D E C M A T I O N

3375 Scott Blvd. Suite 236 Santa Clara, Calif. 95054
(408) 980-1678 Telex: 171596 AAA COM SUVL

13-Mar-84

NOTES ON D201 INSTALLATION

1) Please be careful of the connectors to not cut your fingers when extracting the card. Covering both connectors witht he cable connector before pulling on the card will protect your fingers.

2) If you use the ISIS-X emulator package, you must use RUNIBDM.COM in place of RUNIB.COM when using the memory disk under CP/M. Change the name of RUNIBDM.COM. Also, some releases of RUNIBDM have a bug of not seeing command files except on the default unit. We will send a new ISIS-X package on request with this bug fixed.

3) The connector between the carrier board and the piggyback board is a possible trouble spot. Make sure the piggyback board is not deformed and that the pins slide easily in the connector when the board is flexed slightly.

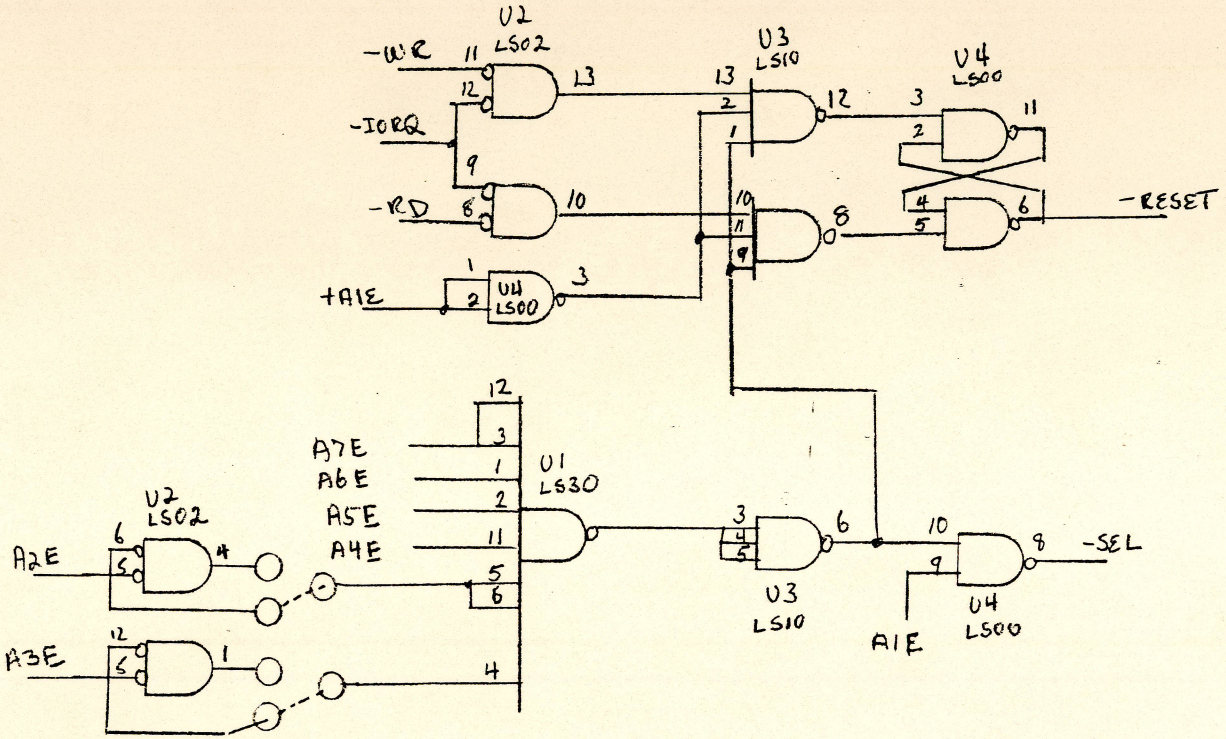
4) The D201 software distribution disk contains a new CP/M system on the system tracks - use it. In some instances you may need to update the control program you are using under your operating system too.

5) Do not make the CP/M disks used as MS-DOS disks a great deal larger (i.e. double) the size of the MS-DOS disk. The program goes by the size of the CP/M disk rather than the size of the file MSDOSIMG.DOS. This will be fixed in a later release.

6) MS-DOS software sent on disks marked RT-11 format must be loaded into non-RT-11/TSX+ systems using a system utility (FLX, FIT, etc.). Use the image mode of copy. The only system lacking such a utility is micro-RSX. This is done because only the 160K MS-DOS disk will fit on a standard 8-inch CP/M disk. Remember, MS-DOS will not work on disks which are CP/M format floppies. It will work on pseudo disks on floppies or any other pseudo disk. If you have a file MSDOSIMG.DOS on a CP/M floppy copy it to an empty psuedo disk which has 160K bytes or a little more of empty space, then you can use it under MS-DOS.

J1, J2

1	-WAITE N.C.
2	DB4
3	
4	-I2D
5	-INTE N.C.
6	
7	A5E
8	A2E
9	
10	A3E
11	A4E
12	DB0
13	DB2
14	DB6
15	DB7
16	A0E
17	A1E
18	DB5
19	A7E
20	A6E
21	DB1
22	-WR
23	+CLKE N.C.
24	-IORQ
25	DB3
26	+MIE N.C.

