters sent to the printer are stored in a queue. When the printer acknowledges a character by dropping the /ACK line, an interrupt is generated and the next character in the queue is output to the printer.

INTERRUPTS

All the interrupts on the board are autovectored. Three of the seven interrupt levels are connected to peripherals on the board, the rest are available on the expansion bus. A level 7 interrupt can also be generated by shorting jumper J3. Level 7 can not be masked, and is treated as a catastrophic event by the BIOS. A level 7 interrupt saves the state of the machine and goes to the ROM monitor. By connecting a de-bounced switch to this interrupt is possible to acquire useful information about a run-away program.

Level 1 autovector	External
Level 2 autovector	Printer acknowledge
Level 3 autovector	68681 DUART
Level 4 autovector	1770 Floppy disk contoller
Level 5 autovector	External
Level 6 autovector	External
Level 7 autovector	External

PAL -- ADDRESS DECODING

The TinyGiant has two Programmable Logic Arrays (PALs). One of their functions is to provide address decoding for the memory and peripherals on the board. The PALs output seven decoded address lines.

$000000-$07FFFF	/RAMCS	Read/write memory
$080000-$0FFFFF	/EXIO1	Expansion
$D80000-$D8FFFF	/PRT	Printer
$E00000-$E7FFFF	/FDC	1770 disk controller
$E80000-$EFFFFF	/SIO	68681 dual serial port
$F00000-$FFFFFF	/ROMCS	Read-only memory

The /RAMCS signal is the bank select for the ramdom-access memory.

The /EXIO1 signal is not used on the board. It is a bank select for the second 512K address space. It is available on the expansion bus and is intended to be used for memory or peripheral expansion.

The /PRT signal is used in several places in the printer port circuitry.

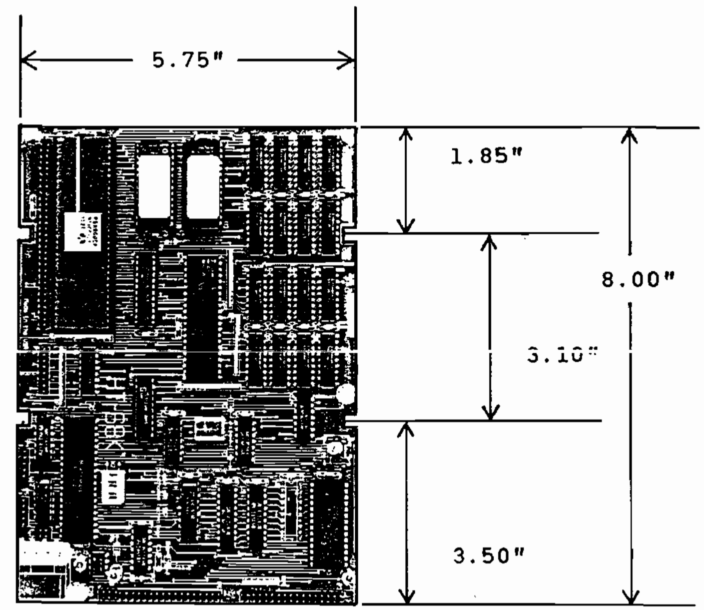
The /FDC signal is the chip select for the 1770 floppy disk controller.

The /SIO signal is the chip select for 68681 serial I/O peripheral.

The /ROMCS signal is the chip select for the EPROMs.

There is an important exception to the memory map above. When the system is running, RAM is in low memory and ROM is in high memory as indicated. When the system is reset however, the RAM is not addressable. Instead, the ROM is selected in both it's normal address space and in the RAM's address space. This allows the reset vector and stack pointer to be read from the EPROMs rather than from RAM. The boot program then accesses a byte in the 68681 DUART's address space, which causes the PALs to restore RAM to low memory.

PHYSICAL DIMENSIONS OF THE TinyGiant BOARD



The TinyGiant board is 8.0 inches long and 5.75 inches wide.

