

# TIMEX COMPUTER FDD3000

## Service Manual



[HTTP://WWW.RETROREVIEW.COM](http://www.retroreview.com)

The retro computing magazine

**TC** TIMEX COMPUTER

**FDD3000**

SERVICE MANUAL

FOR DOUBLE  
&  
SINGLE DRIVE SYSTEM

REV 03

---

## CONTENTS

---

- I- INTRODUCTION
- II- WHAT IS THE FDD 3000
  - 1- HARDWARE
  - 2- OPERATING SYSTEM
  - 3- DISKETTES
- III- OPERATING PROCEDURES
- IV- HOW TO TEST THE FDD
- V- TROUBLESHOOTING

## APPENDIX

- I- ERROR MESSAGES
- II- PIN OUT RS232
- III- PARST LIST
- IV- SPARES PARTS LIST
- V- LISTING (MACRO) CBIOS
- VI- LISTING (MACRO) BIOS
- VIII- DATA SHEETS OF COMPONENTS

## INTRODUCTION

---

I) INTRODUCTION

---

It is not the objective of this manual how to teach how to repair but to give some technical guidelines allowing fault finding and isolation of the defect and whenever possible to solve the fault.

The troubleshooter needs to use its technical knowledge on a rational way. He will need patience, a continuity tester, a multimeter and an oscilloscope. As a very basic rule these are the tools required to identify and troubleshoot the faults.

Special attention is required to the non conformities with the product specification.

This manual was conceived to completely cover the product and the project, and is the most updated one.

The way this manual was written will give a progressive knowledge of the product as you will read through it allowing you to solve all the situations.

WHAT IS THE FDD 3000

---

FDD MANUAL

---

II - HARDWARE

---

II-2 THE OPERATING SYSTEM

---

The addition of the floppy disc drive system to the zx Spectrum or its compatibles TC2048 and TC2068 creates a very powerful system, the TOS ( TIMEX OPERATING SYSTEM ). This OPERATING SYSTEM was developed by TIMEX giving the user a capacity from the BASIC that he never thought about.

The secret is the controller that commands the drive operation, another one for external communications that an unimaginable flexibility to TOS.

The controller has 64k Bytes of DINAMIC memory.

The input and output of data is controlled by a Z80A microprocessor at 4 MHZ.

When a TOS command is issued it passes through a parallel bus to the controller that executes the instructions, without overloading the computer memory, since the the controller on its own is a stand alone computer.

The controller and the operating system were developed around the need to communicate. So we can find 2 RS 232 (V24) ports, which can be configured to any type of communication protocol, which means that besides the power to communicate with other users of TOS it is also possible to TALK with users of others computer systems. Other peripherals like telephonic MODEMS, PRINTERS and other devices connectable to a communication port can be driven by the FDD system.

### II-3) THE MEDIA ( Diskettes or simply discs)

The diskettes ( Discs) are manufactured with very thin flexible material coated with a magnetic material, enclosed inside a hard plastic case. A shutter mechanically actuated protects the media when out of the floppy disc drive unit.

This protection opens automatically when the disc is inserted in the drive and when it is completely inside the READ/WRITE head touches the disc.

The disc is of double density type allowing a formatted capacity of 140 K per side ( total 320 K Bytes per disc ).

Although the discs are very reliable certain basic precautions have to be taken:

- Do not expose the discs to very strong or direct sources of heat or light.
- Do not submit disc to high temperatures.
- Do not expose the disc to any type of magnetic field.
- Do not expose to dust.
- Keep discs away of humide environments.
- Do not open the protection shutter and/or touch the media with the fingers or any other object.

The information stored in the disc can be protected in two possible ways: hardware or software. the Hardware protection is achieved trough the protection tabs, that cover or open a hole near the corners of the disc plastic enclosure, which allows or stops the write operations on the disc.

OPEN HOLE.....WRITE PROTECTED DISC

... About CP/M

CP/M means "Control Program Monitor" and was one of the first Operating Systems written in the beginning of the 70' s. Dr. Gary Kindall is considered as the father of CP/M, he was developing Software for the INTEL microprocessor 8080.

With the development of different types and sizes of floppy disc drives the need for a system that would interconnect or make all of them was created. This is one of the reasons of CP/M flexibility and transportability from one system to another. CP/M Operating System consists off 3 parts, to know:

BIOS ( BASIC INPUT/OUTPUT SYSTEM )

BDOS ( BASIC DISC OPERATING SYSTEM )

CCP ( CONSOLE COMMAND PROCESSOR )

the BIOS is the only part of the Operating System that depends on the Hardware where CP/M is going to run. Is the only part of the Operating System that can be costumised by the Hardware manufacturer to have mind particularities of the Floppy Disc Drives, Discs, Terminals, and Printers that he produces.

BDOS can not be relocated or modified since it is legally protected. It is the part of the Operating System that controls all the Drives and is independent of the Hardware.

The CCP interfaces the user with the software since it translates the keyboard instructions into instructions that the microprocessor can understand. It is also legally protected and can not be relocated or modified.

When CP/M is loaded into computer memory it is done in a well organized way occupying segments of memory. The BIOS is stored in the top of the memory.

The CCP stays immediatly below and CCP occupies the bottom part of the memory pile. The remaining memory space is reserved for the application programs and is called TPA ( TRANSIT PROGRAM AREA ).

The amount of memory used by the BIOS depends on the Hardware configuration and is typically of 3K Bytes.

The BDOS as a fixed length of 3.5 K Bytes. This happens because when CP/M transfer information from the RAM to the Disc, or Vice-versa does it always in blocks of 128 Bytes up to a maximum of 16 drives. ( only for maximum on the case of the TIMEX FDD ). The physical operation between the DISCS and the RAM is handled by the BIOS.

Below the BIOS and BDOS location in memory we find the CCP and the TPA, the lowest part of memory is formed by a block of 256 K Bytes the contains the system variables.

When the Computer is Switch-On, the first operation performed is to load the CP/M Operation System, from its location in the Disc, to the RAM of the Computer. This is called "BOOTSTRAPPING" or simply system "BOOT". Depending on the system this can be achieved through a special combination of keystrokes on the keyboard. in the case of the FDD 3000 this happens automatically after the insertion of the disc into the floppy disc drive. We know that this operation is finished when the screen displays the log-on message from the Hardware manufacturer followed by the prompt:

A>

This symbol indicates that the CCP is functioning and is waiting to answer any command line.

If by any reason it is necessary re-initialize CP/M after the power-on of the computer, it is possible to execute a "WARN-BOOT" warm reset of the Operating System. This is some times required after a system error and the subsequent Operating System "error message". This procedure loads BDOS and CCP into their memory location meaning that the computer is ready to accept new commands again.

CP/M is a registered trademark of DIGITAL RESEARCH.

Some passages of "USING CP/M" by PETER GOSLING where quoted above.

## OPERATING PROCEDURES

### III- OPERATION PROCEDURES.

In this chapter we will through the installation procedure of the Floppy Disc Drive unit.

The FDD 3000 supports two different Operating systems:

TOS - TIMEX OPERATING SYSTEM

CP/M - CONTROL PROGRAM MONITOR

The Hardware configuration depends directly of the Operating System to be used. Lets see why:

If we want to use TOS software we need the following Hardware:

- MICROCOMPUTER (Z80 based)
- TV SET OR MONITOR
- INTERFACE FOR COMMUNICATION
- FLOPPY DISC DRIVE SYSTEM FDD 3000
- PRINTER ( RS232 ) (OPTIONAL)

In this configuration the interface changes according to the computer to be used.

If your request is to run Software in CP/M format two approaches can be used, one using a computer as console and the other is to use a TERMINAL 3000, in this case the configuration will be:

- TERMINAL 3000
- MONITOR
- FLOPPY DISC DRIVE SYSTEM FDD 3000
- PRINTER ( RS232 SERIE ) ( OPTIONAL )

III-2) THE CONNECTIONS BY SEQUENCE

- 1- Connect the curly cable from the FDD 3000 to the right communication interface for your computer.
- 2- Carefully engage the interface connector of the interface in the computer edge-connector.
- 3- Connect the RS232 cable for the printer into the chosen channel "A" or "B".
- 4- If you have a ZX SPECTRUM you can use a monitor, just connect the monitor RCA plug from the monitor into the socket at the back of the FDD 3000 system also marked "MONITOR". If you have a ZX SPECTRUM and not a MONITOR you have to use the TV set, connecting the TV cable to the appropriate plug on the computer. If you any other TIMEX computer and a MONITOR you have to plug the RCA jack into your computer, socket marked with MONITOR.  
It is completely impossible to connect the tv cable to the video plug at the back of the FDD 3000 since no picture will be produced at all in the TV screen.  
If you are using a TERMINAL 3000 to work with this system the monitor cable RCA must be connected to the MONITOR plug at the back of the FDD 3000 system.
- 5- Connect all the power supplies to the mains socket 220 VOLT. If you are usin the ZX SPECTRUM or TC computers connect the plugs from the power supplies to the correct sockets at the back pannels of the computers. Connect the TV set or the MONITOR to the mains.
- 6- Then power-on of:
  - POWER ON TV SET OR MONITOR
  - POWER ON OF THE FDD 3000
  - POWER ON OF THE COMPUTER
  - POWER ON OF THE PRINTER

HOW TO TEST THE FDD 3000

IV- HOW TO TEST THE FDD 3000

Configure the system to operate in the TOS mode:

- 1- POWER ON THE TV SET OR THE MONITOR
- 2- POWER ON THE FDD 3000
- 3- POWER ON THE PRINTER
- 4- POWER ON THE COMPUTER
- 5- PUT THE TOS FORMATTED DISC INTO DRIVE A AND THE CP/M DISC IN DRIVE B
- 6- DO RESET ON THE FDD 3000 CONTROLLER RESET BOTTON
- 7- DO RESET ON THE COMPUTER INTERFACE BY PRESSING RESET BOTTON ON THE INTERFACE

ON THE SCREEN THE FOLLOWING MESSAGE WILL APPEAR AFTER A SECOND

C 1982 SINCLAIR RESEARCH LTD  
C 1984 TIMEX - TOS VA.2

- 8- THE SCREEN WILL CLEAR AND THE FOLLOWING MESSAGE WILL BE DISPLAYED

"DAMAGE'S CHANNEL'S" ( if there is no RS 232  
communicatio between channel  
"A" and "B")

If the communication is possible between the two channels the following message is shown:

" TAKE OUT THE PLUG RS 232 &  
CONNECT THE PRINTER CORD! "

PRESS C TO CONTINUE

- 9- PRESS KEY C,  
THE SCREEN WILL THEN SHOW:

" MONITOR EMULATOR FOR TC2048/TC2068  
Copyright TMX PORTUGAL  
VERSION V A1.1 11, APRIL 1986

PLEASE PUT THE CP/M DISKETTE IN DRIVE A:  
AND PRESS THE CCONTROLLER RESET BOTTON: "

- 10- REMOVE THE DISC IN DRIVE "A". TURN TO SIDE B AND INTRODUCE IT IN DRIVE "A" AGAIN. PRESS THE RESET BUTTON ON THE CONTROLLER: THE SCREEN WILL SHOW:

```
CP/M VERSION 2.2
COPYRIGHT DIGITAL RESEARCH INC!

CBIOS VERSION VAI.1 COPYRIGHT TMX PORTUGAL

3, JUNE 1986
```

- 11- TYPE "T" + "RETURN"  
THE SCREEN DISPLAYS:

```
TESTE DO FDD 3000
PREPARADO POR F.FARIA & Z.SILVA
C 1986 TMX PORTUGAL
```

TESTE DE ACESSO A DRIVE A CONSTITUIDO POR:

- 1- CRIACAO DO FICHEIRO "TEMP.TMP"
- 2- ESCRITA THE 128 K BYTES (ASCII"?) NO FICHEIRO
- 3- FECHO DO FICHEIRO
- 4- ABERTURA DE FICHEIRO
- 5- LEITURA DO FICHEIRO
- 6- APAGAR AO FICHEIRO

TESTE DO CANAL ( :CH\_A) DO TIMEX FDD 3000

VERIFIQUE SE A SEQUENCIA DE CARACTERES ESTA CORRECTA!

1234567890QWERTYUIOPASDFGHJKLZXCVBNM

TESTE DE MEMORIA DO TIMEX FDD 3000

MEMORIA OK !!!

A>

- 12- CONNECT THE "RCA" PLUG AND VERIFY IF THE SCREEN IMAGE IS CORRECT.

IF THE ANSWER IS yes THE CONTROLLER IS APPROVED !!!

---

FDD MANUAL

---

IMPORTANT !            IMPORTANT !            IMPORTANT !            IMPORTANT !

BEFORE STARTING THE REPAIR SESSION PLEASE MAKE SHURE  
THAT YOU HAVE BACKUP COPIES OF THE TEST SOFTWARE

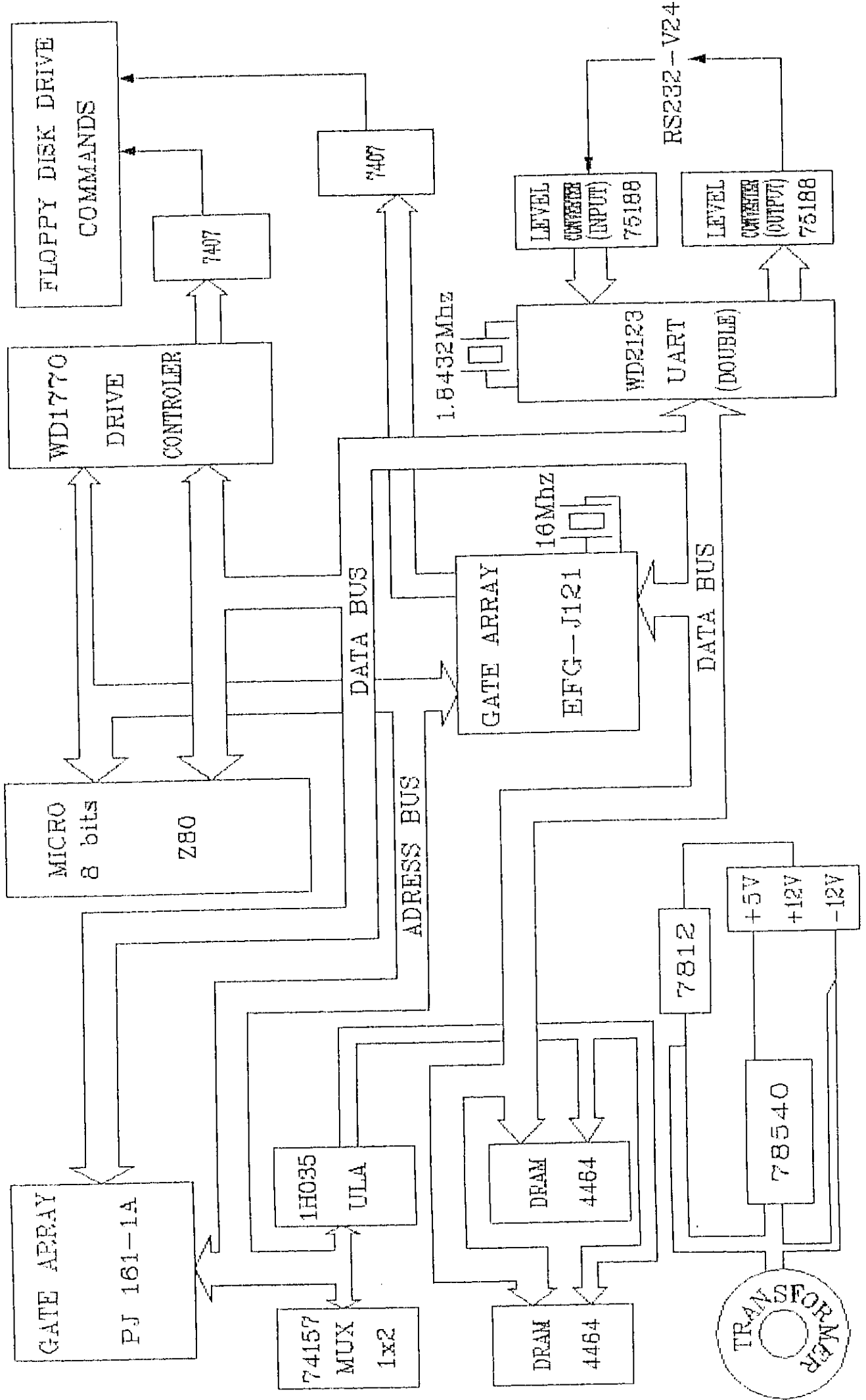
IMPORTANT !            IMPORTANT !            IMPORTANT !            IMPORTANT !

ELECTRICAL SCHEMATICS

## V - ELECTRICAL SCHEMATICS

## LIST OF SIGNALS:

- 1 - CLOCK 16 MHZ (XIN) (PIN 45 GA121)
- 2 - CLOCK 16 MHZ (XOUT) (PIN 44 GA121)
- 3 - CLOCK 4 MHZ (XOUT) (PIN 42 GA121)
- 4 - CLOCK 8 MHZ (XOUT) (PIN 42 GA121)
- 5 - CLOCK 1.8432 MHZ (X1) (PIN 31 WD2123 - U10)
- 6 - CLOCK 1.8432 MHZ (X2) ( PIN 32 WD2123 - U10)
- 7 - D10A/D10K (PIN 15,16 - U17)
- 8 - DRCV/SWC B (PIN 15,16 - U17)
- 9 - BASE Q1 ( D43C1 )
- 10- TIME ( PIN 12 - U17)
- 11- C1 ( PIN 10 - U17)
- 12- COLLECTOR Q1 ( D43C1 )
- 13- VOUT ( U16 )
- 14- VAC



## TROUBLESHOOTING

## VI - TROUBLESHOOTING

Since the FDD 3000 is composed of several blocks of Hardware and Software, it is necessary to fully identify the origin of the fault if it exists.