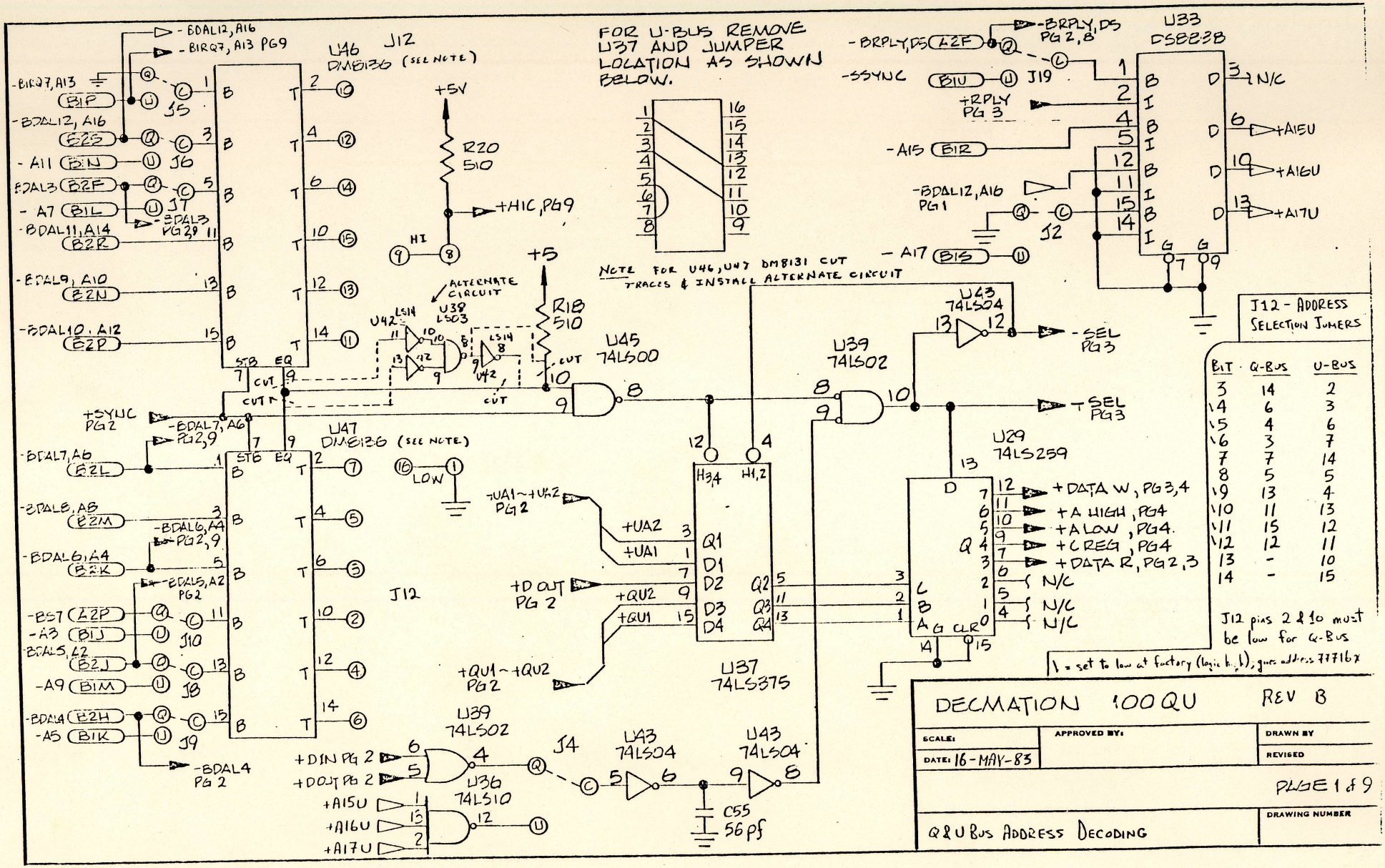


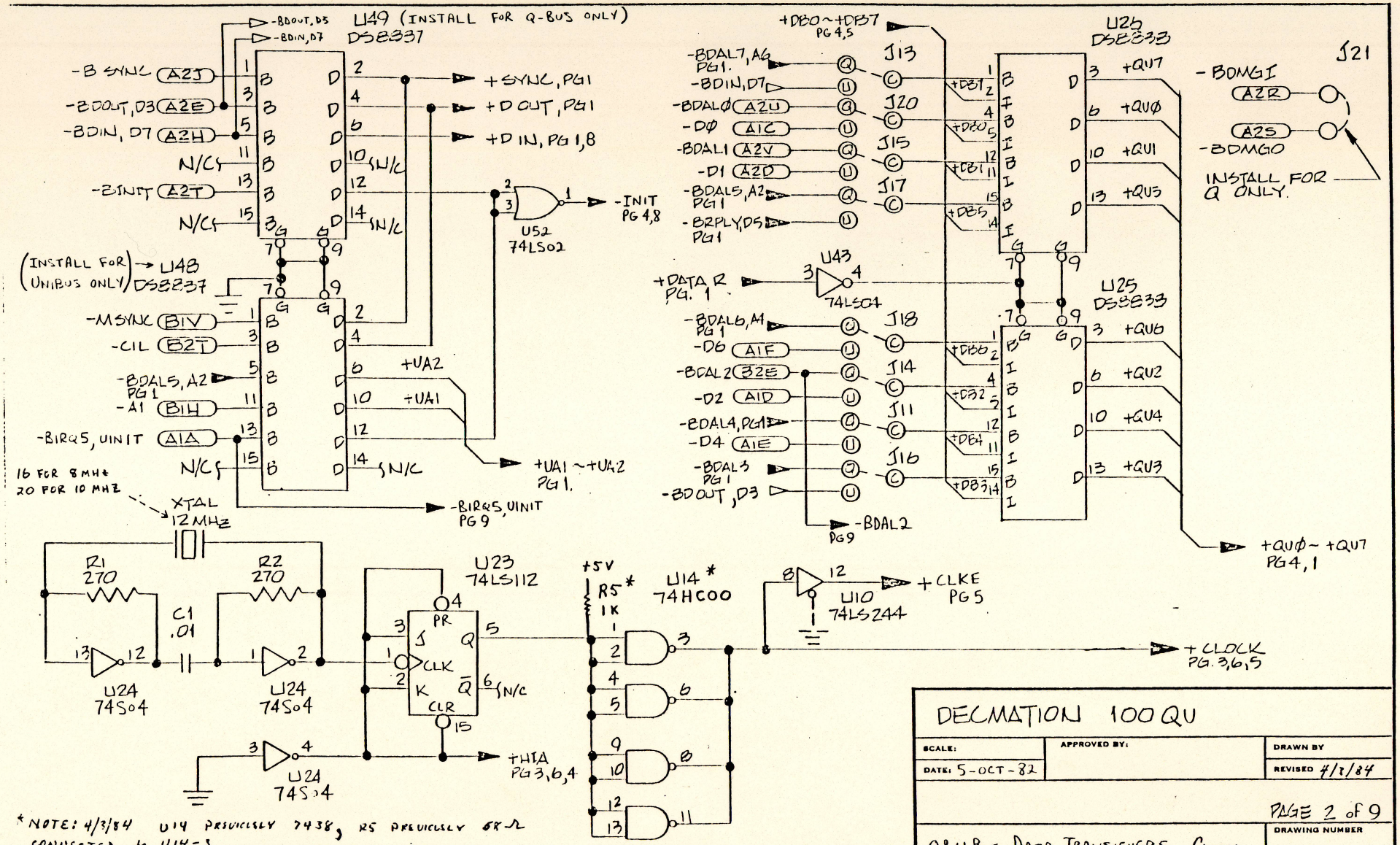
J3

A0E	1
-WR	2
-RD	3
DB1	4
DB3	5
DB5	6
DB7	7
	8
-SEL	9
DB6	10
DB4	11
DB2	12
DB0	13
N.C.	14
-IORQ	15
-RESET	16

DECMATION
D201 CARRIER

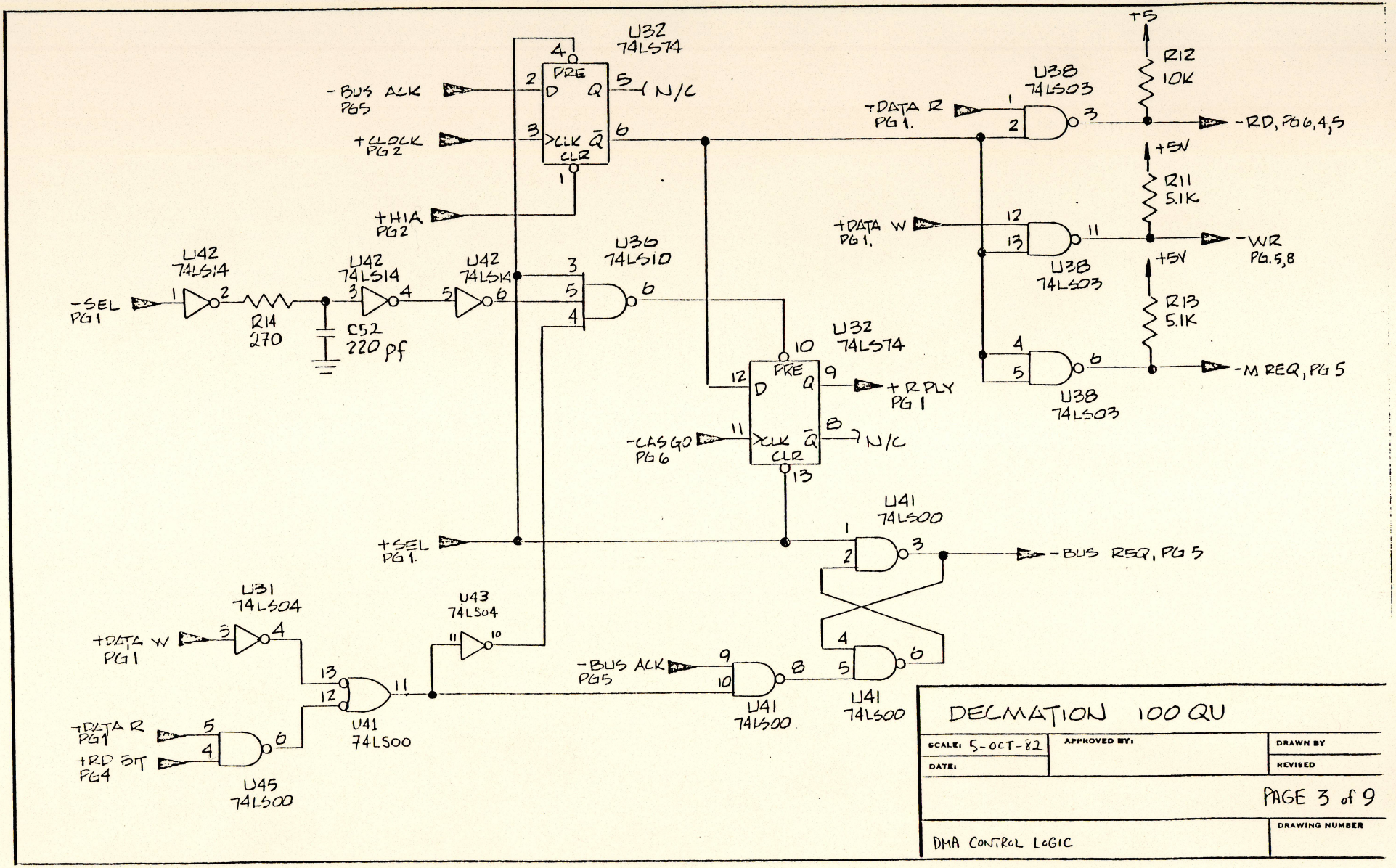
3/8/85

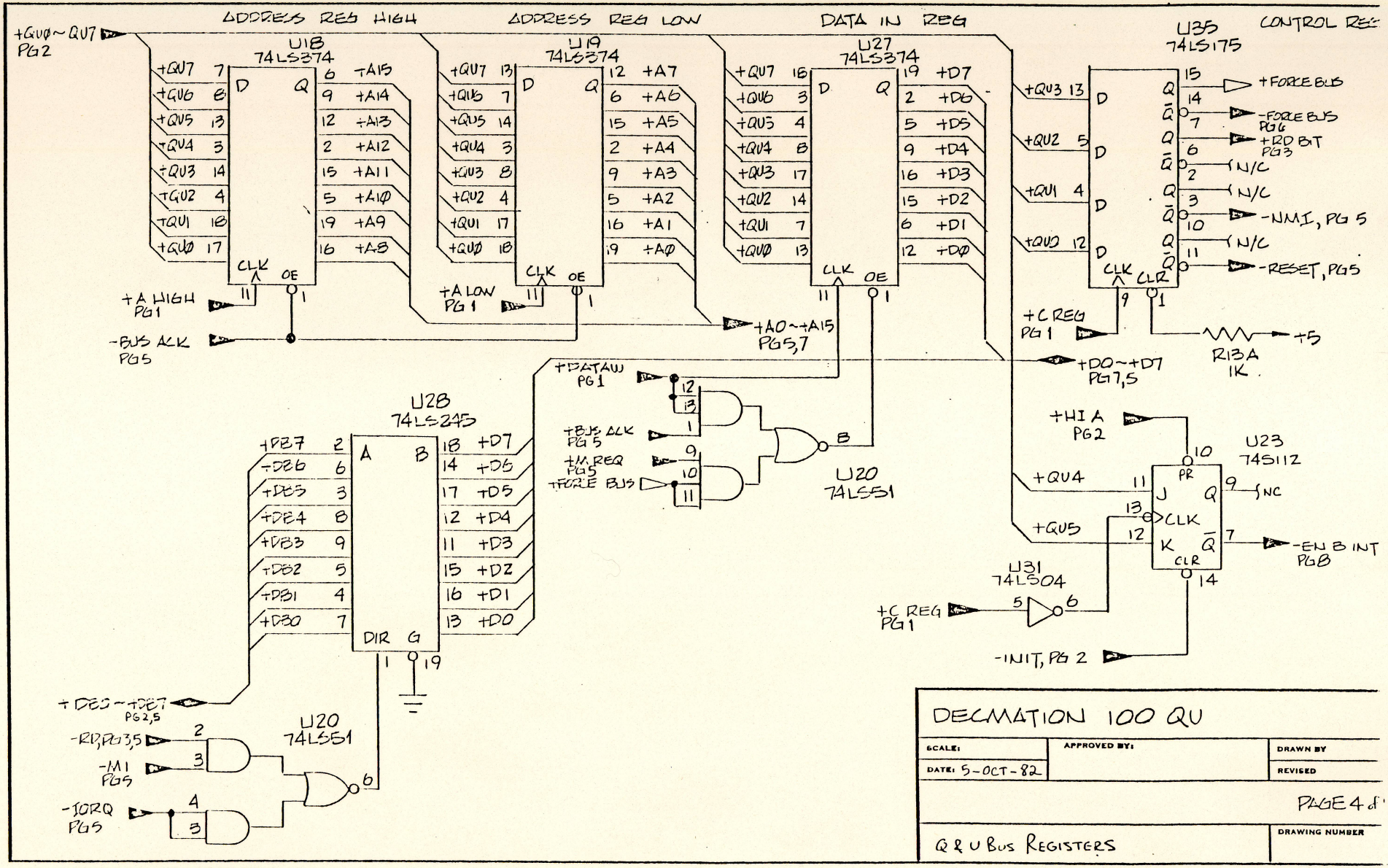

FOR U-BUS REMOVE U37 AND JUMPER LOCATION AS SHOWN BELOW.
NOTE FOR U46,U47 DM8131 CUT TRACES & INSTALL ALTERNATE CIRCUIT
ALTERNATE CIRCUIT
U46 DM8136 (SEE NOTE)
U47 DM8136 (SEE NOTE)
U45 74LS00
U39 74LS02
U43 74LS04
U29 74LS259
U37 74LS375
U36 74LS10
U33 DS8833B
J12
J4
J12 - ADDRESS SELECTION JUMERS
BIT Q-BUS U-BUS
3 14 2