The board was designed to be mounted with four screws. The holes are spaced so the board could be mounted to the side of a standard 5 1/4" floppy disk drive.

CONNECTOR SIGNAL SPECIFICATIONS

The power connector on the TinyGiant is the same as used on a floppy disk drive. The power used is +5 and +12 volts. The +12 volts is converted to -12 by an on-board power converter. Most PC clone power supplies already have the connectors wired for several floppy disks so one of them can be used to power the TinyGiant board.

There are 7 connectors on the TinyGiant board. Two of these are for expansion, One for the console serial port, one for the auxillary serial port, one for the printer, one for the disk drives, and one for power.

The expansion bus is divided between two connectors, P1 and P2. The P1 connector brings out all the basic 68000 control signals. These come directly from the 68000 cpu chip and are not buffered. The remaining signals come from the P2 connector. Any signal should be buffered before it is used on an expansion board.

P1 CONNECTOR -- EXPANSION

1 = D4	64 = D5
2 = D3	63 = D6
3 = D2	62 = D7
4 = D1	61 = D8
5 = D0	60 = D9
6 = AS*	59 = D10
7 = UDS*	58 = D11
8 = LDS*	57 = D12
9 = R/W*	56 = D13
10 = DTACK*	55 = D14
11 = BG*	54 = D15
12 = BGACK*	53 = GND
13 = BR*	52 = A23
14 = VCC, 5 V	51 = A22
15 = CLK, 8 MHZ	50 = A21
16 = GND	49 = VCC, 5 V
17 = HALT*	48 = A20
18 = RESET*	47 = A19
19 = VMA*	46 = A18
20 = E	45 = A17
21 = VPA*	44 = A16
22 = BERR*	43 = A15
23 = IPL2*	42 = A14
24 = IPL1*	41 = A13
25 = IPL0*	40 = A12
26 = FC2	39 = A11
27 = FC1	38 = A10
28 = FC0	37 = A9
29 = A1	36 = A8
30 = A2	35 = A7
31 = A3	34 = A6
32 = A4	33 = A5

P2 CONNECTOR -- EXPANSION

1 = EXIO1*	2 = N/C*
3 = +12	4 = DREQ*
5 = -12	6 = IRQ1*
7 = VCC, 5 V	8 = IRQ5*
9 = VCC, 5 V	10 = IRQ6*
11 = GND	12 = IRQ7*
13 = GND	14 = GND
15 = GND	16 = 16 MHZ CLOCK

P3 CONNECTOR -- DISK

1 = GND	2 = NC
3 = GND	4 = NC
5 = GND	6 = DS0
7 = GND	8 = IP
9 = GND	10 = DS3
11 = GND	12 = DS2
13 = GND	14 = DS1
15 = GND	16 = MD
17 = GND	18 = DIR
19 = GND	20 = STEP
21 = GND	22 = WD
23 = GND	24 = WG
25 = GND	26 = TK0
27 = GND	28 = WP
29 = GND	30 = RD
31 = GND	32 = SIDE SEL
33 = GND	34 = NC

