

VERSION 0.00

D I M E N S I O N 6 8 0 0 0
S Y S T E M R E F E R E N C E M A N U A L
M i c r o C r a f t C o r p o r a t i o n

6 8 0 - 0 0 0 1 - 2 0 0

PRELIMINARY

SYSTEM REFERENCE MANUAL

N O T I C E

Micro Craft Corporation reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

MICRO CRAFT CORPORATION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. MICRO CRAFT CORPORATION SOFTWARE IS SOLD OR LICENSED "AS IS." THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT MICRO CRAFT CORPORATION, ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL MICRO CRAFT CORPORATION BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF MICRO CRAFT CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from Micro Craft Corporation.

Copyright 1983 by Micro Craft Corporation

Micro Craft Corporation
4747 Irving Blvd.
Dallas, Texas 75247
(214) 630-2562

Additional copies of this manual
may be ordered from your DEALER
by using the MICRO CRAFT part number 680-0001-200.

Ask your DEALER also for a
free brochure with a complete list of a
Micro Craft manuals and products

MICRO CRAFT CORPORATION
Customer Support Department
4747 Irving Blvd.
Dallas, Texas 75247

TABLE OF CONTENTS

Pg. 1	INTRODUCTION
Pg. 3	WELCOME
Pg. 5	CHAPTER 1 - OVERVIEW
Pg. 9	POWER SUPPLY
Pg. 9	MAIN BOARD
Pg. 9	KEYBOARD
Pg. 9	KEYBOARD
Pg. 10	KEYBOARD INTERFACE
Pg. 10	CRT
Pg. 10	PROCESSOR
Pg. 10	MEMORY
Pg. 10	I/O
Pg. 11	EXPANSION SLOTS
Pg. 13	CHAPTER 2 - HARDWARE
Pg. 15	POWER SUPPLY
Pg. 15	MAIN BOARD
Pg. 15	J1 - POWER
Pg. 16	J2 - RESET SWITCH ASSEMBLY
Pg. 16	J3 - KEYBOARD
Pg. 16	J4 - VIDEO OUTPUT
Pg. 17	J5 - MINI STANDARD DISKETTE CONNECTOR (34 Pin)
Pg. 18	J6 - PARALLEL PRINTER PORT
Pg. 19	J7 - STANDARD DISKETTE CONNECTOR (50 Pin)
Pg. 20	J8 - RS-232C INTERFACE
Pg. 21	J9 - GAME CONTROL (A/D)
Pg. 21	J10 - SPEAKER
Pg. 22	J11 - LIGHT PEN
Pg. 22	J12 - NOT USED AND NOT PHYSICALLY PRESENT
Pg. 22	J13 - AUTO LINE FEED
Pg. 22	J14 - AUXILIARY VIDEO
Pg. 23	KEYBOARD
Pg. 23	KEYBOARD
Pg. 23	KEYBOARD INTERFACE
Pg. 24	PRINTER OUTPUT
Pg. 24	RS-232
Pg. 25	REAL TIME CLOCK
Pg. 25	CRT
Pg. 26	SPEAKER
Pg. 26	DISK DRIVES
Pg. 27	MC68000
Pg. 27	INTERRUPTS
Pg. 28	MEMORY
Pg. 29	EXPANSION SLOTS
Pg. 29	CO-PROCESSORS
Pg. 31	CHAPTER 3 - MEMORY ALLOCATION
Pg. 33	MEMORY CIRCUITRY
Pg. 33	MEMORY MAP
Pg. 35	USES

Pg. 37	CHAPTER 4 - CP/M-68K, CPMBIOS, and ROMBIOS
Pg. 39	FLOW
Pg. 39	ADDITIONAL CALLS
Pg. 40	FUNCTION 256 - Change Diskette Drive Format
Pg. 41	FUNCTION 257 - Read Diskette Drive Format
Pg. 42	FUNCTION 258 - Set the Physical Parameter Table, Set the CP/M Disk Parameter Table, or Set the Disk Interlace Table
Pg. 43	TERMINAL ATTRIBUTES
Pg. 43	CONTROL FUNCTIONS
Pg. 43	ESCAPE SEQUENCES
Pg. 44	ROMBIOS
Pg. 44	ROMBIOS FUNCTIONS
Pg. 45	INPUT/OUTPUT
APPENDICES	
Pg. A-1	TOOL-KIT
Pg. B-1	KEYBOARD CODES
Pg. C-1	ASCII CODES
Pg. D-1	EXPANSION SLOT PINOUTS
Pg. E-1	DIRECT I/O
Pg. F-1	ROMBIOS

I N T R O D U C T I O N

WELCOME TO THE DIMENSION 68000

You are now the owner of the most powerful most compatible, most flexible micro computer available in the world today. You, also, may be assured that the designers of Dimension 68000 intend that the statement above will be true for many years to come!

This manual is a tool for gaining an in-depth understanding of the operation of the DIMENSION 68000. It is NOT a tutorial. It is a collection of facts and other data.

C H A P T E R 1

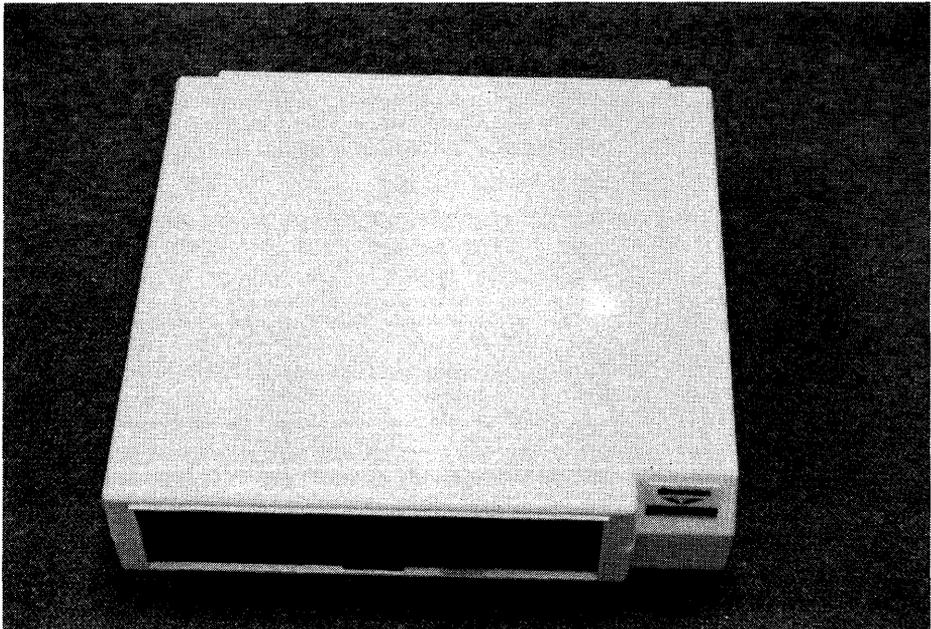
O V E R V I E W

The DIMENSION 68000 system is made up of several functional components or systems. Those systems are:

- The Power Supply
- The Main Board
- The Keyboard
- The CRT Interface
- The Processor (CPU)
- The Memory
- The I/O (Input/Output)
- The Expansion Slots
- The Floppy Disks and Controllers

These systems, together, handle all of the operations that are performed by the DIMENSION 68000.

This manual often gives directional instructions for the location, by the user, of various systems, parts, and components. In this manual, all directional instructions will refer to the orientation depicted in the view (looking down on the top) below:

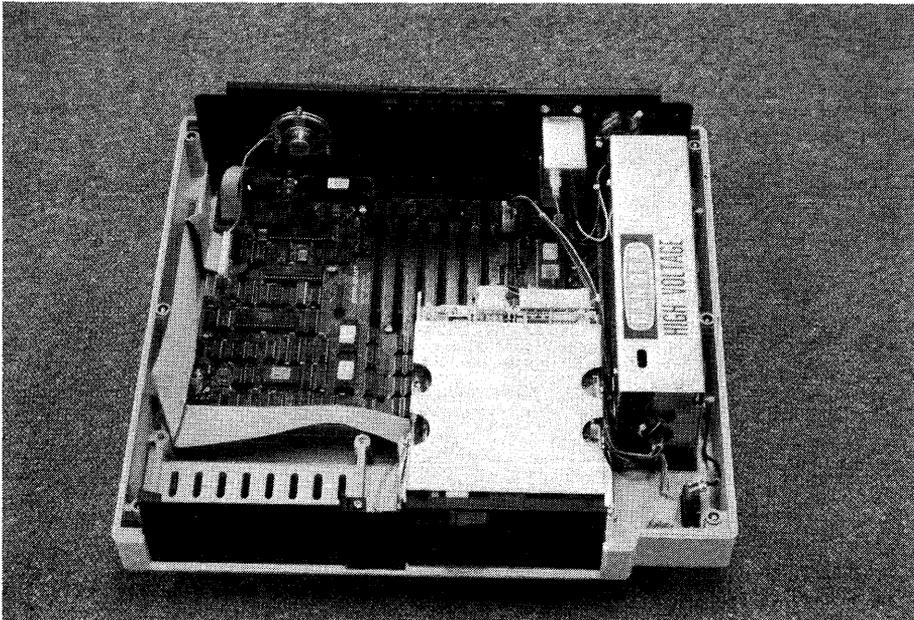


The directions "Front" and "down" are towards the diskette drives and the reset switch. The directions "back" and "up" are away from the drives and toward the REAR PANEL.

To remove the top cover from the system unit of the DIMENSION 68000, use the following steps:

- TURN OFF the POWER
- TURN the system unit over (upside down)
- USING a PHILLIPS HEAD SCREWDRIVER, LOOSEN the SIX (6) SCREWS that are holding the top cover. These screws are NOT removable from the bottom cover of the system unit. The screws are designed to be held captive in the bottom cover so that they will not be lost.
- With the loosened cover still in place, TURN the system unit back over (right side up)
- LIFT off the top cover

Looking down upon the system unit (with the top cover removed), the system unit looks like the picture below.



The POWER SUPPLY

The POWER SUPPLY is enclosed in the metal cover on the right side inside the system unit. It supplies +5 VDC @ 8 A., +12 VDC @ 2.5 A., and -12 VDC @ 2 A. It is a high-frequency "switching"-type power supply.

The MAIN BOARD

The MAIN BOARD is the large printed circuit board which takes up most of the bottom of the case of the system unit. The MAIN BOARD is the actually the computer.

On the MAIN BOARD are the integrated circuits, components, and the expansion slots. The microprocessor (or "brain") of the DIMENSION 68000 is the Motorola (or equivalent) MC68000L8 device. The 68000 has an addressing capacity of 16 M Bytes. The MAIN BOARD can have up to 512 K Bytes of RAM (Random Access Memory) in 128 K Byte increments.

Also, on the MAIN BOARD are 8 K Bytes of ROM (Read Only Memory). The ROM is used to hold the system MONITOR and the ROMBIOS.

The six long peripheral slots that are toward the middle back of the MAIN BOARD can each hold a card. The cards can be any one of the following types:

- An Emulator Card (examples; Z-80, 8086, and 6512)
- A Memory Expansion Card
- A Hard Disk Controller
- A Peripheral Controller
- A Peripheral Device

The KEYBOARD System

The Keyboard system is composed of two parts; the Keyboard and the Keyboard Interface.

The Keyboard

The Keyboard unit, that plugs into the back of the DIMENSION 68000 system unit, is a microprocessor controlled assembly that is capable of handling in excess of 30 characters per second. The Keyboard assembly transmits the characters, that are typed in, to the DIMENSION 68000 system using the following characteristics:

- 30 characters/second
- 8 bit characters
- ASCII (American Standard Code for Information Interchange) encoding
- asynchronous transmission

The Keyboard Interface

The Keyboard Interface circuitry receives the characters from the Keyboard assembly. The asynchronous transmission technique adds a "start" bit and adds a "stop" bit to each character, and then transmits the character, with the added bits, in a serial mode, from the Keyboard assembly to the Keyboard Interface. The Keyboard Interface removes the added "start" and "stop" bits from each character received, and converts the character from a serial mode to a parallel mode. This is implemented using one serial channel of the Signetics 2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) chip.

The CRT Interface

The CRT Interface provides a video output signal for the connection of a CRT or other video display device. The CRT interface supplies a video signal with the following characteristics:

- RS-170 compatible signal
- Composite video (sync + video)
- Capable of COLOR and MONOCHROME operation

The Processor (CPU)

The Processor (or CPU - Central Processing Unit) is a Motorola MC68000 (or equivalent) microprocessor integrated circuit (or chip). The Motorola MC68000 device is internally a 32 Bit microprocessor. The MC68000 has 16 Bit wide data paths.

The Memory

There are two (2) types of memory in the DIMENSION 68000; RAM (Random Access Memory) and ROM (Read Only Memory). The Main Board can hold 128 K Bytes, 256 K Bytes, 384 K Bytes, or 512 K Bytes of RAM. The amount of RAM can be expanded from 512 K Bytes up to 16 M Bytes by using memory expansion cards. There are 8 K Bytes of ROM.

The I/O (Input/Output)

The I/O (Input/Output) for the DIMENSION 68000 system unit consists of the following devices:

- The Keyboard Interface
- The CRT Interface
- The Diskette Drives
- The Parallel Printer Port
- The RS-232C Interface
- The Game Control Interface

Optionally, there can be connectors on the rear panel to connect the following:

- Additional Mini (5 1/4", 3 1/2", 3 1/4", etc.) Diskette Drives
- Additional 8 Inch (Full Size) Diskette Drives (up to four total)

The Expansion Slots

There are six (6) Expansion Slots inside the system unit. The Expansion Slots provide the capability to plug in various types of cards. Some of the types of cards are:

- Emulator Card(s)
- Hard Disk Interface Card
- RS-232 Ports Card(s) - 8 ports per card
- Analog/Digital Interface Card(s)
- 512 K Byte RAM Memory Cards

It is expected that, in the future, Micro Craft Corporation and other manufacturers will be providing many types of cards to plug into the Expansion Slots.

C H A P T E R 2

H A R D W A R E

The POWER SUPPLY

The POWER SUPPLY for the DIMENSION 68000 is a high-frequency "switching"-type unit. It supplies the following voltages and currents:

- +5 VDC +, - 1% @ 8 Amps
- +12 VDC +, - 5% @ 2.5 Amps
- 12 VDC +, - 5% @ 2 Amps

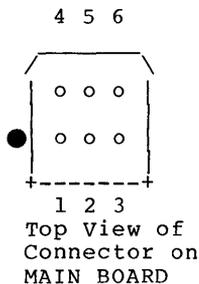
These voltages are supplied to the MAIN BOARD and to the Disk Drives. Power supplied to the MAIN BOARD is available to the Expansion Slots, and to the Game Control connector (J9) on the MAIN BOARD.

The POWER SUPPLY requires 115 VAC @ 3 Amps (MAX.) to operate. The POWER SUPPLY for the DIMENSION 68000, optionally, may be configured to operate from 220 VAC power. The POWER SUPPLY input power, whether 115 VAC or 220 VAC, must be of a frequency in the range of from 50 Hz. to 400 Hz. The input power to the POWER SUPPLY is filtered for RFI (Radio Frequency Interference). The input power to the POWER SUPPLY also has power line surge suppression applied.

The MAIN BOARD

The MAIN BOARD has the Expansion Slots, the integrated circuits (including the 68000 microprocessor), the components, and the connectors that link the MAIN BOARD to the POWER SUPPLY, the Keyboard, etc. The connectors on the MAIN BOARD are detailed below.

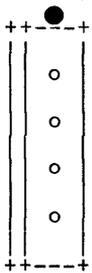
J1 - POWER



PIN	USAGE
1	+5 VDC
2	GROUND
3	+12 VDC
4	+5 VDC
5	GROUND
6	-12 VDC

The Connector Necessary to Mate With the Connector on the MAIN BOARD is made using a MOLEX P/N 03-09-1064 recepticle and MOLEX P/N 02-09-1118 pins (or the equivalent).

J2 - RESET SWITCH ASSEMBLY



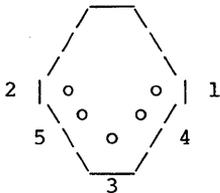
Top View of Connector on MAIN BOARD

PIN	USAGE
1	RESET
2	GROUND
3	NORMAL
4	LED POWER

The PAINT DOT indicates pin 1.

The Connector Necessary to Mate With the Connector on the MAIN BOARD is a MOLEX P/N 22-26-9041 (or the equivalent).

J3 - KEYBOARD



View of Front of Connector at rear of SYSTEM UNIT

PIN	USAGE
1	SIGNAL FROM KEYBOARD
2	
3	
4	GROUND
5	+5 VDC POWER TO KEYBOARD

The Connector Necessary to Mate With the Connector at rear of SYSTEM UNIT is a 5 PIN, "DIN"-type, MALE

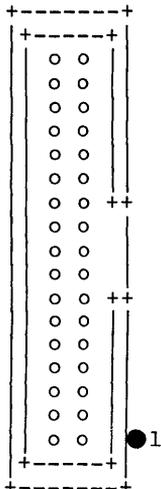
J4 - VIDEO OUTPUT



PIN	USAGE
Center	SIGNAL
Outside	GROUND

The Connector Necessary to Mate With the Connector on the REAR PANEL is an RCA type MALE

J5 - MINI STANDARD DISKETTE CONNECTOR (34 Pin)



View of the Connector on the MAIN BOARD

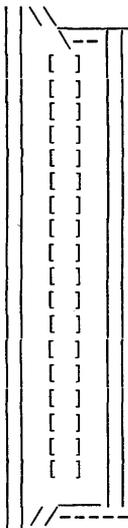
The PAINT DOT indicates pin 1

PIN	USAGE
1	GROUND
2	
3	GROUND
4	
5	GROUND
6	DRSEL 3 *
7	GROUND
8	INDEX*
9	GROUND
10	DRSEL 0 *
11	GROUND
12	DRSEL 1 *
13	GROUND
14	DRSEL 2 *
15	GROUND
16	
17	GROUND
18	M DIR *
19	GROUND
20	M STEP *
21	GROUND
22	M W DATA *
23	GROUND
24	WR ENA *
25	GROUND
26	
27	GROUND
28	
29	GROUND
30	RD DATA
31	GROUND
32	M HEAD *
33	GROUND
34	

The Connector Necessary to Mate With the Connector on the MAIN BOARD is an ANSLEY P/N 609-34000 (or the equivalent)

NOTE: An * after a signal name indicates that the logic for that signal is inverted, that is to say that the signal is active on a "0" instead of a "1".