4 6 3
5 4 6
6 3 7
7 7 14
8 5 5
9 13 4
10 11 13
11 15 12
12 12 11
13 - 10
14 - 15
J12 pins 2 & 10 must be low for Q-Bus
1 = set to low at factory (logic high), gives address 77716x
DECMATION 100QU REV B
SCALE:
APPROVED BY:
DRAWN BY
DATE: 16-MAY-83
REVISED
PAGE 1 of 9
Q&U Bus Address Decoding
DRAWING NUMBER
11X17 PRINTED ON NO. 1000H-10 CLEARPRINT FADE-OUT

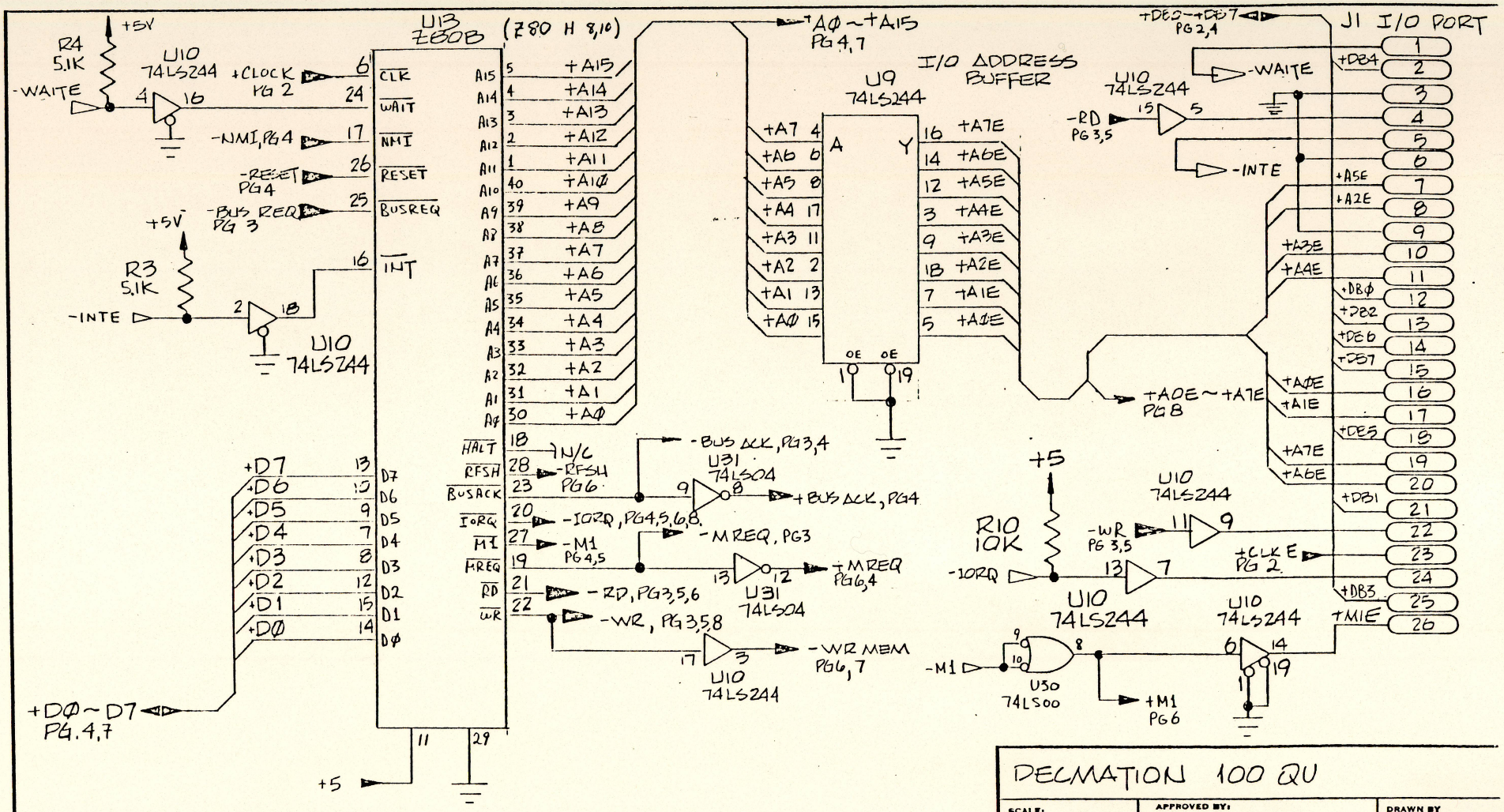


*NOTE: 4/3/84 U14 PREVIOUSLY 7438, R5 PREVIOUSLY 68Ω CONNECTED TO U14-3

DECMATION 100 QU		
SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED 4/3/84
		PAGE 2 of 9
Q & U BUS DATA TRANSCEIVERS, CLOCK		DRAWING NUMBER