A good rule to troubleshoot is to always use a disc that you are absolutely shure that is 100 % working.

Connect all the different parts of the FDD 3000 according to what was described in chapter III as follows:

Power On the TV set or the MONITOR,  
switch on the FDD 3000 and computer.

Insert the System Disc into drive "A".

Verify if the power LED is ON:

If the LED in the FDD 3000 is OFF  
preform the operations described above in  
the reverse order.

Verify if the FUSE located in the rear  
pannel of the FDD conduts.

Verify if all the plugs are fully  
engaged, well inserted, and are in good  
conditions.

Verify if all the terminals are well  
crimped and inserted in the terminals  
and housings.

---

---

## FDD MANUAL

---

---

The LED on the Floppy Disc Drive remains ON but the systems fully operates, i.e. responds correctly to the commands

Verify if the line "IN USE" is not interrupted.

Make shure that the DIODE that connects the "IN USE" signal and ground is good.

Remember that this signal is active LOW meaning a very low voltage level near 0 V (0.4 v).

Verify that the signals in U8 change.

Verify if PIN 11 of the gatte array GA 121 is not short circuited.

Note: Configure the system to work in CP/M mode and load the the Operating System. In good circunstances the LED located in the drive blinks when loading the Operating System, in the end it should stay in the OFF stage. The Led would lit again only in case of a disc access. The "IN USE" signal is set ON or OFF after the drive selection preformed by the Operating System.

The LED remains always ON and the drive can only recognize the TRACK 00.

If the system is still configured to run CP/M verify if the communication cable with the console or computer is in perfect conditions ( all the lines warrantying galvanic connection between plugs).

Do a system RESET checking if the system is able to locate other tracks then only TRACK 00. If that happens the problem is located at the CONSOLE/COMPUTER level.

---

---

FDD MANUAL

---

---

---

After power on nothing happens:

Verify if the address lines of the controller are not stuck to a fix logic value.

Verify if the DATA lines are active changing logic values.

Verify if the PINS of RD and WR of the processor ( Z80 ) are accessed ( PINS 21 and 22 of U1 ).

---

---

FDD MANUAL

---

---

---

The system loads TOS but not the CP/M Operating System:

Verify if the SELECT lines of the different MEMORY banks are correctly selected by the signals "ASEL", "CP/M3", "CP/M2", "CP/M1" ( PINS 22, 23, 24, 25, of gatte array GA121 ).

Note: After POWER ON the bank automatically selected is the one corresponding to "ASEL".

Have in mind that for a correct selection of the MEMORY banks only one can be selected at the time ( EXOR ).

To check if the MEMORY selects are correct procede as follows: after insering the CP/M disc in drive "A" check, with the help of an Oscilloscope, PIN 22 of gatte array GA 121. If it must go LOW when trying to load the Operating System. Try to load the Operating System by pressing thr RESET button on the controller, all the lines should stay HIGH.

---

---

FDD MANUAL

---

---

---

It does not communicate with the interface although it loads TOS.

Verify if PIN 21 of gate array GA121 (MODE) is correctly selected in J1B and J1A. This PIN selects the type of communication to be performed ( BYTES or NIBLES).

Verify if all the DATA lines DO ( DATA OUT ) oscilate within the specified limits.

Verify if the DI lines ( DATA IN ) oscilate inside the correct values.

---

---

FDD MANUAL

---

---

---

After POWER ON nothing happens.

Verify if the system is powered.

Verify if there is no excess power consumption in any of the lines.

Verify the clock signal between PINS 44 and 45 of the GA 121.

Verify the 8 MHz signal at PIN 42 of gate array GA 121.

Verify if the contacts of the RESET button are free from any short-circuit.

Verify if the RESET line ( PIN 34) of the gate array GA 121 is at the logic state "1" HIGH.

Press the RESET button and check if the signal across PIN 34 of the same gate array varies inside the correct values.

Verify if the line " INT" is HIGH. Analyse the response of this signal with changes in the RESET.

Verify if the "HALT" command is activated.

Verify if the ON/OFF swicht is operational cutting or conducting when actuated.

Check if the soldered points between the the circuit and the secondary terminals of the transformer are in good conditions.

Check the primary of the transformer.

Measure the DC value of the SPSU output in the open circuit situation.

With the help of resistive loads on the secondary power lines check the voltage drop.

Verify that the power LED is in good conditions and the connecting wires also assure a reliable connection.

---

---

FDD MANUAL

---

---

---

Impossible to FORMAT a disc in drive "A".

Verify if the line "WD" is assuring a good connection between the drive and the PCB.

Check if the LOGIC level of PIN 22 of WD 1770 varies during the format operation.

Verify the inverter U-7 ( Gatte IN-11, Gatte OUT-10).

Even with the Hardware protection ( protection TAB of the disc - open ) it is possible to write in the disc.

Verify if the line "WRPT" assures the connection between the drive and PIN 10 of U-11 ( GA 121 ).

Verify the contact of pin of U-11 with the PCB.

The controller outputs to the RS 232 line but does not receive through the same line.

Verify if the +12V and -12V voltages are applied to the BUFFER's CONVERTERS U9 ( 1489 or 75189 ).

Check if the decoupling capacitors on the ENABLE pins are correctly inserted and in good working conditions.

Check if the ripple measured in the power lines are inside reasonable limits.

Verify if the clock signal that drives the communication has no other interfering signal in top of it.

Produce an artificial shunt between the input and output pins of the converter, with a very simple software program open a file sending after ASCII characters through the channel. this helps to check the efficiency of the level converters.

-----  
FDD MANUAL  
-----  
-----  
-----

After RESET the Controller reads TRACK 00 and the following 3 TRACKS

Verify if any line of U-3 ( ULA 1h035) is not stuck LOW.

Check if U-3 do not heat.

Verify the address lines of the microprocessor before and after the multiplex circuit.

Make shure that A0 and A1 change value allowing the execution of the first 4 TOS instructions addressed to the floppy disc drive.

| A0 | A1 |      | A0 | A1 |      |
|----|----|------|----|----|------|
| 0  | 0  | DS   | 0  | 0  | TK00 |
| 0  | 1  | MD   | 0  | 1  | IF   |
| 1  | 0  | DIR  | 1  | 0  | RD   |
| 1  | 1  | STEP | 1  | 1  | WFTR |

TRANSMIT

RECEIPT

Verify if the SELECT lines of the MEMORY banks of the gate array GAL21, after RESET are accessed. Check the status of PIN 9 of U-3 (256CS).

Verify if the Z80 line INT is active.

Verify PIN 28 ( WD 1770 - INT ).

Verify the oscillations in PIN 27 of Z80 M1 ( Machine Code Cycle ).

---

FDD MANUAL

---

---

---

The characters sent to the printer do not to the ones displayed on the screen.

Verify if the printer is configured in the same way as the controller.

BAUD RATE  
PARITY  
STOP BITS  
TEXT/BYTES  
BITS/CHARACTER

Check the ripple of the power supply are not out of the specified.

Check if there are no spurious signals on top of the clock signal of the communication IC.

Verify if the UART clock signal is stable.

Check if the decoupling capacitors of the level converters are in good conditions.

Check all the supply voltages at the communication converters.

---

FDD MANUAL

---

---

---

Loads TOS incompletely

Verify if the address lines multiplexed in U4 and U5 are stuck and do not oscillate.

Does not load TOS and does not select the drive.

Check if the multiplexer MUX U2 (74157) is commutating the address A4 and A15.

Verify if the control signal HPX PIN 14 of U3 (1H035) is not stuck LOW by GA121 or U3.

No VIDE0 output

Check if the pins of the RCA plug are well soldered.

Assure that the grounding of the cable is good.

Verify the connections of the curly cable.

Does not execute RESET because the RESET line is hold LOW.

Verify if C16 is not LOW.

Check if the PINS of the RESET Button are not short-circuited.

---

FDD MANUAL

---

MECHANICAL FAULTS

---

APPENDIX

---

APPENDIX I-1

---

The Operating System when invoked by mistake will produce an error message, which could be:

HARDWARE FAULT ON DISK

This message will be display whenever a READ or WRITE error occur during access to the disc.

The error can be located in the disc drive Hardware or by the disc itself. If the disc in drive A is not present this error may in some circumstances be generated. Note that the system can not operate without a disc in drive A.

A faulty disc can also generate this message.

DISC CORRUPTED

This message may be displayed if a disc is partially destroid due to magnetic fields, heat etc.

The Operating System loaded in the controller memory is corrupted by noise generated by electrical interference in the power line.

The new disc inserted in drive A has a new version of the Operating System or no Operating System was previuosly saved in the disc. In any of those re-BOOT the system.

If the message remains after checking al the above is because the disc is fault. try to FORMAT it again but before try to transfer the files to another disc which will work in the majority of the cases.

----- IS WRITE PROTECTED

Appears after a trial to erase a file or directory that is protected via the instruction "ATTR \* " ( software protection ).

This error may also be generated by "SAVE \*", "ERASE \*", "MOVE \*" and "OPEN \*".

---

APPENDIX I-2

---

DISK-----IS SOFTWARE W/P

It is not possible to create, update, erase, rename or even write protect a file or directory in a software protected disc with the instruction "ATTR \*" this error may be generated by the following instructions: "SAVE \*", "OPE \*", "CLOSE\* ", "DIM \*", "ERASE \*", "LET\* ", "MOVE\*", "PRINT\*", or "ATTR \*". Note that the instruction FORMAT does not produce this error since it formats the disc even if the same is software write protected.

DISC-----IS HARDWARE W/P

The disc is Hardware protected by the use of the protection TABS.

The instruction FORMAT \* (DISC) does not format a Hardware write protected disc but generates the error message "HARDWARE FAULT ON DISC -----".

DIRECTORY FULL ON DISC----

Generated whenever the systems tries to update a file however there is no space in the disc or directory. This error may occur with the following instructions "DIM \*", "CLOSE \*", "MOVE \*" and "SAVE \*".

A drive directory accepts up to 128 files entry. The disc itself uses one entry for its own name. Any other directory uses one entry. The files occupy one entry for each 16k length. This means that a file with 7k uses only one entry and a file with 50k uses 4 .

---

---

FDD MANUAL

---

---

---

ILLEGAL USE OF A ROOT NAME

it is impossible to specify a name for the root or from the root. This error may be generated by the instructions that have a PATHNAME as argument; "GOTO \*", "GOSUB \*" AND "CAT \*" ( AT THE ROOT LEVEL), also "DIM \*", "ERASE \*", "SAVE \*", (FROM THE ROOT).

CHANNEL BUSY

The instruction "OPEN \*" generates this an error message when trying to specify this channel. Change channel or firts close the OPEN file.

CHANNEL NOT OPEN

The instructions that generate this error message are "CLOSE \*", "LIST \*", "PRINT \*", "INPUT \*" or "RESTORE \*" when trying to use a channel not previously OPEN.

ILLEGAL DRIVE NAME

The instruction FORMAT \* generates this error if the specified drive is wrong.

ILLEGAL DISC NAME

The instruction FORMAT \* generates this error message when a non valid name is specified for the drive in the second ALFANUMERIC entry. It is necessary to specify a disc name or directory and not a PATHNAME.

DISC -----FULL

The disc as no free space, this error is generated by the instructions "SAVE \*", "MOVE \*" OR "PRINT \*".

THIS ARE THE TOS ERROR MESSAGES



---

APPENDIX III

---

PARTS LIST

---

APPENDIX IV

---

SPARE PARTS LIST

| ITEM | PART NAME                             | P/NUMBER   | %  |
|------|---------------------------------------|------------|----|
| 0035 | DISKETTE, TOS VA.2                    | 398-917020 | 5% |
| 0040 | DISKETTE, CP/VH VA.2                  | 398-909464 | 5  |
| 0170 | DRIVE, HITACHI HFD350SX               | 394-970100 | 1  |
| 0180 | FUSE, SLOW BLOW 250V/0A5              | 396-920033 | 10 |
| 0275 | ASSY, CONN. CENTER END 21 WAYS        | 301-970110 | 5  |
| 0285 | CORD, WITH MOULD'D" PLUG FEM 15WAYS   | 395-930035 | 2  |
| 0295 | SUB. ASSY, PUSH BOTTOM                | 301-975110 | 8  |
| 0405 | CONN.34 POSITIONS EDGE CARD           | 395-930055 | 1  |
| 0415 | CABLE, FLAT 34 COND. 28AWG, GRAY      | 395-930040 | 2  |
| 0440 | PLUG, HOUSING, 4 POS                  | 396-920050 | 3  |
| 0460 | CONN. IN PIN HEADER 2x17 PINS DBL ROW | 301-945004 | 2  |
| 0465 | PLUG, PCB MOUNTING, 4 WAYS            | 395-930069 | 5  |
| 0470 | PLUG, RGT. ANG. PCB MOUNTING 4 WAYS   | 395-930070 | 5  |
| 0490 | CONN. "D" TYPE 9 PINS SOCKET DGR MALE | 301-940200 | 5  |
| 0505 | CONN. LOCK HD. STRAIGHT POS 21 WAYS   | 395-930084 | 5  |
| 0510 | XTAL. 16MHZ, HC-18/U                  | 395-930060 | 2  |
| 0515 | XTAL. 1.8432MHZ, HC-18U               | 395-930066 | 2  |
| 0520 | IC, CPU HK3880-280                    | 330-910201 | 5  |
| 0530 | IC, WD1170                            | 395-930010 | 5  |
| 0535 | IC, WD2123                            | 395-930015 | 5  |
| 0600 | IC, 75188                             | 395-930085 | 3  |
| 0605 | IC, 75189                             | 395-930090 | 3  |
| 0635 | IC, 74LS273                           | 395-930025 | 3  |
| 0645 | IC, DRAW 4454-150NS                   | 335-912115 | 5  |
| 0650 | IC, ULA 311035                        | 334-812014 | 3  |
| 0811 | HUB ASSY-PCB                          | 301-975130 |    |
| 0812 | IC, EFGJ121                           | 395-930105 | 5  |
| 0813 | IC, EFGJ151                           | 395-930065 | 5  |

---

APPENDIX V

---

LISTING (BASIC)UBIOS

\*\*\*\*\*

BIOS

Versao para densidade dupla

VERSAD FINAL EM : 85/10/29

ACTUALIZACAO : 3/6/86

BY : Antonio Nobrega

Esta versao substitui a versao de 20/1/83. O novo BIOS tem processamento de erro para recuperacao de problemas de hardware. Emendado a nao detecao de erro de read a segunda tentativa

\*\*\*\*\*

CP/M REFERENCE CONSTANTS

|       |     |                      |                                   |
|-------|-----|----------------------|-----------------------------------|
| IMAG  | EQU | 100H                 | ;offset para a REL -1 ou 0        |
| MSIZE | EQU | 20                   | ;Capacidade da memoria em K bytes |
| IAS   | EQU | (MSIZE-20)*1024+IMAG | ;Deslocamento em relacao aos      |
| CCP   | EQU | BIAS                 | ;20 K bytes                       |
| BDOS  | EQU | CCP+806H             |                                   |
| CBIOS | EQU | CCP+1600H            |                                   |

|        |     |       |                                       |
|--------|-----|-------|---------------------------------------|
| IOBYTE | EQU | 0003H | ;Estrutura do IOBYTE nao implementado |
| CDISK  | EQU | 0004H | ;Drive corrente                       |
| NDRIVE | EQU | 4     | ;Numero de drives previsto            |
| NTENT  | EQU | 10    | ;Numero de tentativas em caso de erro |
|        |     |       | ;no acesso a disco                    |
| SECLDG | EQU | 128   | ;Sector logico                        |
| SECFIS | EQU | 256   | ;Sector fisico                        |
| SECI   | EQU | 0     | ;Sector fisico representado em 1 byte |

Caracteres ASCII

|        |     |     |                  |
|--------|-----|-----|------------------|
| LF     | EQU | 0AH | ;Line feed       |
| CR     | EQU | 0DH | ;Carriage return |
| FFFEED | EQU | 0CH | ;Form feed       |
| EOF    | EQU | 1AH | ;End of file     |
| ESCAPE | EQU | 1BH |                  |

Protocolo comunicacoes CONIN CONOUT

|         |     |       |  |
|---------|-----|-------|--|
| ASTAT   | EQU | 0     |  |
| BSTAT   | EQU | 40H   |  |
| CSTAT   | EQU | 10H   |  |
| DSTAT   | EQU | 50H   |  |
| STMSK   | EQU | DSTAT |  |
| COM     | EQU | 2FH   | ;Endereco do porto de comunicacoes       |
| SERVICO | EQU | 0FH   |  |
| MASK    | EQU | 5FH   |  |
| BYTEON  | EQU | 0AH   |  |
| STANDBY | EQU | 05H   |  |
| WSTA    | EQU | 06H   | ;Flag indica conin esta espera caractere |

\*\*\*\*\*  
\* Descricao do Hardware \*  
\*\*\*\*\*

comunicacao serie  
=====

Chip WD 2123

=====

Enderecos:

```
CANALA EQU 80H ;Endereco do porto A
SI0ASTS EQU CANALA + 1 ;Endereco do registro de status
SI0ACHD EQU CANALA + 1 ;Endereco do registro de comando
SI0ADAT EQU CANALA ;Endereco do registro de data
SI0ABR EQU 10H ;Endereco do registro de baud/rate
```

```
CANALB EQU 40H ;Endereco do porto B
SI0BSTS EQU CANALB+1 ;Endereco do registro de status
SI0BCMD EQU CANALB+1 ;Endereco do registro de comando
SI0BDAT EQU CANALB ;Endereco do registro de data
SI0BBER EQU 11H ;Endereco do registro de baud/rate
```