J6 - PARALLEL PRINTER PORT

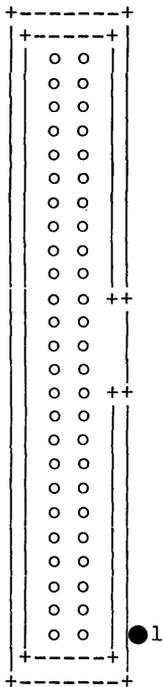


PIN	USAGE
----	-----
1	STROBE*
2	D0*
3	D1*
4	D2*
5	D3*
6	D4*
7	D5*
8	D6*
9	D7*
10	ACKNLG*
11	RDY*
12	Not Connected
13	Not Connected
14	AUTO LINE FEED *
15	Not Connected
16	Not Connected
17	GROUND
18	Not Connected
19	GROUND
20	GROUND
21	GROUND
22	GROUND
23	GROUND
24	GROUND
25	GROUND
26	GROUND
27	GROUND
28	GROUND
29	GROUND
30	GROUND
31	INIT*
32	Not Connected
33	GROUND
34	Not Connected
35	Not Connected
36	Not Connected

The Connector Necessary to Mate With the Connector on the REAR PANEL is an Amphenol 36 Pin "Blue Ribbon" type

NOTE: An * after the signal name indicates that the logic for that signal is inverted, that is to say that the signal is active on a "0" instead of a "1".

J7 - STANDARD DISKETTE CONNECTOR (50 Pin)



View of the Connector on the MAIN BOARD

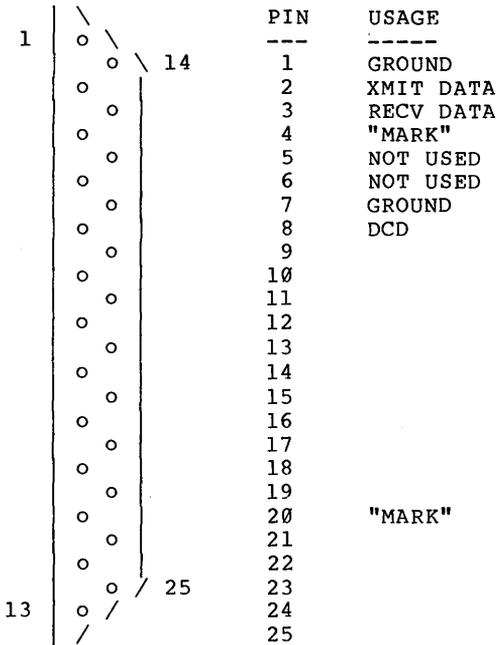
The PAINT DOT indicates pin 1

PIN	USAGE
1	GROUND
2	M LOW C *
3	GROUND
4	MOTOR OFF 0 *
5	GROUND
6	MOTOR OFF 1 *
7	GROUND
8	MOTOR OFF 2 *
9	GROUND
10	2 SIDED *
11	GROUND
12	
13	GROUND
14	M HEAD *
15	GROUND
16	
17	GROUND
18	M HEAD LD *
19	GROUND
20	INDEX *
21	GROUND
22	
23	GROUND
24	MOTOR OFF 3 *
25	GROUND
26	DR SEL 0 *
27	GROUND
28	DR SEL 1 *
29	GROUND
30	DR SEL 2 *
31	GROUND
32	DR SEL 3 *
33	GROUND
34	M DIR *
35	GROUND
36	M STEP *
37	GROUND
38	M W DATA *
39	GROUND
40	WR ENA *
41	GROUND
42	TK 00 *
43	GROUND
44	W PROT *
45	GROUND
46	RD DATA
47	GROUND
48	
49	GROUND
50	

The Connector Necessary to Mate With the Connector on the MAIN BOARD is an ANSLEY P/N 609-5000 (or the equivalent)

NOTE: An * after a signal name indicates that the logic for that signal is inverted, that is to say that the signal is active on a "0" instead of a "1".

J8 - RS-232C INTERFACE

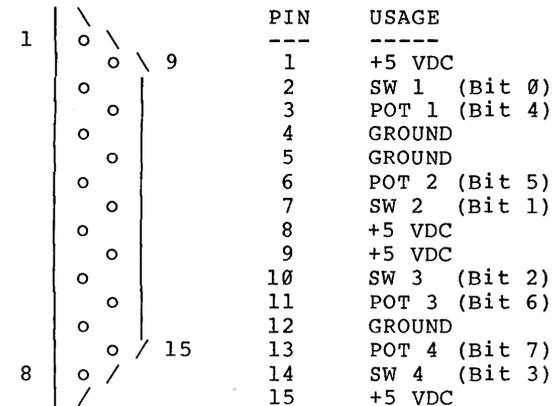


The Connector Necessary to Mate With the Connector on the REAR PANEL is a 25 PIN, "D Subminiature" type MALE (DB25P)

NOTE: "MARK" indicates a high logic signal (+12 VDC)

View of Connector on REAR PANEL

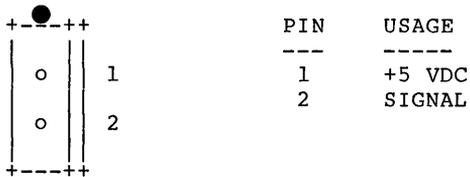
J9 - GAME CONTROL (A/D)



The Connector Necessary to Mate With the Connector on the REAR PANEL is a 15 PIN, "D Subminiature" type MALE (DB15P)

View of Connector on REAR PANEL

J10 - SPEAKER

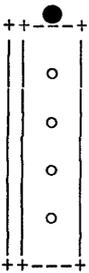


The Connector Necessary to Mate With the Connector on the MAIN BOARD is a MOLEX P/N 22-26-9021 (or the equivalent).

View of Connector on MAIN BOARD

The PAINT DOT indicates pin 1

J11 - LIGHT PEN



View of Connector
on MAIN BOARD

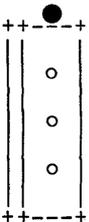
PIN	USAGE
1	RESET
2	GROUND
3	NORMAL
4	LED POWER

The PAINT DOT
indicates pin 1

The Connector Necessary
to Mate With the Connector
on the MAIN BOARD is a MOLEX
P/N 22-26-9041 (or the
equivalent).

J12 - NOT USED AND NOT PHYSICALLY PRESENT

J13 - AUTO LINE FEED



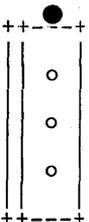
View of Connector
on MAIN BOARD

PIN	USAGE
1	LOGIC "1" (+5 VDC BFFRD)
2	CONTROL LINE
3	GROUND (LOGIC "0")

The PAINT DOT
indicates pin 1

The Connector Necessary
to Mate With the Connector
on the MAIN BOARD is a MOLEX
P/N 22-26-9031 (or the
equivalent).

J14 - AUXILIARY VIDEO



View of Connector
on MAIN BOARD

PIN	USAGE
1	+5 VDC
2	SIGNAL (Signal is NOT Adjustable)
3	GROUND

The PAINT DOT
indicates pin 1

The Connector Necessary
to Mate With the Connector
on the MAIN BOARD is a MOLEX
P/N 22-26-9031 (or the
equivalent).

A schematic of the MAIN BOARD is included in the APPENDICES.

The KEYBOARD System

The Keyboard system is composed of two parts; the Keyboard and the Keyboard Interface.

The Keyboard

The Keyboard is a microprocessor controlled assembly that is capable of handling in excess of 30 characters per second. The Keyboard transmits characters to the DIMENSION 68000 system using the following characteristics:

- 30 characters/second
- 8 bit characters
- ASCII (American Standard Code for Information Interchange) encoding
- asynchronous transmission
- Power required = +5 VDC at 250 mA.
- Rollover = 2 key
- Special Keys =

The Keyboard Interface

The Keyboard Interface receives the characters from the Keyboard assembly. The asynchronous transmission that is used between the Keyboard and the Keyboard Interface adds a "start" bit and a "stop" bit to each character. The Keyboard Interface removes the added "start" and "stop" bits from each character received.

- Memory Mapped Locations
 - FFC401 = Keyboard Mode Register
 - FFC403 = Keyboard Status
 - FFC405 = Keyboard Interface Command Register
 - FFC407 = Keyboard Data
 - FFC409 = Auxilliary Command Register

A list of the keys, on the keyboard, and the codes that each key generates is included in the KEYBOARD CODES APPENDIX. Also, a detailed description of the keyboard and the keyboard interface is included in the DIRECT I/O APPENDIX.

The Keyboard Interface, the Printer, the RS-232C Interface, and the Real Time Clock are all handled by a Signetics (or equivalent) 2681 type Dual Universal Asynchronous Receiver / Transmitter (DUART) integrated circuit (chip).

The PRINTER OUTPUT

The PRINTER OUTPUT is handled through the DUART chip that is described above. The PRINTER OUTPUT is unusual in that the user writes the character to be printed to the SET OUTPUT Register, and then writes the character to the RESET OUTPUT Register.

- Memory Mapped Locations

FFC41B = Printer Port Configuration Register

FFC41B = Printer Port Status

FFC41D = Printer Port Data SET Register

FFC41F = Printer Port Data RESET Register

The AUTO LINE FEED * signal is strappable for either enabling the auto line feed function or disabling it. This strapping is done at J13 on the MAIN BOARD. The strapping options for J13 are discussed above under the topic MAIN BOARD.

A detailed discussion of the printer port is contained in the DIRECT I/O APPENDIX.

The RS-232 Interface

The RS-232C Interface is handled through the DUART chip that is described above. The RS-232 Interface is capable of handling bit-per-second rates of from 75 BPS to 19.2 KBPS. (It is not proper to speak of Baud rate on an RS-232 Interface. Baud rate is properly defined as the number of signalling changes per second on a telephone or telegraph communications line. Baud rate is normally considered to be the signalling changes per second between modems. Baud rate and bit rate are not always the same.)

- Memory Mapped Locations

FFC409 = Auxilliary Control Register

FFC40B = Interrupt Mask Register

FFC411 = Mode Registers (1 and 2)

FFC413 = Clock Select Register

FFC413 = Status Register

FFC415 = Command Register

FFC417 = Output Data Register

FFC417 = Input Data Register

FFC41B = Input Port for DCD

A detailed discussion of the RS-232C interface is contained in the DIRECT I/O APPENDIX.

The REAL TIME CLOCK

The REAL TIME CLOCK is handled through the DUART chip that is described above. Neither the ROMBIOS nor the CP/M BIOS use interrupts, and therefore they do not use the REAL TIME CLOCK.

- Memory Mapped Locations

- FFC409 = Auxilliary Control Register
- FFC409 = Input Port Change Register
- FFC40B = Interrupt Mask Register
- FFC40B = Interrupt Status Register
- FFC40D = Counter/Timer Upper Register (Most Significant Byte)
- FFC40F = Counter/Timer Lower Register (Least Significant Byte)

A detailed discussion of the REAL TIME CLOCK is contained in the DIRECT I/O APPENDIX.

The CRT Interface

The CRT Interface provides a video output signal to connect a CRT or other video display device. The CRT interface supplies a video signal with the following characteristics:

- EIA RS-170 compatible signal
(EIA = Electronic Industries Association)
- NTSC Positive Composite Color Video
(NTSC = National Television Standards Committee)
- Capable of COLOR and MONOCHROME operation
- Adjustable output level (0 to 1 Volt Pk to Pk)

The tint of the color signal is adjustable by means of a trimmer capacitor that is located on the MAIN BOARD. Also, there is potentiometer that is used to adjust the level of the output signal that is routed to the main video connector on the rear panel.

On the MAIN BOARD is an auxiliary video connector which is described above. The signal available on the auxiliary video connector is an NTSC compatible positive video signal. The black level is about 0.75 Volts, the white level is about 2.0 Volts, and the sync tip level is 0 Volts. The output level is NOT adjustable, and the output is NOT protected against short circuits.

- Memory Mapped Locations

- FF8001 = CRT Controller Register Address Selection Register
- FF8003 = CRT Controller Addressed Register Data
- FF8005 = Display Buffer Address High Byte
- FF8009 = CRT Display Write Mode Register
- FF800A = Reset CRT Controller

A detailed description of the CRT Interface is included in the DIRECT I/O APPENDIX.

The SPEAKER

The SPEAKER is inside the case of the system unit. It is driven by half of a 74LS74 flip-flop through a Darlington amplifier circuit. The SPEAKER connection is described above. The SPEAKER used is a one. It is an 8 ohm unit.

The SPEAKER is controlled by a soft switch. The switch can put the cone of the SPEAKER into the "in" position or the "out" position. This switch operates in the same fashion as a TOGGLE switch. Each time a program references the memory address associated with the SPEAKER switch, the speaker will change state: move from "in" to "out" or from "out" to "in". Each time the state of the SPEAKER is changed, a tiny "click" is produced. By referencing the address of the SPEAKER switch very frequently and continuously, a program can cause the generation of a steady tone from the speaker. A program can reference the address of the SPEAKER switch by performing either a "read" or a "write" to that address.

- Memory Mapped Locations
FFC800 = Speaker Toggle Address

The SPEAKER is also described in the DIRECT I/O APPENDIX.

The DISK DRIVES

The DISK DRIVES that are standard on the DIMENSION 68000 are half height, 5 1/4 inch, double-sided, double-density, half-steppable, 40 track diskette drives. They are capable of storing up to 400 KBytes of data.

Optionally available, are 80 track diskette units that are capable of storing up to 800 KBytes. As well as 8 inch diskette drives, 3 1/2 inch diskette drives, and 3 1/4 inch diskette drives.

The controller for the diskettes is the NEC uPD765A integrated circuit.

- Memory Mapped Locations
FFD001 = Diskette Controller Status
FFD003 = Data Register
FFD005 = Control Register

Further information is available in the PINOUT APPENDIX and the DIRECT I/O APPENDIX. A detailed description of the controller is available in the NEC DATA SHEET for the uPD765A.

The MC68000

The MC68000 microprocessor used in the DIMENSION 68000 is an 8 MHz. device. It has 16 bit wide data paths (but all of the internal architecture is 32 bits wide). The 68000 chip has the following registers:

8 DATA REGISTERS that are 32 bits wide
 7 ADDRESS REGISTERS that are 32 bits wide
 1 USER STACK POINTER that is 32 bits wide
 1 SUPERVISOR STACK POINTER that is 32 bits wide
 1 PROGRAM COUNTER that is 32 bits wide
 1 STATUS REGISTER that is 16 bits wide

The 68000 allows 5 data types:

Bits
 BCD Digits (4 bits/BCD Digit - 1 BCD DIGIT = 1 Nybble)
 Bytes (8 bits)
 Words (16 bits)
 Long Words (32 bits)

The 68000 has the capability of directly addressing 16 MBytes of memory. There are 14 addressing modes on 61 basic instructions for over 1000 total instruction types.

For a detailed discussion of the 68000 microprocessor, see the MOTOROLA MC68000 USER'S MANUAL.

INTERRUPTS

The DIMENSION 68000 comes with six maskable levels of interrupts and one non-maskable interrupt level. The maskable interrupt levels are levels 1, 2, 3, 4, 5, and 6. The non-maskable interrupt level is level 7. Level 1 is the lowest priority and level 7 is the highest priority. The DUART (which is described above) is set by the hardware circuitry to be the non-maskable interrupt on priority level 7. The DUART has the capability to mask for interrupts the Keyboard, the RS-232, the Printer, and the Real Time Clock. All of the interrupts on the DIMENSION 68000 utilize the Autovector type of interrupt processing.

The Autovector processing computes the interrupt vector number (in hex) by the following formula:

$$\text{VECTOR NUMBER} = (\text{PRIORITY-LEVEL} + 024)$$

The above formula gives the interrupt vector number in hexadecimal. The Autovector processing computes the location of the interrupt vector (in hex) by the following formula:

$$\text{VECTOR LOCATION} = \text{VECTOR NUMBER} * 4$$

The location of the vector address contains the address of an interrupt routine. The following chart shows the assignments for priority level, vector number, and vector address for the DIMENSION 68000.

DEVICE	PRIORITY LEVEL (decimal)	VECTOR NUMBER (decimal)	LOCATION OF VECTOR ADDRESS (hex)
SLOT 6	1	25	064
SLOT 5	2	26	068
SLOT 4	3	27	06C
SLOT 3	4	28	070
SLOT 2	5	29	074
SLOT 1	6	30	078
DUART	7	31	07C

Additionally, there are exception vectors. Vectors 0 and 1 are taken as a pair and not as two separate vectors. The following chart shows the assignments for the vector number, the vector address, and its usage.

VECTOR NUMBER (decimal)	VECTOR ADDRESS (hex)	ASSIGNMENT	USED BY
0	000	Reset: Initial Supervisor Stack Pointer	BOOT
	004	Reset: Initial Program Counter	
2	008	Bus Error	CP/M
3	00C	Address Error	CP/M
4	010	Illegal Instruction	CP/M
5	014	Zero Divide	CP/M
6	018	CHK Instruction	CP/M
7	01C	TRAPV Instruction	CP/M
8	020	Privelege Violation	CP/M
9	024	Trace	CP/M
10	028	Line 1010 Emulator	CP/M
11	02C	Line 1111 Emulator	CP/M
12 - 24	030-060	NOT USED	NO USER
32 - 255	080-3FF	NOT USED	NO USER

MEMORY

The DIMENSION 68000 memory is byte addressable. Each memory address has associated with it an additional bit that is used for emulation trapping. If a memory address is accessed, and the emulation trap bit is on, then the emulation strobe bit, on the bus and at the slots, is toggled.

The DIMENSION 68000 can address up to 16 MBytes of memory. This is made possible by the use of a 24 Bit wide address.