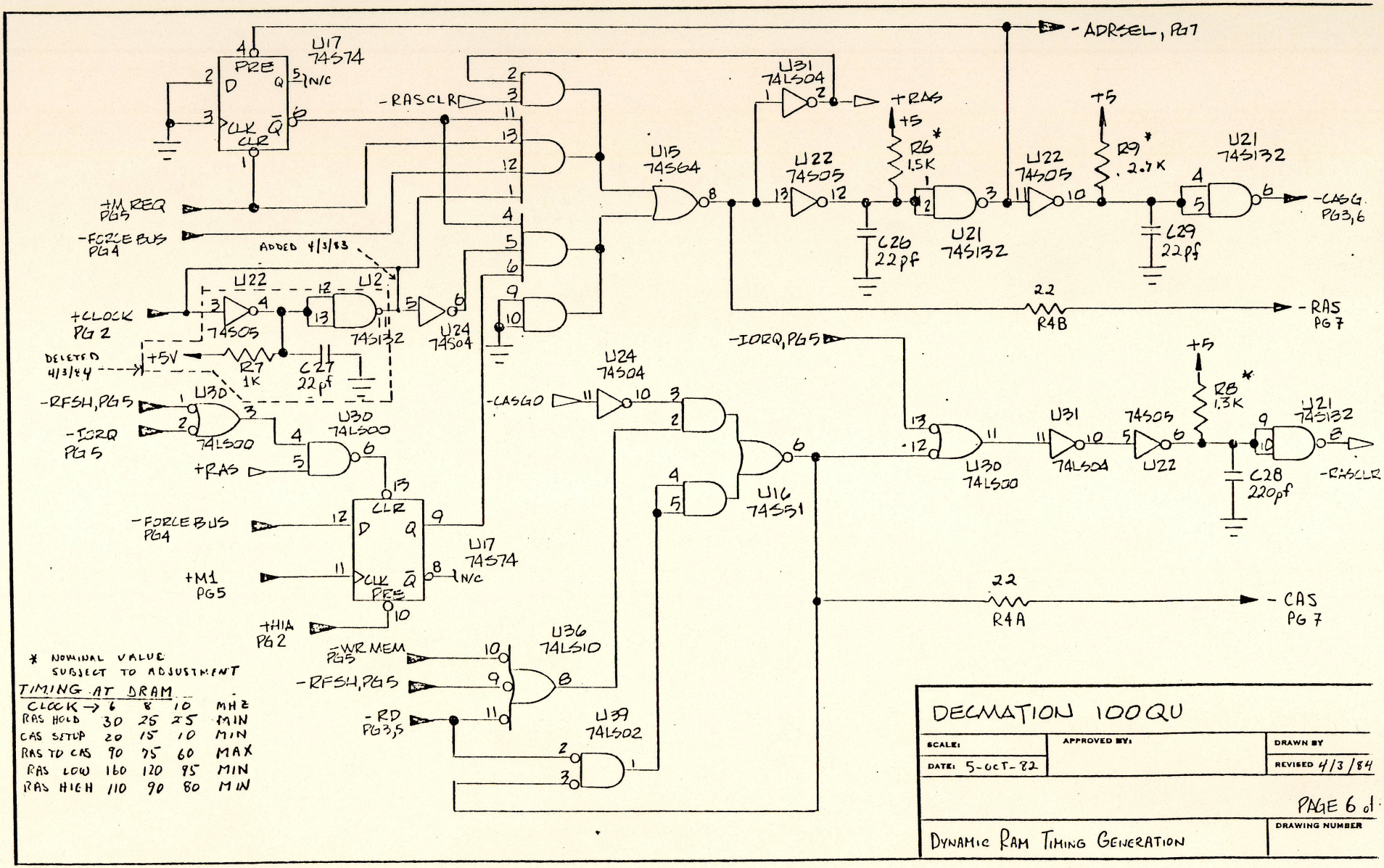
DECMATION 100 QU

SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED 4/3/84

PAGE 5 of 9

Z80 & I/O PORT

DRAWING NUMBER



* NOMINAL VALUE
SUBJECT TO ADJUSTMENT

TIMING AT DRAM

CLOCK →	6	8	10	MHz
RAS HOLD	30	25	25	MIN
CAS SETUP	20	15	10	MIN
RAS TO CAS	90	75	60	MAX
RAS LOW	160	120	95	MIN
RAS HIGH	110	90	80	MIN