Comandos de programacao

```
ACDH EQU 00000111B ;input e output enable
```

Flags do registro de status

```
RXRDY EQU 00000001B ;Receiver READY
RR EQU 01111000B ;Erro na recepcao ou na emissao
IDERR EQU 37H ;Palavra de comando com reset flag
SI0WORD EQU 27H ;palavra de comando com RQT="0"
PTS EQU 20H
IOMASK EQU 81H
```

Controlador de disco

=====

Chip FD1770

=====

Enderecos:

```
FDC EQU 000H
FDCSTS EQU FDC ;Reg de stat
FDCMD EQU FDC ;Reg de comando
FDCTRK EQU FDC + 1 ;Reg da pista
FDCSEC EQU FDC + 2 ;Reg de sector
FDCDAT EQU FDC + 3 ;Reg de dados
```

Comandos de programacao

```
STCHD EQU 05H ;Restore
ERCHD EQU 15H ;Seek c/ verify
STICHD EQU 55H ;Step in
PTOCHD EQU 75H ;Step out
RDSCHD EQU 80H ;Read sector
RDMCHD EQU 98H ;Read multiple sectors
WRSCHD EQU 0A0H ;Write sector c/ precompensacao
RACHD EQU 0C0H ;Read address
RICHD EQU 0D0H ;Force interrupt
```

Flags do status

```
BUSY EQU 0 ;Busy
DLOST EQU 2 ;Data lost
FOUND EQU 4 ;Record not found
PROTE EQU 6 ;Write protect
```

```

;
; Outros registos
; =====
DRQ EQU 02FH ;Registo de comunicacoes
; bit 0-6 usados comunicacao em nibbles
; bit 7 DRQ
HRD EQU 0EOH ;Registo de controle
; bit 0 - drive sel 0
; bit 1 - drive sel 1
; bit 2 - drive sel 2
; bit 3 - drive sel 3
; bit 4 - side select
; bit 5 - /EDEN ( densidade)
; bit 6 - /BOOT
; bit 7 - IN USE

```

```

;
; Programacao
;
DENSID EQU 01011111B ;Programacao do porto de controle
INTPRG EQU 11011111B
MSKDRV EQU 00001111B ;Mascara para os drives
LUZ EQU 7 ;Acende luz no drive seleccionado

```

```

;
; *****
; * Tabela de JUMPs do BIOS *
; *****

```

```

JP BOOT
JP WBOOT
JP CONST
JP CONIN
JP CONOUT
JP IHPR
JP PUNCH
JP READER
JP HOME
JP SELDSK
JP SELTRK
JP SETSEC
JP SETDMA
JP READ
JP WRITE
JP IMPRST
JP SECTRAN

```

```

;
; LOGO inicial da versao do bios
;

```

```

SIGNON:: DB ESCAPE
DEFM "H"
DB ESCAPE
DEFM "J"
DEFM "CP/M Version 2.2"
DB CR,LF
DEFM "Copyright by DIGITAL RESEARCH, Inc."
DB CR,LF,LF
DEFM "Cbios Version A1.1 Copyright by THX PORTUGAL"
DB CR,LF,LF
DEFM "3, June 1986"
DB CR,LF,LF
DEFM "$"

```

```

;
; *****
; * Subrotinas do BIOS *
; *****

```

\*\*\*\*\*

\*\*\*\*\*  
\* BOOT \*  
\*\*\*\*\*

; Procedimento a ser executado depois do LOADER  
; em que inicializa o IOBYTE o disco corrente a  
; zero, e afixa a mensagem inicial. Por fim ini-  
; cializa a page 0 do CP/M com os valores conve-  
; nientes passando em seguida o comando ao CCP.  
;

```
BOOT:: XOR A
LD (IOBYTE),A ;Reset IOBYTE
LD (CDISK),A ;Inicializa o drive selecionado
LD (UNIT),A
CALL INIT
CALL INITRS ;inicializa canais rs_232
LD HL,SIGNON
CALL PMSG ;Imprimir a mensagem inicial
JP GOCPM
```

\*\*\*\*\*  
\* WBOOT \*  
\*\*\*\*\*

; Inicializa o hardware, carrega o CCP e o BDOS  
; e inicializa a pagina 0 do CP/M passando em se-  
; guida o comando ao CCP.

```
WBOOT:: LD SP,STACK
DI
CALL INIT
XOR A
CALL LIGHT
LD HL,CCP ;Endereco de base do CP/M
LD C,FDCDAT
LD B,0
LD D,1
WBLOP:: LD A,D
CALL RDLOOP
JR NZ,WBOOT
LD A,(OH
INC D
CP D
JR NZ,WBLOP
CALL STEPIN ;pista 1
XOR A
CALL RDLOOP ;Sector 0 pista 1
CALL LIGHTOF
```

```
GOCPM:: LD A,0C3H
LD (00H),A
LD HL,CBIOS+3
LD (01H),HL
LD (05H),A
LD HL,BDOS
LD (06H),HL
XOR A
LD (BLREAD),A ;Nada no BUFFER de blocagem
LD (BLALT),A
LD BC,0080H
CALL SETDHA
LD A,(CDISK)
LD C,A
```

```

*****
*                               Subrotinas de I/O de caracteres                               *
*****

```

```

*****
*                               CONST                               *
*****

```

```

; Indica o status da consola
; .parametros de saida
;     reg. A - FF se existe caracter
;           - 00 se nao existe caractere
; .altera o registo A

```

```

CONST:: IN      A,(COM)
        AND     MASK
        CP      04
        JR      Z,CONSON
        XOR     A
        RET

CONSON:: LD      A,OFFH
        RET

```

```

*****
*                               CONIN                               *
*****

```

```

CONIN:: LD      (SVSTK),SP
        LD      SP,STACK
        CALL    RXBYTE ;Recebe byte do terminal
        LD      C,A
        LD      SP,(SVSTK)
        RET

```

```

*****
*                               CONOUT                              *
*****

```

```

CONOUT:: LD      (SVSTK),SP
        LD      SP,STACK
        LD      A,C
        CALL    TXBYTE ;Transmite byte
        LD      SP,(SVSTK)
        RET

```

```

*****
*                               IMPR                               *
*****

```

```

IMPR:: LD      (SVSTK),SP
        LD      SP,STACK
        CALL    PRINTER
        LD      SP,(SVSTK)
        RET

```



```
LD      (TRACK), A
RET
```

```
*****
*                               SETSEC                               *
*****
```

```
SETSEC:: LD      A, C
DEC      A
LD      (SECTOR), A
RET
```

```
*****
*                               SETDMA                               *
*****
```

```
SETDMA:: LD      (POINTR), BC
RET
```

```
*****
*                               SELDSK                               *
*****
```

```
SELDISK:: LD      HL, 0
LD      A, C
CP      NDRIVE
RET     NC
LD      (UNITDR), A
LD      L, A
ADD     HL, HL
ADD     HL, HL
ADD     HL, HL
ADD     HL, HL
LD      DE, DPHTAB
ADD     HL, DE
RET
```

```
*****
*                               HOME                                *
*****
```

```
HOME:: LD      A, (UNITDR)
LD      C, A
CALL   LIGHT
LD      A, RSTCMD
OUT    (FDCMD), A
CALL   TEMP
HOMLP:: IN     A, (FDCSTS)
BIT    BUSY, A
JR     NZ, HOMLP
CALL   LIGHTOF
XOR    A
RET
```

```
*****
*                               READ                                *
*****
```

\*\*\*\*\*

```
ZAD:: LD      (SVSTK),SP
      LD      SP,STACK
      CALL   SAVJMP ;guarda salto de int e coloca novo jmp
      LD      A,(BLREAD) ;Testar se o BUFFER ja foi
      AND     A ;lido alguma vez
      JR      Z,RDCONT
      CALL   IGUAL ;Testa se os sectores no BUFFER
                  ;e para ser lido sao iguais
                  ;se SIM
                  ;          entao A = 0
                  ;          senao A = FF

      AND     A
      JR      Z,REXISTE
      LD      A,(BLALT) ;Testa se o sector no BUFFER
                  ;foi alterado
                  ;se SIM
                  ;          entao "blalt" = FF
                  ;          senao "blalt" = 0

      AND     A
      JR      Z,RDCONT
      CALL   WR256
      AND     A
      JR      NZ,RDFIM
CONT:: LD      HL,BLDRIV
      LD      A,(UNITDR)
      LD      (HL),A ;Bl driv ← UNITdr
      INC     HL
      LD      A,(TRACK)
      LD      (HL),A ;Bl pist ← track
      INC     HL
      LD      A,(SECTOR)
      LD      (HL),A ;Bl sect ← sector
      INC     HL
      XOR     A
      LD      (HL),A ;Blalt ← 0 (* nao alterado *)
      DEC     A
      LD      (BLREAD),A
      CALL   RD256
      AND     A
      JR      NZ,RDFIM ;Erro
      LD      (BLALT),A
REXISTE:: CALL   RTRANSF
          XOR     A
DFIM:: LD      (ERRORD),A ;flag de erro de leitura
      CALL   PUTJMP ;repoi salto
      LD      SP,(SVSTK)
      RET
```

```
*****
*                               WRITE                               *
*****
```

```
WRITE:: LD      (SVSTK),SP
        LD      SP,STACK
        CALL   SAVJMP ;guarda 32h
        PUSH   BC ;Salvar se e acesso a directoria
                  ;se e o primeiro ou se e um acesso
                  ;normal
        LD      A,(BLREAD) ;Testa se o BUFFER ja foi
        AND     A ;lido alguma vez
        JR      Z,WRCONT
```

```

CALL    IGUAL    ;Testa se os sectores no BUFFER
                ;e para ser lido sao iguais
                ;se SIM
                ;
                ;      entao A = 0
                ;      senao A = FF

```

```

AND     A
JR      Z,WEXISTE
LD      A,(BLALT) ;Testa se o sector no BUFFER
                ;foi alterado se SIM
                ;      entao "blalt" = FF
                ;      senao "blalt" = 0

```

```

AND     A
JR      Z,WRCONT
CALL    WR256
AND     A
JR      NZ,WRFIM

```

```

WRCONT:: LD      HL,BLDRIV
LD      A,(UNITDR)
LD      (HL),A      ;Bl driv <-- UNITdr
INC     HL
LD      A,(TRACK)
LD      (HL),A      ;Bl pist <-- track
INC     HL
LD      A,(SECTOR)
LD      (HL),A      ;Bl sect <-- sector
INC     HL
XOR     A
LD      (HL),A      ;Bl alt <-- 0 (* nao alterado *)
DEC     A
LD      (BLREAD),A
CALL    RD256
AND     A
JR      NZ,WRFIM    ;Erro

```

```

WEXISTE:: CALL    WTRANSF
POP     BC
LD      A,C
CP      1
JR      NZ,WRNDIR
CALL    WR256
AND     A
JR      NZ,WRFIM
LD      (BLREAD),A

```

```

WRNDIR:: XOR     A
JR      WRFIM

```

```

WRFIM:: POP     BC
WRFIM:: CALL   PUTJMP
LD      SP,(SVSTK)
RET

```

```

;
; *****
;                               OUTRAS
; *****
;

```

```

;      *** init ***
;

```

```

; Inicializa todo o hardware
; .controlador de disco
;

```

```

INIT:: LD      A,INTPRG
OUT     (HRD),A
LD      B,NDRIVE
LD      A,OFEH
INTLP:: OUT    (HRD),A

```

```

RLC      A
DJNZ    INTLP
LD      A,DENSID
OUT     (HRD),A
AND     OFEH
OUT     (HRD),A
LD      (IMAGSEL),A
LD      A,RSTCMD
OUT     (FDCMD),A
CALL    TEMP
INITLP:: IN      A,(FDCSTS)
        BIT     BUSY,A
        JR     NZ,INITLP
        XOR    A
        LD     (BLREAD),A
        LD     (BLALT),A
        LD     (UNIT),A
        DEC    A
        LD     B,NDRIVE
        LD     HL,DRVTBL
INLP::  LD     (HL),A
        INC    HL
        DJNZ  INLP
        CALL  LIGHTOF
        RET

;
;
;      *** RD256 ***
;
;Le 256 bytes para o BUFFER
;
RD256:: LD     A,(BLDRIV)
        LD     C,A
        CALL  LIGHT
        LD     A,NTENT
        LD     (RETRY),A
        CALL  PREPARA ;tenta dez vezes o seek
        AND   A
        JR   NZ,RDERR
RTRY::  LD     HL,BLBUFF ;endereco buffer
        CALL  RDDSK ;tenta leitura
        AND   A
        JR   Z,RFIM ;se o ok sai
        LD     A,(RETRY)
        DEC    A
        LD     (RETRY),A
        JR   NZ,RTRY
RDERR:: LD     B,255
RDOOUT:: CALL  LIGHTOF
        LD     A,B
        RET
RFIM::  LD     B,00
        JR   RDOOUT
RDDSK:: IM     1
        EI
        LD     A,RDSCMD
        OUT   (FDCMD),A
RLP::  IN     A,(DRQ)
        RLA
        JR   NC,RLP
        INI
        JR   RLP
;
;
;      *** WR256 ***

```

```

;
;
WR256:: LD      A,(BLDRIV)      ;Acesso ao drive do buffer
        LD      C,A
        CALL    LIGHT
        LD      A,NTENT
        LD      (RETRY),A      ;10 tentativas
        CALL    PREPARA ;tenta escrita
        AND     A      ;erro?
        JR      NZ,WRERR      ;sim jump sai
WRTRY:: LD      HL,BLBUFF
        CALL    WRDSK      ;tenta escrita
        AND     A      ;erro?
        JR      Z,WFIM
        BIT     6,A      ;disco protegido?
        CALL    NZ,DSKPRT      ;sim logo para consola
        CP     "R"      ;retry?
        JR      Z,WRTRY ;sim tenta nova escrita
        LD      A,(RETRY)      ;tenta dez vezes
        DEC     A
        LD      (RETRY),A
        JR      NZ,WRTRY
WRERR:: LD      B,255      ;flag erro
WRQUT:: CALL    LIGHTOF ;desliga led
        LD      A,B
        RET
WFIM::  LD      B,0      ;flag write ok
        JR      WRQUT
WRDSK:: EI
        LD      C,FDCDAT      ;endereço porto data
        LD      A,WRSCMD
        OUT     (FDCMD),A
WRLP::  IN      A,(DRQ)
        RLA
        JR      NC,WRLP
        OUTI
        JR      WRLP
;
;

```

```

;
;      *** PRINTER ***
;

```

```

;Transmite byte recebido registo C
;para canal serie seleccionado
;

```

```

PRINTER:: LD      A,SIOWORD
        OUT     (SIOACMD),A
        CALL    ERROA
IMPHR::  IN      A,(SIOASTS)
        BIT     7,A      ;cts low ?
        JR      Z,IMPHR ;nao espera
        BIT     0,A      ;registo de tx. esta ready?
        JR      Z,IMPHR ;nao espera
IMPTX::  LD      A,C
        OUT     (SIOADAT),A
        RET
;
;

```

```

;
;      *** PRINSTS ***
;

```

```

;Da o estado do periferico (rs232)
;associado com o canal seleccionado
;

```

```

;
;
PRINSTS:: CALL    ERROA

```

```

IN      A,(SIOASTS)
BIT     7,A
JR      NZ,LPTON
XOR     A
RET
PTON:: LD  A,OFFH
RET

;
;          *** FUN: ***
;
;Out do caractere pelo canal B
;
PUN::  LD  A,SIOWORD
      OUT (SIOBCMD),A
UNCTS:: IN  A,(SIOBSTS)
      AND 81H ;espera cts low e thr empty
      CP  81H
      JR  NZ,PUNCTS
      LD  A,C
      OUT (SIOBDAT),A
      RET

;
;          *** RDR: ***
;
;Le caractere do canal B
DR::  LD  A,SIOWORD
      OUT (SIOBCMD),A ;forca CTS low
RDRLP:: IN  A,(SIOBSTS) ;get status
      AND 7FH ;limpa cts
      CP  8 ;testa erro
      JR  NC,RDRERR
      BIT RXRDY,A ;byte ready?
      JR  Z,RDRLP ;espera loop
      IN  A,(SIOBDAT) ;le byte
      LD  C,A
      LD  A,ACOM ;forca CTS a high
      OUT (SIOBCMD),A
      LD  A,C
      RET
RDRERR:: LD  A,37H
      OUT (SIOBCMD),A
      JR  RDRLP

;
;          *** WTRANSF ***
;
;Transfer 128 bytes do endereco POINTR
;para o BUFFER - high ou low.
;
TRANSF:: LD  HL,BLEBUFF
      LD  BC,SECLOG
      LD  A,(HIGH)
      AND A
      JR  Z,WLOW
      ADD HL,BC
LOW::  EX  DE,HL
      LD  HL,(POINTR)
      LDIR
      XOR A
      DEC A
      LD  (BLALT),A ;PROVISORIO

```

RET

\*\*\* rtransf \*\*\*

;Transfer 128 bytes do BUFFER -high ou low-  
; para o POINTR.

```
RTRANSF::      LD      HL,BLBUFF
               LD      BC,SECLOG
               LD      A,(HIGH1)
               AND     A
               JR      Z,RLOW
               ADD     HL,BC
RLOW::         LD      DE,(POINTR)
               LDIR
               RET
```