The DIMENSION 68000 memory has, in its associated circuitry, bus arbitration logic. The bus arbitration priorities are:

HIGHEST	CRT CONTROLLER CIRCUITRY
	EXPANSION SLOT 6
	EXPANSION SLOT 5
	EXPANSION SLOT 4
	EXPANSION SLOT 3
	EXPANSION SLOT 2
	EXPANSION SLOT 1
LOWEST	MC68000 CPU

EXPANSION SLOTS

Each EXPANSION SLOTS can accept a plug in card. Each slot directly connects to the system's bus. A detailed discussion of the connections for the EXPANSION SLOTS is contained in the EXPANSION SLOT PINOUT APPENDIX.

Each EXPANSION SLOT has four signals that are unique to that slot. They are:

- BOARD ENABLE	FROM SYSTEM
- BOARD REQUEST	TO SYSTEM
- BOARD ACKNOWLEDGE	FROM SYSTEM
- BOARD INTERRUPT	TO SYSTEM

The Micro Craft Corporation supplies an expansion slot board prototyping kit. This kit contains a prototyping board, a set of expansion slot board drawings, data regarding the signals available on the expansion slot connector, and data about the DIMENSION 68000 system circuitry.

CO-PROCESSORS

The DIMENSION 68000 has been designed to allow the simultaneous use of more than one processor. All of the emulator cards have a microprocessor chip on them. This makes all of the emulator cards, co-processors. Some other possible co-processor uses are:

- an array processor
- a mathematical floating point processor
- a mathematical multiply and divide processor
- a specific machine emulator
- an intelligent device controller

C H A P T E R 3

M E M O R Y A L L O C A T I O N

MEMORY CIRCUITRY

The MEMORY CIRCUITRY contains bus arbitration logic circuitry and emulation trapping circuitry. The bus arbitration logic circuitry is designed to give to the CRT Controller circuitry the highest priority. The next highest priority to EXPANSION SLOT 6, then, EXPANSION SLOTS 5, 4, 3, 2, and 1. The lowest priority is the MC68000 CPU.

The emulation trapping circuitry for the memory is implemented by including in the memory design an additional bit with each byte of memory. The additional bit is used as an emulation trapping flag. If the bit is on for a specific byte, and that byte is accessed, then the emulation circuitry causes the emulation strobe bit in the system bus to be pulsed. This does NOT hold true for accesses by the CRT Controller circuitry, NOR does it hold true for accesses by the MC68000 CPU.

MEMORY MAP

The DIMENSION 68000 memory contains both Read Only Memory (ROM) and Random Access Memory (RAM). The system memory is allocated as follows.

ADDRESS	FUNCTION
-----	-----
*****	INTERRUPT VECTORS *****
000000	RESET - Initial Stack Pointer
000004	RESET - Initial Program Counter (PC)
000008	BUS ERROR
00000C	ADDRESS ERROR
000010	ILLEGAL INSTRUCTION
000014	DIVIDE BY 0
000018	CHECK INSTRUCTION
00001C	TRAPV INSTRUCTION
000020	PRIVILEGE INSTRUCTION
000024	TRACE
000028	OPCODE 1010 EMULATION
00002C	OPCODE 1111 EMULATION
000030 - 00005F	Reserved
000060	SPURIOUS INTERRUPT
000064	LEVEL 1 INTERRUPT
000068	LEVEL 2 INTERRUPT
00006C	LEVEL 3 INTERRUPT
000070	LEVEL 4 INTERRUPT
000074	LEVEL 5 INTERRUPT
000078	LEVEL 6 INTERRUPT
00007C	LEVEL 7 INTERRUPT - DUART - Keyboard, Printer, etc.
000080 - 0000BF	TRAP VECTORS
0000C0 - 0000FF	Reserved

ADDRESS	FUNCTION
***** PERIPHERAL CONTROLLER BOOT RAM *****	
000100 - 000113	CONTROLLER 1
000114 - 000127	CONTROLLER 2
000128 - 00013B	CONTROLLER 3
00013C - 00014F	CONTROLLER 4
000150 - 000163	CONTROLLER 5
000164 - 000177	CONTROLLER 6
***** RESERVED PERIPHERAL RAM *****	
000178 - 00019B	MONITOR RESERVED LOCATIONS
00019C	TOP OF RAM
0001A0 - 0001A7	ADDRESS of ROM BASED RESET VECTOR
0001A8 - 0001CB	CRT PARAMETER TABLE
0001CC - 0001E5	BOOT DISK PARAMETER TABLE
0001E6 - 0001E9	SYSTEM CONFIGURATION WORD
0001EA - 0001ED	ADDRESS of TEXT SCREEN TABLE
0001EE - 0001F1	ADDRESS of KEYBOARD TABLE
0001F2 - 0001F5	ADDRESS of GRAPHICS CHARACTER SET #1 TABLE
0001F6 - 0001F9	ADDRESS of GRAPHICS CHARACTER SET #2 TABLE
0001FA - 0001FD	ADDRESS of PRESENT SCREEN TABLE
0001FE	INTERRUPT MASK FOR 2861 I/O CHIP
0001FF	END OF RESERVED PERIPHERAL RAM
***** USER RAM AREA *****	
000200 - 00119F	NORMAL (80 x 50) SCREEN TEXT AREA
001200 - 01FFFF	MINIMUM SYSTEM USER RAM
001300	Start of CP/M-68K TRANSIENT PROGRAM AREA
010000 - 01FFFF	Co-Processor RAM Area, During Emulation (Min)
010000 - 07FFFF	Co-Processor RAM Area, During Emulation (Max)
020000 - FEFFFF	Expanded User RAM Area
***** RESERVED AREAS *****	
FF0000 - FF1FFF	Basic Processor ROM Area (ROMBIOS and MONITOR)
FF2000 - FF7FFF	Reserved Area (Graphics RAM)
***** INPUT/OUTPUT REGION *****	
FF8000 - FF87FF	CRT Controller
FF8800 - FF8FFF	Slot 1
FF9000 - FF97FF	Slot 2
FF9800 - FF9FFF	Slot 3
FFA000 - FFA7FF	Slot 4
FFA800 - FFAFFF	Slot 5
FFB000 - FFB7FF	Slot 6
FFB800 - FFBFFF	Reserved
FFC000 - FFC3FF	Reserved
FFC400 - FFC7FF	Keyboard/Printer/Modem/Real Time Clock
FFC800 - FFCBFF	Speaker
FFCC00 - FFCFFF	Game Control
FFD000 - FFD3FF	Floppy Disk
FFD400 - FFD7FF	Emulation Controller
FFD800 - FFDBFF	Reserved
FFDC00 - FFDDFF	ROM Toggle
FFDE00 - FFDEFF	Reserved
FFE000 - FFEFFF	Reserved

USES

The USES for the DIMENSION 68000 system memory are described by the above memory map. Some of the descriptions will be further amplified here.

The area from FE0000 to FF1FFF comprise the ROMBIOS area. This name is a combination of ROM and BIOS, where BIOS is short for Basic Input / Output System. The ROMBIOS provides the routines for I/O and for the DIMENSION 68000 system MONITOR. The MONITOR is a collection of routines that are used to "boot" the system and to communicate with the Keyboard and the CRT.

Each interrupt vector is a location in memory that contains the address of a software routine (or a ROMBIOS routine) that handles the interrupt that is associated with that specific location. The interrupt vectors are used by CP/M, by the EXPANSION SLOTS, and by the DUART which handles the Keyboard, the Printer Port, the RS-232 Interface, and by the Real Time Clock.

The highest address of the RAM varies depending on the amount of RAM that is installed. The following chart shows the relationship between the amount of memory installed and the highest address in RAM.

AMOUNT OF MEMORY	HIGHEST MEMORY ADDRESS
-----	-----
128 KB	01FFFF
256 KB	02FFFF
384 KB	03FFFF
512 KB	04FFFF
16 MB	FEFFFF

CHAPTER 4

CP/M - 68K, CPMBIOS, AND ROMBIOS

FLOW

The CP/M-68K operating system is basically composed of three parts; the Console Command Processor (CCP), the Basic Disk Operating System (BDOS), and the Basic Input / Output System (BIOS). The CCP handles the commands that are typed in from the console (or are entered from a submit file) such as DIR, or SUB.

The BDOS handles the functions such as file access, drive access, system/program control, and exceptions. The BDOS partially handles character Input/Output (I/O) functions.

The largest part of the character I/O handling and all of the basic I/O handling is performed by the CP/M BIOS. The CP/M BIOS accesses the I/O functions that are provided in the DIMENSION 68000 Read Only Memory BIOS (ROMBIOS). The ROMBIOS actually controls the I/O controllers and I/O devices.

The CP/M operating system is described more fully in the CP/M-68K Operating System User's Guide and in the CP/M Operating System Programmer's Guide. Both manuals are from Digital Research, Inc. Digital Research, Inc., is the company that developed CP/M and it is the company that licenses the use of CP/M.

ADDITIONAL CALLS

There is a list of the CP/M-68K functions in the CP/M Operating System Programmer's Guide. The Micro Craft Corporation has added three additional functions to the CP/M-68K operating system for use on the DIMENSION 68000 computer. The additional functions are presented in the following pages.

FUNCTION 256: Change Diskette Drive Format

Entry Parameters:

Register D0.W = 100H = 256 (decimal)

Register D1.W = New Disk Drive Unit Number

where 0 = Drive A:

1 = Drive B:

2 = Drive C:

3 = Drive D:

4 = Drive E:

5 = Drive F:

6 = Drive G:

7 = Drive H:

Register D2.W = New Disk Drive Format

where 0 = null

1 = Std 40 track

2 = Std 80 track

3 = IBM single side (CPM 86) 8 sector

4 = IBM double side (CPM 86) 8 sector

5 = TRS 80

6 = KayPro

7 = Cromemco

8 = Osborne

9 = IBM 3740 8 Inch Single Side, Density

A = TRS 16 Double Side, Density

B = User Defined

Returned Values:

NONE

FUNCTION 257: Read Diskette Drive Format

Entry Parameters:

Register D0.W = 101H = 257 (decimal)

Register D1.W = Disk Unit Number

where 0 = Drive A:

1 = Drive B:

2 = Drive C:

3 = Drive D:

4 = Drive E:

5 = Drive F:

6 = Drive G:

7 = Drive H:

Returned Values:

Register D0.W = Selected Drive Type

where 0 = null

1 = Std 40 track

2 = Std 80 track

3 = IBM single side (CPM 86) 8 sector

4 = IBM double side (CPM 86) 8 sector

5 = TRS 80

6 = KayPro

7 = Cromemco

8 = Osborne

9 = IBM 3740 8 Inch Single Side, Density

A = TRS 16 Double Side, Density

B = User Defined

FUNCTION 258: Set the Physical Parameter Table,
Set the CP/M Disk Parameter, or
Set the Disk Interlace Table

Entry Parameters:

Register D0.W = 102H = 258 (decimal)

Register D1.W = Command Select

where 0 = Ignore

1 = Set Physical Parameters

(DPRMTX 16 words)

2 = Set CP/M Disk Parameter Block Table

(DPBX 8 words)

3 = Set Interlace Table

(XLTX 64 words)

Register D2.L = Table Address

Returned Values:

NONE

TERMINAL ATTRIBUTES

The DIMENSION 68000 CP/M BIOS has been augmented to provide a set of "ESCAPE SEQUENCES" that will handle various screen functions on the CRT display. Also, several "CONTROL" screen functions have been implemented. The following charts list the "ESCAPE SEQUENCES" and the "CONTROL" functions. These functions are accessed by outputting the code or codes through the CP/M "CONOUT" function.

CONTROL FUNCTIONS

HEX VALUE	FUNCTION
B1	Clear Screen
07	Bell
B7	Home
91	Cursor Up
92	Cursor Left
93	Cursor Right
94	Cursor Down (Line Feed)

ESCAPE SEQUENCES (Escape plus the hex value listed)

HEX VALUE	ASCII CODE	FUNCTION
48	H	Home Cursor
3D	=	Position Cursor
76	v	Clear Screen & Home Cursor
41	A	Cursor Up
42	B	Cursor Down
43	C	Cursor Right
44	D	Cursor Left
4B	K	Erase End of Line
4A	J	Erase End of Page
78	x	Clear End of Line
79	y	Clear End of Page
63	c	Turn On Cursor Flag
64	d	Turn Off Cursor Flag
65	e	Turn On Inverse Flag
66	f	Turn Off Inverse Flag

ROMBIOS

The CP/M functions operate by performing calls to the appropriate routines in the ROMBIOS. ROMBIOS is an acronym for the ROM Based Input / Output System. The usage of the ROMBIOS is fully described in the ROMBIOS Appendix to this manual. A chart showing the ROMBIOS functions is included to illustrate the contents of the ROMBIOS.

ROMBIOS FUNCTIONS

-
- 1 = Initialize CRT Controller
 - 2 = Output a character to the screen
 - 3 = Clear the TEXT screen
 - 4 = Read a key stroke
 - 5 = Get keyboard status
 - 6 = Boot from disk
 - 7 = Start the built-in monitor
 - 8 = Read selected disk drive status
 - 9 = Select disk drive and turn motor ON
 - 10 = Deselect disk drive and turn motor OFF
 - 11 = Seek to selected disk track
 - 12 = Seek to track 0 and recalibrate disk - rezero
 - 13 = Seek to track and read a sector
 - 14 = Seek to track and write a sector
 - 15 = Seek to track and format a track
 - 16 = Format the entire disk
 - 17 = Read the sector ID information
 - 18 = Read the disk status
 - 19 = Reserved
 - 20 = Reserved
 - 21 = Read Printer Status
 - 22 = Output a character to Printer Interface
 - 23 = Position CRT cursor
 - 24 = Read character at present CRT cursor location
 - 25 = Clear Graphics Screen
 - 26 = Set graphics color
 - 27 = Plot X, Y point
 - 28 = Reserved
 - 29 = Read color at designated point
 - 30 = Output Character to CRT with No Control Characters

INPUT / OUTPUT

To fully appreciate the functions supplied in the ROMBIOS, requires a knowledge of the actual input and output circuitry used inside the DIMENSION 68000 computer. A detailed discussion of the programming requirements for the input and the output circuitry is included in the DIRECT I/O Appendix to this manual.

C H A P T E R 5

C P / M - 6 8 K

BDOS AND BIOS

The CP/M operating system is composed of two parts; the Basic Disk Operating System (BDOS) and the Basic Input / Output System (BIOS). The BDOS is that part of the operating system that is independent of the Input/Output.

The BIOS is that part of the operating system that is concerned with Input/Output. The BIOS for the DIMENSION 68000 system uses the ROMBIOS as much as is possible. The CP/M BIOS includes a table that handles diskettes and also handles the RS-232 port. The label that has been assigned to that table is "drivetb". Following is a description of the "drivetb" table.

TABLE OFFSETS

OFFSET	LENGTH	DESCRIPTION
+0	1 byte	Number of Disk types
+1	1 byte	DUART reset RS-232 command
+2	1 byte	DUART select RS-232 command
+3	1 byte	DUART set 9600 bits per second
+4	1 byte	DUART set no parity/8 bits
+5	1 byte	DUART set normal/1 stop bit
+6	1 byte	RS-232 XON/XOFF flag
+7	1 byte	Old Track Number
+8	1 byte	Old Physical Sector Number
+9	1 byte	New Track Number
+10	1 byte	New Sector Number
+11	4 bytes	New DMA Address
+15	1 byte	Old Disk Unit
+16	1 byte	Retry Count
+17	4 bytes	Save Location for the D1 register
+20	1 byte	Save Location
+21	1 byte	Save Location
+22	2 bytes	Drive
+24	2 bytes	Warm Boot Flag
+26	2 bytes	Number of Memory Regions = 1
+28	4 bytes	Starting Address of the Memory Region(s)
+32	4 bytes	Ending Address of the Memory Region(s)

ADDITIONAL CALLS

The DIMENSION 68000 version of CP/M has three functions added to the normal CP/M calls. The functions are:

256 - Change Diskette Drive Format

Entry Parameters:

Register D0.W = 100H = 256 (decimal)

Register D1.W = New Disk Drive Unit Number

where 0 = Drive A:

1 = Drive B:

2 = Drive C:

3 = Drive D:

4 = Drive E:

5 = Drive F:

6 = Drive G:

7 = Drive H:

Register D2.W = New Disk Drive Format

where 0 = null

1 = Std 40 track

2 = Std 80 track

3 = IBM single side

4 = IBM double side

5 = TRS 80

6 = KayPro

7 = Cromemco

8 = Osborne

9 = IBM 3740 8 Inch Single Side, Density

A = TRS 16 Double Side, Density

B = User Defined

Returned Values:

Register D0.W

257 - Read Diskette Drive Format

Entry Parameters:

Register D0.W = 101H = 257 (decimal)

Register D1.W = Disk Unit Number

where 0 = Drive A:

1 = Drive B:

2 = Drive C:

3 = Drive D:

4 = Drive E:

5 = Drive F:

6 = Drive G:

7 = Drive H:

Returned Values:

Register D0.W

258 - Set the Physical Parameter Table,
 Set the CP/M Disk Parameter, or
 Set the Disk Interlace Table

 Entry Parameters:

Register D0.W = 102H = 258 (decimal)

Register D1.W = Command Select

where 0 = Ignore

1 = Set Physical Parameters
 (DPRMTX 16 words)

2 = Set CP/M Disk Parameter Block Table
 (DPBX 8 words)

3 = Set Interlace Table
 (XLTX 64 words)

Register D2.L = Table Address

Returned Values:

Register D0.W

TERMINAL ATTRIBUTES

The DIMENSION 68000 CP/M BIOS has been augmented to provide a set of "ESCAPE SEQUENCES" that will handle various screen functions on the CRT display. Also, several "CONTROL KEY" screen functions have been implemented. The following charts list the "ESCAPE SEQUENCES" and the "CONTROL KEY" functions.