TV950

P4 CONNECTOR -- TERMINAL SERIAL CONNECTOR

1 GND → 1 = GND
2 TXD → 2 = RXD
3 RXD → 3 = TXD

1 = GND	6 = NC
2 = RXD	7 = GND
3 = TXD	8 = NC
4 = CTS	9 = NC
5 = RTS	10 = NC

P5 CONNECTOR -- MODEM SERIAL CONNECTOR

1 = GND	6 = NC
2 = RXD	7 = GND
3 = TXD	8 = DCD
4 = CTS	9 = NC
5 = RTS	10 = RTS

P6 CONNECTOR -- PRINTER

1 = STROBE
2 = D0
3 = D1
4 = D2
5 = D3
6 = D4
7 = D5
8 = D6
9 = D7
10 = ACK
11-40 = GND

POWER CONNECTOR

1 = +12 volts DC
2 = Ground (+12 return)
3 = Ground (+5 return)
4 = +5 volts DC

MEMORY MAP

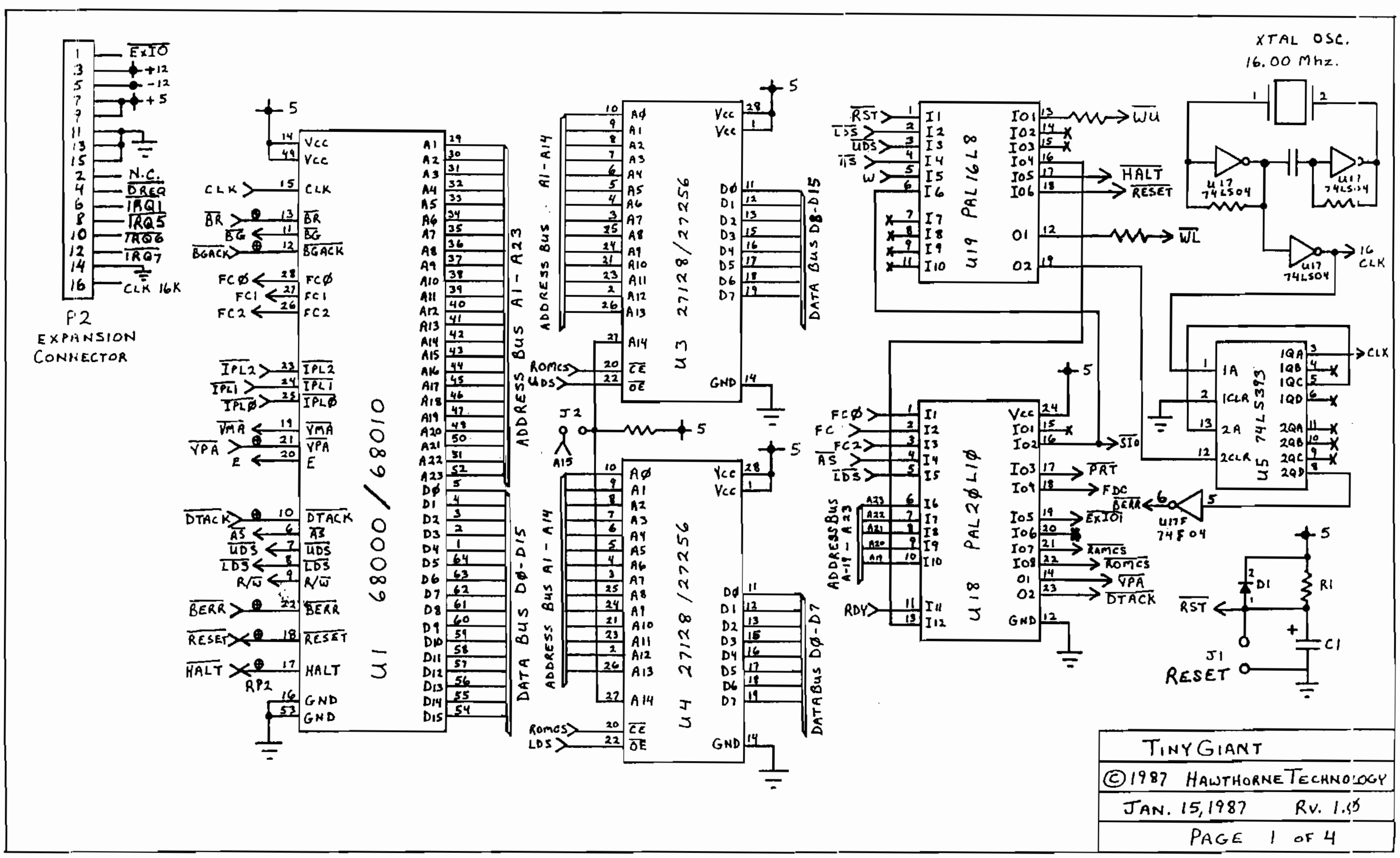
$000000 - $07FFFF		RAM ON BOARD
$080000 - $0FFFFF		EXPANSION 1
$D80000 - $D8FFFF		PRINTER
$E00000 - $E7FFFF		FLOPPY DISK
$E80000 - $EFFFFF		SERIAL I/O
$F00000 - $FFFFFF		PROM ON BOARD