\*\*\* igual \*\*\*

;Testa se o sector que esta no BUFFER e  
;igual ao que se quer ler ou escrever

```
IGUAL::        PUSH    IX
               LD      IX,BLDRIV
               LD      A,(UNITDR)
               CP      (IX+0)
               JR      NZ,DIFER
               LD      A,(TRACK)
               CP      (IX+1)
               JR      NZ,DIFER
               LD      A,(SECTOR)
               CP      (IX+2)
               JR      NZ,DIFER
               LD      A,(ERRORD) ;ve se houve erro no read anterior
               AND     A
               JR      NZ,DIFER
               POP     IX
               RET
DIFER::        LD      A,OFFH
               POP     IX
               RET
```

;Entry: B=Sector Count, C=Starting Sector, HL=Start Address  
;Reads every other sector (1 sector interleve)

\*\*\* RDLOOP \*\*\*

```
RDLOOP::      OUT      (FDCSEC),A
               LD      A,RDSCMD
               OUT     (FDCMD),A
RLLP::        IN      A,(DRQ)
               RLA
               JR      NC,RLLP
               INI
               JR      NZ,RLLP
RELLP::       IN      A,(FDCSTS)
               BIT     BUSY,A
               JR      NZ,RELLP
               AND     1CH
               RET
```

```

;
;
;          *** point ***
;
;Devolve no reg HL o apontador do
;descriptor do drive especificado
;no reg. A
;
;
POINT:: LD      HL,DRVTBL
        AND     A
PNT1::  RET     Z
        INC     HL
        DEC     A
        JR      PNT1
;
;
;          *** setpin ***
;
;Faz avançar a cabeça do drive de uma
;posição
;
;
STEPIN:: LD      A,STICMD
        OUT     (FDCMD),A
        CALL   TEMP
SETPLP:: IN      A,(FDCSTS)
        BIT    BUSY,A
        JR     NZ,SETPLP
        RET
;
;
;          *** light ***
;
;Seleciona o drive e acende a luz
;do mesmo.
;
;
LIGHT:: LD      C,A
        LD      (TUNIT),A
        LD      A,(IMAGSEL)
        RES    LUZ,A
        OR     MSKDRV
        LD      (IMAGSEL),A
        OUT    (HRD),A
        LD      A,(UNIT)
        CALL   POINT
        IN     A,(FDCTRK)
        LD      (HL),A
        LD      B,OFEH
LIGCNT:: DEC     C
        JP     M,LIGEND
        RLC    B
        JR     LIGCNT
LIGEND:: LD      A,(IMAGSEL)
        AND    B
        OUT    (HRD),A
        LD      (IMAGSEL),A
        LD      A,(TUNIT)
        LD      (UNIT),A
        CALL   POINT
        LD      A,(HL)
        CP     OFFH
        JR     NZ,LIGNRST
        LD      A,RSTCMD
        OUT    (FDCMD),A

```

```

CALL      TEMP
LIGLP::  IN      A,(FDCSTS)
        BIT     BUSY,A
        JR     NZ,LIGLP
        XOR    A
LIGNRST:: OUT     (FDCTRK),A
        RET

```

```

;
;
;      *** Inirs ***
;

```

```

;Inicializa os canais com os parametros
;dados pela tabela SIOTAB
;

```

```

INITRS:: LD      HL,SIOTAB      ;Inicializa HL com ADD tabela
CALL     PUTMOD ;Constroi a palavra de modo
OUT      (SIDACMD),A
LD      A,ACOM
OUT      (SIDACMD),A      ;Palavra de comando
LD      A,(HL)
OUT      (SIDABR),A      ;Baud rate
;
;Canal A programada
;
INC      HL      ;Inicializa hl com parametros canal B
CALL     PUTMOD
OUT      (SIDBCMD),A
LD      A,ACOM
OUT      (SIDBCMD),A
LD      A,(HL)
OUT      (SIDBBR),A
RET

```

```

;
;
;      *** Putmod ***
;

```

```

;Forma a palavra de modo
;

```

```

PUTMOD:: LD      A,(HL) ;Baud rate factor
INC      HL
ADD     A,(HL) ;Bits/char
INC      HL
ADD     A,(HL) ;Parity enable
INC      HL
ADD     A,(HL) ;Parity select
INC      HL
ADD     A,(HL) ;Stop bits
INC      HL ;Endereca baud rate
RET

```

```

;
;
;      *** lightoff ***
;

```

```

;A apaga a luz do drive selecionado
;

```

```

LIGHTDF:: LD      A,(IMAGSEL)
SET     LUZ,A
LD      (IMAGSEL),A
OR      MSKDRV
OUT     (HRD),A
LD      A,(IMAGSEL)
OUT     (HRD),A

```

```
RES    LUZ,A
LD     (IMAGESEL),A
RET
```

```
;      ***TXBYTE***
;Transmite um byte para a consola
```

```
TXBYTE::    PUSH    DE
            PUSH    BC
            LD     (CHATMP),A      ;Guarda caractere temporario
            LD     B,A      ;Caractere em A
TX0::      CALL   TXBY
            JR     Z,TXOUT
TX1::      IN     A,(COM)
            AND    5FH
            CP     07
            JR     NZ,TX1
            LD     A,(CHATMP)
            LD     B,A
            JR     TX0
TXOUT::    XOR    A
            POP    DE
            POP    BC
            RET
```

```
TXBY::     LD     A,B
            AND    0FH
            LD     E,A      ;E Fica com 1.o nibble
            LD     A,B
            AND    0F0H
            RRCA
            RRCA
            RRCA
            RRCA
```

```
READY::   LD     D,A      ;D Fica com 2.o nibble
            LD     A,BYTEON
            OUT   (COM),A
            IN    A,(COM)
            AND   MASK     ;Deixa passar bits 6,4,3,2,1,0
            CP   SERVICO
            JR    NZ,READY
            LD   A,E
            OR   CSTAT
            OUT  (COM),A ;Tx. 1.o nibble e bit 4 a "1"
            LD   B,A
            LD   C,0
```

```
TXTW1::   IN    A,(COM) ;Espera eco
            DEC   C
            JR    Z,TXERR
            AND   MASK     ;Limpa bits 7 e 5
            CP   B
            JR    NZ,TXTW1      ;eco diferente espera
            LD   A,D
            OR   BSTAT     ;Set flag
            OUT  (COM),A ;TX 2.o nibble e bit 6 a "1"
            LD   B,A
            LD   C,0
```

```
TXTW2::   IN    A,(COM) ;Espera eco
            DEC   C
            JR    Z,TXERR
            AND   MASK
```

```

CP      B
JR      NZ, TXTW2
LD      A, STMSK ;Envia fim de caractere
OUT     (COM), A
XOR     A
RET
TXERR:: LD      A, 07H
OUT     (COM), A
AND     A          ;SET FLAG ERRO
RET
;
;
;      ***RXBYTE***
;Recebe um byte
;
;
RXBYTE:: PUSH    BC
        DE
RX0::   CALL    RXBY
JR      Z, RXOUT ;Byte ok sai
RX1::   IN      A, (COM)
AND     5FH
CP      7
JR      Z, RX1
JR      RX0
RXOUT:: LD      A, B          ;COLOCA CARACTERE EM A
        DE
        POP    BC
        RET
;
;
;
RXBY::  LD      A, WSTA ;Flag estou espera byte
OUT     (COM), A
RXTW0:: IN      A, (COM)
AND     MASK
CP      STANDBY
JR      NZ, RXBY
LD      A, 0FH
OUT     (COM), A ;Transmite flag ready
LD      C, 0
RXTW1:: IN      A, (COM)
DEC     C
JR      Z, RXERR
LD      E, A
AND     STMSK
CP      CSTAT
JR      NZ, RXTW1
LD      A, E
OUT     (COM), A ;Transmite eco
AND     0FH
LD      E, A          ;Guarda 1.o nibble
LD      C, 0
RXTW2:: IN      A, (COM)
DEC     C
JR      Z, RXERR
BIT     6, A          ;Testa se e 2.o nibble
JR      Z, RXTW2
OUT     (COM), A ;Transmite eco
AND     0FH
RLCA
RLCA
RLCA
RLCA
OR      E

```

```

AND      7FH
LD       B,A      ;Guarda caractere
RXTW3::  IN       A,(COM)
AND      5FH
CP       7
JR       Z,RXERR
CP       STMSK
JR       NZ,RXTW3
LD       A,STMSK ;fora de servico
OUT      (COM),A
XOR      A
RET
RXERR::  LD       A,07H
OUT      (COM),A
AND      A
RET
;       *** temp ***
;
;Temporizador de 50 uS
;
;
TEMP::   PUSH     BC
LD       B,18
DJNZ    $
POP      BC
RET
;
;
;       *** seek ***
;
;Posiciona a cabeca do drive selecionado na
;pista especificada na posicao de memoria
;"track".
;
;
SEEK::   PUSH     BC
LD       A,NTENT
LD       (TENTAT),A      ;n.o de tentativas de seek
IN       A,(FDCTRK)
LD       B,A
LD       A,(BLPIST)
CP       B
JR       Z,SEKOK        ;ja la estava !!
SEEKLP:: LD       A,(BLPIST)
OUT      (FDCDAT),A
CALL    SEKINT      ;tenta seek
AND     10111111B   ;mascara bit write protected
AND     A           ;erro?
JR       Z,SEKOK    ;nao jump
CALL    HDME       ;forca restor
LD       A,(TENTAT) ;n.o de tentativas.
DEC     A
LD       (TENTAT),A
JR       NZ,SEEKLP   ;ate dez tentativas
INC     A           ;set erro seek
JR       SEKERR
SEKOK::  XOR      A
SEKERR:: POP      BC
RET
SEKINT:: LD       A,SEKCMD
IM      1
EI
OUT     (FDCMD),A      ;Comando de ir para a pista
;selecionada
JR      $           ;espera saida interrupt
;

```

```

;
;     *** prepara ***
;
;Prepara a leitura ou a escrita de
;um sector. Para isso actualiza o reg.
;e sector posiciona a cabeca na pista
;e prepara os registos
;
;
PREPARA::      LD      A,(BLSECT)
              OUT     (FDCSEC),A
              LD      C,FDCDAT      ;Endereco do reg. de dados
              JP      SEEK          ;Posiciona a cabeca
;
ERRDSK::      LD      HL,ABORT
              CALL    PMSG
ERRAGN::      CALL    RXBYTE
              RES     5,A
              PUSH   AF
              CALL    TXBYTE
              POP    AF
              CP     "A"
              JP     Z,WBOOT
              CP     "I"
              RET    Z
              CP     "R"
              JR     NZ,ERRAGN
              RET
;
; DSKPRT : coloca msg erro diskette protegida
;
DSKPRT::      LD      HL,MSGPRO
              CALL    PMSG
              JR     ERRDSK
;
; TIMEOUT : timeout do pun, e printer
;
; SAVJMP : guarda salto do int e coloca novo salto
;
SAVJMP::      PUSH   AF
              PUSH   BC
              LD     HL,38H
              LD     DE,BUFINT
              LD     BC,0003
              LDIR
              DEC   HL
              LD   DE,INTRPT
              LD   A,D
              LD   (HL),A
              LD   A,E
              DEC  HL
              LD   (HL),A
              DEC  HL
              LD   (HL),0C3H      ;jmp
              POP  BC
              POP  AF
              RET
;
; PUTJMP : repoi salto 38h
;
PUTJMP::      PUSH   AF          ;salva reg
              LD     DE,38H
              LD     HL,BUFINT
              LD     BC,0003
              LDIR

```

POP AF  
RET

INTRPT : devolve subrotina status do disco

INTRPT:: POP AF ;destroi ret  
IN A,(FDCSTS) ;le status  
AND SCH ;mascara bits erro  
RET

\*\*\* pmsg \*\*\*

;Imprime uma string que e apontada pelo  
;par HL e termina em \$.

PMSG:: LD A,(HL)  
LD C,A  
CP "\$"  
INC HL  
RET Z  
CALL TXBYTE ;transmite log cpm  
JR PMSG

ERROA:: IN A,(SIOASTS)  
AND ERR  
RET Z  
LD A,37H  
OUT (SIOACMD),A  
RET

ERROB:: IN A,(SIOBSTS)  
AND ERR  
RET Z  
LD A,37H  
OUT (SIOBCMD),A  
RET

\*\*\*\*\*  
\* Estruturas de dados, tabelas e mensagens \*  
\*\*\*\*\*  
\*\*\*\*\*  
\* MENSAGENS \*  
\*\*\*\*\*

MSGPRO:: DB ESCAPE,"Y",55,32  
DB LF,LF  
DEFM "Diskette hardware protected\$"

ABORT:: DB CR,LF  
DEFM "Abort, Ignore, Retry?#"

\*\*\*\*\*  
\* Tabelas \*  
\*\*\*\*\*

SECTBO:: DB 01,06,11,16,05,10,15,04  
DB 09,14,03,08,13,02,07,12

DISK PARAMETER BLOCK FOR STANDARD 3" FLOPPY  
Densidade dupla

```

DPBLKO:: DW      32
          DB      3
          DB      7
          DB      0
          DW     151
          DW      63
          DE     11110000B
          DE     00000000B
          DW      16
          DW      2
;
;
;

```

DISK PARAMETER HEADERS FOR A 4 DISK SYSTEM

```

DPHTAB:: DW      SECTB0,0000H      ;DPH FOR UNIT 0
          DW     0000H,0000H
          DW     DIRBUF,DPBLKO
          DW     CHK0,ALLO
;
          DW     SECTB0,0000H      ;DPH FOR UNIT 1
          DW     0000H,0000H
          DW     DIRBUF,DPBLKO
          DW     CHK1,ALL1
;
          DW     SECTB0,0000H      ;DPH FOR UNIT 2
          DW     0000H,0000H
          DW     DIRBUF,DPBLKO
          DW     CHK2,ALL2
;
          DW     SECTB0,0000H      ;DPH FOR UNIT 3
          DW     0000H,0000H
          DW     DIRBUF,DPBLKO
          DW     CHK3,ALL3
;
;
;

```

```

*****
*                               *
*      Estruturas de dados      *
*                               *
*****
;
;
;

```

```

SIDTAB:: DB      02,0CH,00,00,80H,08      ;Parametros canais RS-232
          DB      02,0CH,00,00,80H,08
BUFINT:: DS      3      ;buffer que guarda 3 bytes 38h
RETRY:: DS      1      ;Numero de tentativas de leitura/escrita
TENTAT:: DS      1      ;n.o de tentativas de seek
IMAGSEL:: DS     1      ;Imagem do registo de controle
HIGH1:: DS      1      ;Determ. sector HIGH ou LOW
DRVTBL:: DS      4      ;Tabela com a imagem dos drives
UNIT:: DS      1
TUNIT:: DS      1
UNITDR:: DS      1      ;Drive selecionado
TRACK:: DS      1      ;Pista selecionada
SECTOR:: DS      1      ;Sector selecionado
POINTR:: DS      2      ;Endereco selecionado
BLDRIV:: DS      1      ;Estrutura de dados para a blocagem/
BLPIST:: DS      1      ;Desblocagem
BLSECT:: DS      1
BLREAD:: DS      1
BLALT:: DS      1
BLBUFF:: DS      SECFIS
          DS      32      ;Stack
STACK:: DS      1
SVSTK:: DS      2      ;Variavel para armazenar o SP
CHATMP:: DS      1
;
;
;

```

\*\*\*\*\*  
\* Estruturas de dados do BDOS definidas no BIOS \*  
\*\*\*\*\*

```
DIRBUF::          DS          128          ;SCRATCH DIRECTORY BUFFER
ALLO::           DS          32          ;UNIT 0 ALLOCATION BUFFER
CHK0::           DS          16          ;UNIT 0 CHECK VECTOR
ALL1::           DS          32          ;UNIT 1 ALLOCATION VECTOR
CHK1::           DS          16          ;UNIT 1 CHECK VECTOR
ALL2::           DS          32          ;UNIT 2 ALLOCATION VECTOR
CHK2::           DS          16          ;UNIT 2 CHECK VECTOR
ALL3::           DS          32          ;UNIT 3 ALLOCATION VECTOR
CHK3::           DS          16          ;UNIT 3 CHECK VECTOR
ERRORD::         DS          1          ;FLAG DE READ READ ERRROR
;
END
```

LISTING (MACRO) BIOS

MDS-800 I/O Drivers for CP/M 2.2  
(four drive single density version)

Version 2.2 February, 1980

vers equ 22 ;version 2.2

Copyright (c) 1980  
Digital Research  
Box 579, Pacific Grove  
California, 93950

true equ 0ffffh ;value of "true"  
false equ not true ;"false"  
test equ false ;true if test bios

if test  
bias equ 03400h ;base of CCP in test system  
endif

if not test  
bias equ 0000h ;generate relocatable cp/m system  
endif

patch equ 1600h

org patch  
cpmb equ \$-patch ;base of cpm console processor  
bdos equ 806h+cpmb ;basic dos (resident portion)  
cpml equ \$-cpmb ;length (in bytes) of cpm system  
nsects equ cpml/128 ;number of sectors to load  
offset equ 2 ;number of disk tracks used by cp/m  
edisk equ 0004h ;address of last logged disk on warm start  
buff equ 0080h ;default buffer address  
retry equ 10 ;max retries on disk i/o before error

perform following functions

boot cold start

wboot warm start (save i/o byte)

(boot and wboot are the same for mds)

const console status

reg-a = 00 if no character ready

reg-a = ff if character ready

conin console character in (result in reg-a)

conout console character out (char in reg-c)

list list out (char in reg-c)

punch punch out (char in reg-c)

reader paper tape reader in (result to reg-a)

home move to track 00

(the following calls set-up the io parameter block for the  
mds, which is used to perform subsequent reads and writes)

seldisk select disk given by reg-c (0,1,2...)