CONTROL KEYS

HEX VALUE	KEYS TO PRESS		FUNCTION
B1	Alt	1	Clear Screen
07	F8		Bell
B7	Alt	7	Home
91	Alt	End 1	Cursor Up
92	Alt	Dwn 2	Cursor Left
93	Alt	Pgd 3	Cursor Right
94	Alt	Lft 4	Cursor Down (Line Feed)

To generate the escape code (27Hex) Press CTRL plus the '"' key

ESCAPE SEQUENCES (Escape plus the key value listed)

HEX VALUE	KEYS TO PRESS	FUNCTION
48	H	Home Cursor
3D	=	Position Cursor
76	v	Clear Screen & Home Cursor
41	A	Cursor Up
42	B	Cursor Down
43	C	Cursor Right
44	D	Cursor Left
4B	K	Erase End of Line
4A	J	Erase End of Page
78	x	Clear End of Line
79	y	Clear End of Page
63	c	Turn On Cursor Flag
64	d	Turn Off Cursor Flag
65	e	Turn On Inverse Flag
66	f	Turn Off Inverse Flag

USAGES

The CP/M call that is used to "Change Diskette Drive Format" is one of the functions that is used by the "RESET" utility program to set the disk formats. The additional calls that have been added to the DIMENSION 68000 CP/M system are accessed in the same way as the standard CP/M function calls.

I N D E X

Backup
BASIC
Central Processor
Connecting Dimension Together
Connectors
Copy
CP/M-68K
Creating Backup Disks
CRT
Cursor
DIMENSION
Disk Drive Configurations
Disk Drive Expansion
Disk Test
Emulators
 6512
 8086
 Z80
Expansion Slots
Firmware
Formatting New Disks
FORTRAN 77+
Installation Checklist
I/O Devices
Keyboard
Languages
Location
Making Disk Copies
MC68000L8
Monitor
Motherboard
OFF/ON Switch
Power
Power Supply Assembly
Program Loading
RESET Switch
ROMBIOS
Startup
System Unit
Test
Unpacking
VDU

A P P E N D I X A

T h e T O O L - K I T

Some users of the DIMENSION 68000 require detailed information on the circuitry, on the software used in the CP/M BIOS, or on the software used in the ROMBIOS. For these users, the Micro Craft Corporation provides a "Tool-Kit" that contains the following items:

- A complete set of the schematic diagrams for the Main Printed Circuit Board of the DIMENSION 68000 computer.
- A diskette that contains the following items:
 - A copy of the source code for the CP/M BIOS.
 - A copy of the source code for the ROMBIOS.

The Micro Craft Corporation wishes to encourage other parties that wish to manufacture equipment for the DIMENSION 68000. In order to provide that encouragement, the Micro Craft Corporation will license the use of the design information for the DIMENSION 68000 and the software used in the DIMENSION 68000. The license for this use will be ROYALTY FREE.

To arrange for a license for the use of the design information and the software for the DIMENSION 68000, contact the Micro Craft Corporation at the following address.

MICRO CRAFT CORPORATION
OEM SALES
4747 IRVING BLVD.
DALLAS, TX. 75247

Also, all purchasers of the "Tool-Kit" will be notified of any updates to the software or major changes to the hardware. Copies of these changes will be made available to the purchasers of the "Tool-Kit" for a small SHIPPING AND HANDLING FEE.

If your application for the DIMENSION 68000 requires any of the following items:

- A copy of the schematic diagram
- A copy of the listing for the ROMBIOS
- A copy of the listing for the CP/M BIOS

These items are available from the Micro Craft Corporation as "The Tool-Kit" for \$99.95 (Residents of Texas add 6 % sales tax). To order "The Tool-Kit", please sign the attached Non-Disclosure Agreement, fill out and mail this form.

- - - - - F O L D H E R E - - - - -

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone__ (____) _____

- - - - - F O L D H E R E - - - - -

I certify that I will hold as confidential, all proprietary information that I receive from Micro Craft Corporation, and will not divulge that information to anyone without written permission, nor use that information for any purpose not in the best interests of the Micro Craft Corporation.

Date

Signature

MICRO CRAFT CORPORATION
CUSTOMER SERVICE DEPARTMENT
4747 IRVING BLVD.
DALLAS, TX. 75247

A P P E N D I X B

D I M E N S I O N K E Y B O A R D C O D E S

KEYBOARD CODES

Key	Un-shift	Shft	Cntl	Cntl/Shft	Alt	Alt Shft	Alt Cntl	Alt Cntl/Shft
A	61	41	C1	C1	E1	C1	C1	C1
B	62	42	C2	C2	E2	C2	C2	C2
C	63	43	C3	C3	E3	C3	C3	C3
D	64	44	C4	C4	E4	C4	C4	C4
E	65	45	C5	C5	E5	C5	C5	C5
F	66	46	C6	C6	E6	C6	C6	C6
G	67	47	C7	C7	E7	C7	C7	C7
H	68	48	C8	C8	E8	C8	C8	C8
I	69	49	C9	C9	E9	C9	C9	C9
J	6A	4A	CA	CA	EA	CA	CA	CA
K	6B	4B	CB	CB	EB	CB	CB	CB
L	6C	4C	CC	CC	EC	CC	CC	CC
M	6D	4D	CD	CD	ED	CD	CD	CD
N	6E	4E	CE	CE	EE	CE	CE	CE
O	6F	4F	CF	CF	EF	CF	CF	CF
P	70	50	D0	D0	F0	D0	D0	D0
Q	71	51	D1	D1	F1	D1	D1	D1
R	72	52	D2	D2	F2	D2	D2	D2
S	73	53	D3	D3	F3	D3	D3	D3
T	74	54	D4	D4	F4	D4	D4	D4
U	75	55	D5	D5	F5	D5	D5	D5
V	76	56	D6	D6	F6	D6	D6	D6
W	77	57	D7	D7	F7	D7	D7	D7
X	78	58	D8	D8	F8	D8	D8	D8
Y	79	59	D9	D9	F9	D9	D9	D9
Z	7A	5A	DA	DA	FA	DA	DA	DA
!	31	21	31	21	B1	A1	B1	A1
@	32	40	32	C0	B2	C0	B2	C0
#	33	23	33	23	B3	A3	B3	A3
\$	34	24	34	24	B4	A4	B4	A4
%	35	25	35	25	B5	A5	B5	A5
^	36	5E	36	DE	B6	DE	B6	DE
&	37	26	37	26	B7	A6	B7	A6
*	38	2A	38	2A	B8	AA	B8	AA
(39	28	39	28	B9	A8	B9	A8
)	30	29	30	29	B0	A9	B0	A9

Key	Un- shift	Shft	Cntl	Cntl/ Shft	Alt	Alt Shft	Alt Cntl	Alt Cntl/Shft
-	2D	5F	DF	DF	AD	DF	DF	DF
=+	3D	2B	3D	2B	BD	AB	BD	AB
{	5B	7B	DB	DB	DB	FB	DB	DB
}	5D	7D	DD	DD	DD	FD	DD	DD
~	60	7E	60	7E	E0	FE	E0	FE
:	3B	3A	3B	3A	BB	BA	BB	BA
"	27	22	27	22	A7	A2	A7	A2
\	5C	7C	DC	DC	DC	FC	DC	DC
,<	2C	3C	2C	3C	AC	BC	AC	BC
.>	2E	3E	2E	3E	AE	BE	AE	BE
/?	2F	3F	2F	3F	AF	BF	AF	BF
SPC	20	20	20	20	A0	A0	A0	A0
INS-0	10	0A	E0	E0	90	8A	E0	E0
DEL-	8B	0B	7F	7F	8B	8B	FF	FF
END-1	11	0C	BE	BE	91	8C	BE	BE
DWN-2	12	0D	FE	FE	92	8D	FE	FE
PGD-3	13	0E	BF	BF	93	8E	BF	BF
LFT-4	14	0F	AC	AC	94	8F	AC	AC
5	15	1A	BA	BA	95	9A	BA	BA
RGT-6	16	1B	BB	BB	96	9B	BB	BB
HOM-7	17	1C	FB	FB	97	9C	FB	FB
UP-8	18	1D	FD	FD	98	9D	FD	FD
PGU-9	19	1E	FC	FC	99	9E	FC	FC
-	2D	2D	2D	2D	AD	AD	AD	AD
+	2B	2B	2B	2B	AB	AB	AB	AB
RTRN	CD	CD	CA	CA	CD	CD	CA	CA
BKSPC	C8	C8	7F	7F	C8	C8	FF	FF
ESC	DB	DB	DB	DB	DB	DB	DB	DB
BRK	8C	8D	8D	8D	8C	8D	8D	8D
PRT-*	2A	1F	BC	BC	AA	9F	BC	BC
TAB	C9	AA	AB	AB	C9	AA	AB	AB
ENTR	8A	8A	8A	8A	8A	8A	8A	8A

Key	Un- shift	Shft	Cntl	Cntl/ Shft	Alt	Alt Shft	Alt Cntl	Alt Cnt/Shft
F1	00	A0	9A	9A	80	A0	9A	9A
F2	01	A1	9B	9B	81	A1	9B	9B
F3	02	A2	9C	9C	82	A2	9C	9C
F4	03	A3	9D	9D	83	A3	9D	9D
F5	04	A4	9E	9E	84	A4	9E	9E
F6	05	A5	9F	9F	85	A5	9F	9F
F7	06	A6	AE	AE	86	A6	AE	AE
F8	07	A7	AF	AF	87	A7	AF	AF
F9	08	A8	8E	8E	88	A8	8E	8E
F10	09	A9	8F	8F	89	A9	8F	8F

A P P E N D I X C

A S C I I C O D E S

American Standard Code for Information Interchange

A S C I I C H A R A C T E R S E T

(7 - B I T C O D E)

MSD	0	1	2	3	4	5	6	7
	000	001	010	011	100	101	110	111
LSD	-----+-----+-----+-----+-----+-----+-----+-----							
0 0000	NUL	DLE	SP	0	@	P	`	p
1 0001	SOH	DC1	!	1	A	Q	a	q
2 0010	STX	DC2	"	2	B	R	b	r
3 0011	ETX	DC3	#	3	C	S	c	s
4 0100	EOT	DC4	\$	4	D	T	d	t
5 0101	ENQ	NAK	%	5	E	U	e	u
6 0110	ACK	SYN	&	6	F	V	f	v
7 0111	BEL	ETB	'	7	G	W	g	w
8 1000	BS	CAN	(8	H	X	h	x
9 1001	HT	EM)	9	I	Y	i	y
A 1010	LF	SUB	*	:	J	Z	j	z
B 1011	VT	ESC	+	;	K	[k	{
C 1100	FF	FS	,	<	L	\	l	
D 1101	CR	GS	-	=	M]	m	}
E 1110	SO	RS	.	>	N	^	n	~
F 1111	SI	US	/	?	O	_	o	DEL

A P P E N D I X D

E X P A N S I O N S L O T

P I N O U T

EXPANSION SLOT PINOUT

The following illustration identifies the pin locations and pin signatures of the edge connectors (* = Low TRUE signal, 0VDC).

CONNECTOR

	Vcc	1	2	Vcc	+5VDC +,- 1%
	Vcc	3	4	Vcc	@ 1A Max/Pin
	RST*	5	6	R/W*	(slot max 3A)
	MEMCYC*	7	8	CRTREQ*	
	D0	9	10	D1	
	D2	11	12	D3	
Data Bus	D4	13	14	D5	
	D6	15	16	D7	
	D8	17	18	D9	
	D10	19	20	D11	
	D12	21	22	D13	
	D14	23	24	D15	
	LOWC*	25	26	DRSEL1*	
Disk	2SIDE*	27	28	HLOAD*	
Drive Exp.	INDEX*	29	30	HEAD*	
Cnt'l	STEP*	31	32	DIRC*	
	WENA*	33	34	WDATA*	
	RDATA	35	36	TK00*	
	DRSEL0*	37	38	WPROT*	
	LDS*	39	40	ACK0*	
	DISPEN*	41	42	E	
	INTRP*	43	44	EMULATION STROBE	
	VMA*	45	46	HLT*	
	F1	47	48	F2	
	(A0) UDS*	49	50	F0	
	A22	51	52	A23	
	A20	53	54	A21	
	A18	55	56	A19	
Address Bus	A16	57	58	A17	
	A14	59	60	A15	
	A12	61	62	A13	
	A10	63	64	A11	
	A8	65	66	A9	
	A6	67	68	A7	
	A4	69	70	A5	
	A2	71	72	A3	
	-12 VOLTS	73	74	A1	
	S1-6*	75	76	ACK1-6*	
	+12 VOLTS	77	78	+12 VOLTS	
	INT1-6*	79	80	DATAVAL*	
	14.31818 MHZ	81	82	BD1-6ENA*	
	GROUND	83	84	GROUND	
	GROUND	85	86	GROUND	

- 25 - 38 DISK CONTROL
14 control and data lines connected directly to the mini-floppy interface connector. The built-in System Disk Controller may be disabled and replaced by another Expansion Slot resident Controller (under software control). This feature permits the usage of special purpose Disk Controllers.
- 25 LCUR* LOWCURRENT*
When the Controller is in either a Read or a Write Mode, this signal will cause the lowering of the write current, when this signal is set true (0VDC). This signal is for use on the inner tracks of a diskette only.
- 26 DRSEL1* DRIVE 1 SELECT*
Selects drive number 1, when set true (0VDC).
- 27 2SIDE* DISK TWO SIDED*
When a drive has been selected and the diskette is spinning, this signal will either be set true (0VDC), for two sided media, or this signal will be set false (+5VDC), for single sided media.
- 28 HLOAD* HEAD LOAD*
When set true (0VDC), causes the currently addressed disk drive head to load.
- 29 INDEX* INDEX*
This pin supplies a true signal (0VDC), each time an index or sector hole is detected by the index/sector photo-detector. This pin is set false (+5VDC) when no index or hole is sensed. When soft sector media is used there will be one pulse (200ms) per revolution. For soft sector media, this pulse is used to indicate the physical beginning of a track.
- 30 HEAD* HEAD*
Selects head 1 when the pin is true (0VDC) and head 0 when false (+5VDC).

- 31 STEP* STEP*
 Moves the head assembly in the direction defined by the DISK DIRECTION pin (pin 32). The signal must be true (0VDC), for a minimum pulse width of 1 microSecond, followed by a false (+5VDC), for a minimum pulse width of 5 milliSeconds between adjacent pulses. The access motion is initiated on the transition from logical 0 to logical 1, or at the trailing edge of the pulse. Any change in the signal on the DISK DIRECTION pin (pin 32) must be made, at least, 1 microSecond before the trailing edge of the DISK STEP pulse. Further information on the timing relationships of these signals is available in the Shugart SA455/465 Double-sided Minifloppy (TM) Diskette Storage Drive OEM Manual.
- 32 DIRC* DISK DIRECTION*
 This pin defines the direction of motion for the head assembly when the DISK STEP (pin 30) pin is pulsed. Either a false (+5VDC) or open circuit defines the direction as out. With this condition established, the motion of the head assembly will be away from the center of the disk. Conversely, if this pin is true (0VDC), the direction of motion is inward. With this condition established, the motion of the head assembly will be toward the center of the disk.
- 33 WENA* WRITE ENABLE*
 A true (0VDC) condition enables the Write Data Function, permitting data to be written to the disk. A false (+5VDC) on this pin enables the Read Data Function, permitting data to be read from the disk.

- 34 WDATA* WRITE DATA*
This pin carries the data that is to be written on the disk. Each transition, from true (+5VDC) to false (0VDC), will cause a current reversal to the head, causing a bit to be written. This pin is enabled when DISK WRITE ENABLE (pin 33) is true (0VDC). The DISK WRITE ENABLE pin must be false (+5VDC) during a Read Data Function. Transitions the WRITE DATA pin should not occur until 8 micro Seconds after the DISK WRITE ENABLE goes true. Similarly, transitions on the WRITE DATA pin must stop 8 micro Seconds before the DISK WRITE ENABLE goes false. Further information on the timing of these signals is available in the Shugart SA455/465 Double-sided Minifloppy (TM) Diskette Storage Drive OEM Manual.
- 35 RDATA READ DATA
This pin provides the raw disk data (data and clock), as detected by the disk head. Normally this pin is false (+5VDC) and changes to true (0VDC) when active. Transitions on the READ DATA pin will not be valid until 2 micro seconds after the drive has been selected. Valid READ DATA can continue up to 1 milli second after the DISK WRITE ENABLE pin goes to 0V. A detailed explanation of the timing relationships for these signals is available in the Shugart SA455/465 Double-sided Minifloppy (TM) Diskette Storage Drive OEM Manual.
- 36 TK0* TRACK 0*
This pin is true (0VDC) when the head assembly is positioned at track 0 (outermost track) and the access circuitry is driving current through phase A of the positioner (stepper) motor. This pin is false (+5VDC) when the head assembly is not positioned at track 0. However, the TRACK 0 signal will go to a logical 0. this reaction is caused by the positioner motor changing from phase A to phase C. One additional Step Out Pulse (on pin 31) will change the positioner motor back to phase A and the TRACK 0 * pin will again go active.
- 37 DRSEL0* DRIVE 0 SELECT*
This pin selects drive 0 when true (0VDC).

38 WPROT* WRITE PROTECT*
 Provided by the disk drive to give the user an indication when a Write Protected disk is used. The pin is true (0VDC) when the disk is Write Protected and false (+5VDC) when the disk is not protected. Under normal operation, the drive will inhibit the Write Function when a Write Protected disk is used, in addition to notifying the interface.

39 LDS* LOWER DATA STROBE*
 Used in conjunction with the UPPER DATA STROBE (pin 49) to control data on the Data Bus, as shown in the table below.

The abbreviations used below are:
 Undef = Undefined, Valid = Valid data bits
 (D8-D15 for bits 8-15, D0-D7 for bits 0-7)

UDS*	LDS*	R/W	D8-D15	D0-D7
+5V	+5V	---	Undef	Undef
0V	0V	+5V	Valid	Valid
+5V	0V	+5V	Undef	Valid
0V	+5V	+5V	Valid	Undef
0V	0V	0V	Valid	Valid
+5V	0V	0V	Undef	Valid
0V	+5V	0V	Valid	Undef

40 ACK0* ACKNOWLEDGE 0*
 True (0VDC) when the MC68000 has control of the Data and Address Busses. This signal is the equivalent of a MEMORY ACKNOWLEDGE * signal to the Expansion Slots.