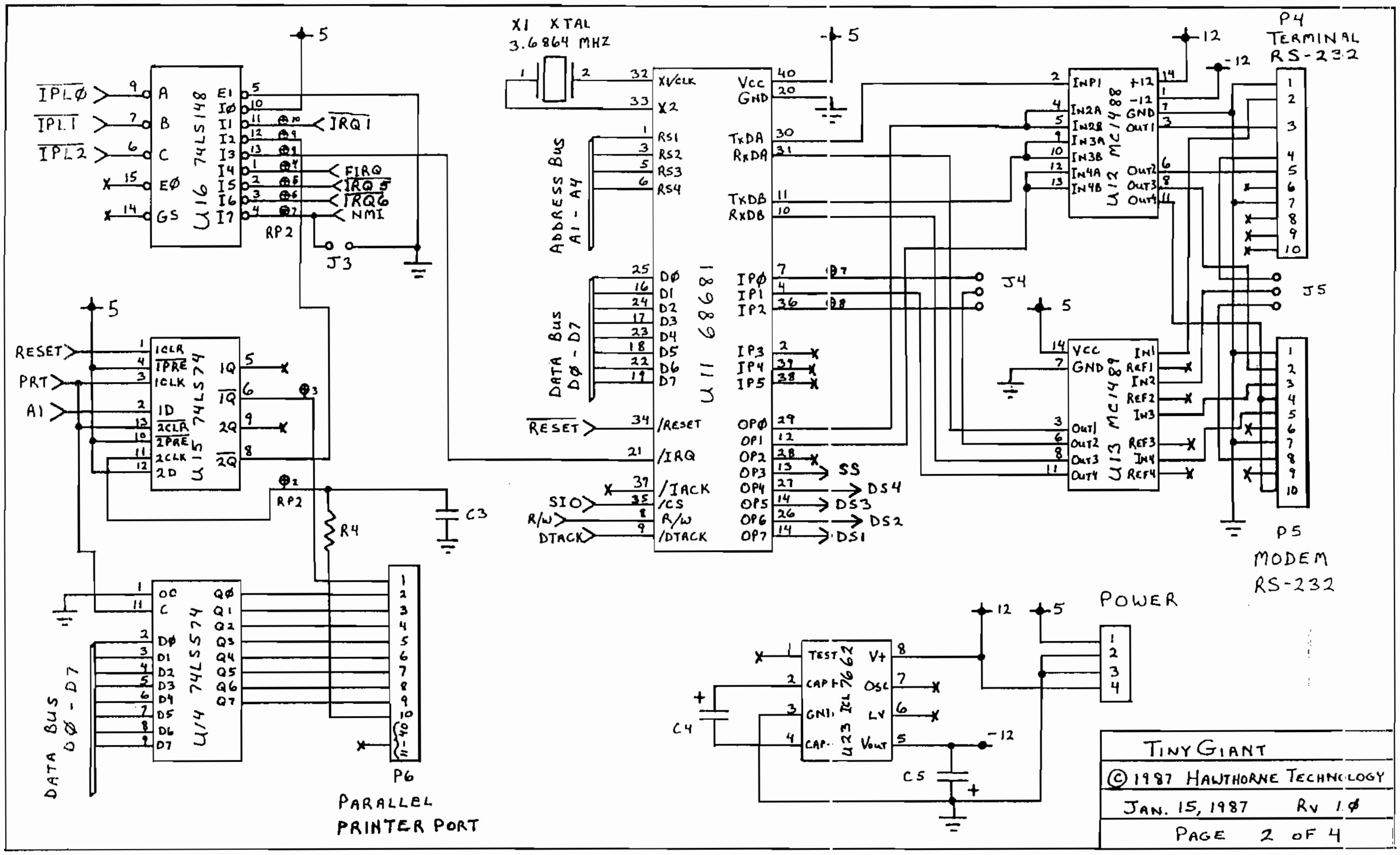
When power is applied the PROM is at location 0. As soon as the serial port is addressed the PROM is shifted into high memory. This is taken care of in the boot prom so all the user will see is RAM memory at location 0. The BIOS is in PROM so it is assumed to run at $F00000.