settrk set track address (0,...76) for subsequent read/write

setsec set sector address (1,...,26) for subsequent read/write

setdma set subsequent dma address (initially 80h)

(read and write assume previous calls to set up the io parameters)

read read track/sector to preset dma address

write write track/sector from preset dma address

jump vector for individual routines

jmp boot

wboote: jmp wboot

jmp const

jmp conin

```

jmp      conout
jmp      list
jmp      punch
jmp      reader
jmp      home
jmp      seldsk
jmp      settrk
jmp      setsec
jmp      setdma
jmp      read
jmp      write
jmp      listst ;list status
jmp      sectran

```

```

maclib  diskdef ;load the disk definition library
disks   4       ;four disks
diskdef 0,1,25,6,1024,243,64,64,offset
diskdef 1,0
diskdef 2,0
diskdef 3,0
endef occurs at end of assembly

```

end of controller - independent code, the remaining subroutines are tailored to the particular operating environment, and must be altered for any system which differs from the intel mds.

the following code assumes the mds monitor exists at 0f800h and uses the i/o subroutines within the monitor

we also assume the mds system has four disk drives

```

evrt    equ      0fdh      ;interrupt revert port
ntc     equ      0fch      ;interrupt mask port
con     equ      0f3h      ;interrupt control port
te      equ      0111$1110b ;enable rst 0(warm boot), rst 7 (monitor)

```

mds monitor equates

```

mon80   equ      0f800h ;mds monitor
mon80   equ      0ff0fh ;restart mon80 (boot error)
i       equ      0f803h ;console character to reg-a
        equ      0f806h ;reader in to reg-a
        equ      0f809h ;console char from c to console out
io      equ      0f80ch ;punch char from c to punch device
        equ      0f80fh ;list from c to list device
        equ      0f812h ;console status 00/ff to register a

```

disk ports and commands

```

base    equ      78h      ;base of disk command io ports
stat    equ      base     ;disk status (input)
type    equ      base+1   ;result type (input)
byte    equ      base+3   ;result byte (input)

low     equ      base+1   ;iopb low address (output)
high    equ      base+2   ;iopb high address (output)

readf   equ      4h      ;read function
writef  equ      6h      ;write function
cal     equ      3h      ;recalibrate drive
rdy     equ      4h      ;i/o finished mask
cr      equ      0dh      ;carriage return
lf      equ      0ah      ;line feed

```

```

;signon: ;signon message: xxxk cp/m vers y.y
db      cr,lf,lf
if      test
db      '32' ;32k example bios
endif

```

```

if      not test
db      '00'      ;memory size filled by relocater
endif
db      'k CP/M vers '
db      vers/10+'0','.',vers mod 10+'0'
db      cr,lf,0

```

```

boot:   ;print signon message and go to ccp
        (note: mds boot initialized iobyte at 0003h)
        lxi      sp,buff+80h
        lxi      h,signon
        call     prmsg      ;print message
        xra     a          ;clear accumulator
        sta     cdisk      ;set initially to disk a
        jmp     gocpm      ;go to cp/m

```

boot;; loader on track 0, sector 1, which will be skipped for warm read cp/m from disk - assuming there is a 128 byte cold start start.

```

        lxi      sp,buff ;using dma - thus 80 thru ff available for stack

```

```

        mvi     c,retry ;max retries
        push    b
boot0:  ;enter here on error retries
        lxi     b,cpmb  ;set dma address to start of disk system
        call    setdma
        mvi     c,0     ;boot from drive 0
        call    seldisk
        mvi     c,0
        call    settrk  ;start with track 0
        mvi     c,2     ;start reading sector 2
        call    setsec

```

```

        read sectors, count nsects to zero
        pop     b       ;10-error count
        mvi     b,nsects

```

```

dsec:   ;read next sector
        push    b       ;save sector count
        call    read
        jnz     booterr ;retry if errors occur
        lhld   iod      ;increment dma address
        lxi     d,128   ;sector size
        dad    d        ;incremented dma address in hl
        mov     b,h
        mov     c,l     ;ready for call to set dma
        call    setdma
        lda     ios     ;sector number just read
        cpi    26      ;read last sector?
        jc     rd1
        must be sector 26, zero and go to next track
        lda     iot     ;get track to register a
        inr    a
        mov     c,a     ;ready for call
        call    settrk
        xra    a        ;clear sector number
ll:     inr    a        ;to next sector
        mov     c,a     ;ready for call
        call    setsec
        pop     b       ;recall sector count
        dcr    b        ;done?
        jnz    rdsec

```

done with the load, reset default buffer address

```

gocpm:  ;(enter here from cold start boot)

```

enable rst0 and rst7

```
di
mvi    a,12h    ;initialize command
out    revrt
xra    a
out    intc     ;cleared
mvi    a,inte   ;rst0 and rst7 bits on
out    intc
xra    a
out    icon     ;interrupt control
```

set default buffer address to 80h

```
lxi    b,buff
call   setdma
```

reset monitor entry points

```
mvi    a,jmp
sta    0
lxi    h,wboote
shld   1        ;jmp wboot at location 00
sta    5
lxi    h,bdos
shld   6        ;jmp bdos at location 5
if     not test
sta    7*8      ;jmp to mon80 (may have been changed by ddt)
lxi    h,mon80
shld   7*8+1
endif
```

leave iobyte set

previously selected disk was b, send parameter to cpm

```
lda    cdisk   ;last logged disk number
mov    c,a     ;send to ccp to log it in
ei
jmp    cpmb
```

error condition occurred, print message and retry

boterr:

```
pop    b       ;recall counts
dcr    c
jz     booter0
try again
push   b
jmp    wboot0
```

booter0:

otherwise too many retries

```
lxi    h,bootmsg
call   prmsg
jmp    rmon80  ;mds hardware monitor
```

bootmsg:

```
db    '?boot',0
```

onst: ;console status to reg-a

(exactly the same as mds call)

```
jmp    csts
```

conin: ;console character to reg-a

```
call   ci
ani    7fh     ;remove parity bit
ret
```

conout: ;console character from c to console out

```
jmp    co
```

```

dad      b      ;translate(sector) address
mov      a,m    ;translated sector number to A
sta      ios
mov      l,a    ;return sector number in L
ret

;
setdma:  ;set dma address given by regs b,c
mov      l,c
mov      h,b
shld    iod
ret

;
read:    ;read next disk record (assuming disk/trk/sec/dma set)
mvi      c,readf ;set to read function
call     setfunc
call     waitio  ;perform read function
ret      ;may have error set in reg-a

;
;
write:   ;disk write function
mvi      c,writf
call     setfunc ;set to write function
call     waitio
ret      ;may have error set

;
;
utility subroutines
prmsg:  ;print message at h,l to 0
mov      a,m
ora      a      ;zero?
rz

;
more to print
push     h
mov      c,a
call     conout
pop      h
inx      h
jmp      prmsg

;
setfunc:
;
; set function for next i/o (command in reg-c)
lxi      h,iopf ;io function address
mov      a,m    ;get it to accumulator for masking
ani      11111000b ;remove previous command
ora      c      ;set to new command
mov      m,a    ;replaced in iopb
;
; the mds-800 controller requires disk bank bit in sector byte
; mask the bit from the current i/o function
ani      00100000b ;mask the disk select bit
lxi      h,ios   ;address the sector select byte
ora      m      ;select proper disk bank
mov      m,a    ;set disk select bit on/off
ret

;
waitio:
mvi      c,retry ;max retries before perm error

rewait:
;
; start the i/o function and wait for completion
call     intype ;in rtype
call     inbyte ;clears the controller

;
lda      dbank   ;set bank flags
ora      a      ;zero if drive 0,1 and nz if 2,3
mvi      a,iopb and 0fff ;low address for iopb
mvi      b,iopb shr 8 ;high address for iopb
jnz     iodrl   ;drive bank 1?

```

```

list:    ;list device out
;        (exactly the same as mds call)
        jmp     lo
;
listst:  ;return list status
        xra     a
        ret                    ;always not ready
;
punch:   ;punch device out
;        (exactly the same as mds call)
        jmp     po
;
reader:  ;reader character in to reg-a
;        (exactly the same as mds call)
        jmp     ri
;
home:    ;move to home position
;        treat as track 00 seek
        mvi     c,0
        jmp     settrk
;
seldsk:  ;select disk given by register c
        lxi     h,0000h ;return 0000 if error
        mov     a,c
        cpi     ndisks  ;too large?
        rnc                    ;leave HL = 0000
;
        ani     10b          ;00 00 for drive 0,1 and 10 10 for drive 2,3
        sta     dbank        ;to select drive bank
        mov     a,c          ;00, 01, 10, 11
        ani     1b          ;mds has 0,1 at 78, 2,3 at 88
        ora     a            ;result 00?
        jz     setdrive
        mvi     a,00110000b  ;selects drive 1 in bank
setdrive:
        mov     b,a          ;save the function
        lxi     h,iop        ;io function
        mov     a,m
        ani     11001111b    ;mask out disk number
        ora     b            ;mask in new disk number
        mov     m,a          ;save it in iopb
        mov     l,c
        mvi     h,0          ;HL=disk number
        dad     h            ;*2
        dad     h            ;*4
        dad     h            ;*8
        dad     h            ;*16
        lxi     d,dpbase
        dad     d            ;HL=disk header table address
        ret
;
;
settrk:  ;set track address given by c
        lxi     h,iot
        mov     m,c
        ret
;
setsec:  ;set sector number given by c
        lxi     h,ios
        mov     m,c
        ret
sectran:
;        ;translate sector bc using table at de
        mvi     b,0          ;double precision sector number in BC
        xchg                    ;translate table address to HL

```

```

out      ilow          ;low address to controller
mov      a,b
out      ihigh        ;high address
jmp      wait0        ;to wait for complete
;
iodr1:   ;drive bank 1
out      ilow+10h     ;SS for drive bank 10
mov      a,b
out      ihigh+10h
;
wait0:   call      instat      ;wait for completion
        ani      iordy      ;ready?
        jz       wait0
;
;       check io completion ok
call     intype        ;must be 00 complete (00) unlinked
;       00 unlinked i/o complete, 01 linked i/o complete (not used)
;       10 disk status changed 11 (not used)
        cpi     10b        ;ready status change?
        jz       wready
;
;       must be 00 in the accumulator
        ora     a
        jnz     werror     ;some other condition, retry
;
;       check i/o error bits
call     inbyte
        ral
        jc      wready     ;unit not ready
        rar
        ani     11111110b   ;any other errors? (deleted data ok)
        jnz     werror
;
;       read or write is ok, accumulator contains zero
        ret
;
wready:  ;not ready, treat as error for now
        call    inbyte     ;clear result byte
        jmp     trycount
;
werror:  ;return hardware malfunction (crc, track, seek, etc.)
;       the mds controller has returned a bit in each position
;       of the accumulator, corresponding to the conditions:
;       0 - deleted data (accepted as ok above)
;       1 - crc error
;       2 - seek error
;       3 - address error (hardware malfunction)
;       4 - data over/under flow (hardware malfunction)
;       5 - write protect (treated as not ready)
;       6 - write error (hardware malfunction)
;       7 - not ready
;       (accumulator bits are numbered 7 6 5 4 3 2 1 0)
;
;       it may be useful to filter out the various conditions,
;       but we will get a permanent error message if it is not
;       recoverable. in any case, the not ready condition is
;       treated as a separate condition for later improvement
trycount:
;       register c contains retry count, decrement 'til zero
        dcr     c
        jnz     await     ;for another try
;
;       cannot recover from error
        mvi     a,1       ;error code
        ret
;

```

```

; intype, inbyte, instat read drive bank 00 or 10
intype: lda dbank
      ora a
      jnz intyp1 ;skip to bank 10
      in rtype
      ret
intyp1: in rtype+10h ;78 for 0,1 88 for 2,3
      ret
;
inbyte: lda dbank
      ora a
      jnz inbyt1
      in rbyte
      ret
inbyt1: in rbyte+10h
      ret
;
instat: lda dbank
      ora a
      jnz instal
      in dstat
      ret
instal: in dstat+10h
      ret

```

```

;
; data areas (must be in ram)
dbank: db 0 ;disk bank 00 if drive 0,1
      ; 10 if drive 2,3
;
iopb: ;io parameter block
      db 80h ;normal i/o operation
iof: db readf ;io function, initial read
ion: db 1 ;number of sectors to read
iot: db offset ;track number
ios: db 1 ;sector number
iod: dw buff ;io address
;
; define ram areas for bdos operation
endif
end

```

---

APPENDIX VII

---

DATA SHEETS OF COMPONENTS

# WESTERN DIGITAL CORPORATION

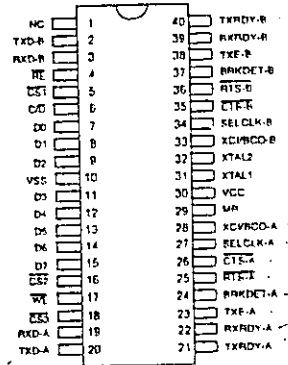
## WD2123 DEUCE Dual Enhanced Universal Communications Element

FINAL

WD2123

### FEATURES

- TWO INDEPENDENT ASYNCHRONOUS FULL DUPLEX DATA COMMUNICATION CHANNELS (2 BOARDS)
- TWO INDEPENDENT BAUD RATE GENERATORS (ONE PER CHANNEL)
- EACH CHANNEL WITH FOLLOWING FEATURES:
  - SELECTABLE 5 TO 8 BIT CHARACTERS
  - 1X, 16X, 64X CLOCK RATES
  - 16 SELECTABLE BAUD RATE CLOCK FREQUENCIES (INTERNAL)
  - LINE BREAK DETECTION AND GENERATION
  - 1, 1W, OR 2 STOP BIT SELECTION
  - FALSE START BIT DETECTION
  - ODD OR EVEN PARITY GENERATE AND DETECTION
  - OVERRUN AND FRAMING DETECTION
  - DOUBLE BUFFERING OF DATA
  - TTL COMPATIBLE INPUTS AND OUTPUTS
  - COMPATIBLE WITH 8251A (ASYNC ONLY) AND WD1983 DEVICES
- DIAGNOSTIC LOCAL LOOP-BACK MODE
- RXD INITIALIZATION UPON MASTER RESET
- ON-BOARD OSCILLATOR FOR EASE OF USE WITH A CRYSTAL
- VERSATILE CLOCK SELECT OPTIONS FOR INDEPENDENT TRANSMIT AND RECEIVE RATES



PIN DESIGNATION

### DESCRIPTION

The Western Digital WD2123 Dual Enhanced Universal Communications Element (DEUCE) is a single chip MOS/LSI Data Communications Controller Circuit that contains two independent full-duplex asynchronous RECEIVER/TRANSMITTER CHANNELS and two independent BAUD RATE GENERATORS. The WD2123 is fabricated in N-Channel silicon gate technology and is packaged in a 40 pin plastic or ceramic package. All inputs and outputs are TTL compatible. The WD2123 Block Diagram is shown in Figure 1. The WD2123 is a merger of two WD1983s and one WD1941 from WDC's line of communications devices on one piece of silicon. The 1983 is an asynchronous only version of the 8251A and the 1941 is a baud rate generator. In this manner, 8251A compatibility is maintained with the WD2123 with the added features of 2 channels and 2 baud rate generators on a single chip.

As depicted from the block diagram, the channels are referred to as CHANNELS A and B. CHANNEL A, which is an asynchronous 8251A, is addressed or controlled by the input signal CS1. CHANNEL B is similarly controlled by CS2. Finally, the BAUD RATE GENERATORS are controlled by CS3.

Each channel of the WD2123 can be programmed to receive and transmit asynchronous serial data. The WD2123 per-

forms serial-to-parallel conversion on data characters received from an input/output device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the status of either channel at any time. Status information on a per channel basis reported includes the type and the condition of the transfer operations being performed by the WD2123 as well as any transmission error conditions (parity, overrun, or framing). Programming the WD2123 is identical to the 8251A in the asynchronous mode, remembering that CS1, when low, selects CHANNEL A and when CS2 is low, selects CHANNEL B.

The WD2123 BAUD RATE GENERATORS may be selected either internally or externally. The clock select logic includes a clock select control bit CR1 (CS) in each COMMAND INSTRUCTION REGISTER. This control bit allows selection of the internal baud clock or an externally applied clock and works in conjunction with the select clock pin, "SELCLK" and the external clock input/baud clock output pin, "XCI/BCO". When CS is logic 1, the external clock select mode is selected. This means that the transmit and receive clocks (TXC and RXC) are internally tied together and the select clock pin, SELCLK, will determine whether those clocks are driven from the internal baud rate generator (SELCLK is high) or from the external clock input pin, "XCI/BCO", (SELCLK is low).