41 DISPEN* DISPLAY ENABLE*
 True (0VDC) when the CRT controller circuitry is not in a horizontal or a vertical blanking or retrace time (i.e. when the CRT controller circuitry is in the active display area on the screen).

42 E E SIGNAL
 The standard enable signal, common to all MC6800 type peripheral devices. The period of the output, of this pin, is ten (10) MC68000 clock periods; consisting of 6 low clocks and 4 high clocks. The MC68000 uses a 7.15909 MHz clock (one-half of a 14.31818 MHz clock).

43 INTRP* INTERRUPT*
 True (0VDC) when any interrupt request (INT1*,
 INT2*, INT3*, INT4*, INT5*, or INT6*) is as-
 serted.

44 EMULATION STROBE
 When active (+5VDC), this signal halts the co-
 processor whenever that processor has just
 accessed a memory location that has been
 tagged by the Emulation Control circuitry. The
 procedure to set up these Emulation Traps is
 as follows:

- 1 - Set the addresses that are to be trapped
 by writing, to address FFD403, a byte that
 has data bit 0 (which is the least signif-
 icant bit) set to 1.

Then, read from the address in memory of
each word whose address is to be trapped.
The addresses to be read must be in the
lowest 512K bytes of memory (i.e. in the
address range of from 000000 to 07FFFF).

When all the locations have been read that
are to be trapped, then write, to address
FFD401, a byte that contains all zeroes.

- 2 - Set the addresses that are not to be trap-
ped by writing, to address FFD403, a byte
that has data bit 0 set to 0.

Then, read from the address in memory of
each word whose address is not to be trap-
ped.

When all the locations have been read that
are not to be trapped, then write, to ad-
dress FFD401, a byte that contains all ze-
roes.

**** NOTICE ****

Once a write to location FFD403 has occurred, the emulation trapping system will be activated. And ANY access to a memory location (by any source - program, CRT controller, DMA device - any source - will set that memory location to the TRAPPED or NOT-TRAPPED condition (according to the byte that was written). The emulation trapping will remain activated until a write (of all zeroes) to location FFD401 occurs.

When a TRAP on a memory location is enabled, any access to that memory location will cause the EMULATION STROBE line to be pulsed to a high (+5 logic level) so as to stop the co-processor. The emulation routine will have to determine the address that was trapped by the steps that follow:

- 1 - Read, from address FFD400, the LSW (Least Significant Word) of the address of the location that was trapped.
- 2 - Read, from address FFD402, the MSB (Most Significant Byte) of the address of the location that was trapped.

After execution of the above steps, the emulation routine will have the 24 bit address of the location in memory that was trapped. The emulation routine will then have to determine the appropriate action to take. After the routine has taken the appropriate action, the routine must re-start the co-processor.

45

VMA* VALID MEMORY ADDRESS*

Used to signal the presence of a valid address, on the Address Bus, and that the processor is synchronized to the E SIGNAL (pin 42). This signal is used for MC6800 peripheral devices.

46 HLT* HALT*
 A bi-directional signal that is driven by an (open collector output) external device and causes the MC68000 to stop at the completion of the current bus cycle. After the processor stops executing instructions, as in a double bus fault condition, the HALT pin is driven by the processor, to indicate to the external devices that the processor has stopped.

47 F1, pin 48 F2, pin 50 F0

These pins indicate the status and the cycle type currently being executed by the MC68000. The table below defines the relationships, which are valid whenever the ADDRESS STROBE * is active (0VDC).

F2	F1	F0	Cycle Type
0	0	0	undefined
0	0	1	User Data
0	1	0	User Program
0	1	1	undefined
1	0	0	undefined
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

49 UDS* UPPER DATA STROBE*

Refer to the LOWER DATA STROBE (pin 39) description.

51 - 72, and 74 A0 - A23 ADDRESS BUS

This bus is a tri-state, 23 bit wide bus. The connected devices should be at tri-state high impedance until such time as they have been granted a Memory Access Cycle.

73 -12 Volts DC

-12 VDC +,- 5% @ 1.0 Amp Max/Slot

- 75 MREQ* MEMORY REQUEST*
 True (0VDC) when ever a memory cycle is being initiated and is held low 0 until the cycle is completed. Bus conflicts may occur if the CRT is not allowed to have regular memory requests granted (every 1.1 us) so it may be desirable to synchronize MEMORY REQUEST with CRT REQUEST. Refer to the following diagram for an illustration of the relationships between MEMORY REQUEST * (pin 75), MEMORY ACKNOWLEDGE * (pin 76), and DATA VALID * (pin 80). Each Expansion Slot has an individual MEMORY REQUEST * lead. Slot number 6 has the highest priority, and slot 1 has the lowest priority.
- 76 ACK1* MEMORY ACKNOWLEDGE 1 *
 ACK2* MEMORY ACKNOWLEDGE 2 *
 ACK3* MEMORY ACKNOWLEDGE 3 *
 ACK4* MEMORY ACKNOWLEDGE 4 *
 ACK5* MEMORY ACKNOWLEDGE 5 *
 ACK6* MEMORY ACKNOWLEDGE 6 *
 This signal, when true (0VDC) indicates to the slot that it has been granted a memory cycle and therefore has control of the address bus and the data bus. This signal is useful to gate address information onto the address bus and data onto the data bus. Each slot has an individual MEMORY ACKNOWLEDGE * pin.
- 77 - 78 +12 Volts DC
 +12 VDC +,- 5% @ 1 Amp Max/Pin 1.5 Amps Max per Expansion Slot
- 79 INT1* INTERRUPT 1 *
 INT2* INTERRUPT 2 *
 INT3* INTERRUPT 3 *
 INT4* INTERRUPT 4 *
 INT5* INTERRUPT 5 *
 INT6* INTERRUPT 6 *
 This signal is an input to the bus from a specific slot. This signal is used to interrupt the MC68000. Each slot has an individual interrupt line. Slot 6 has the highest priority and slot 1 has the lowest.
- 80 DATAVAL8 DATA VALID *
 This common signal, to all Expansion Slots and the CRT and the MC68000, is meaningful only if MEMORY REQUEST * and MEMORY ACKNOWLEDGE * are also true (0VDC). This signal indicates that the data being written into or read from memory is valid.

81 14.31818 MHz
 This clock is buffered and is used by the
 68000 circuitry to derive the 7 MHz clock that
 is used by the 68000 processor.

82 BD1ENA* BOARD 1 ENABLE *
 BD2ENA* BOARD 2 ENABLE *
 BD3ENA* BOARD 3 ENABLE *
 BD4ENA* BOARD 4 ENABLE *
 BD5ENA* BOARD 5 ENABLE *
 BD6ENA* BOARD 6 ENABLE *
 This signal goes active (0VDC) whenever the
 memory address range for the specific slot is
 addressed. The address ranges for each slot
 are as follows:

SLOT 1 = FF8800 to FF8FFF
SLOT 2 = FF9000 to FF97FF
SLOT 3 = FF9800 to FF9FFF
SLOT 4 = FFA000 to FFA7FF
SLOT 5 = FFA800 to FFAFFF
SLOT 6 = FFB000 to FFB7FF

When this signal goes active, then the addresses for this slot may be read, written, or executed as required.

83 - 86 GROUND GROUND reference for system.

A P P E N D I X E

D I R E C T I / O

Direct I/O Description

This section describes the various I/O commands available .

ADDRESS READ/WRITE FUNCTION

The following abbreviations are used throughout this section:

- Read Word = RW
- Read Byte = RB
- Write Word = WW
- Write Byte = WB
- Hor = Horizontal
- Vert = Vertical
- Char = Character
- B/W = Black and White
- T/G = Text/Graphics

CRT Controller

The CRT controller circuitry is based on the MC6845 chip.

FF8001	WB		Write CRT Controller Register Selection Input (controls the Address Register for the CRT Controller)
		CRT Cntl'r Register	CPU Register
			CRT Controller Address Register
		R0	00 Hor Char Total
		R1	01 Hor Char Displayed
		R2	02 Hor Sync Position
		R3	03 Hor Sync Width
		R4	04 Vert Char Rows Total
		R5	05 Vert Scan Line Adj
		R6	06 Vert Char Rows Displayed
		R7	07 Vert Sync Position
		R8	08 Interlace Mode
		R9	09 Number of Scan lines/Char -1
		RA	0A T/G Mode Scan Start
		RB	0B T/G Mode Scan Stop
		RC	0C Buffer Start Address Mid Byte
		RD	0D Buffer Start Address Low Byte
		RE	0E T/G Change Char Add. Hi Byte
		RF	0F T/G Change Char Add. Low Byte
		R10	10 Light Pen Char Add. Hi Byte
		R11	11 Light Pen Char Add. Low Byte
FF8003	RB		Read Data from the Register Addressed (Addressed by writing to FF8001)
FF8003	WB		Write Data to the Register Addressed (Addressed by writing to FF8001)
FF8005	WB		Write High Byte of Address in Memory of Start of Display Buffer

CRT Controller (cont'd)

FF8009

WB

Write Mode Register

Bit 7 (MSB)

- 0 = HIRES GRAPHICS
- 1 = LORES/TEXT

Bit 6

- 0 = 8 dot wide character on screen
- 1 = 7 dot wide character on screen

Bit 5

- 0 = CRT Controller operates ASYNCHRONOUSLY with CPU
- 1 = CRT Controller operates SYNCHRONOUSLY with CPU

Bits 4 and 3

- 00 = 14 MHz dot clock
- 01 = 3.58 MHz dot clock
- 10 = 7 MHz dot clock
- 11 = 1.79 MHz dot clock

Bit 2

- 0 = Screen Display ON
- 1 = Screen Display OFF

Bit 1

- 0 = Standard Text Character Set and LORES
- 1 = Alternate Text Character Set and HIRES

Bit 0 (LSB)

- 0 = Non-Mixed Screen Mode (i.e. all TEXT or all Graphics)
- 1 = Mixed Screen Mode (i.e. Color Graphics and Monochrome TEXT)

MODE DESCRIPTION

- 44 Off (Refresh Only)
- A0 Hi Res Text (80 Column)
- 21 Hi Res Text/Graphics
- 90 Med Res Text (40 Column)
- 91 Med Res Text/Graphics
- 20 Hi Res Graphics
- 21 Med Res Graphics (B/W)
- XX Lo Res Graphics (B/W)
- XX Lo Res Text/Graphics (B/W)

CRT Controller (cont'd)

FF800A RB Reset Controller

Procedure:

The following steps are given as an example of the usage of the CRT controller:

- Execute a loop to do the following sequence of steps 18 times:
 - Write to the CRT Controller Address Register (FF8001).
 - Write the Data that goes to the Register Addressed.
- Write the High Order Byte of the Address (in memory) of the Start of the Display Buffer to the High Order Byte Address Register (FF8005).
- Write the mode desired to the Mode Register (FF8009).

WARNING

Data may be destroyed in the text or graphics buffer area when the setup procedure is executed.

Typical Parameters (hex)

CRT CTRL REGISTER	40 Col/24Ln	80 Col/25Ln	80 col/48Ln	92 col/24Ln
R0	04	00	00	00
R1	37	6F	6F	6F
R2	28	50	50	50
R3	2E	5A	5A	5A
R4	1E	1E	1E	1E
R5	00	00	00	00
R6	18	19	18	18
R7	1B	1B	1B	1B
R8	00	00	00	00
R9	08	08	09	09
RA	00	00	00	00
RB	01	01	01	00

Peripheral Slot Selects

FF8800 - FF8FFF	RW/BW	Read Controller ROM in slot 1
FF9000 - FF97FF	RW/BW	Read Controller ROM in slot 2
FF9800 - FF9FFF	RW/BW	Read Controller ROM in slot 3
FFA000 - FFA7FF	RW/BW	Read Controller ROM in slot 4
FFA800 - FFB7FF	RW/BW	Read Controller ROM in slot 5
FFB000 - FFB7FF	RW/BW	Read Controller ROM in slot 6
FFB800 - FFBFFF		RESERVED

Above are the memory addresses that are reserved for each controller. As each group of addresses are accessed, then the ENABLE line (for the slot that corresponds to the memory addresses accessed) will go active (TTL logic LOW). When the ENABLE line goes active, then the addresses for the slot may be read, written, or executed as required by each individual controller. NOTICE that the first seven words are reserved as System Information and must be implemented as follows:

RELATIVE SYSTEM INFORMATION
ADDRESS FUNCTION DESCRIPTION

000	16 Bit MANUFACTURER'S CODE
002	16 Bit PRODUCT SUBCODE
004	8 Character Controller Description (ASCII)
008	
00C	I/O VECTOR

The data is read 1 byte at a time, in the byte mode.

The Manufacturer's Code is assigned by Micro Craft and uniquely identifies the peripheral manufacturer, for automatic configuration purposes.

The Subcode is defined by the manufacturer to identify the particular model, options or devices attached.

The 8 byte ASCII data is manufacturer definable and will be displayed when the System is "Booted", to identify the particular controller that is installed.

If the I/O Vector is used, it points to the local (Controller Resident) standard I/O routine start address. If the Vector is not used, a null vector is placed in the I/O Vector location. A null vector is defined as FFFF or 0000 (hex). All accesses to the Controller addresses (in the slot) use the memory timing mode; that is, data transferred into and out of the boards must accept or present data in 250 nanoSeconds from BOARD ENABLE.

Keyboard, Printer, RS-232, and Real Time Clock

The Keyboard, the Printer, the RS-232, and the Real Time Clock (RTC) are all connected into the DIMENSION 68000 system through a Signetics (or equivalent) SCN2681AC1N40 Dual Universal Asynchronous Receiver / Transmitter (DUART) chip. For a detailed description of the operation of the Signetics SCN2681 DUART chip, see the Signetics Dual Asynchronous Receiver / Transmitter (DUART) data sheet. The registers on the DUART are all one byte in width, and they are addressed as follows:

Address	Description for Read	Description for Write
	** Channel A is the KEYBOARD	**
FFC401	Mode Register A (MRLA,MR2A)	Mode Register A (MRLA,MR2A)
FFC403	Status Register A (SRA)	Clock Select Register A (CSRA)
FFC405	* RESERVED *	Command Register A
FFC407	Recv. Holding Reg. A (RHRA)	Transmit Holding Reg. A (THRA)
	** Input Port has the Input Lines	**
	** IP4 = RS-232 Data Carrier Detect	**
	** IP1 = Printer ACKNLG*	**
	** IP0 = Printer READY*	**
FFC409	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
FFC40B	Interrupt Status Reg. (ISR)	Interrupt Mask Register (IMR)
	** Counter/Timer is the Real Time Clock	**
FFC40D	Counter/Timer Upper (CTU)	Counter/Timer Upper Reg. (CTUR)
FFC40F	Counter/Timer Lower (CTL)	Counter/Timer Lower Reg. (CTLR)
	** Channel B is the RS-232	**
FFC411	Mode Register B (MRLB,MR2B)	Mode Register B (MRLB,MR2B)
FFC413	Status Register B (SRB)	Clock Select Register B (CSRB)
FFC415	* RESERVED *	Command Register B (CRB)
FFC417	Recv. Holding Reg. B (RHRB)	Transmit Holding Reg. B (THRB)
FFC419	* RESERVED *	* RESERVED *
	** Input Port has the Input Lines	**
	** IP4 = RS-232 Data Carrier Detect	**
	** IP1 = Printer ACKNLG*	**
	** IP0 = Printer READY*	**
FFC41B	Input Port	Output Port Config. Reg. (OPCR)
	** Counter/Timer is the Real Time Clock	**
	** Output Port is the Printer	**
FFC41D	Start Counter Command	Set Output Port Bits Command
FFC41F	Stop Counter Command	Reset Output Port Bits Command

The DIMENSION 68000 system does not use interrupts in the ROMBIOS, nor are interrupts used in the CP/M-68K BIOS.

Following are descriptions of the usage of the Keyboard, the RS-232, the Real Time Clock, and the Printer.

Keyboard

FFC403 RB Read Keyboard Status
 FFC407 RB Read Keyboard Data

Procedure:

Read Keyboard Status byte (byte wide), if data bit 0 = 1 then status is not ready. When the status is ready (data bit 0 = 0), then read the keyboard data to get the character entered.