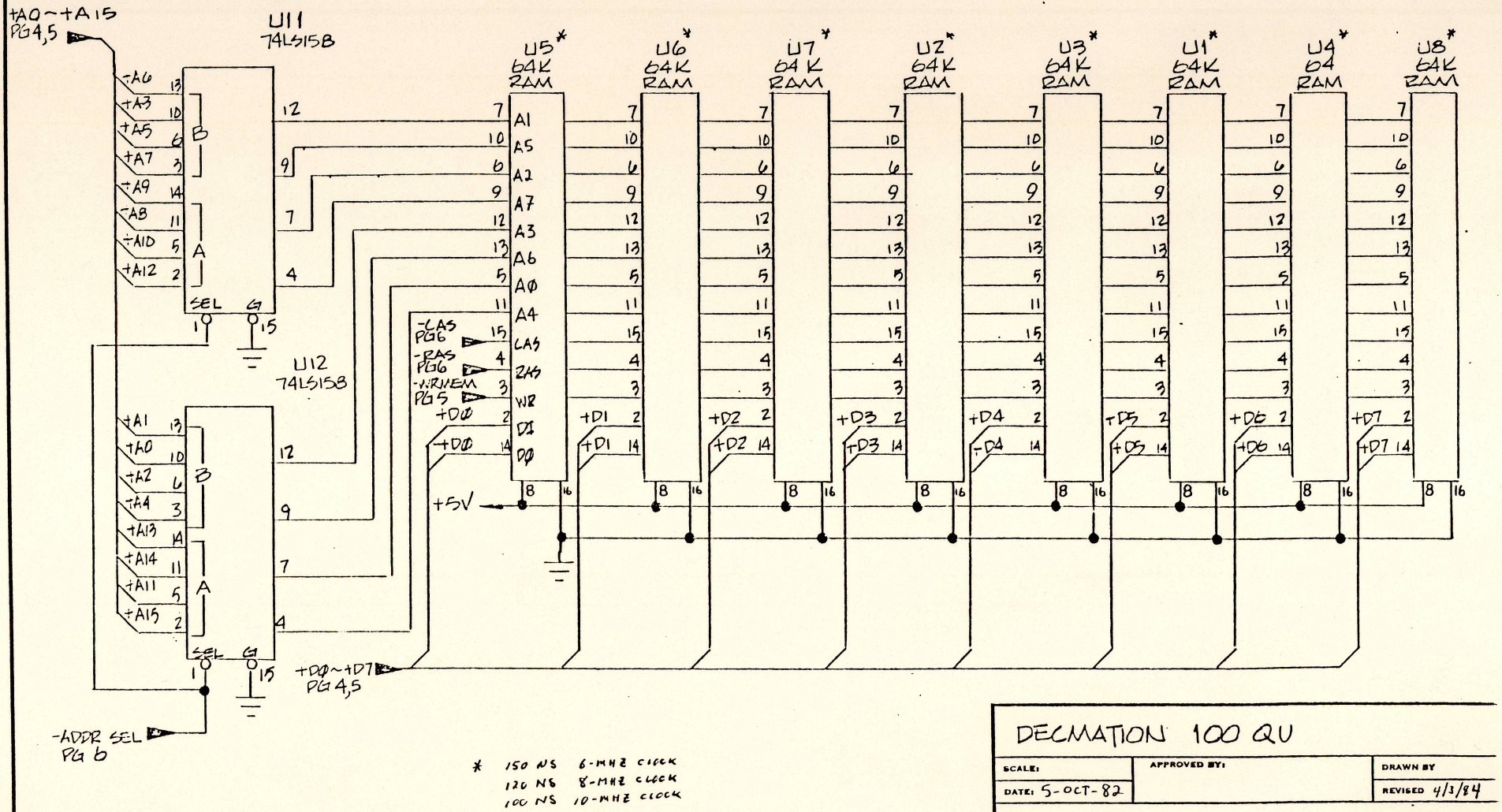
DECMATION 100QU

SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED 4/3/84

PAGE 6 of

DYNAMIC RAM TIMING GENERATION

DRAWING NUMBER



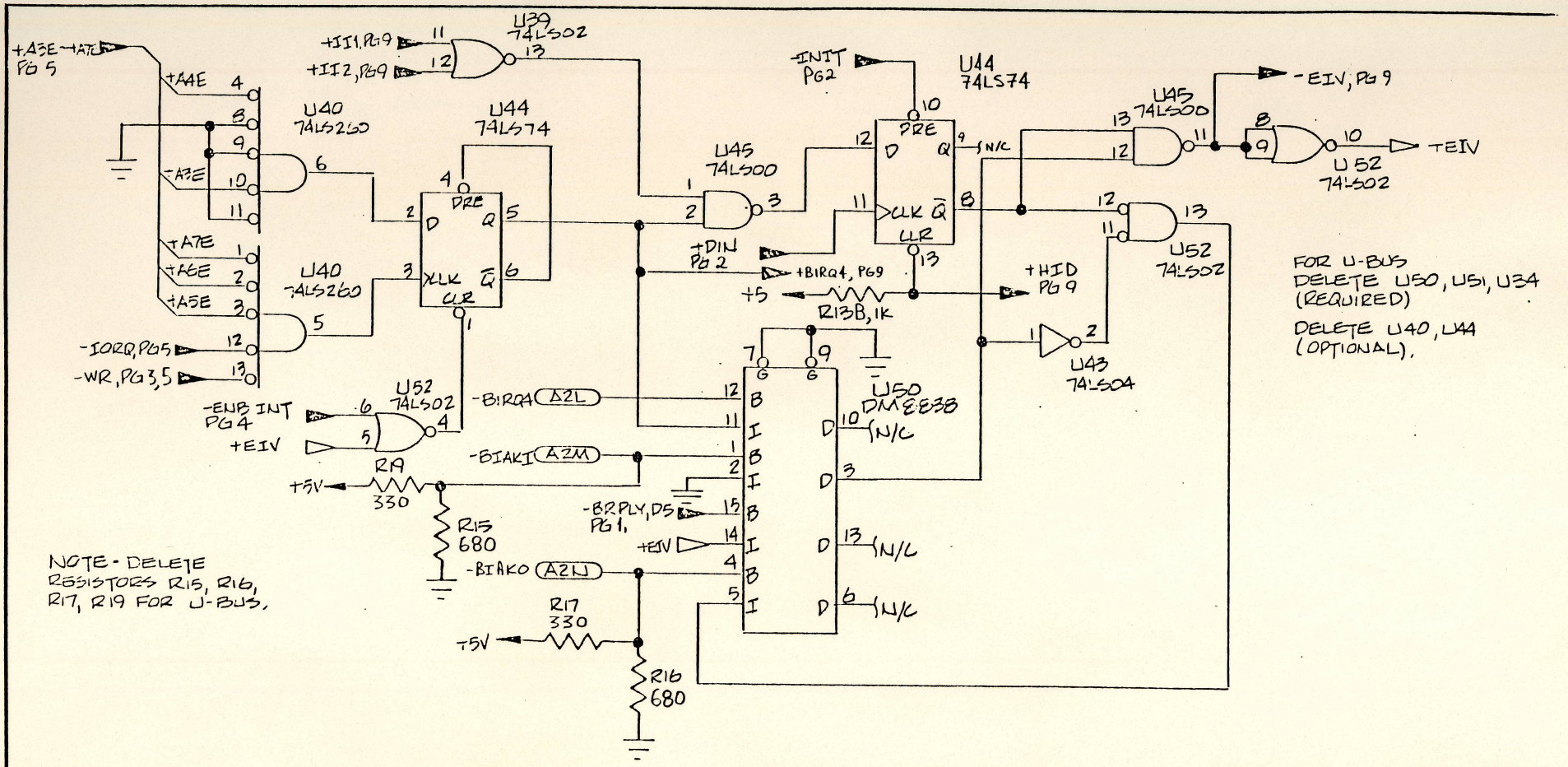
* 150 NS 6-MHZ CLOCK
120 NS 8-MHZ CLOCK
100 NS 10-MHZ CLOCK

DECMATION 100 QU

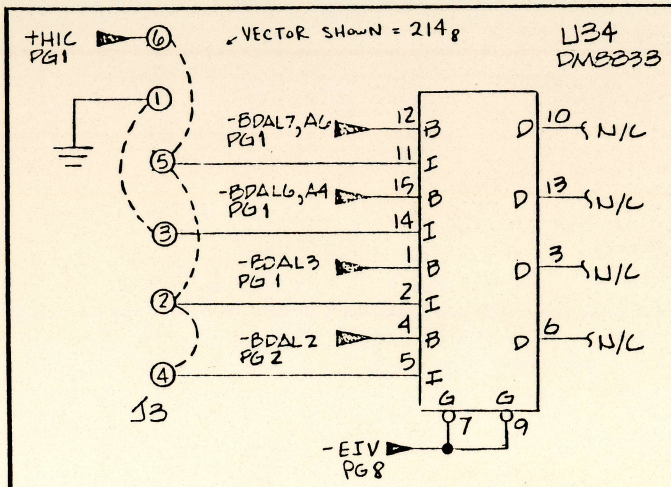
SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED 4/3/84

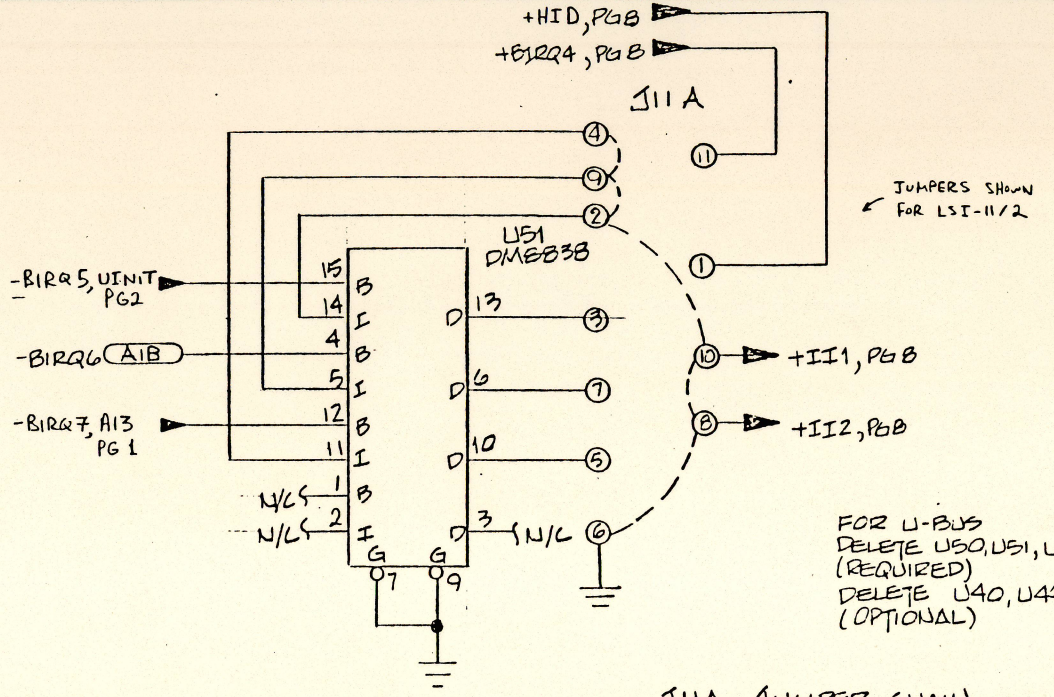
DYNAMIC RAMS

DRAWING NUMBER



DECMATION 100 QU		
SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED
		PAGE 8 of 9
Q-BUS INTERRUPT LOGIC		DRAWING NUMBER





FOR U-BUS
DELETE U50, U51, U34 (REQUIRED)
DELETE U40, U44 (OPTIONAL)

INTERRUPT VECTOR (OCTAL)	INTERRUPT VECTOR SELECTION J3 JUMPER CHAIN
100	6-3 / 1-2-4-5
104	6-3-4 / 1-2-5
110	6-3-2 / 1-4-5
114	6-3-2-4 / 1-5
200	6-5 / 1-2-3-4
204	6-5-4 / 1-2-3
210	6-5-2 / 1-3-4
214	6-5-4-2 / 1-3
300	6-3-5 / 1-2-4
304	6-3-5-4 / 1-2
310	6-3-5-2 / 1-4
314	6-2-3-4-5 / 1-N/C