WD2123

| PIN DESCRIPTION |                             |  |  |
|-----------------|-----------------------------|--|--|
| PIN NUMBER      | SIGNAL MNEMONIC             | SIGNAL NAME  | FUNCTION   |
| 10              | VSS                         | GROUND   | Ground   |
| 30              | VCC                         | POWER SUPPLY                                       | +5VDC power supply input.  |
| 7               | D0                          | DATA BUS   | This is the 8 bit Bidirectional Data Bus. It is the means of communication between the WD2123 and the CPU. Data, control, mode and status registers are accessed via this bus.                       |
| 8               | D1                          |  |  |
| 9               | D2                          |  |  |
| 11              | D3                          |  |  |
| 12              | D4                          |  |  |
| 13              | D5                          |  |  |
| 14              | D6                          |  |  |
| 15              | D7                          |  |  |
| 5               | $\overline{CS1}$            | $\overline{CHIP SELECT ONE}$                       | $V_{IL}$ on this input selects Channel A and enables computer communications with Channel A Data, control and status registers.  |
| 16              | $\overline{CS2}$            | $\overline{CHIP SELECT TWO}$                       | $V_{IL}$ on this input selects Channel B and enables computer communications with Channel B Data, control and status registers.  |
| 18              | $\overline{CS3}$            | $\overline{CHIP SELECT THREE}$                     | $V_{IL}$ on this input select the Baud Rate registers for programming.   |
| 6               | $\overline{C/\overline{D}}$ | CONTROL or DATA SELECT                             | This Input is used in conjunction with the appropriate Chip Select and an active read or write operation to determine register access via the Data Bus.  |
| 4               | $\overline{RE}$             | $\overline{READ ENABLE}$                           | $V_{IL}$ on this Input allows the CPU to read data, or status information from the selected register.  |
| 17              | $\overline{WE}$             | $\overline{WRITE ENABLE}$                          | $V_{IL}$ on this Input allows the CPU to write data or control information into the selected register.   |
| 29              | MR                          | MASTER RESET                                       | $V_{IH}$ on this Input resets both channels to the idle state and resets the status, command, mode and Data registers.   |
| 31              | XTAL1                       | CRYSTAL OSCILLATOR INPUT                           | This is the input side of the on-chip oscillator. It can also be driven by an external clock source.   |
| 32              | XTAL2                       | CRYSTAL OSCILLATOR OUTPUT                          | This is the output side of the on-chip oscillator.   |
| 27              | SELCLK-A                    | SELECT CLOCK (Channel A)                           | This input is used in conjunction with the Clock Select bit (CR1) in the command register to determine the baud clock source for Channel A.  |
| 34              | SELCLK-B                    | SELECT CLOCK (Channel B)                           | This input is used in conjunction with the Clock Select bit (CR1) in the command register to determine the baud clock source for Channel B.  |
| 28              | XC/BCO-A                    | EXTERNAL CLOCK INPUT/BAUD CLOCK OUTPUT-(Channel A) | This is a bidirectional port, which is used as the externally applied baud clock input or the internal baud rate generator output depending on the states of SELCLK and CR1 command bit. (Channel A) |
| 33              | XC/BCO-B                    | EXTERNAL CLOCK INPUT/BAUD CLOCK OUTPUT-(Channel B) | This is a bidirectional port, which is used as the externally applied baud clock input or the internal baud rate generator output depending on the states of SELCLK and CR1 command bit. (Channel B) |
| 26              | $\overline{CTS-A}$          | $\overline{CLEAR-TO-SEND}$ (Channel A)             | $V_{IL}$ on this Input enables Channel A to transmit serial data if the Transmitter is enabled.  |

| PIN NUMBER | SIGNAL MNEMONIC           | SIGNAL NAME  | FUNCTION   |
|------------|---------------------------|--|--|
| 35         | $\overline{\text{CTS-B}}$ | $\overline{\text{CLEAR-TO-SEND}}$<br>(Channel B)   | $V_{IL}$ on this input enables Channel B to transmit serial data if the Transmitter is enabled.  |
| 20         | TXD-A                     | TRANSMIT DATA<br>(Channel A)                       | This is the Serial Data Output from Channel A.   |
| 2          | TXD-B                     | TRANSMIT DATA<br>(Channel B)                       | This is the Serial Data Output from Channel B.   |
| 19         | RXD-A                     | RECEIVE DATA<br>(Channel A)                        | This is the Serial Data Input for Channel A.   |
| 3          | RXD-B                     | RECEIVE DATA<br>(Channel B)                        | This is the Serial Data Input for Channel B.   |
| 21         | TXRDY-A                   | TRANSMITTER READY<br>(Channel A)                   | This output, when high ( $V_{OH}$ ), alerts the CPU that Channel A is ready to accept a new data character. The TXRDY output is automatically reset whenever a character is written into the Transmitt Holding Register and can be used as an interrupt to the system. $\overline{\text{CTS}}$ must be asserted. |
| 40         | TXRDY-B                   | TRANSMITTER READY<br>(Channel B)                   | This output, when high ( $V_{OH}$ ), alerts the CPU that Channel B is ready to accept a new data character. The TXRDY output is automatically reset whenever a character is written into the Transmitt Holding Register and can be used as an interrupt to the system. $\overline{\text{CTS}}$ must be asserted. |
| 22         | RXRDY-A                   | RECEIVER READY<br>(Channel A)                      | This output, when high ( $V_{OH}$ ), alerts the CPU that Channel B contains a data character that is ready to be input. This output is automatically reset whenever the new character is read from the Receive Holding Register and can be used as an interrupt to the system.                                   |
| 39         | RXRDY-B                   | RECEIVER READY<br>(Channel B)                      | This output, when high ( $V_{OH}$ ), alerts the CPU that Channel B contains a data character that is ready to be input. This output is automatically reset whenever the new character is read from the Receive Holding Register and can be used as an interrupt to the system.                                   |
| 23         | TXE-A                     | TRANSMITTER EMPTY<br>(Channel A)                   | This output, when high ( $V_{OH}$ ), indicates that Channel A Transmitter has no new characters to send and is waiting in an idle state.   |
| 38         | TXE-B                     | TRANSMITTER EMPTY<br>(Channel B)                   | This output, when high ( $V_{OH}$ ), indicates that Channel B Transmitter has no new characters to send and is waiting in an idle state.   |
| 24         | BRKDET-A                  | BREAK DETECT<br>(Channel A)                        | This output, when high ( $V_{OH}$ ), indicates that the Receiver for Channel A has detected a break condition.   |
| 37         | BRKDET-B                  | BREAK DETECT<br>(Channel B)                        | This output, when high ( $V_{OH}$ ), indicates that the Receiver for Channel B has detected a break condition.   |
| 25         | $\overline{\text{RTS-A}}$ | $\overline{\text{REQUEST-TO-SEND}}$<br>(Channel A) | A general purpose output that is controlled by the command register bit CRS for Channel A.   |
| 36         | $\overline{\text{RTS-B}}$ | $\overline{\text{REQUEST-TO-SEND}}$<br>(Channel B) | A general purpose output that is controlled by the command register bit CRS for Channel B.   |
| 1          | NC                        |  | No Internal Connection.  |

WD2123

WD2123

If the internal BRG clock is selected, (SELCLK is high) then the external clock input pin becomes a BRG clock output. Hence, the mnemonic, "XCVBCO".

When CR1 (CS) is logic 0, then internal clock select mode is selected. The transmit clock (TXC) is driven by the internal BRG clock and the receive clock is driven by the select clock pin, (SELCLK). The XCVBCO pin becomes the baud clock output (the same signal that is being applied to TXC).

The WD2123 also provides a local loop-back test mode of operation for each channel. This diagnostic mode is independently controlled via the LB(CR7) bit of the COMMAND REGISTER. When LB is logic 1, the channel is programmed for Local Loop-Back. In this diagnostic mode, the TXD output is set to the marking (logic "1") state; the output of the TRANSMIT REGISTER is "looped-back" into the RECEIVER REGISTER input; RTS output is held high; the CTS and RXD inputs are ignored. An additional requirement is that the TEN(CR0) command bit and the REN(CR2) be logic 1. The status and output flags operate normally.

Each channel is also provided with break character generation and detection. (A break character is defined as all zero data bits, parity bit and stop bits after a valid start bit.) For break character generation, SBRK (CR3) command bit is set to a logic 1. This causes the TXD output to be forced low (spacing) for as long as SBRK is programmed high. The break detect output and status bit (SR6) is set to logic 1, indicating that the receiver has detected a break character. The framing error flag is also set to 1 for this condition.

#### ARCHITECTURE

The WD2123 is an eight bit bus-oriented device. Communication between the controlling CPU and the two RECEIVER/TRANSMITTER CHANNELS or the two BAUD RATE GENERATORS occurs via the 8-bit data bus through a common set of bus transceivers. Figure 1 is a Block Diagram of the WD2123.

A diagram of one of the two communication controllers is shown in Figure 2. There are two accessible data registers, which buffers transmit and receive data. They are the TRANSMIT HOLDING REGISTER and the RECEIVE HOLDING REGISTER. There is a parallel-to-serial shift register, the TRANSMIT REGISTER and a serial-to-parallel shift register, the RECEIVE REGISTER.

Operational Control and monitoring of the CHANNEL is performed by two CONTROL REGISTERS (the COMMAND INSTRUCTION REGISTER and the MODE INSTRUCTION REGISTER) and the STATUS REGISTER.

A read/write control circuit allows programming/monitoring or loading/reading of data in the CONTROL, STATUS and HOLDING REGISTERS by activating the appropriate control lines: Chip Select (CS1, CS2, CS3), READ ENABLE (RE), WRITE ENABLE (WE) and CONTROL or DATA SELECT (C/D).

Internal control of each channel is by means of two internal microcontrollers: one for transmit and one for receive. The control registers, various counters and external signals provide inputs to the microcontrollers, which generate the necessary control signals to send and receive serial data according to the programmed protocol.

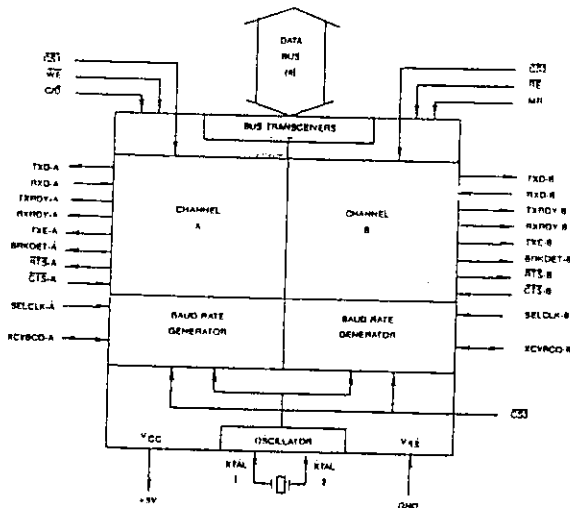


FIGURE 1. WD2123 BLOCK DIAGRAM

A diagram of one of the two BAUD RATE GENERATORS is shown in Figure 3. The 4 low order DATA BUS bits, D0-D3, are used to program the desired rate by loading the RATE REGISTER. Control signals CS3, We and C/D are used to select and load the appropriate register.

The contents of the RATE REGISTER is decoded and addresses a FREQUENCY SELECT ROM for the proper frequency, which is generated by the DIVIDER circuitry and the control logic.

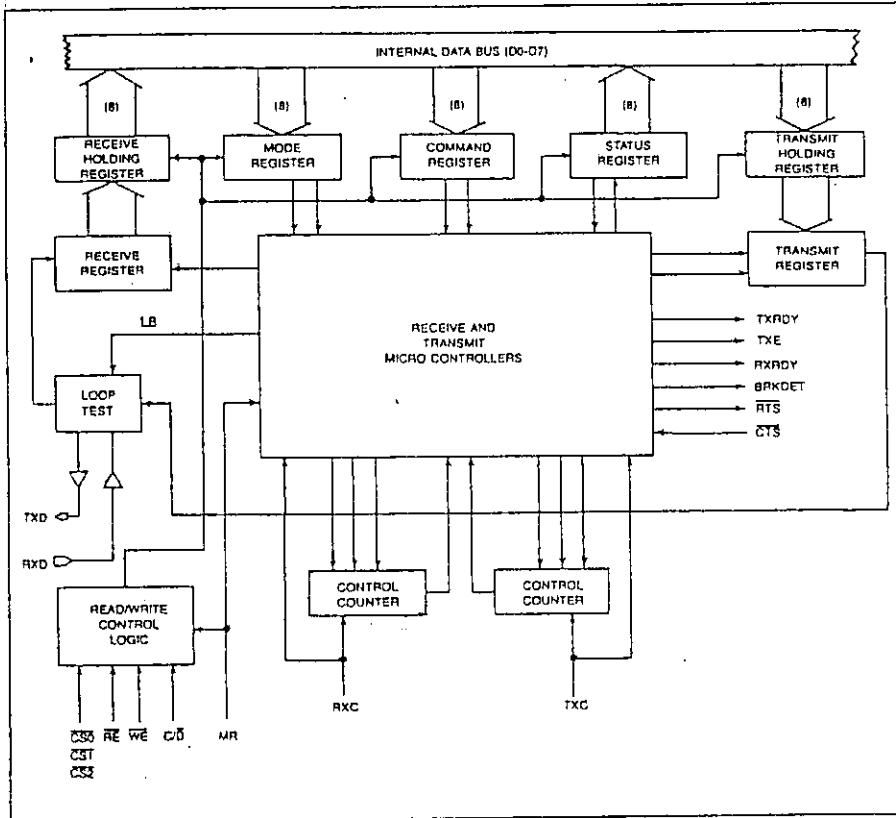


FIGURE 2. RECEIVE/TRANSMIT COMMUNICATIONS CONTROLLER DIAGRAM

WD2123

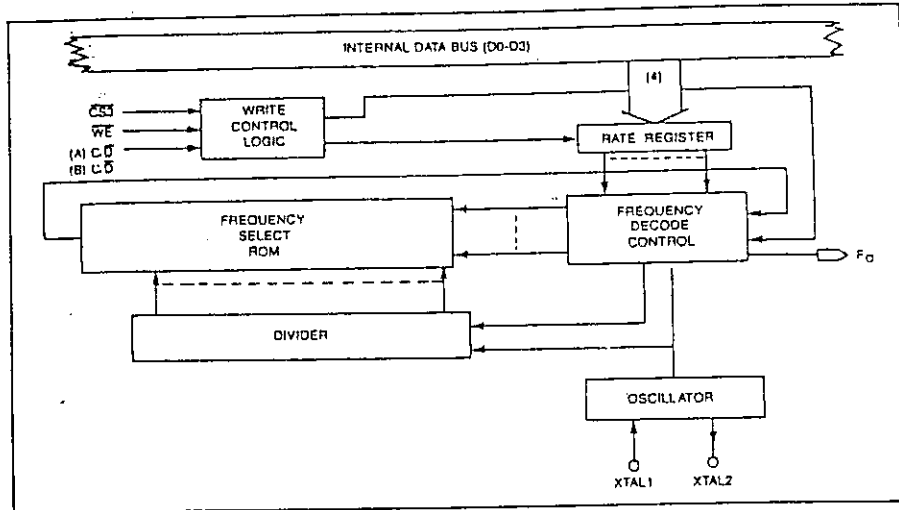


FIGURE 3. WD2123 BAUD RATE GENERATOR DIAGRAM

The WD2123 registers are addressed by the following table:

| C0 | RE | WE | CS1 | CS2 | CS3 | REGISTER SELECTED               |
|----|----|----|-----|-----|-----|---------------------------------|
| L  | L  | H  | L   | H   | H   | RECEIVE HOLDING REG. — CHA      |
| L  | H  | L  | L   | H   | H   | TRANSMIT HOLDING REG. — CHA     |
| H  | L  | H  | L   | H   | H   | STATUS REG. — CHA               |
| H  | H  | L  | L   | H   | H   | MODE AND COMMAND REG. — CHA     |
| L  | L  | H  | H   | L   | H   | RECEIVE HOLDING REG. — CHB      |
| L  | H  | L  | H   | L   | H   | TRANSMIT HOLDING REG. — CHB     |
| H  | L  | H  | H   | L   | H   | STATUS REG. — CHB               |
| H  | H  | L  | H   | L   | H   | MODE and COMMAND REG. — CHB     |
| L  | H  | L  | H   | H   | L   | RATE REG. — CHA                 |
| H  | H  | L  | H   | H   | L   | RATE REG. — CHB                 |
| X  | X  | X  | H   | H   | H   | DATA BUS IN HIGH IMPEDANCE MODE |

TABLE 1. WD2123 REGISTER ADDRESSING

Note:  
 "L" means  $V_{IL}$  at pins.  
 "H" means  $V_{IH}$  at pins.  
 "X" means don't care.