The Keyboard Controller is located in conjunction with the RS-232C Serial port, the Printer Interface and the Real Time Clock. The keyboard must be initialized according to the following parameters;

WB \$19 to \$FFC405 Set Control Register
 - Reset Mode Register Pointer
 - Disable Keybd Xmit
 - Enable Keybd Recv

WB \$44 to \$FFC403 Set 300 Baud Rate for Xmit and Recv
 WB \$F0 to \$FFC409 Set Aux Control Reg.
 - Bit Rate Generator = Set 2
 - Xtal Clock divided by 16
 - NO Delta IP bit Interrupts

WB \$13 to \$FFC401 Set Mode Register 1
 - Error Mode = Char
 - NO RTS Control
 - NO Recv Interrupt
 - NO Parity
 - 8 Bits per Character

WB \$07 to \$FFC401 Set Mode Register 2
 - Channel Mode = Normal
 - NO Xmit RTS Control
 - NO CTS Enable
 - ONE Stop Bit

RS-232

FFC409	WB	Write Aux. Control Register
FFC40B	WB	Write Interrupt Mask Register
FFC411	WB	Write RS-232 Mode Registers (1 and 2)
FFC413	WB	Write RS-232 Clock Select Register
FFC413	RB	Read RS-232 Status Register
FFC415	WB	Write RS-232 Command Register
FFC417	WB	Write RS-232 Output Data Register
FFC417	RB	Read RS-232 Input Data Register
FFC41B	RB	Read Input Port for DCD (IP4)

Procedure:

To set-up the RS-232 Port, the following steps may be used:

WB	\$00	to	\$FFC40B	Set Interrupts OFF
WB	\$F0	to	\$FFC409	Set BRG to SET 2
				Set Counter-Timer to XTAL divided by 16
				Set DELTA Interrupts OFF
WB	\$XX	to	\$FFC413	Set Baud Rate for XMIT and RECV
	\$44	=	300 BPS	
	\$55	=	600 BPS	
	\$66	=	1200 BPS	
	\$88	=	2400 BPS	
	\$99	=	4800 BPS	
	\$BB	=	9600 BPS	
	\$CC	=	19200 BPS	
WB	\$40	to	\$FFC415	Reset Error Status
				Disable XMIT and RECV
WB	\$20	to	\$FFC415	Reset RS-232 Receiver
WB	\$15	to	\$FFC415	Reset Mode Reg. Pointer
				Enable XMIT and RECV
WB	\$XX	to	\$FFC411	Set Mode Register 1
	\$02	=	7 Bits/Char, Even Parity	
	\$06	=	7 Bits/Char, Odd Parity	
	\$0A	=	7 Bits/Char, Mark Parity	
	\$0E	=	7 Bits/Char, Space Parity	
	\$12	=	7 Bits/Char, No Parity	
	\$16	=	7 Bits/Char, No Parity	
WB	\$XX	to	\$FFC411	Set Mode Register 2
	\$07	=	Normal Mode, 1 Stop Bit	
	\$47	=	Auto Echo Mode, 1 Stop Bit	
	\$87	=	Local Loopback Test Mode, 1 Stop Bit	
	\$C7	=	Remote Loopback Test Mode, 1 Stop Bit	

RS-232 (cont'd)

To operate the RS-232 Port, the following commands may be used:

If DCD (Data Carrier Detect) checking is desired, then:

```
RB $FFC41B  Get Input Port Byte
    Test to see if bit 4 is a 0, if so, DCD is ON
```

```
RB $FFC413  Get Status
    Bit 0 = RECV Ready
    Bit 1 = RECV Holding Reg. Full
    Bit 2 = XMIT Ready
    Bit 3 = XMIT Holding Reg. Empty
    Bit 4 = Overrun Error
    Bit 5 = Parity Error
    Bit 6 = Framing Error
    Bit 7 = Recvd Break
```

```
RB $FFC417  Get RECV Character
```

```
WB $FFC417  Put XMIT Character
```

Parallel Printer Port

FFC41D	WB	Write Printer Data Byte to SET Printer Port Latches
FFC41F	WB	Write Printer Data Byte to RESET Printer Port Latches
FFD801	WB	Write Printer Strobe

Procedure:

Initialize the Printer Port by writing a 00 (Hex) to the Output Port Config. Register (FFC41B).

Test the Input Port (FFC41B) to determine if the Printer is READY by examining bit 0 (IP0). If bit 0 is 0, then the Printer is ready.

Reset the Printer Port Latches by writing a FF (Hex) to the RESET Printer Port (FFC41F).

Write the data (the character) that is to be sent to the Printer to the SET Printer Port (FFC41D).

Set the STROBE bit by writing anything to the Write Printer Strobe location (FFD801). The Write Printer Strobe location causes the activation of a one-shot circuit which generates the printer strobe.

On the MAIN Printed Circuit Board (Mother Board) is a Jumper Pin Connector (J13) which allows the Parallel Printer Port to be configured so that the AUTO LINE FEED * signal can be set to a "0" (GROUND or 0V) or to a "1" (LOGIC +5). The Jumper Pins are designated pins 1, 2, and 3. Pin 2 is the middle pin. If the Jumper Plug is placed on pins 3 and 2, then the AUTO LINE FEED * is enabled and a line feed character will be inserted after every carriage return character. If the Jumper Plug is placed on pins 1 and 2, then the AUTO LINE FEED is disabled and NO line feed is inserted.

Speaker

FFC800 RB Toggle Speaker

Procedure:

Each time the speaker is toggled, the speaker cone will make a transition either in or out. Under software control the frequency can be varied between 0 Hz and 25K Hz.

Game Control

```

FFCC00    RB    Reset Timers
FFCC11    RB    Read Switch 0
FFCC13    RB    Read Switch 1
FFCC15    RB    Read Switch 2
FFCC17    RB    Read Switch 3
FFCC19    RB    Read Paddle 1
FFCC1B    RB    Read Paddle 2
FFCC1D    RB    Read Paddle 3
FFCC1F    RB    Read Paddle 4

```

Procedure:

Reading a switch byte will result in a negative number, if the switch is in the closed (pressed) position, and a positive number if the switch is in the open (released) position. To read the paddle position, first the Reset Timers command is executed, followed by reading the desired paddle. A timing loop is executed, to determine the time required for each paddle to change from a negative (not ready) to a positive (ready) number. The timing loop should be set to give a result of 0 when the timer goes positive immediately, and +255 when the timer takes the maximum time.

Example

```

*****
*
*   0 = Low Resistance (CCW) *
*  255 = High Resistance (CW) *
*
*   d2 = Paddle number to check *
*
*****
      and      #3,d2
      lsl      #1,d2
      move     #255,d1
      clr      d0
      move.l   $ffcc19,a0      bare paddle return
a:     tst.b   0(a0,d2)        wait til last timer if finished
      bmi     a
      tst.b   $ffcc00         start timer
b:     tst.b   0(a0,d2)        test if finished
      bpl     c
      nop
      nop
      nop
      nop
      add     #1,d0
      dbra   d1,b
c:     ***** d0 holds result 0 - 255 range *****

```

Floppy Disk

The controller is an NEC uPD765A, and for a detailed description refer to the NEC Data Sheets.

FFD001 RW Read Status

Reading the status returns the status of the INT (Interrupt Flag) in the Most Significant Bit (MSB = 0) is zero then INT is set, MSB = 1 is INT reset. The least significant 8 bits are the contents of the Main Status Register (contents of the disk controller).

FFD003 RB Read Data Register Byte
FFD003 WB Write Data Register Byte

Read and write the Controller Register data registers, as defined in the detailed specification sheets. The Controller may be programmed for 5 1/4" Floppies in the FM (standard) or MFM (double density) modes using non-DMA modes only.

FFD005 WB Write Control BYTE

The control byte selects the drive, turns the motor ON or OFF, and outputs the terminating character flag on or off.

BITS	FUNCTION
0	
x 0	Select Drive 0
x x 1	Select Drive 1
x 0 x	Turn Selected Drive Motor OFF and disable Disk Bus
x 1 x	Turn Selected Drive Motor ON and enable Disk Bus
0 x x	Terminating Character Flag OFF
1 x x	Terminating Character Flag ON

The Terminating Character Flag should be turned ON for one instruction time at the completion of transferring the requested number of bytes of read or write data and then turned immediately OFF. The controller will complete the read or write execution phase by itself. At the completion of the read or write execution phase, the results registers may be read.

Emulation Control

FFD400	RW	Read LSB of Trap Address
FFD401	WB	Disable Write Configuration Memory
FFD402	RB	Read MSB of Trap Address
FFD403	WB	Enable Write Configuration Memory

Procedure:

To set up the Emulation Memory;

- (1) Set the addresses that are to be trapped by writing, to address FFD403, a byte that has data bit 0 (the LSB) set to 1.

Then, read the data from each location in memory whose address is to be trapped. The addresses to be trapped must be in the lowest 512K bytes of memory (i.e. in the range from 000000 to 07FFFF).

When all of the locations have been read that are to be trapped, then write to address FFD401 a byte that contains all zeroes.

- (2) Set the addresses that are not to be trapped by writing, to address FFD403, a byte that has data bit 0 (the LSB) set to 0.

Then, read the data from each location in memory whose address is not to be trapped. These address must also be in the lower 512K bytes of memory.

When all of the locations have been read that are not to be trapped, then write to address FFD401 a byte that contains all zeroes.

When a TRAP on a memory location is enabled, an access to that memory location will cause the EMULATION STROBE line to be pulsed to a high (+5 logic level) so as to stop the co-processor. The emulation routine will have to determine the address that was trapped by the following steps:

- Read, from address FFD400, the LSW (Least Significant Word) of the address of the location that was trapped.
- Read, from address FFD402, the MSB (Most Significant Byte) of the address of the location that was trapped.

After execution of the above steps, the emulation routine will have the 24 bit address of the location that was trapped. The emulation routine then will have to determine the appropriate action to take. After the routine has taken the appropriate action, the routine must restart the co-processor.

Printer Strobe

FFD801 WB Write Printer Strobe

Procedure:

When the Write Printer Strobe location is written to (with any data), a one-shot circuit is activated which generates the printer strobe.

Memory Switch

FFDC00 RB Switch Memory and I/O Addresses

Procedure:

The MC68000 processor requires that the initial power-up reset vectors be located at locations 0000000 through 0000007. Since these are required to be in ROM for initial power-up, the ROM and I/O are initially mapped into the addresses 0000000 through 000FFFF. The initial power-up routine then moves the initialization routines to locations 0010000 up in RAM and jumps to it. At the execution of the Switch Memory command, the ROM addresses will be mapped to locations FF0000 through FF7FFF, the CRT Controller to FF8000 and Peripherals to FF9000 through FFFFFF.

A P P E N D I X F

R O M B I O S

ROMBIOS

This Section contains information on the basic software contained within the System. This software is contained in Read Only Memory (ROM) and is used by the System during start/restart operations.

ROMBIOS is the abbreviation for ROM Based Input/Output System. The contents of ROMBIOS permit the user to create programs utilizing the functions and features of the System without having to repeat the instructions necessary to perform read/write operations to the System's I/O devices.

ROMBIOS is located in the standard System ROMs. The functions available are:

- Screen Display
- Floppy Disk System
- Keyboard

These functions are interfaced through a common standardized structure with a standard address of FF0100 (hex). In each case the D0 register of the CPU is loaded with a 16 bit code which identifies the function.

Additional registers may be loaded with the parameters described below, if required. These registers are loaded before ROMBIOS is called.

ROMBIOS FUNCTION CODES (set in register D0 - word mode)

- 0 = Unused Function - immediately returns
- 1 = Initialize CRT Controller
- 2 = Output a character to the screen
- 3 = Clear the TEXT screen
- 4 = Read a key stroke
- 5 = Get keyboard status
- 6 = Boot from disk
- 7 = Start the built-in monitor
- 8 = Read selected disk drive status
- 9 = Select disk drive and turn motor ON

- 10 = Deselect disk drive and turn motor OFF
- 11 = Seek to selected disk track
- 12 = Seek to track 0 and recalibrate disk - rezero
- 13 = Seek to track and read a sector
- 14 = Seek to track and write a sector
- 15 = Seek to track and format a track
- 16 = Format the entire disk
- 17 = Read the sector ID information
- 18 = Read the disk status
- 19 = Reserved
- 20 = Reserved
- 21 = Read Printer Status
- 22 = Output a character to Printer Interface
- 23 = Position CRT cursor
- 24 = Read character at present CRT cursor location
- 25 = Clear Graphics Screen
- 26 = Set graphics color
- 27 = Plot X, Y point
- 28 = Reserved
- 29 = Read color at designated point
- 30 = Output Character to CRT with No Control Characters

The following is a detailed description of the ROMBIOS Function Codes. In the following descriptions, the contents of the registers are UNDETERMINED, unless specified. Each description provides the contents of the registers used by the Function Codes both at the beginning of the Function and at the end of the Function. The final register contents are provided after a JSR \$FF0100 or JSR ROMBIOS instruction.

Function Code 0 = Unused

Initial Register Contents:

Not Used

Final Register Contents:

Unchanged

Function:

This function returns immediately to the caller.

Function Code 1 = Initialize CRT Controller

Initial Register Contents:

D0 = 1 (word mode)

A0 = Address of CRT Table (longword mode)

Final Register Contents (after JSR \$FF0100 or JSR ROMBIOS.)

A0 = Initial Value (longword mode)

Function:

This function writes the contents of the CRT Table to the CRT Controller and switches the CRT to the requested mode. It also calculates the longword values for MAX LENGTH, LINE LENGTH, PRESENT CHARACTER POINTER and CRT BUFFER ADDRESS and stores them in the CRT Table.

Anytime this function is called, the address of the CRT Table is placed into location 0001FA. Additionally, when this function is called with the Mode Byte set for TEXT, the address of the CRT Table will be placed into location 0001EA.

Function Code 2 = Output a character to the CRT screen

Initial register contents:

D0 = 2 (word mode)

D1 = ASCII character value (output character) (word mode)

Final register contents:

Undetermined

Function:

Places the character (indicated by its ASCII value) on the screen at the next character location and moves the reverse video cursor. The PRESENT CHARACTER POINTER value is incremented by 1.

If a character exceeds the end of a line, then an automatic advance to the start of the next line is generated. If the next character position would exceed the CRT Buffer (off of the bottom of the screen), then the entire CRT Buffer contents will be scrolled upward by 1 line and the last line will be blank filled.

If the value of D0 is equal to 08 (hex) (ASCII backspace), the PRESENT CHARACTER POINTER is decremented by 1 and the character in that position is displayed in reverse video. If the backspace would result in going beyond the start of the CRT Buffer, then the PRESENT CHARACTER POINTER is not decremented.

If the character in D0 is equal to 0D (hex) (ASCII Carriage Return), the next character position will immediately move to the start of the next line.

If the CRT is in graphics mode, then the following modifications occur:

1. Reverse video cursor shown only when present character position exceeds graphics/text switch point. (text CRT parameter table +16 location)
2. When scroll is required, then scroll functions extends from bottom of the screen to the graphics/text switch point.

Function Code 3 = Clear the TEXT Screen

Initial Register Contents:

D0 = 3 (word mode)

Final Register Contents:

Undefined

Function:

Clears the TEXT screen buffer (all blanks). This routine leaves the cursor and the PRESENT CHARACTER POINTER positioned at the upper left hand corner of the screen.

Function Code 4 = Read Key From Keyboard

Initial Register Contents:

D0 = 4 (word mode)

Final Register Contents:

D0 = ASCII code for key (word mode)

Function:

This function does the following steps:

- waits until a key is pressed
- gets the ASCII code for the key
- puts the ASCII code in register D0
- returns

Function Code 5 = Get Keyboard Status

Initial Register Contents:

D0 = 5 (word mode)

Final Register Contents:

D0 = Status (word mode)

Function:

Determines the status of the keyboard and returns the READY/NOT READY Flag to the D0 register.

If a key has been pressed but not read, then the D0 register will be set to 0000 (hex) to denote keyboard ready.

If no key has been pressed since the last Read Keyboard Function, then the keyboard is not ready and D0 will be set to 00FF (hex).

Function Code 6 = Boot From Disk

Initial Register Contents:

D0 = 6 (word mode)

Final Register Contents:

Unknown

Function:

The Boot Routine reads disk sector 0, track 0 from the disk defined by the Boot Disk Parameter Table, located in addresses 0001CC to 0001F5 (hex), and loads the data read into memory starting at location 010000 (hex). At the successful completion of the read, program control is given to the instructions starting at location 010000 (hex). Normally, the first sector contains instructions to read the rest of track 0, which is the Boot Loader for the installed Operating System.

Function Code 7 = Start built-in Monitor

Initial Register Contents:

D0 = 7 (word mode)

Final Register Contents:

Unknown

Function:

This function will invoke the ROM resident Monitor. To use minimum space on the stack this function may be JUMP'ed to instead of being called. The only way to return from this mode is to have the operator reboot the System or to go to a user specified location in the calling program.

Function Code 9 = Select Disk and Turn Motor ON

Initial Register Contents:

D0 = 9 (word mode)
A0 = Address of Selected Disk Parameter Table (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

Using the contents of the Disk Parameter Table, this function selects the appropriate floppy disk drive and turns the motor ON, if required. If the contents of the table +23 = 0 (disk motor ON flag), then the motor was OFF, so turn the motor ON, set the Motor ON Flag and wait until an index mark is detected and then return. If the Motor ON Flag was set to non-zero, then immediately return.

Additional Information:

All of the disk functions require that the A0 register be set to the starting address of the Disk Parameter Table for that particular disk. Normally each disk drive in the System will have it's own Disk Parameter Table. The format for the Disk Parameter Table is given later in this Section.

Function Code 10 = Deselect Disk Drive and Turn OFF Motor

Initial Register Contents:

D0 = 10 (word mode)

A0 = Disk Parameter Table address (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function does the following:

- deselect the Disk Drive defined by the Disk Parameter Table
- turn the motor OFF
- set the MOTOR ON FLAG in the Table to OFF
- return

Function Code 11 = Seek to Selected Track

Initial Register Contents:

D0 = 11 (word mode)

A0 = Disk Parameter Table address (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function does the following:

- assumes that the selected drive motor is ON
- moves the head to the new selected track location on the selected drive.
- returns

This function is of limited value since;

- only one drive at a time can be selected
- the read or write functions automatically perform a seek to track.

Function Code 12 = Seek to Track 0 and Recalibrate (rezero)

Initial Register Contents:

D0 = 12 (word mode)
A0 = Disk Parameter Table address

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function does the following:

- assumes that the selected disk motor is ON
- rezeroes the disk controller by moving the disk head to the track 0 location.

Function Code 13 = Seek to Track and Read a Sector

Initial Register Contents:

D0 = 13 (word mode)
A0 = Disk Parameter Table address (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function does the following:

- assumes that the disk drive motor is ON
- seeks to the requested track
- gets the number of data bytes to read in from either
 - the Bytes/Sector value in the Disk Parameter Table
 - or
 - the Data Bytes/Sector value in the Disk Parameter Table
- reads the number of data bytes requested into the buffer address which is defined in the Disk Parameter Table
- loads the appropriate status bytes and disk parameters into the Disk Parameter Table.
- returns

Function Code 14 = Seek To Track and Write a Sector

Initial Register Contents:

D0 = 14 (word mode)

A0 = Disk Parameter Table address

Final Register Contents:

A0 = Initial value (longword mode)

Function:

This function assumes that the disk drive is selected and that the motor is ON. The drive will seek to the requested track and sector and then the number of user specified bytes will be written from the buffer defined in the Disk Parameter Table. At the completion of the write function, the disk status will be returned to the Disk Parameter Table.

Function Code 15 = Seek To and Format a Track

Initial Register Contents:

D0 = 15 (word mode)
D3 = Disk Interlace Table address (longword mode)
A0 = Disk Parameter Table (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function assumes that the requested disk is selected and that the motor is ON. The disk drive will seek to the requested track and after the index mark is detected will write the ID data and format data (E5 hex) in the sector. The Sector ID will be the value in the first byte in the Disk Interlace Table. At the completion of writing the previous sector, the Sector ID will be taken from the next value in the Disk Interlace Table. This sequence continues until all of the sectors have been written or until another index mark is detected. The use of the Sector Interlace Table allows the logical sectors to be linearly increasing while the physical placement allows for interlace processing time.