All of the inturrupts are auto vectored. Some of them are dedicated to devices on the board and some are available on the expansion connector.

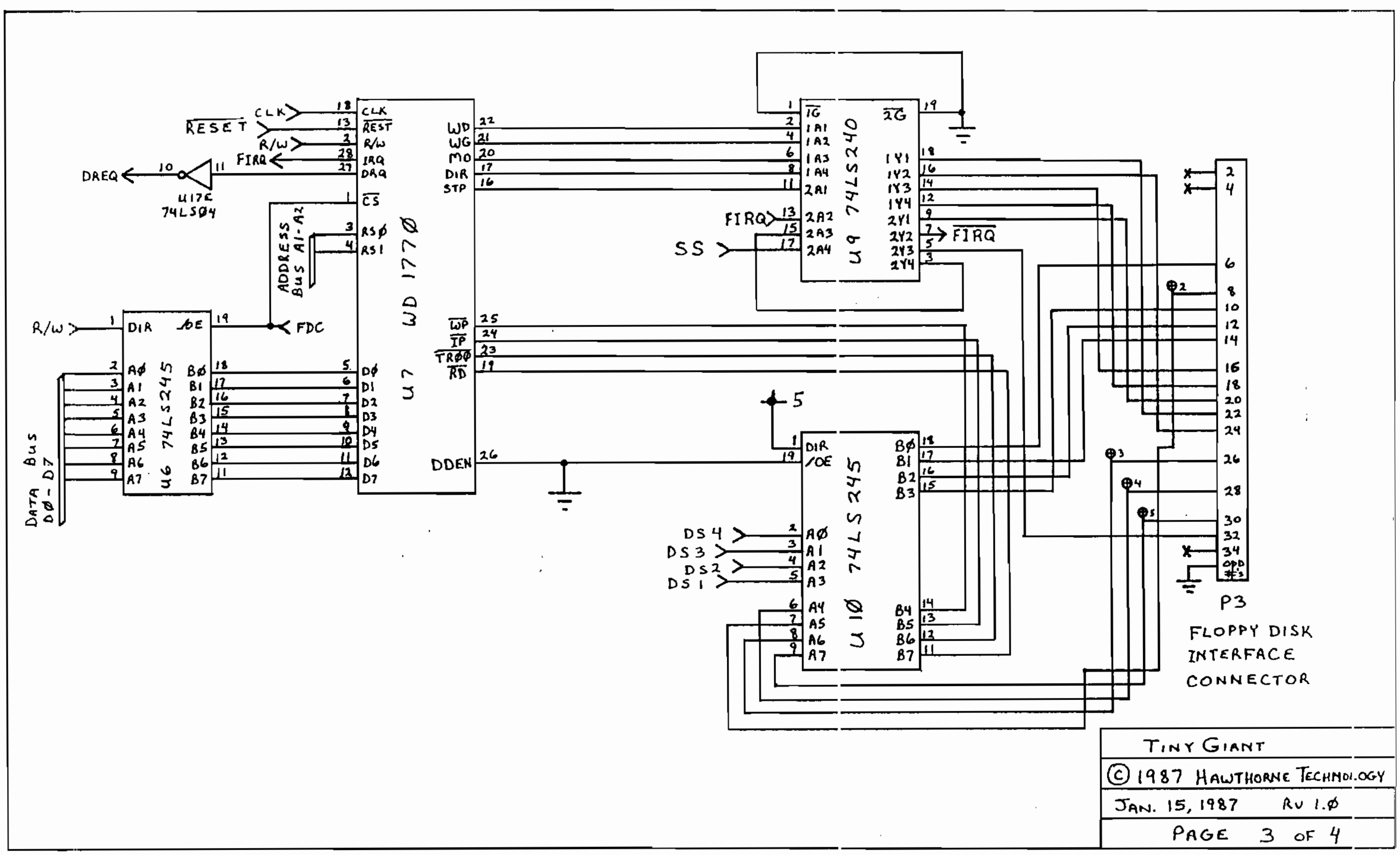
INTERRUPT LEVELS

0 = not an interrupt
1 = external
2 = Printer, internal
3 = Serial, internal
4 = Floppy disk, internal
5 = external
6 = external
7 = NMI, external