The WD2123 contains two MODE REGISTERS—one for each channel. The format and definition of the MODE REGISTERS are shown below:

| MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| S2  | S1  | EP  | PEN | L2  | L1  | B2  | B1  |

The WD2123 contains two COMMAND REGISTERS—one per channel. The format and definition of the COMMAND REGISTERS are shown below:

| CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| LB  | IR  | RTS | ER  | SBK | REN | CS  | TEN |

WD2123

| B2  | B1 | BAUD RATE FACTOR    |
|-----|----|---------------------|
| 0   | 0  | Undefined           |
| 0   | 1  | 1X                  |
| 1   | 0  | 16X                 |
| 1   | 1  | 64X                 |
| L2  | L1 | CHARACTER LENGTH    |
| 0   | 0  | 5 Bits              |
| 0   | 1  | 6 Bits              |
| 1   | 0  | 7 Bits              |
| 1   | 1  | 8 Bits              |
| PEN |    | PARITY ENABLE       |
| 0   |    | Disable Parity      |
| 1   |    | Enable Parity       |
| EP  |    | PARITY SELECT       |
| 0   |    | Odd Parity          |
| 1   |    | Even Parity         |
| S2  | S1 | NUMBER OF STOP BITS |
| 0   | 0  | Invalid             |
| 0   | 1  | 1 Bit               |
| 1   | 0  | 1½ Bits*            |
| 1   | 1  | 2 Bits              |

TABLE 2. WD2123 MODE REGISTERS

\* 16X and 64X only. 1X will be 2 stop bits.

| TEN | TRANSMIT ENABLE |                                      |
|-----|-----------------|--------------------------------------|
| 1   | Enable          |                                      |
| 0   | Disable         |                                      |
| CS  |                 | CLOCK SELECT                         |
| 1   |                 | XMIT and RCV Clock source common     |
| 0   |                 | XMIT and RCV Clock sources different |
| REN |                 | RECEIVE ENABLE                       |
| 1   |                 | Enable                               |
| 0   |                 | Disable                              |
| SBK |                 | SEND BREAK CHARACTER                 |
| 1   |                 | Force TXD Low                        |
| 0   |                 | Normal Operation                     |
| ER  |                 | ERROR RESET                          |
| 1   |                 | Reset Error Flags                    |
| 0   |                 | No Reset                             |
| RTS |                 | REQUEST TO SEND                      |
| 1   |                 | Force RTS pin = 0 (V <sub>OL</sub> ) |
| 0   |                 | Force RTS pin = 1 (V <sub>OH</sub> ) |
| IR  |                 | INTERNAL RESET                       |
| 1   |                 | Next Write to Mode Register          |
| 0   |                 | Next Write to Command Register       |
| LB  |                 | LOOP BACK ENABLE                     |
| 0   |                 | Normal Operation Mode                |
| 1   |                 | Local Loop-Back Mode                 |

TABLE 3. WD2123 CONTROL REGISTERS

WD2123

The WD2123 contains two STATUS REGISTERS—one per channel. The STATUS REGISTER is a read-only register. The format and definition of the STATUS REGISTERS are shown below:

| SR7 | SR6        | SR5 | SR4 | SR3 | SR2 | SR1       | SR0       |
|-----|------------|-----|-----|-----|-----|-----------|-----------|
| CTS | BRK<br>DET | FE  | OE  | PE  | TXE | RX<br>RDY | TX<br>RDY |

|        |  |                   |
|--------|--|-------------------|
| TXRDY  |  | TRANSMITTER READY |
| 1      | Denotes THR is empty and ready for a new character                                     |                   |
| 0      | THR not empty. (Reset when THR is loaded by CPU)                                       |                   |
| RXRDY  |  | RECEIVER READY    |
| 1      | Denotes that the RHR contains a valid character  |                   |
| 0      | RHR does not contain a valid character. (Reset when the CPU reads the RHR)             |                   |
| TXE    |  | TRANSMITTER EMPTY |
| 1      | Denotes that the TR is empty   |                   |
| 0      | Denotes that the TR is not empty   |                   |
| PE     |  | PARITY ERROR      |
| 1      | Denotes Parity Error   |                   |
| 0      | No Parity Error. (Reset by ER bit of command register)                                 |                   |
| OE     |  | OVERRUN ERROR     |
| 1      | Denotes Overrun Error  |                   |
| 0      | No Overrun Error. (Reset by ER bit of command register)                                |                   |
| FE     |  | FRAMING ERROR     |
| 1      | Denotes Framing Error  |                   |
| 0      | No Framing Error. (Reset by ER bit of command register)                                |                   |
| BRKDET |  | BREAK DETECT      |
| 1      | Indicates that the receiver has detected a line break condition. (FE will also be set) |                   |
| 0      | No Break Condition detected for at least one bit time                                  |                   |
| CTS    |  | CLEAR-TO-SEND     |
| 1      | Indicates that the CTS pin is active (V <sub>IL</sub> )                                |                   |
| 0      | Indicates that the CTS pin is not active (V <sub>IH</sub> )                            |                   |

TABLE 4. WD2123 STATUS REGISTERS

The WD2123 contains two RATE REGISTERS that are used to select 16 BAUD rates when CR1 = 1 and SELCLK = 1. The Format of the RATE REGISTERS is shown below. Note that the Receiver and the Transmitter of any channel run off the same Baud clock except when CR1 = 0, then the Transmitter runs off the Baud Clock and the Receiver runs off an externally applied signal input on the SELCLK pin.

|    |   |   |   |   |            |            |            |            |
|----|---|---|---|---|------------|------------|------------|------------|
|    |   |   |   |   |            |            |            | D0         |
| D7 | X | X | X | X | RA3<br>RB3 | RA2<br>RB2 | RA1<br>RB1 | RA0<br>RB0 |

When C/D = 0, RA3 to RA0 are loaded.

When C/D = 1, RB3 to RB0 are loaded.

The C/D line is used in conjunction with CS3 and WE to program the desired BAUD rate. When C/D is low, Channel A is selected, and when C/D is high, Channel B is selected. The low order 4 bits of the DATA BUS are loaded into the selected rate register, and the high order 4 bits are ignored.

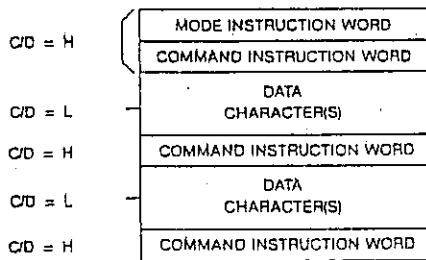
When the crystal frequency equals 1.8432 MHz the following baud rates may be programmed.

| R3 | R2 | R1 | R0 | BAUD RATE |          |          | FREQUENCY<br>(KHZ) | DIVISOR |
|----|----|----|----|-----------|----------|----------|--------------------|---------|
|    |    |    |    | BRF 1X    | BRF 16X  | BRF 64X  |                    |         |
| 0  | 0  | 0  | 0  | 800       | 50.0     | 12.50    | 0.80               | 2304    |
| 0  | 0  | 0  | 1  | 1,200     | 75.0     | 18.75    | 1.20               | 1536    |
| 0  | 0  | 1  | 0  | 1,760     | 110.0    | 27.50    | 1.75               | 1049    |
| 0  | 0  | 1  | 1  | 2,150     | 134.5    | 33.59    | 2.15               | 855     |
| 0  | 1  | 0  | 0  | 2,400     | 150.0    | 37.50    | 2.40               | 768     |
| 0  | 1  | 0  | 1  | 3,200     | 200.0    | 50.00    | 3.20               | 576     |
| 0  | 1  | 1  | 0  | 4,800     | 300.0    | 75.00    | 4.80               | 384     |
| 0  | 1  | 1  | 1  | 9,600     | 600.0    | 150.00   | 9.60               | 192     |
| 1  | 0  | 0  | 0  | 19,200    | 1,200.0  | 300.00   | 19.20              | 96      |
| 1  | 0  | 0  | 1  | 28,800    | 1,800.0  | 450.00   | 28.80              | 64      |
| 1  | 0  | 1  | 0  | 38,400    | 2,400.0  | 600.00   | 38.40              | 48      |
| 1  | 0  | 1  | 1  | 57,600    | 3,600.0  | 900.00   | 57.60              | 32      |
| 1  | 1  | 0  | 0  | 76,800    | 4,800.0  | 1,200.00 | 76.80              | 24      |
| 1  | 1  | 0  | 1  | 115,200   | 7,200.0  | 1,800.00 | 115.20             | 16      |
| 1  | 1  | 1  | 0  | 153,600   | 9,600.0  | 2,400.00 | 153.60             | 12      |
| 1  | 1  | 1  | 1  | 307,200   | 19,200.0 | 4,800.00 | 307.20             | 6       |

TABLE 5. WD2123 BAUD RATE SELECTION

### READ/WRITE OPERATIONS

The WD2123 must be initialized after a MASTER RESET pulse by first writing the MODE INSTRUCTION word and then the COMMAND INSTRUCTION word. Thereafter, every control write to the device is interpreted as a COMMAND word. If it is desired to re-program the MODE REGISTER, a COMMAND REGISTER bit, INTERNAL RESET (CR6), allows the next control write data to be entered into the MODE REGISTER.



TYPICAL DATA BLOCK TRANSFER

### OPERATING DESCRIPTION

The WD2123 is primarily designed to operate in an 8 bit microprocessor environment, although other control logic schemes are easily implemented. The DATA BUS and the interface control signals ( $\overline{CS1}$ ,  $\overline{CS2}$ ,  $\overline{CS3}$ , C/D, RE, WE) should be connected to the microprocessor's data bus and system control bus. A 1.8432 MHz crystal should be connected to the WD2123 as shown in Figure 5. The appropriate TXC (RXC) clock frequencies should be programmed via system software. Different Baud clock configurations are possible, such as separate transmit and receive frequencies, and are outlined in the general description.

For typical data communication applications, the RXD and TXD input/outputs can be connected to RS-232C Interface circuits. Interface control signals,  $\overline{CTS}$  and  $\overline{RTS}$ , are controlled and sensed by the CPU through the COMMAND and STATUS REGISTERS and can be configured in several ways. The  $\overline{CTS}$  input can be used to synchronize the transmitter to external events.

The TXRDY, RXRDY, TXE and BRKDET FLAGS may be connected to the microprocessor system as interrupt inputs or the STATUS REGISTER can be periodically read in a polled environment to support data communication control operations.

The SBRK bit of the COMMAND REGISTER (CR3) is used to send a Break Character. (A Break Character is defined as a start bit, and all zero data, parity and stop bits.) When the CR3 bit is set to a "1", it causes the transmitter output, TXD, to be forced low after the last bit of the last character is transmitted.

The Receiver is equipped with logic to look for a break character. When a break is received, the BREAK DETECT (BRKDET) FLAG and STATUS bit are set to "1". When the receiver input line goes high ( $V_{IH}$ ) for at least one clock period, the receiver resets the BRKDET FLAG and resumes its search for a start bit.

### PROGRAMMING PROCEDURE

The programming sequence of the two channels will be different, depending on whether it is an initialization sequence (that is, one performed right after a hardware master reset occurs) or a re-programming sequence (that is, one performed to change the protocol characteristics (Parity, rate, character length, etc.) after the device has been previously operating in the system). The programming sequence differs, in that, after a master reset, the chip is set to expect the first control write operation (C/D = 1) to contain a mode instruction. Any subsequent control write operations will be transferred to the command instruction register.

Now when it is desired to change the mode instruction register contents, the following re-programming sequence should be performed. A Command Control word of "40" Hex is written to the Chip. This turns off the Receiver and Transmitter and sets the IR (Internal Reset) bit. This bit causes the read/write control logic to expect the next control write operation to be a new mode instruction. After the new mode instruction is written to the chip, all subsequent control write operations will again be interpreted as command instructions. Therefore, after the new mode instruction is performed, the next command would turn the receiver and transmitter back on and resume normal Data operations.

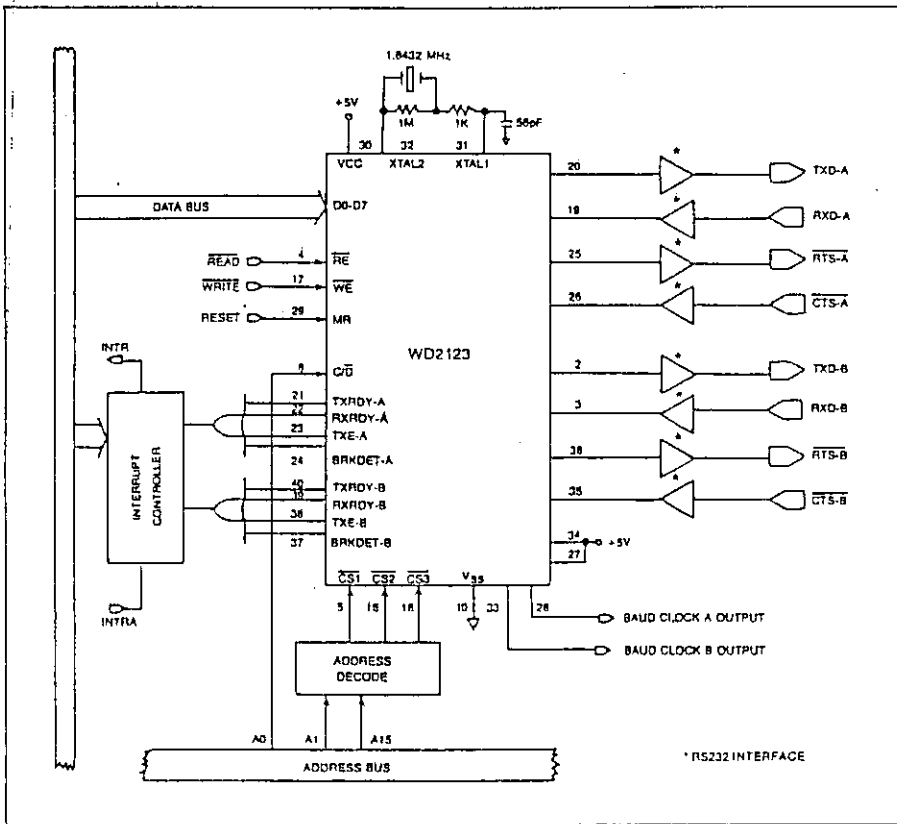


FIGURE 4. WD2123 MICROPROCESSOR APPLICATION

WD2123

**ABSOLUTE MAXIMUM RATINGS**

|   |              |
|---|--------------|
| V <sub>DD</sub> with respect to V <sub>SS</sub> ..... | 0.5V to +12V |
| Voltage on Any Pin with Respect to Ground .....       | -0.5V to +7V |
| Power Dissipation .....                               | 500 Mw.      |
| Lead Temperature (Soldering 10 sec.) .....            | 300°C        |

STORAGE TEMPERATURE:  
 Ceramic: -85°C to +150°C  
 Plastic: -55°C to +125°C

**CRYSTAL SPECIFICATIONS:**

|                         |              |
|-------------------------|--------------|
| Temperature range ..... | 0°C to +70°C |
| Series resistance ..... | 300Ω to 500Ω |
| Overall tolerance ..... | ±0.01%       |

Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under dc electrical characteristics.

**TABLE 6. DC ELECTRICAL CHARACTERISTICS**T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5.0V ±5%; GND = 0V

| SYMBOL          | PARAMETER                                  | MIN  | TYP | MAX             | UNIT | TEST CONDITIONS  |
|-----------------|--|------|-----|-----------------|------|--|
| V <sub>IL</sub> | Input Low Voltage                          | -0.5 |     | 0.8             | V    |  |
| V <sub>IH</sub> | Input High Voltage                         | 2.0  |     | V <sub>CC</sub> | V    |  |
| V <sub>OL</sub> | Output Low Voltage                         |      |     | 0.45            | V    | I <sub>OL</sub> = 1.6 mA                                       |
| V <sub>OH</sub> | Output High Voltage                        | 2.4  |     |                 | V    | I <sub>OH</sub> = -100 μA                                      |
| I <sub>DL</sub> | Data Bus Leakage<br>(High Impedance State) |      |     | -50<br>10       | μA   | V <sub>OUT</sub> = 0.45V<br>V <sub>OUT</sub> = V <sub>CC</sub> |
| I <sub>IL</sub> | Input Leakage                              |      |     | 10              | μA   | V <sub>IN</sub> = V <sub>CC</sub>                              |
| I <sub>CC</sub> | Power Supply Current                       |      | 100 | 125             | mA   | V <sub>CC</sub> = 5.25V<br>No Load                             |

**TABLE 7. CAPACITANCE**T<sub>A</sub> = 25°C; V<sub>CC</sub> = GND = 0V

| SYMBOL           | PARAMETER         | MIN | TYP | MAX | UNIT | TEST CONDITIONS                        |
|------------------|-------------------|-----|-----|-----|------|--|
| C <sub>IN</sub>  | Input Capacitance |     |     | 10  | pF   | f <sub>C</sub> = 1MHz                  |
| C <sub>I/O</sub> | I/O Capacitance   |     |     | 20  | pF   | Unmeasured pins<br>returned to<br>GND. |

## AC ELECTRICAL CHARACTERISTICS

TABLE 8. A.C. CHARACTERISTICS

T<sub>A</sub> = 0°C to + 70°C; V<sub>CC</sub> = 5.0V ± 5%; GND = 0V

| SYMBOL                | CHARACTERISTIC   | MIN        | MAX        | UNITS      | CONDITIONS   |
|-----------------------|--|------------|------------|------------|--|
| <b>BUS PARAMETERS</b> |  |            |            |            |  |
| Read Cycle            |  |            |            |            |  |
| t <sub>AR</sub>       | Address Stable Before READ ( $\overline{CS}, C/\overline{D}$ )               | 50         |            | ns         |  |
| t <sub>RA</sub>       | Address Hold Time for READ ( $\overline{CS}, C/\overline{D}$ )               | 50         |            | ns         |  |
| t <sub>RE</sub>       | READ Pulse Width   | 230        |            | ns         |  |
| t <sub>RD</sub>       | Data Delay from READ   |            | 200        | ns         | C <sub>L</sub> = 50 pF                                       |
| t <sub>RDH</sub>      | READ to Data Floating  | 25         | 200        | ns         | C <sub>L</sub> (Max) = 50 pF<br>C <sub>L</sub> (Min) = 15 pF |
| Write Cycle           |  |            |            |            |  |
| t <sub>AW</sub>       | Address Stable Before WRITE  | 50         |            | ns         |  |
| t <sub>WA</sub>       | Address Hold Time for WRITE  | 50         |            | ns         |  |
| t <sub>WE</sub>       | WRITE Pulse Width  | 230        |            | ns         |  |
| t <sub>DS</sub>       | Data Set-Up Time for WRITE   | TWE        |            | ns         |  |
| t <sub>WDH</sub>      | Data Hold Time for WRITE   | 100        |            | ns         |  |
| <b>OTHER TIMINGS</b>  |  |            |            |            |  |
| t <sub>TXC</sub>      | Transmit Clock Period  | 1.6        |            | us         |  |
| t <sub>DTX</sub>      | TxD Delay from Falling Edge of TxC   |            | 1000       | ns         | C <sub>L</sub> = 100 pF                                      |
| t <sub>SRX</sub>      | Rx Data Set-Up Time to Sampling Pulse  | 200        |            | ns         | C <sub>L</sub> = 100 pF                                      |
| t <sub>HRX</sub>      | Rx Data Hold Time to Sampling Pulse  | 200        |            | ns         | C <sub>L</sub> = 100 pF                                      |
| t <sub>TX</sub>       | Transmitter Input Clock Frequency<br>1x Baud Rate<br>16x and 64x Baud Rate   | DC<br>DC   | 500<br>600 | kHz<br>kHz | Clock<br>50% Duty<br>Cycle                                   |
| t <sub>TPW</sub>      | Transmitter Input Clock Pulse Width<br>1x Baud Rate<br>16x and 64x Baud Rate | 1.0<br>800 |            | us<br>ns   |  |
| t <sub>TPD</sub>      | Transmitter Input Clock Pulse Delay<br>1x Baud Rate<br>16x and 64x Baud Rate | 1.0<br>800 |            | us<br>ns   |  |