Function Code 16 = Format Entire Disk

Initial Register Contents:

D0 = 16 (word mode)
D3 = Disk Interlace Table address (longword mode)
A0 = Disk Parameter Table (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function will select the requested drive, and turn the motor ON. The function will format the entire disk, starting at track 0 on the selected head. At the completion of the format function the motor will be turned OFF and the drive deselected.

Function Code 17 = Read Sector ID Information

Initial Register Contents:

D0 = 17 (word mode)

A0 = Disk Parameter Table address (longword mode)

Final Register Contents:

A0 = Initial Value (longword mode)

Function:

This function assumes that the requested drive is selected and that the motor is ON. When the first sector mark is located, the data in the Sector ID is read and written into the location specified by the Disk Parameter Table.

Function Code 18 = Return Disk Error Status

Initial Register Status:

D0 = 18 (word mode)

A0 = Disk Parameter Table address (longword mode)

Final Register Contents:

D0 = 0 if last operation completed without error,
non-zero if last operation had an error

A0 = Initial Value (longword mode)

Function:

This function sets the D0 register if an error is detected in the previous disk function. If the D0 register is non-zero, then the appropriate status bytes in the Disk Parameter Table can be interrogated to determine the cause of the error.

Function Code 19 = Reserved

Initial Register Contents:

Not Used

Final Register Contents:

Unchanged

Function:

This function is Reserved for later use.

Function Code 20 = Reserved

Initial Register Contents:

Not Needed

Final Register Contents:

Unchanged

Function:

This function is Reserved for later use.

Function Code 21 = Read Printer Status

Initial Register Contents:

D0 = 21 (word mode)

Final Register Contents:

Unknown

Function:

This function returns the status of the built-in Printer Interface. If D0 = 0 then the printer is ready. If bit 7 is set, then the printer is busy. If bit 6 is set, then the printer is off-line or not available.

Function Code 22 = Output Character to Printer Interface

Initial Register Contents:

D0 = 22 (word mode)

D1 = ASCII character to be sent to the Printer Interface (word mode)

Final Register Contents:

Unknown

Function:

Waits until the Printer Interface can accept a character and then outputs the character to the Centronics Type Parallel Interface.

Function Code 23 = Position CRT Cursor

Initial Register Contents:

D0 = 23 (word mode)
D1 = Column coordinate of cursor (X axis) (word mode)
D2 = Line coordinate of cursor (Y axis) (word mode)

Final Register Contents:

A0 = Address of CRT Table (longword mode)

Function:

This function moves the CRT cursor to the position indicated by the D1 and D2 registers. If the cursor is positioned outside the allowable dimensions, then the cursor will be placed at the closest legal X or Y position, respectively (if D1 is greater than characters per line or D2 is greater than the number of lines, or if either is less than 1).

The upper left hand corner of the screen is position (1,1).

Function Code 24 = Read Present Character

Initial Register Contents:

D0 = 24 (word mode)

Final Register Contents:

D0 = Present Character (byte mode)

A0 = Address of CRT Table (longword mode)

Function:

This function will return the ASCII value of the character that is located at the position that is defined by the present cursor position.

Function Code 25 = Clear Graphics Screen

Initial Register Contents:

D0 = 25

Final Register Contents:

Unknown

Function:

This function clears the graphics screen.

NOTES:

This function should only be used in GRAPHICS MODE.

The CRT controller MUST be initialized in GRAPHICS MODE before using this function.

Function Code 26 = Set Color

Initial Register Contents:

D0 = 26 (word mode)
D1 = Color Code (word mode)

Final Register Contents:

A0 = Present Screen Address (longword mode)

Function:

This function sets the Color Code into the Present Color Byte

NOTES:

This function should only be used in GRAPHICS MODE.

The CRT Controller MUST be initialized in GRAPHICS MODE before using this function.

Function Code 27 = Plot X, Y Point

Initial Register Contents:

D0 = 27 (word mode)
D1 = X Coordinate (word mode)
D2 = Y Cooridante (word mode)

Final Register Contents:

A0 = Present Screen Address(longword mode)

Function:

This function plots a point at the designated X and Y coordinates using the color identified in the Present Color Byte. If the CRT mode is black and white, then the point will be a single pixel at the closest actual coordinate. If the mode is color, then the point will be on the correct scan line but will be placed in the closest horizontal point that can be aliased into the color. Multiple color points, plotted close together, may exhibit color changes depending on the equivalent pixel color point. The values of the X, Y coordinates are defined as a minimum pixel, therefore in the highest resolution mode the X coordinate may be from 0 through 700.

Function Code 28 = Reserved

Initial Register Contents:

D0 = 28 (word mode)

Final Register Contents

Unchanged

Function

This function is Reserved for later use.

Function Code 29 = Read a Pixel

Initial Register Contents:

D0 = 29 (word mode)
D1 = X coordinate (word mode)
D2 = Y coordinate (word mode)

Final Register Contents:

D0 = Color Mode (word mode)
A0 = Present Screen Address (longword mode)

Function:

This function reads the color value at the requested X,Y coordinate and returns the value in the D0 register.

Function Code 30 = Output a Character Directly to CRT
with NO Control Characters

Initial Register Contents:

D0 = 30 (word mode)
D1 = character (word mode)

Final Register Contents:

A0 = Address of CRT Table

Function:

This function outputs the character in the D1 register to the CRT screen, directly, with NO control characters.

CRT Table Address Offsets

+0 = 1 byte MODE

- Bit 7 = 0 Text mode (byte mapped, hardware character generation)
- = 1 Graphics mode (bit mapped, 1 bit per pixel)
- Bit 6 = 0 8 bits per character or graphics byte
- = 1 7 bits per character or graphics byte
- Bit 5 = 0 CRT and CPU asynchronous with each other
- = 1 CRT and CPU operate synchronously (used in 80 - 100 column mode)
- Bits 4,3 CRT dot clock frequency
- = 00 14 MHz
- = 01 7 MHz
- = 10 4.77 MHz
- = 11 3.58 MHz
- Bit 2 = 0 CRT ON
- = 1 CRT BLANKED
- Bit 1 = 0 Low Order 256 Character Set
- = 1 High Order 256 Character Set
- Bit 0 = 0 Monochrome (Black & White)
- = 1 Color

+1 = 1 byte Most Significant Byte (bits 23 - 16) of CRT Buffer addr.

+2 = 1 byte Maximum horizontal character count (R0)

+3 = 1 byte Number of displayed characters per line (R1)

+4 = 1 byte Start of horizontal sync position, in character times (R2)

+5 = 1 byte Horizontal sync width, in character times (R3)

+6 = 1 byte Number of character rows, in vertical direction (R4)

+7 = 1 byte Number of scan lines, to adjust vert. freq. (50/60Hz) (R5)

+8 = 1 byte Number of vertically displayed rows (R6)

+9 = 1 byte Vertical sync position, in row count (R7)

+10 = 1 byte Interlace mode (R8)

- = 0 for non-interlace
- = 3 for interlace

+11 = 1 byte Number of scan lines for a character box minus one (R9)

(must be an odd value for interlace mode)

+12 = 1 byte Position in character scan to start graphics/text switch (R10) (normally set = 0)

+13 = 1 byte Width in scan lines -1 of graphics/text start (R11) (normally set = 0)

+14 = 2 byte Low order CRT Buffer Address (R12-R13) (bits 15 - 0)

+16 = 2 byte Low order CRT Buffer Address where switch from graphics to text takes place (R14-R15)

+18 = 2 byte Low order CRT Buffer Address of Light Pen position (R16-R17)

CRT Table Address Offsets (cont'd)

- +20 = 4 byte Length of CRT Buffer in bytes
calculated by the initialize CRT function
Value = Chars/Line * Lines/Screen
- +24 = 4 byte Length of CRT line in bytes
calculated by initialize CRT function
The value is equal to characters/line.
- +28 = 4 byte Present character position address in CRT Buffer
must be within CRT buffer area.
- +32 = 4 byte CRT Buffer starting address
calculated by initialize CRT function.

Graphics Extensions to CRT Table

These parameters are set up by the Initialize CRT Controller Function (function code = 1) when the Mode Byte is set to Graphics (bit 7 = 1).

- +36 = 1 byte Present Color Byte
Values (hex) for COLOR

00 = Black	08 = Dark Aqua
01 = Dark Blue	09 = Bright Blue
02 = Red	0A = Grey
03 = Magenta	0B = Light Blue
04 = Dark Green	0C = Green
05 = Grey	0D = Aqua
06 = Orange	0E = Yellow
07 = Pink	0F = White
- +38 = 2 byte Xmax = Number (n) of Bytes (columns) Displayed * 8
(Xmax = n * 8)
= Number of Bits Displayed
- +40 = 2 byte Ymax = Number of Rows * Number of Scans *
Interlace Factor (1 or 2)

Disk Parameter Table

Reference: The NEC μ P765A Data Sheets

+0 = 1 byte Disk Select Code

Bits 7 (LSB) & 6 - DON'T CARE

Bit 5 - TC

pulsed to 1 to indicate end of block transfer to the
765 controller chip

Bit 4 - DATA RATE

0 = 8 Inch Data Rate

1 = 5 1/4 Inch Data Rate

Bits 3 & 2 - DRIVE NUMBER

DRIVE NUMBER	DRIVE SIZE	
	5 1/4 INCH	8 INCH
00	DRIVE A	DRIVE E
01	DRIVE B	DRIVE F
10	DRIVE C	DRIVE G
11	DRIVE D	DRIVE H

Bits 1 & 0 (LSB) - DRIVE SIZE

00 = INVALID

01 = 5 1/4 Inch

10 = INVALID

11 = 8 Inch

+1 = 1 byte Disk Step Rate and Head Unload Time parameters
used by the SPECIFY command

Bits 7 (MSB) to 4 - Disk Step Rate

where DSR = Disk Step Rate

= number of 1 mS. steps minus 1

example 0010 = 2 + 1 = 3 mS.

Bits 3 to 0 (LSB) - Head Unload Time

where HUT = Head Unload Time

= number of 16 mS. steps minus 1

example 0010 = (2 + 1) * 16
= 3 * 16 = 48 mS.

Disk Parameter Table (cont'd)

+2 = 1 byte Disk Head Load Time value and Non-DMA Flag

Bits 7 (MSB) to 1 - Head Load Time
 where HLT = Head Load Time
 = number of 16 mS. steps minus 1
 example 0000010 = (2 + 1) * 2
 = 3 * 2
 = 6 mS. Load Time

+3 = 1 byte Disk Mode and Command parameter, set up by ROMBIOS
 this is the uP765 command most significant byte

Bit 7 (MSB) - Multi-Track
 0 = Not Multi-Track
 1 = Multi-Track Read or Write

Bit 6 - MF
 0 = FM Mode
 1 = MFM Mode

Bit 5 - Skip Deleted Data
 0 = Terminate on a Deleted Data Address Mark
 1 = Skip over a Deleted Data Address Mark

Bits 4 to 0 (LSB) - Command
 00000 - 00001 = INVALID
 00010 = READ A TRACK
 00011 = SPECIFY
 00100 = SENSE DRIVE STATUS
 00101 = WRITE DATA
 00110 = READ DATA
 00111 = RECALIBRATE
 01000 = SENSE INTERRUPT STATUS
 01001 = WRITE DELETED DATA
 01010 = READ ID
 01011 = INVALID
 01100 = READ DELETED DATA
 01101 = FORMAT A TRACK
 01110 = INVALID
 01111 = SEEK
 10000 = INVALID
 10001 = SCAN EQUAL
 10010 - 11001 = INVALID
 11001 = SCAN LOW OR EQUAL
 11010 - 11100 = INVALID
 11101 = SCAN HIGH OR EQUAL
 11110 - 11111 = INVALID

Disk Parameter Table (cont'd)

-
- +4 = 1 byte Disk Head Select/Drive Select Code, set up by ROMBIOS
this is the uP765 command least significant byte
- Bits 7 (MSB) to 3 - Zeroes
00000 = Zeroes
- Bit 2 - Head Select
0 = Head 0
1 = Head 1
- Bit 1 to 0 - Unit Select
00 = Drive 0
01 = Drive 1
10 = Drive 2
11 = Drive 3
- +5 = 1 byte Disk Cylinder Address (track)
the range is 0 to 255
- +6 = 1 byte Disk Head Select
the range is 0 to 1
- +7 = 1 byte Disk Sector Address
- +8 = 1 byte Disk Sector Size = $128 * 2^N$ bytes/sector
where DSS = Disk Sector Size in 128 Byte steps minus 1
example 03 (hex) = $128 * 2^3$
= $128 * 8$
= 1024 Bytes/Sector
- +9 = 1 byte Disk Sectors/Track (or End Of Track count value)
- +10 = 1 byte Disk Sector Gap Length
- +11 = 1 byte Sector Data Length, set to FF (hex), unless Disk Sector Size is set to 0, then this is bytes per sector
- +12 = 4 byte Disk Data Buffer Address, starting address of buffer to be read/written

Disk Parameter Table (cont'd)

+16 = 1 byte Status 0 byte, from last operation
+17 = 1 byte Status 1 byte, from last operation
+18 = 1 byte Status 2 byte, from last operation
+19 = 1 byte Cylinder Address, read from Disk ID
+20 = 1 byte Next Head Address
+21 = 1 byte Next Sector Address
+22 = 1 byte Disk Sector Size value, read from Disk ID
+23 = 1 byte Motor ON Flag, set to 0 if OFF, set to 1 if ON
+24 = 1 byte Disk Steps/Track value = 2^N steps/track
+25 = 1 byte Number of Tracks Per Side value

D I M E N S I O N 6 8 0 0 0
S Y S T E M R E F E R E N C E M A N U A L

I N D E X

I N D E X

Central Processor	7
Connectors	15, 17, 18, 20, 21, 22
CP/M-68K	34, 37, 39
CRT	10, 25, 35, 44
Cursor	43
Disk Drive Expansion	17, 19
Emulators	9
6512	9
8086	9
Z80	9
Expansion Slots	7, 9, 11, 29
I/O Devices	7, 10, 39, 45
Keyboard	7, 9, 23, 33, 35
MC68000L8	27
Monitor	34
Power	8
Power Supply Assembly	7, 9, 15
RESET Switch	8, 16
ROMBIOS	9, 25, 34, 35, 37, 39
System Unit	16

A P P E N D I X G

M C 6 8 0 0 0 A S S E M B L Y L A N G U A G E

The following material is excerpted from the MOTOROLA MC68000 MICROPROCESSOR USER'S GUIDE.

3.1 INTRODUCTION

This section contains an overview of the form and structure of the MC68000 instruction set. The instructions form a set of tools that include all the machine functions to perform the following operations:

- Data Movement
- Integer Arithmetic
- Logical
- Shift and Rotate
- Bit Manipulation
- Binary Coded Decimal
- Program Control
- System Control

The complete range of instruction capabilities combined with the flexible addressing modes described in Section 2 provide a very flexible base for program development. Detailed information about each instruction is given in Appendix B.

3.2 DATA MOVEMENT OPERATIONS

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction, there are several special data movement instructions: move multiple registers (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 3-1 is a summary of the data movement operations.

3.3 INTEGER ARITHMETIC OPERATIONS

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are

Table 3-1. Data Movement Operations

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	—	An → SP@ – SP → An SP + d → SP
MOVE	8, 16, 32	(EA)s → EAd
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → EA
MOVEQ	8	#xxx → Dn
PEA	32	EA → SP@ –
SWAP	32	Dn [31:16] ↔ Dn [15:0]
UNLK	—	An → SP SP@ + → An

NOTES:

s = source
d = destination
[] = bit numbers

limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 3-2 is a summary of the integer arithmetic operations.

3.4 LOGICAL OPERATIONS

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 3-3 is a summary of the logical operations.

3.5 SHIFT AND ROTATE OPERATIONS

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand

Table 3-2. Integer Arithmetic Operations

Instruction	Operand Size	Operation
ADD	8, 16, 32	$D_n + (EA) \rightarrow D_n$ $(EA) + D_n \rightarrow EA$ $(EA) + \#xxx \rightarrow EA$ $A_n + (EA) \rightarrow A_n$
	16, 32	
ADDX	8, 16, 32	$D_x + D_y + X \rightarrow D_x$
	16, 32	$A_x@ - A_y@ - + X \rightarrow A_x@$
CLR	8, 16, 32	$0 \rightarrow EA$
CMP	8, 16, 32	$D_n - (EA)$ $(EA) - \#xxx$ $A_x@ + - A_y@ +$ $A_n - (EA)$
	16, 32	
DIVS	$32 \div 16$	$D_n / (EA) \rightarrow D_n$
DIVU	$32 \div 16$	$D_n / (EA) \rightarrow D_n$
EXT	$8 \rightarrow 16$	$(D_n)_8 \rightarrow D_{n16}$
	$16 \rightarrow 32$	$(D_n)_{16} \rightarrow D_{n32}$
MULS	$16 * 16 \rightarrow 32$	$D_n * (EA) \rightarrow 32 D_n$
MULU	$16 * 16 \rightarrow 32$	$D_n * (EA) \rightarrow D_n$
NEG	8, 16, 32	$0 - (EA) \rightarrow EA$
NEGX	8, 16, 32	$0 - (EA) - X \rightarrow EA$
SUB	8, 16, 32	$D_n - (EA) \rightarrow D_n$ $(EA) - D_n \rightarrow EA$ $(EA) - \#xxx \rightarrow EA$ $A_n - (EA) \rightarrow A_n$
	16, 32	
SUBX	8, 16, 32	$D_x - D_y - X \rightarrow D_x$ $A_x@ - - A_y@ - - X \rightarrow A_x@$
TAS	8	$(EA) - 0, 1 \rightarrow EA [7]$
TST	8, 16, 32	$(EA) - 0$

NOTE:

[] = bit number

sizes and allow a shift count specified in the instruction of one or eight bits, or 0 to 63 bits specified in a data register.

Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates.

Table 3-4 is a summary of the shift and rotate operations.