J11A JUMPER CHAIN

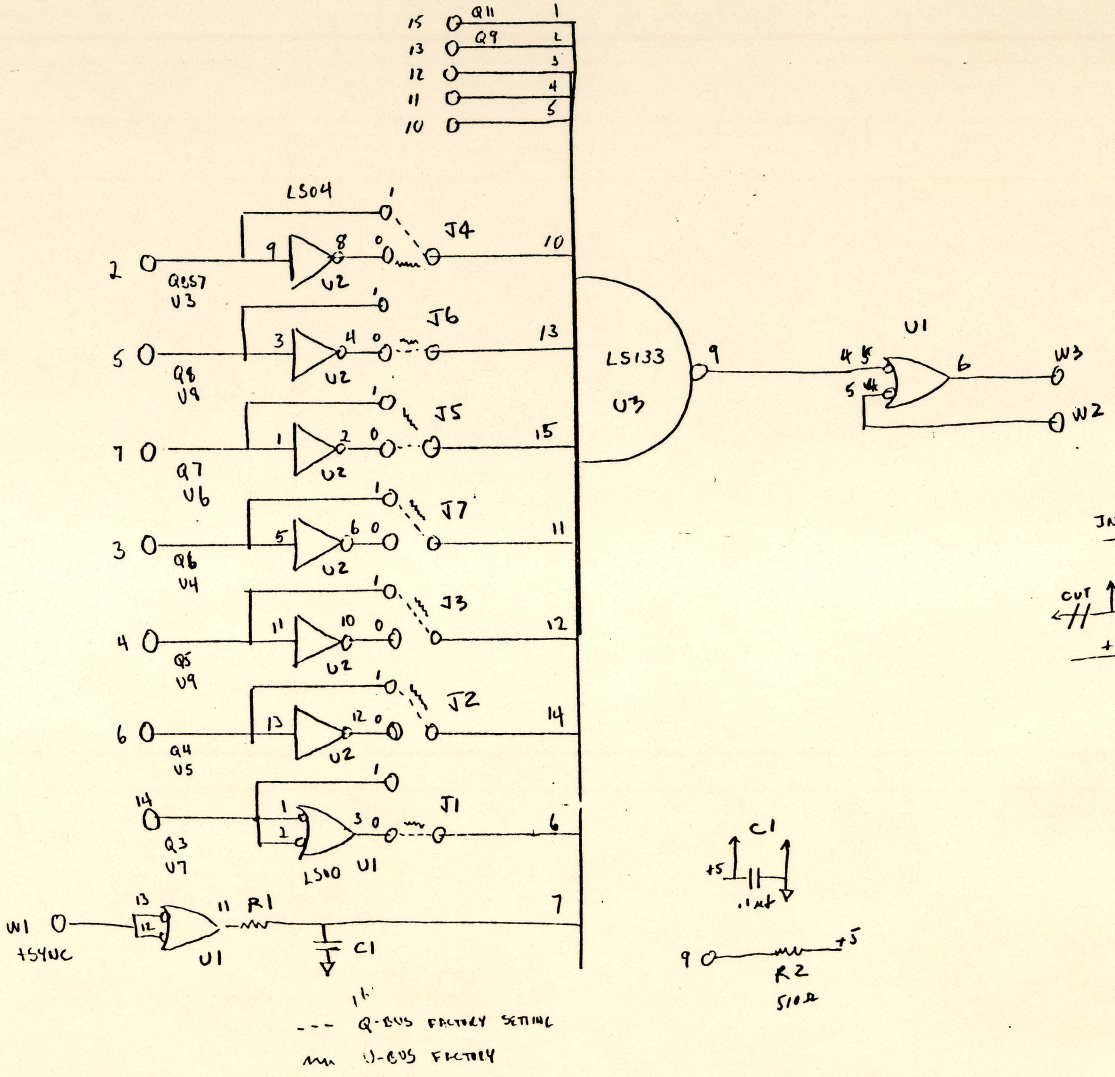
LSI-11/2	6-2-9-4-10-8 / N.C. 5,7,3,11,1
LSI-11/23 P4	6-2-9-4/3-10/7-8 / N.C 5,1,11
INTERRUPT PRIORITY SELECTION: P.5	6-4-9-10/11-2/7-8/ N.C. 3,5,1
P.6	6-2-4-10/ 11-9/5-8 / N.C 3,7,1
P.7	6-2-8-10/ 11-4-9 / N.C

DECMATION.100QU

SCALE:	APPROVED BY:	DRAWN BY
DATE: 5-OCT-82		REVISED

PG 9 of 9

INTERRUPT VECTOR & PRIORITY SELECT — DRAWING NUMBER



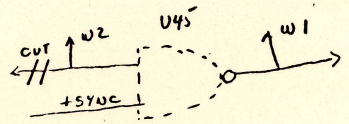
JUMPERS

Q-BUS	J7	J6	J5	J4	J3	J2	J1
177160	1	0	0	1	1	1	0
360	1	0	1	1	1	1	0
370	1	0	1	1	1	1	1
400	1	1	0	0	0	0	0
410	1	1	0	0	0	0	1
420	1	1	0	0	0	1	0

UNIBUS	J8	J6	J1	J5	J2	J7	J4
177160	1	0	0	1	1	1	0
360	1	0	1	1	1	1	0
370	1	0	1	1	1	1	1
400	1	1	0	0	0	0	0
410	1	1	0	0	0	0	1
177420	1	1	0	1	0	1	0
176360	0	0	1	1	1	1	0