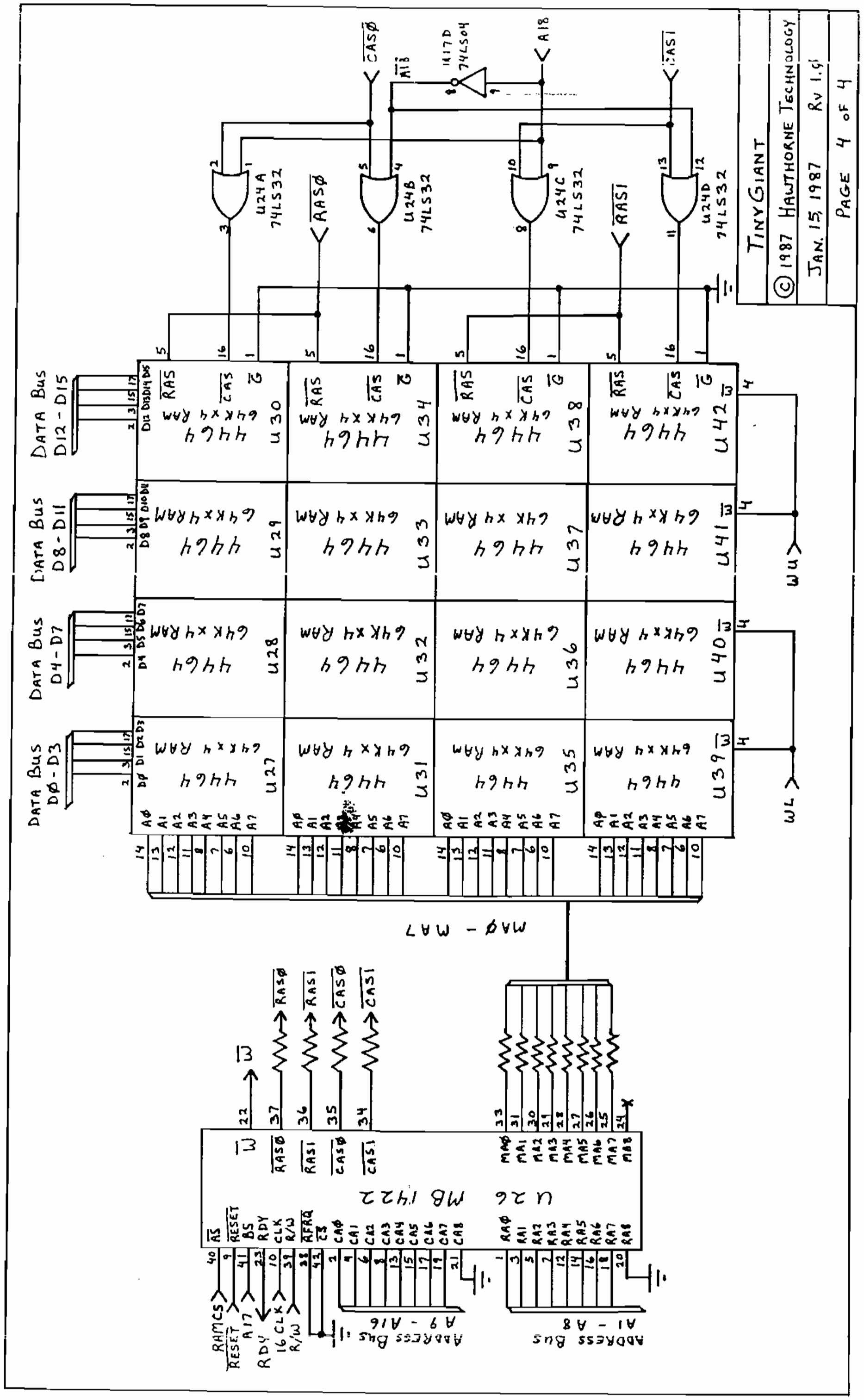

TINY GIANT
©1987 HAWTHORNE TECHNOLOGY
JAN. 15, 1987
PAGE 1 OF 4
P2
EXPANSION CONNECTOR
XTAL OSC.
16.00 Mhz.
U1 68000/68010
U3 27128/27256
U4 27128/27256
U19 PAL16L8
U18 PAL20L10
U5 74LS393
ADDRESS BUS A1 - A23
DATA BUS D0 - D15
ADDRESS BUS A1 - A14
DATA BUS D8 - D15
DATA BUS D0 - D7
RESET
J1
J2
D1
R1
C1



U16 74LS148
U15 74LS74
U14 74LS574
U11 68681
U12 MC1488
U13 MC1489
U23 ICL7662
X1 XTAL 3.6864 MHZ
ADDRESS BUS A1 - A4
DATA BUS DØ - D7
P4 TERMINAL RS-232
P5 MODEM RS-232
P6 PARALLEL PRINTER PORT
POWER
TINY GIANT
© 1987 HAWTHORNE TECHNOLOGY
JAN. 15, 1987 RV 1.Ø
PAGE 2 OF 4



U7 WD 1770
U9 74LS240
U6 74LS245
U10 74LS245
U17C 74LS04
ADDRESS BUS A1-A2
DATA BUS DØ - D7
P3 FLOPPY DISK INTERFACE CONNECTOR
TINY GIANT
© 1987 HAWTHORNE TECHNOLOGY
JAN. 15, 1987 RV 1.Ø
PAGE 3 OF 4



REFERENCE BOOK LIST

M68000 16/32-bit Microprocessor Programmer's Reference Manual, by Motorola, Prentice Hall

68000, 68010, 68020 Primer, by Kelly-Bootle and Fowler, The Waite Group, Howard W. Sams

The 68000: Principles and Programming, by Leo Scanlon, Howard W. Sams

68000 Assembly Language: Techniques for Building Programs, by Donald Krantz and James Stanley, Addison-Wesley Publishing Company

68000 Assembly Language Programming, by Lance Leventhal, Doug Hawkins, Gerry Kane and William D. Cramer, Osborne/McGraw-Hill

The 68000 Microprocessor: Architecture, Software and Interfacing Techniques, by W. Triebel, and A. Singh, Prentice Hall

68000 Microprocessor Handbook, by William Cramer, AVM Systems, Gerry Kane, Osborne/McGraw-Hill

68000 User's Manual, by J. Carr, Prentice Hall

MC68020 32 bit Microprocessor User's Manual, by Motorola, Prentice Hall

Basic Microprocessors and the 68000, by Ron Bishop & the Motorola Semiconductor Group, Hayden Books

Dr. Dobb's Toolbook of 68000 Programming, Edited, Prentice Hall

Mastering the 68000 Microprocessor, by Robinson, Tab Books

Programming the 68000, by Rosenzweig and Harrison, Hayden Books

Programming the 68000, by Steve Williams, Sybex Computer Books

Programming the 68000 Microprocessor, by T. King and B. Knight, Addison-Wesley

Self Guided Tour through the 68000, by M. Andrews, Prentice Hall