Table 3-3. Logical Operations

Instruction	Operand Size	Operation
AND	8, 16, 32	$D_n \wedge (EA) \rightarrow D_n$
		$(EA) \wedge D_n \rightarrow EA$
		$(EA) \wedge \#xxx \rightarrow EA$
OR	8, 16, 32	$D_n \vee (EA) \rightarrow D_n$
		$(EA) \vee D_n \rightarrow EA$
		$(EA) \vee \#xxx \rightarrow EA$
EOR	8, 16, 32	$(Ea) \oplus D_y \rightarrow EA$ $(EA) \oplus \#xxx \rightarrow EA$
NOT	8, 16, 32	$\sim (EA) \rightarrow EA$

NOTE:

 \sim = invert

Table 3-1. Data Movement Operations

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	An → SP@ – SP → An SP + d → SP
MOVE	8, 16, 32	(EA)s → EAd
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → EA
MOVEQ	8	#xxx → Dn
PEA	32	EA → SP@ –
SWAP	32	Dn [31:16] ↔ Dn [15:0]
UNLK	–	An → SP SP@+ → An

NOTES:

s = source

d = destination

[] = bit numbers

limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 3-2 is a summary of the integer arithmetic operations.

3.4 LOGICAL OPERATIONS

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 3-3 is a summary of the logical operations.

3.5 SHIFT AND ROTATE OPERATIONS

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand

Table 3-4. Shift and Rotate Operations

Instruction	Operand Size	Operation
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

3.6 BIT MANIPULATION OPERATIONS

Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 3-5 is a summary of the bit manipulation operations.

Table 3-5. Bit Manipulation Operations

Instruction	Operand Size	Operation
BTST	8, 32	~ bit of (EA) → Z
BSET	8, 32	~ bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~ bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~ bit of (EA) → Z ~ bit of (EA) → bit of EA

3.7 BINARY CODED DECIMAL OPERATIONS

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using the following instructions: add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 3-6 is a summary of the binary coded decimal operations.

Table 3-6. Binary Coded Decimal Operations

Instruction	Operand Size	Operation
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$ $Ax@ - 10 + Ay@ - 10 + X \rightarrow Ax@$
SBCD	8	$Dy_{10} - Dx_{10} - X \rightarrow Dx$ $Ax@ - 10 - Ay@ - 10 - X \rightarrow Ax@$
NBCD	8	$0 - (EA)_{10} - X \rightarrow EA$

3.8 PROGRAM CONTROL OPERATIONS

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions. These instructions are summarized in Table 3-7.

The conditional instructions provide setting and branching for the following conditions:

CC — Carry Clear	LS — Low or Same
CS — Carry Set	LT — Less Than
EQ — Equal	MI — Minus
F — Never True	NE — Not Equal
GE — Greater or Equal	PL — Plus
GT — Greater Than	T — Always True
HI — High	VC — No Overflow
LE — Less or Equal	VS — Overflow

Table 3-7. Program Control Operations

Instruction	Operation
Conditional	
Bcc	Branch conditionally (14 conditions), 8- and 16-bit displacement
DBcc	Test condition, decrement and branch. 16-bit displacement
Scc	Set byte conditionally (16 conditions)
Unconditional	
BRA	Branch always, 8- and 16-bit displacement
BSR	Branch to subroutine, 8- and 16-bit displacement
JMP	Jump
JSR	Jump to subroutine
Returns	
RTR	Return and restore condition codes
RTS	Return from subroutine

3.9 SYSTEM CONTROL OPERATIONS

System control operations are accomplished by using privileged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in Table 3-8.

Table 3-8. System Control Operations

Instruction	Operation
Privileged	
ANDI to SR	Logical AND to status register
EORI to SR	Logical EOR to status register
MOVE EA to SR	Load new status register
MOVE USP	Move user stack pointer
ORI to SR	Logical OR to status register
RESET	Reset external devices
RTE	Return from exception
STOP	Stop program execution
Trap Generating	
CHK	Check register against bounds
TRAP	Trap
TRAPV	Trap on overflow
Status Register	
ANDI to CCR	Logical AND to condition codes
EORI to CCR	Logical EOR to condition codes
MOVE EA to CCR	Load new condition codes
MOVE SR to EA	Store status register
ORI to CCR	Logical OR to condition codes

INSTRUCTION FORMAT SUMMARY

C.1 INTRODUCTION

This appendix provides a summary of the first word in each instruction of the instruction set. Table C-1 is an operation code (op-code) map which illustrates how bits 15 through 12 are used to specify the operations. The remaining paragraph groups the instructions according to the op-code map.

Table C-1. Operation Code Map

Bits 15 through 12	Operation	Bits 15 through 12	Operation
0000	Bit Manipulation/MOVEP/Immediate	1000	OR/DIV/SBCD
0001	Move Byte	1001	SUB/SUBX
0010	Move Long	1010	(Unassigned)
0011	Move Word	1011	CMP/EOR
0100	Miscellaneous	1100	AND/MUL/ABCD/EXG
0101	ADDQ/SUBQ/ScC/DBcc	1101	ADD/ADDX
0110	Bcc/BSR	1110	Shift/Rotate
0111	MOVEQ	1111	(Unassigned)

Table C-2. Effective Address Encoding Summary

Addressing Mode	Mode	Register
Data Register Direct	000	register number
Address Register Direct	001	register number
Address Register Indirect	010	register number
Address Register Indirect with Postincrement	011	register number
Address Register Indirect with Predecrement	100	register number
Address Register Indirect with Displacement	101	register number
Address Register Indirect with Index	110	register number
Absolute Short	111	000
Absolute Long	111	001
Program Counter with Displacement	111	010
Program Counter with Index	111	011
Immediate or Status Register	111	100

Table C-3. Conditional Tests

Mnemonic	Condition	Encoding	Test
T	true	0000	1
F	false	0001	0
HI	high	0010	$\overline{C+Z}$
LS	low or same	0011	$C+Z$
CC(HS)	carry clear	0100	\overline{C}
CS(LO)	carry set	0101	C
NE	not equal	0110	\overline{Z}
EQ	equal	0111	Z
VC	overflow clear	1000	\overline{V}
VS	overflow set	1001	V
PL	plus	1010	\overline{N}
MI	minus	1011	N
GE	greater or equal	1100	$N \cdot V + \overline{N} \cdot \overline{V}$
LT	less than	1101	$N \cdot \overline{V} + \overline{N} \cdot V$
GT	greater than	1110	$N \cdot V \cdot \overline{Z} + \overline{N} \cdot \overline{V} \cdot \overline{Z}$
LE	less or equal	1111	$Z + N \cdot \overline{V} + \overline{N} \cdot V$

C.2 BIT MANIPULATION, MOVE PERIPHERAL, IMMEDIATE INSTRUCTIONS

Dynamic Bit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Register	1	Type	Effective Address								
							Mode	Register							

Static Bit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	Type	Effective Address						
							Mode	Register							

Type field: 00 = TST 10 = CLR
 01 = CHG 11 = SET

MOVEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Register	Op-Mode	0	0	1	Register						

Op-Mode field: 100 = transfer word from memory to register
 101 = transfer long word from memory to register
 110 = transfer word from register to memory
 111 = transfer long word from register to memory

OR Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Size	Effective Address						
							Mode	Register							

Size field: 00 = byte
 01 = word
 11 = long word

OR Immediate to CCR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

OR Immediate to SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

AND Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	Size	Effective Address						
									Mode	Register					

*

AND Immediate to CCR

5	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0

AND Immediate to SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0

SUB Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	Size	Effective Address						
									Mode	Register					

*

ADD Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	Size	Effective Address						
									Mode	Register					

*

EOR Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0	Size	Effective Address						
									Mode	Register					

*

*Size field: 00 = byte
 01 = word
 11 = long word

EOR Immediate to CCR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0

EOR Immediate to SR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0	0	1	1	1	1	1	0	0

CMP Immediate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0	Size		Effective Address					
										Mode			Register		

Size field: 00 = byte
 01 = word
 11 = long word

C.3 MOVE BYTE INSTRUCTION**MOVE Byte**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	Destination				Source							
				Register		Mode		Mode		Register					

Note Register and Mode location

C.4 MOVE LONG INSTRUCTION**MOVE Long**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	Destination				Source							
				Register		Mode		Mode		Register					

Note Register and Mode location

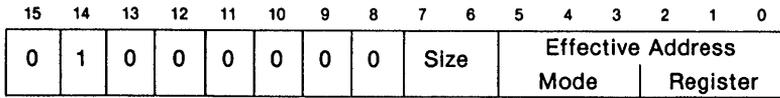
C.5 MOVE WORD INSTRUCTION**MOVE Word**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	Destination				Source							
				Register		Mode		Mode		Register					

Note Register and Mode location

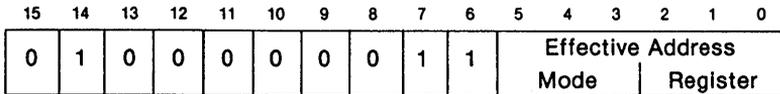
C.6 MISCELLANEOUS INSTRUCTIONS

NEGX

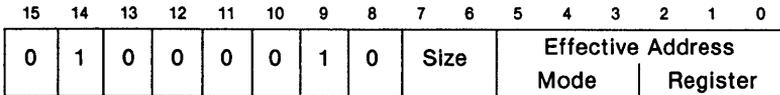


*

MOVE from SR

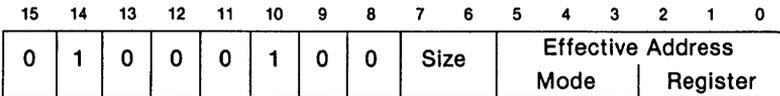


CLR



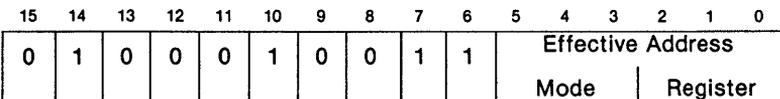
*

NEG

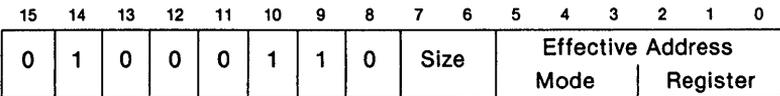


*

MOVE to CCR

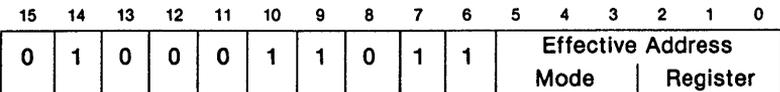


NOT



*

MOVE to SR



NBCD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	0	Effective Address					
										Mode	Register				

PEA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	0	1	Effective Address				
										Mode	Register				

SWAP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	0	1	0	0	0	Register	

MOVEM Registers to EA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	1	Sz	Effective Address				
										Mode	Register				

Sz field: 0 = word transfer
 1 = long word transfer

EXTW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	1	0	0	0	0	Register	

EXTL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	1	1	0	0	0	Register	

TST

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	0	Size		Effective Address					
										Mode	Register				

Size field: 00 = byte
 01 = word
 11 = long word

TAS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	0	1	1	Effective Address Mode Register					

ILLEGAL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	0	1	1	1	1	1	1	0	0

MOVEM EA to Registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	0	1	Sz	Effective Address Mode Register					

Sz field: 0 = word transfer
1 = long word transfer

TRAP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	0	Vector			

LINK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	1	0	Register		

UNLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	1	1	Register		

MOVE to USP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	0	0	Register		

MOVE from USP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	0	1	Register		

RESET

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0

NOP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	0	1

STOP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0

RTE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1

RTS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1

TRAPV

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	1	1	0

RTR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	1	1	1

JSR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	1	0	Effective Address Mode		Register			

JMP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	1	1	Effective Address Mode		Register			

CHK

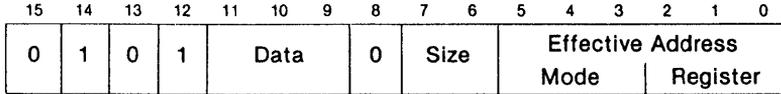
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	Register			1	1	0	Effective Address Mode		Register			

LEA

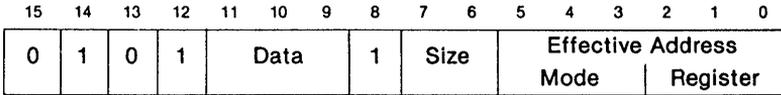
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	Register			1	1	1	Effective Address Mode		Register			

C.7 ADD QUICK, SUBTRACT QUICK, SET CONDITIONALLY, DECREMENT INSTRUCTIONS

ADDQ



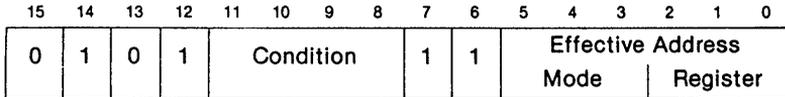
SUBQ



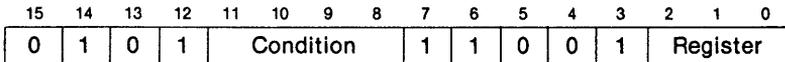
Data field: Three bits of immediate data, 0, 1-7, representing a range of: 8, 1-7, respectively.

Size field: 00 = byte
 01 = word
 10 = long word

Scc



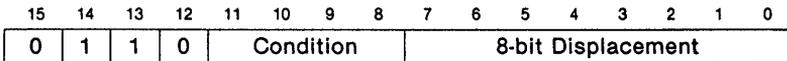
DBcc



Condition field: See Table C-3

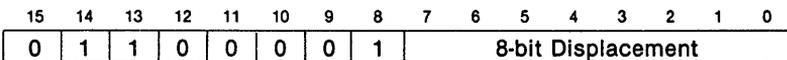
C.8 BRANCH CONDITIONALLY INSTRUCTION

Bcc



Condition field: See Table C-3

BSR



8-bit Displacement field: Two's complement integer representing the relative distance in bytes. If 0, 16-bit displacement is in extension word.

MOV^{EQ}

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	1		1	Register				0	Data*							

*Data is sign extended to a long operand, and all 32 bits are transferred to the data register.

C.10 OR, DIVIDE, SUBTRACT DECIMAL INSTRUCTIONS

OR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	Register				Op-Mode		Effective Address					
										Mode		Register			

Op-Mode field: Byte Word Long Operation
 000 001 010 (<Dn>) \wedge (<ea>) \rightarrow <Dn>
 100 101 110 (<ea>) \wedge (<Dn>) \rightarrow <ea>

DIVU

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	0	0	0	Register				0	1	1	Effective Address					
										Mode		Register				

DIVS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	0	0	0	Register				1	1	1	Effective Address					
										Mode		Register				

SBCD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	0	0	0	Destination Register*				1	0	0	0	0	R/M	Source Register*		

R/M field: 0 = register-register
 1 = memory-memory

*If R/M = 0, specifies a data register.

If R/M = 1, specifies an address register for the predecrement addressing mode.

C.11 SUBTRACT, SUBTRACT EXTENDED INSTRUCTIONS

SUB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	Register				Op-Mode		Effective Address					
										Mode		Register			

Op-Mode field: Byte Word Long Operation
 000 001 010 (<Dn>) + (<ea>) \rightarrow <Dn>
 100 101 110 (<ea>) + (<Dn>) \rightarrow <ea>

SUBX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	Destination Register*	1	Size	0	0	R/M	Source Register*					

R/M field: 0 = register-register
 1 = memory-memory

*If R/M = 0, specifies a data register.
 If R/M = 1, specifies an address register for the predecrement addressing mode.

C.12 COMPARE, EXCLUSIVE OR INSTRUCTIONS

CMP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	Register	Op-Mode	Effective Address									
						Mode	Register								

Op-Mode field: Byte Word Long Operation
 000 001 010 (<Dn>)-(<ea>)

CMPM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	Register	1	Size	0	0	1	Register					

EOR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	Register	1	Size	Effective Address								
						Mode	Register								

Size field: 00 = byte
 01 = word
 10 = long word

C.13 AND, MULTIPLY, ADD DECIMAL, EXCHANGE INSTRUCTIONS

AND

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Register	Op-Mode	Effective Address									
						Mode	Register								

Op-Mode field: Byte Word Long Operation
 000 001 010 (<Dn>)-(<ea>)-> <Dn>
 100 101 110 (<ea>)-(<Dn>)-> <ea>

MULU

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Register			0	1	1	Effective Address			Mode		Register

MULS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Register			1	1	1	Effective Address			Mode		Register

ABCD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Destination Register*			1	0	0	0	0	R/M	Source Register*		

R/M field: 0 = register-register
 1 = memory-memory

*If R/M = 0, specifies a data register.

If R/M = 1, specifies an address register for the predecrement addressing mode.

EXGD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Data Register			1	0	1	0	0	0	Data Register		

EXGA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Address Register			1	0	1	0	0	1	Address Register		

EXGM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Data Register			1	1	0	0	0	1	Address Register		

C.14 ADD, ADD EXTENDED INSTRUCTIONS**ADD**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	Register			Op-Mode			Effective Address			Mode		Register

Op-Mode field: Byte Word Long Operation

000	001	010	(<Dn> v (<ea>) → <Dn>
100	101	110	(<ea> v (<Dn>) → <ea>

ADDX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	Destination Register*			1	Size		0	0	R/M	Source Register*		

R/M field: 0 = register-register
 1 = memory-memory

*If R/M = 0, specifies a data register

If R/M = 1, specifies an address register for the predecrement addressing mode.

C.15 SHIFT/ROTATE INSTRUCTIONS

Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	Count/ Register		dr	Size		i/r	Type		Register			

Count/Register field: If i/r field = 0, specifies count
 If i/r field = 1, specifies a data register that contains the count.

dr field: 0 = right
 1 = left

Size field: 00 = byte
 01 = word
 11 = long word

i/r field: 0 = immediate count
 1 = register count

Type field: 00 = arithmetic shift
 01 = logical shift
 10 = rotate with extend
 11 = rotate

Memory

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	Type		dr	1	1	Effective Address Mode		Register			

Type field: 00 = arithmetic shift
 01 = logical shift
 10 = rotate with extend
 11 = rotate

dr field: 0 = right
 1 = left