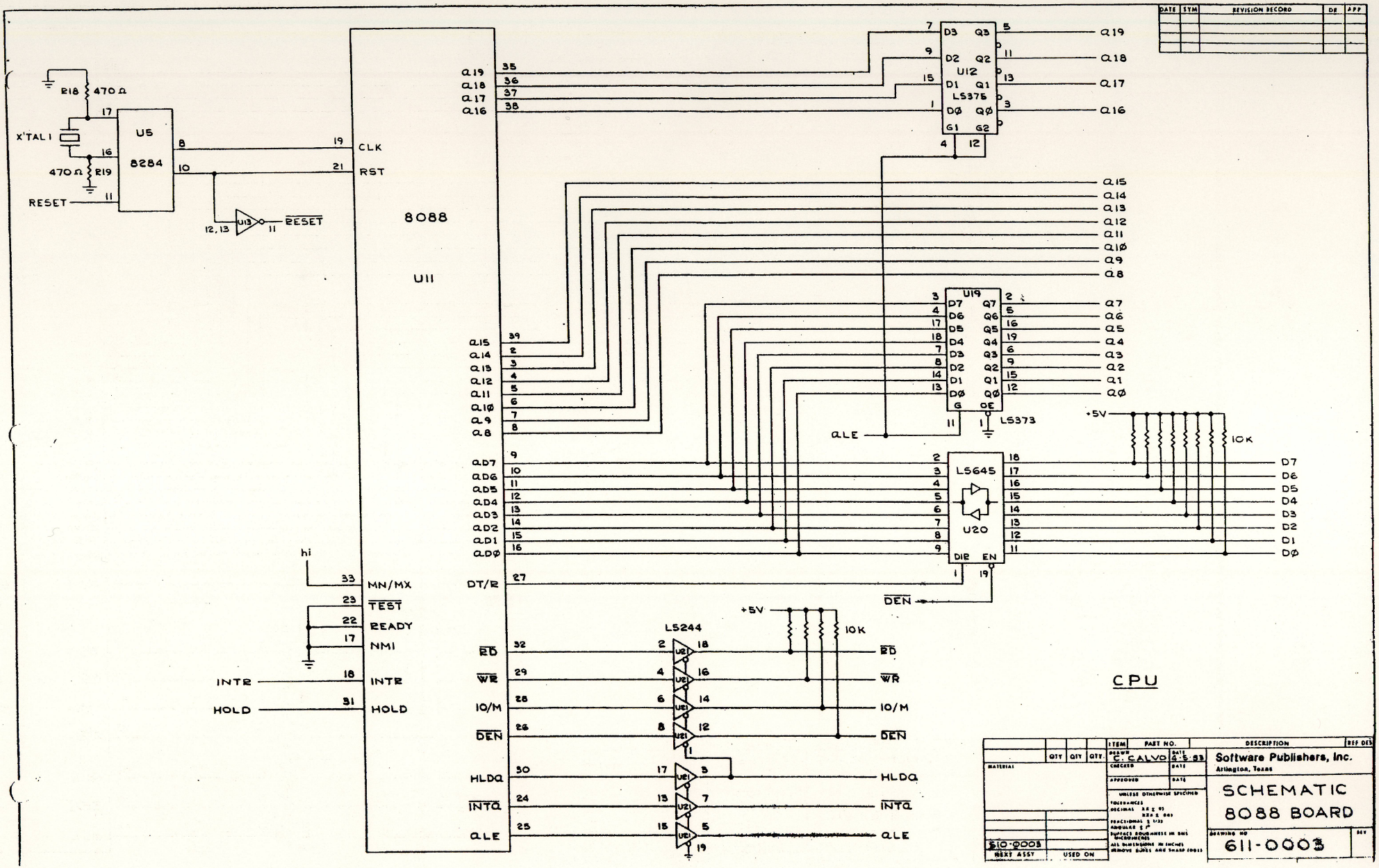
changed from c
J4 J5 -
J1 J4 J5

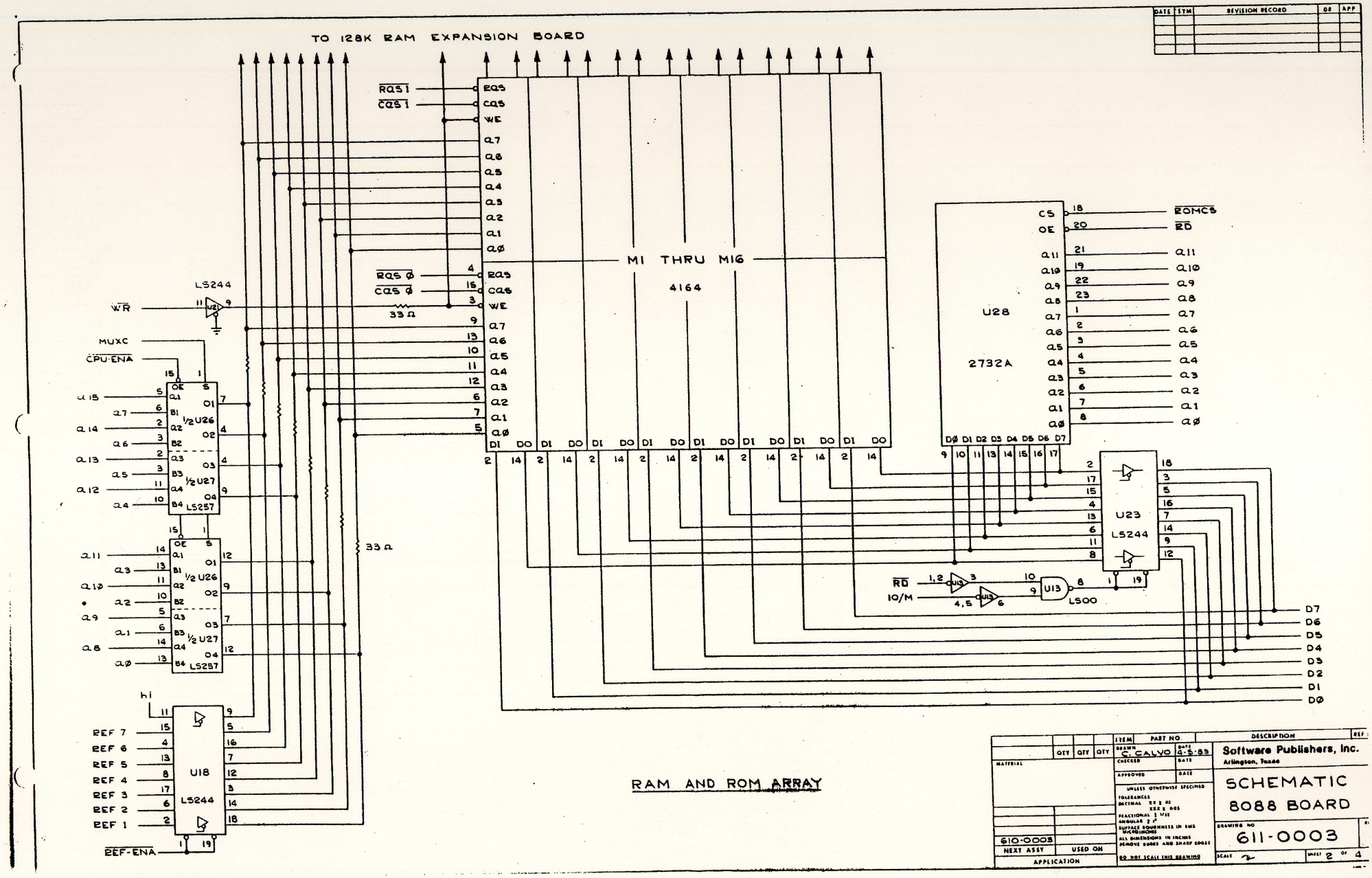
INSTALLATION



ISOLATE & GROUND
U46, U47 7, 9
SUBSTITUTE 8837 for U46, U47
REMOVE R14

SCHEMATIC PIGGY-BACK BOARD
DECMATION 12-MAY-84
Revised 23-JUL-84


R18 470Ω
X'TAL 1
R19 470Ω
RESET
U5
8284
CLK
RST
RESET
U13
8088
U11
Q19
Q18
Q17
Q16
U12
LS375
Q15
Q14
Q13
Q12
Q11
Q10
Q9
Q8
U19
LS373
Q7
Q6
Q5
Q4
Q3
Q2
Q1
Q0
ALE
+5V
10K
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
LS645
U20
DIR
EN
D7
D6
D5
D4
D3
D2
D1
D0
DT/R
DEN
hi
MN/MX
TEST
READY
NMI
INTR
HOLD
LS244
RD
WR
IO/M
DEN
HLDA
INTA
ALE
CPU
DATE
SYM
REVISION RECORD
DR
APP
ITEM
PART NO.
DESCRIPTION
DES DES
QTY
DRAWN
C. CALVO
4-5-83
CHECKED
APPROVED
UNLESS OTHERWISE SPECIFIED
MATERIAL
NEXT ASSY
USED ON
Software Publishers, Inc.
Arlington, Texas
SCHEMATIC
8088 BOARD
DRAWING NO
611-0003
611-0003
REV



RAM AND ROM ARRAY

DATE	SYM	REVISION RECORD	DR	APP

ITEM	PART NO	DESCRIPTION	REF

	QTY	QTY	QTY
MATERIAL			

DRAWN C. CALVO DATE 4-5-83
CHECKED DATE
APPROVED DATE

UNLESS OTHERWISE SPECIFIED
TOLERANCES
DECIMAL .XX ± .02
.XXX ± .005
FRACTIONAL ± 1/32
ANGULAR 2°
SURFACE ROUGHNESS IN RMS MICROINCHES
ALL DIMENSIONS IN INCHES
REMOVE BURRS AND SHARP EDGES
DO NOT SCALE THIS DRAWING

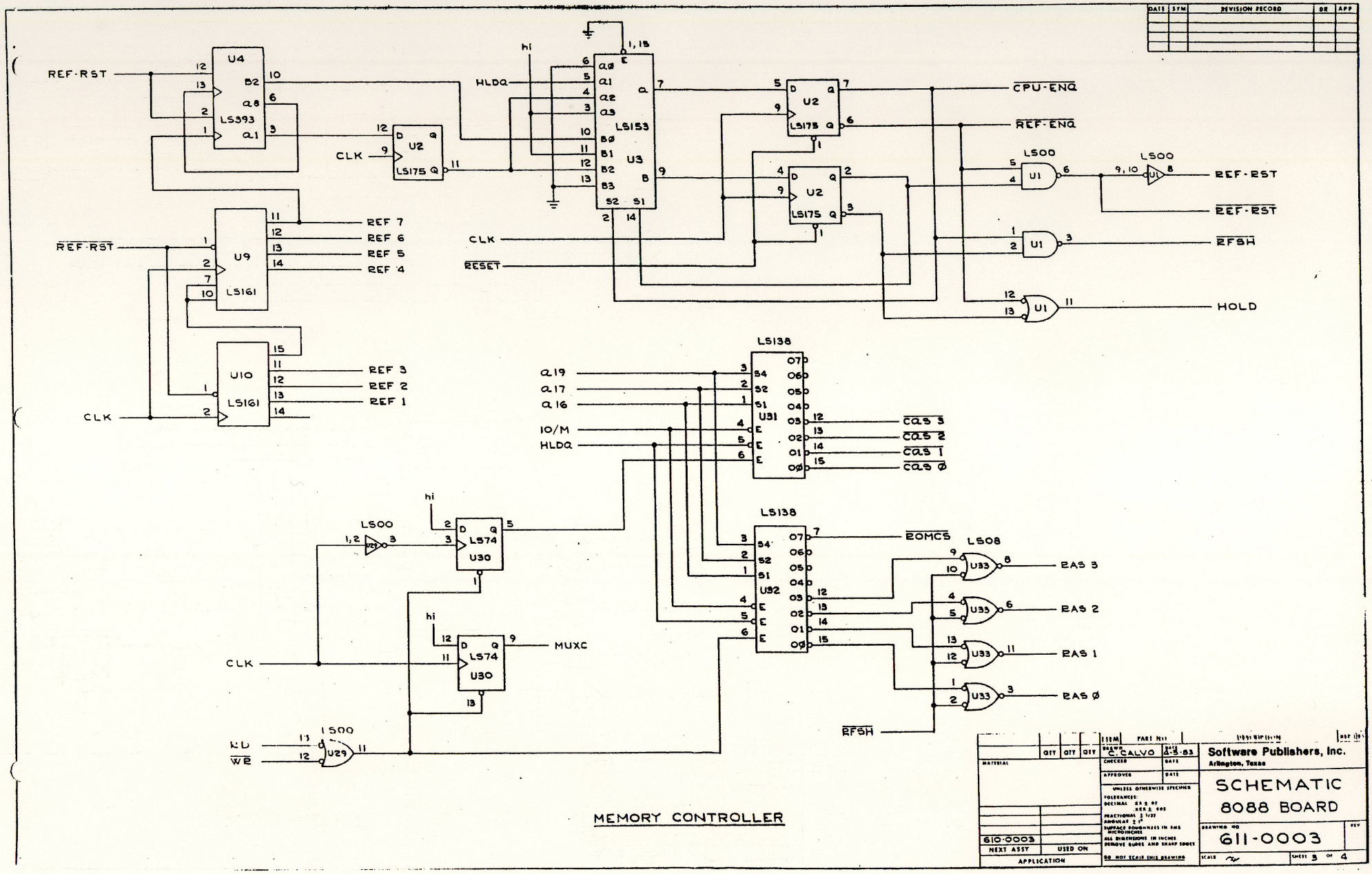
610-0003
NEXT ASSY | USED ON
APPLICATION