WD2123

WD2123

TABLE 9. A.C. CHARACTERISTICS (CONTINUED)

| SYMBOL    | CHARACTERISTICS   | MIN        | MAX        | UNIT          | TEST CONDITION             |
|-----------|---|------------|------------|---------------|----------------------------|
| $t_{RX}$  | Receiver Input Clock Frequency<br>1x Baud Rate<br>16x and 64x Baud Rate   | DC<br>DC   | 500<br>600 | kHz<br>kHz    | Clock<br>50% Duty<br>Cycle |
| $t_{RPW}$ | Receiver Input Clock Pulse Width<br>1x Baud Rate<br>16x and 64x Baud Rate | 1.0<br>800 |            | $\mu$ s<br>ns |                            |
| $t_{RPD}$ | Receiver Input Clock Pulse Delay<br>1x Baud Rate<br>16x and 64x Baud Rate | 1.0<br>800 |            | $\mu$ s<br>ns |                            |
| $t_{TX}$  | TxDY Delay from Center of Stop Bit  |            | 8          | $t_{RXC}$     | $C_L = 50pF$ (16X)         |
| $t_{RX}$  | RxDY Delay from Center of Stop Bit  |            | 1/2        | $t_{RXC}$     |                            |
| $t_{RS}$  | Internal BRKDET Delay from Center<br>of Data Bit                          |            | 1          | RXC           |                            |
| $t_{TRD}$ | TxDY Delay from Falling Edge of<br>WRITE                                  |            | 450        | ns            |                            |
| $t_{TOD}$ | TxD Output from Falling Edge of<br>WRITE                                  |            | 1/2        | $t_{TXC}$     |                            |
| $t_{WC}$  | Control Delay from Rising Edge of<br>WRITE ( $\overline{RTS}$ )           |            | 200        | ns            |                            |
| $t_{CR}$  | Control to READ Set-Up Time ( $\overline{CTS}$ )                          |            | 1          | $t_{TXC}$     |                            |
| $t_{MR}$  | Master Reset  | 500        |            | ns            |                            |

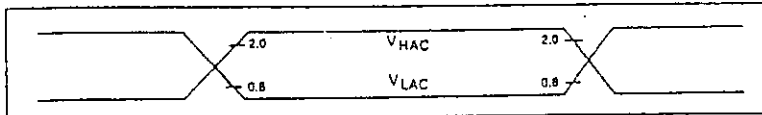


FIGURE 5. A.C. TEST POINTS

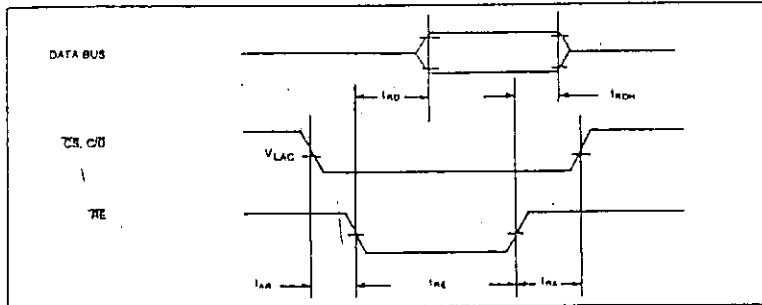


FIGURE 6. READ TIMING

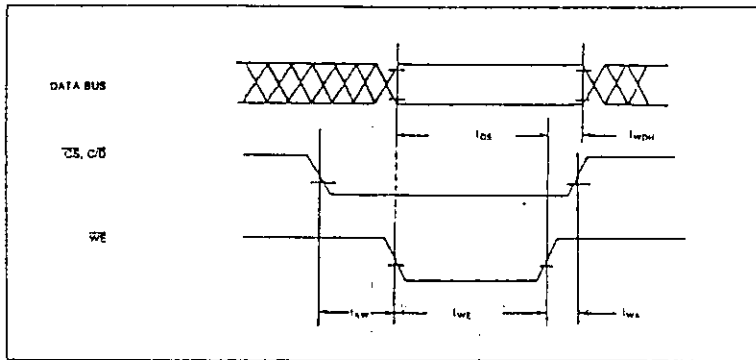


FIGURE 7. WRITE TIMING

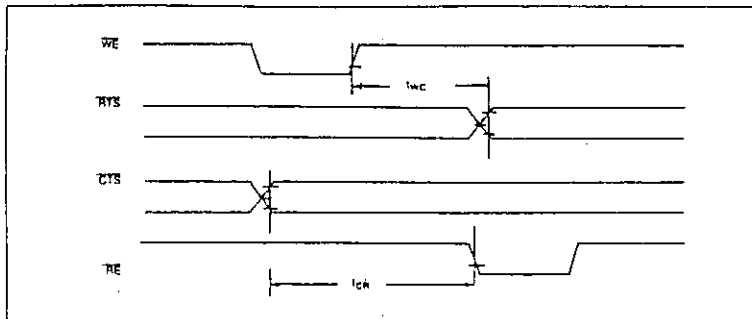


FIGURE 8. INTERFACE CONTROL TIMING

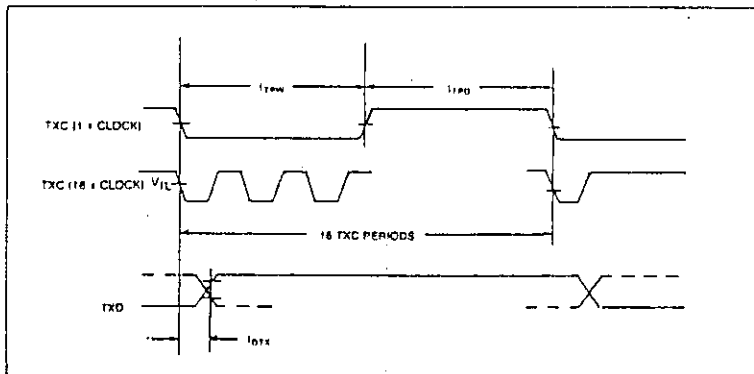


FIGURE 9. TRANSMITTER CLOCK AND DATA TIMING

WD2123

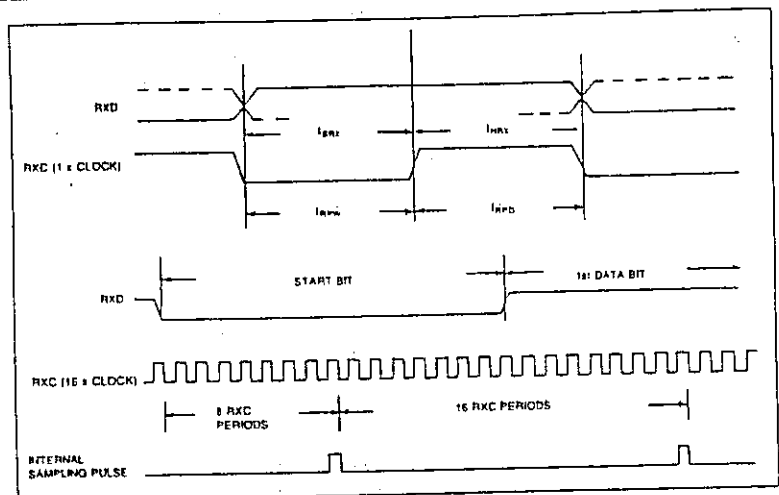


FIGURE 10. RECEIVER CLOCK AND DATA TIMINGS

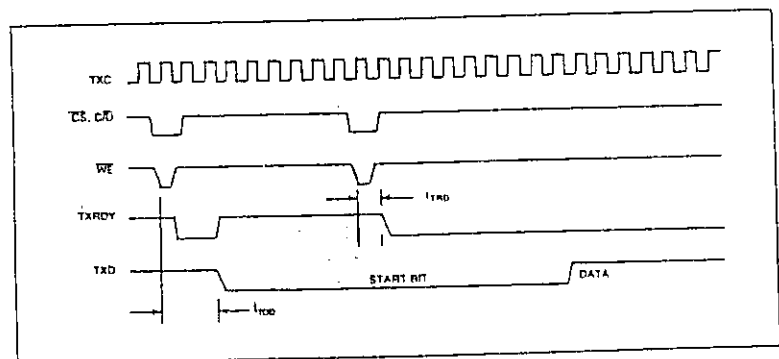


FIGURE 11. TRANSMITTER OUTPUT TIMINGS WITH RESPECT TO TRANSMIT CLOCK

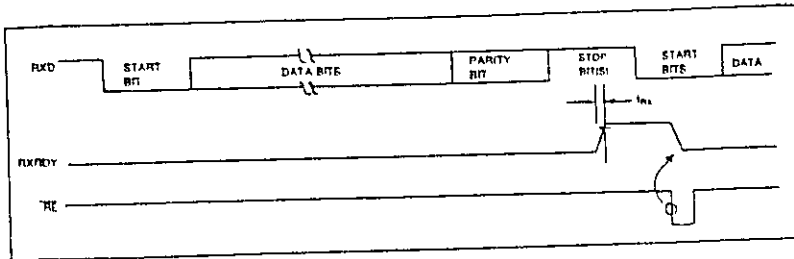


FIGURE 12. RXRDY TIMING

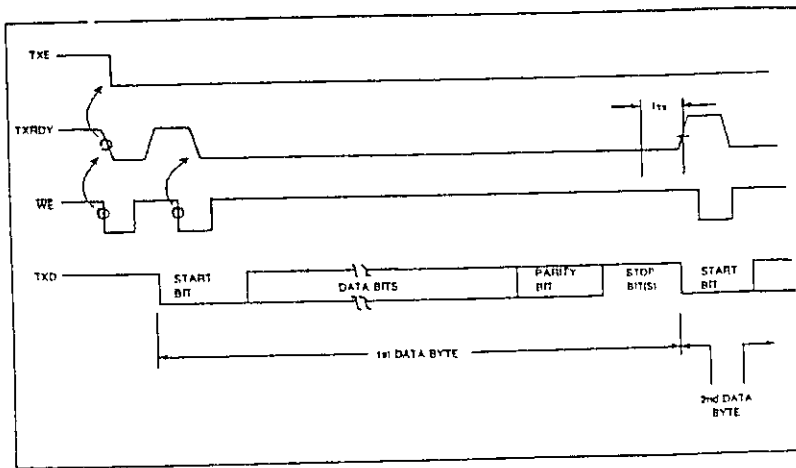
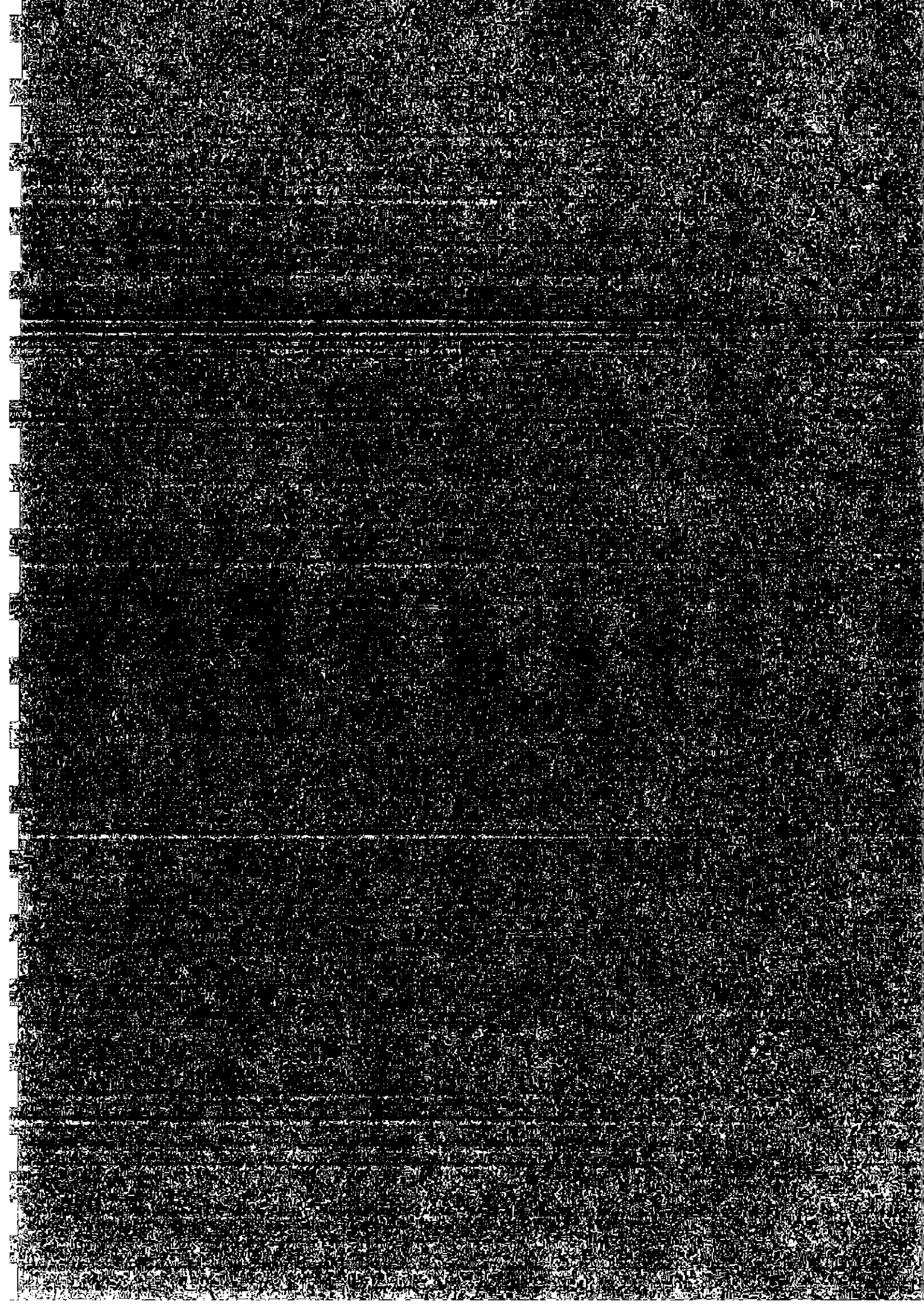


FIGURE 13. TXRDY TIMING



1 - Hardware.

O hardware do sistema de diskettes TIMEX, daqui para a frente referido apenas por FDD, e constituído por duas partes basicas: o conjunto formado pelo controlador, as unidades de disco e a fonte de alimentacao e o "interface" que estabelece a ligacao entre esta e o Spectrum e inclui a extensao do BASIC do Spectrum ( FDD-BASIC ).

1.a - Interface.

O interface como se pode ver no esquema anexo, dispoe de 4Kb de ROM ou EPROM, onde residem as rotinas de extensao do BASIC e comunicacao com a unidade de discos, 1Kb de RAM, utilizada para armazenamento de variaveis de sistema e buffers de dados, para as rotinas de comunicacao, um porto de comunicacoes bidireccional e logica de comando.

O mapa de memoria do interface e o seguinte:

|               |                       |
|---------------|-----------------------|
| 0000h - 1FFFh | ROM                   |
| 2000h - 3FFFh | RAM                   |
| 0EFh          | PORTO de COMUNICACOES |

O espaco de enderecos ocupado, nao corresponde ao tamanho das memorias utilizadas, existindo portanto varios ecos para cada memoria.

Como se pode observar, o espaco usado e o mesmo da ROM do Spectrum. O conflito e resolvido pela logica do interface que selecciona uma ou outra ROM, conforme o estado de um biestavel interno. Este mecanismo e actuado pelo processador, ao executar um ciclo de "operation code fetch", em determinados enderecos. Assim 0000h e 0008h seleccionam a ROM de extensao e 0604h a ROM do Spectrum.

A escolha dos enderecos de entrada na ROM de extensao, foi feita de modo a sanhar o controle do computador, apos um "cold start" ( 0000h ) e sempre que o Spectrum detecte um erro de funcionamento ( 000Ch ). A reentrada no Spectrum e feita por um salto indirecto, atraves da instrucao RET que se encontra na posicao 0604h da nova ROM.

1.b - Unidade de discos.

1.b.1 - Controlador.

Esta unidade e relativamente complexa pelo que a descricao sera sumaria.

O controlador, e um computador autonomo baseado no microprocessador Z80, com 16 Kb de RAM, 1 Kb de ROM, controlador de discos, controlador de canais asincronos, porto bidireccional de comunicacoes e logica de comando.

A ROM, que esta activa logo apos um reset, tem um pequeno programa que le para memoria, a partir de 3F00h, o conteudo do sector 0, da pista 0, do disco II e termina com uma instrucao JP 3F00h.

Fazendo o set do bit 6 do porto 0EFh do controlador, esta ROM e desactivada e a RAM passa a ocupar os primeiros 16 Kb de espaco de