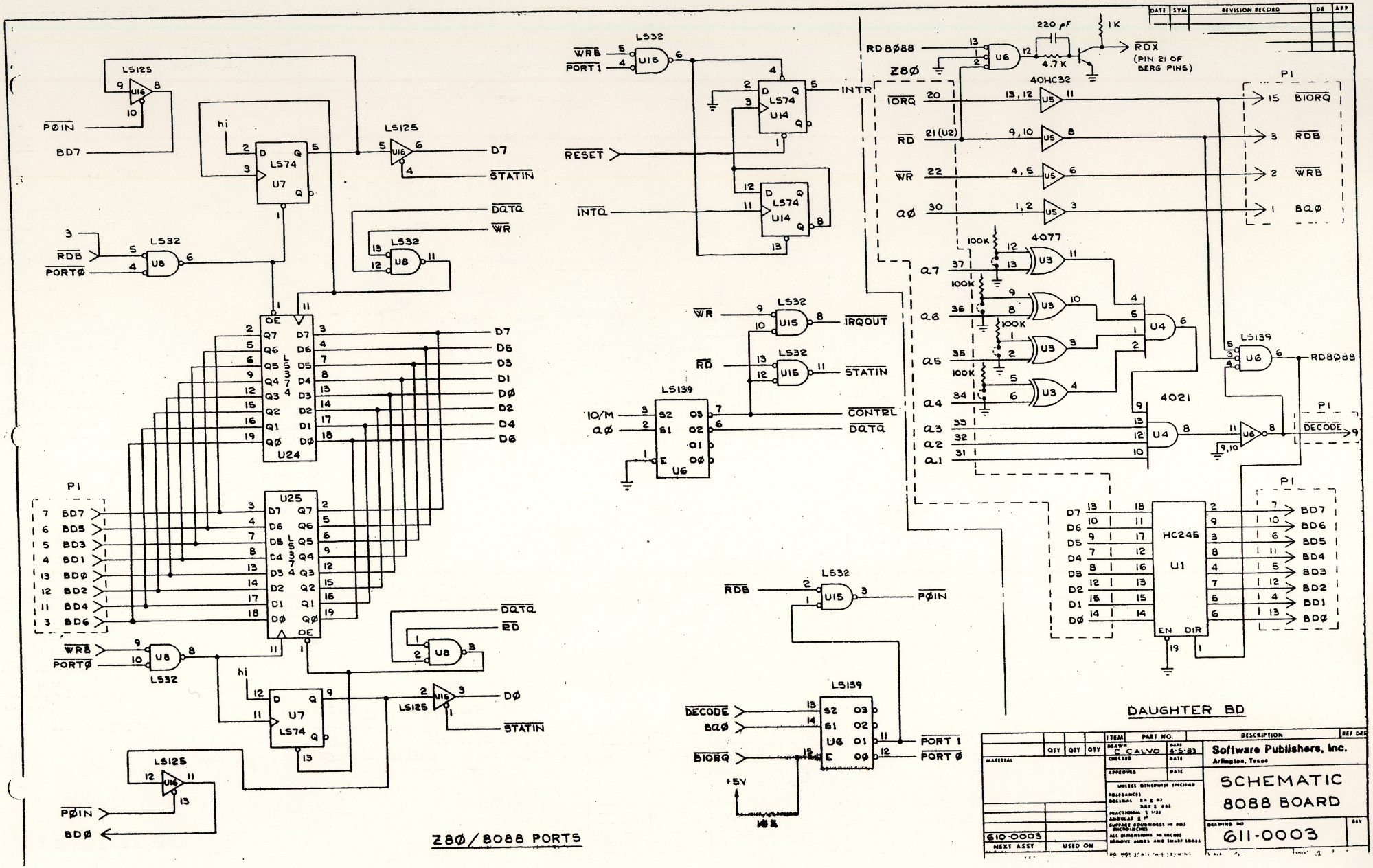
Software Publishers, Inc.
Arlington, Texas

SCHEMATIC
8088 BOARD

DRAWING NO 611-0003

SCALE | SHEET 2 OF 4


REF-RST
U4
LS393
CLK
U2
LS175
HLDQ
hi
LS153
U3
CPU-ENQ
REF-ENQ
LS00
U1
REF-RST
REF-RST
RFSH
HOLD
RESET
U9
LS161
REF 7
REF 6
REF 5
REF 4
U10
LS161
REF 3
REF 2
REF 1
LS138
U31
a19
a17
a16
IO/M
HLDQ
CAS 3
CAS 2
CAS 1
CAS Ø
U30
LS74
MUXC
U32
EOMCS
LS08
U33
BAS 3
BAS 2
BAS 1
BAS Ø
U29
RD
WE
MEMORY CONTROLLER
Software Publishers, Inc.
Arlington, Texas
SCHEMATIC
8088 BOARD
611-0003
610-0003
C. CALVO
4-5-83
SHEET 3 OF 4


Z8Ø/8088 PORTS
DAUGHTER BD
SCHEMATIC
8088 BOARD
611-0003
Software Publishers, Inc.
Arlington, Texas
C CALVO
4-5-83
610-0003
PØIN
BD7
BD6
BD5
BD4
BD3
BD2
BD1
BDØ
D7
D6
D5
D4
D3
D2
D1
DØ
STATIN
DATQ
WR
RDB
PORTØ
WRB
PORT1
RESET
INTQ
INTR
IRQOUT
RD
CONTRL
IO/M
ED
BIORQ
DECODE
PORT 1
PORT Ø
RD8Ø88
RDX
(PIN 21 OF BERG PINS)
220 pF
1K
4.7K
100K
4077
4021
HC245
LS125
LS532
LS574
LS139
40HC32

CP/M-86 SOFTWARE INSTALLATION

Supplement to D201 Installation Guide

Decmation, Inc.
3375 Scott Blvd. #236
Santa Clara, CA 95054

(408) 980-1678

DOC 1004-12/26/84

SOFTWARE INSTALLATION

CP/M-86 and MS-DOS are distributed on 8-inch floppy disks in standard CP/M format. For users who do not have diskette drives on their system the software can be provided on other media, such as 9-track tape. In this case the alternate media will contain files which serve as CP/M disks.

The following files are used as indicated:

CPM.SYS -CP/M-86 operationing system which must be on disk A: to run CP/M-86.

Z88.COM -A CP/M-80 program which is executed to start CP/M-86.

Z80.CMD -A CP/M-86 program which must be executed to return to the CP/M-80 environment.

Media Compatibility

CP/M-80 and CP/M-86 share the same disk structure. The best distribution media for ordering software is standard 8-inch CP/M format for those DEC users with floppy disks. Rainbow CP/M format, on 5-inch, RX50 disks, can be read under most operating systems with Decmation software.

How to Run CP/M-86

The file CPM.SYS must be on disk A:. To start CP/M-86 run the CP/M-80 program Z88.

A>Z88
(CP/M-86 boots with message)

To return to CP/M-80 run the CP/M-86 program Z80.

A>Z80
(CP/M-80 prompt returns)

Running under CP/M-86 is almost the same as under CP/M-80. The disk assignments are the same and the commands are very similar. CP/M-86 executable files have the file type .CMD, rather than .COOM.

A. Baldwin

D100Q CSR = 177360_8
VEC = 214_8

HARDWARE INSTALLATION

D100Q, Z80 PROCESSOR CARD

16-May-83

Decmation
3375 Scott Bl. #236
Santa Clara, CA 95051

(408) 980-1678

HARDWARE INSTALLATION

This document tells how to install the D100Q procesor card in DEC Q-bus computers, including the LSI-11, LSI-11/23, PDP-11/23, Micro PDP-11 and PDP-11/73.

The D100Q is a dual-high, standard DEC format card. It plugs into a standard bus slot wired for the Q-bus. It makes no difference if 16, 18 or 22-bit addressing is used on your computer.

The D100Q has an associated bus address. This is factory set to the octal address 177160, unless the card is marked otherwise. The D100Q's bus address must not conflict with any pre-existing device's bus address. The 8 addresses 177160-177167 are used by the D100Q.

If more than one D100Q is to be plugged into the same system, then each unit must have its own independent bus address. The standard, suggested bus addresses for successive cards are as follows:

D100Q #1 177160
D100Q #2 177360 ✓ 12 July 85 ARB
D100Q #3 177370
D100Q #4 177400 VECTOR = 214 8
D100Q #5 177410
etc.

None of these addresses conflict with standard DEC, Q-bus devices except 177400 and 177410 which conflict with the RK05 disk drive.

Many installations will have a list of the devices and their bus addresses inside the computer cabinet, or with the computer documentation. If you have no list of the existing use of bus

addresses you can test the address before the D100Q is plugged in by using the microcode ODT feature of Q-bus processors. It may be difficult to enter microcode ODT on processors which boot automatically when powered up. If your processor powers up with an "@" symbol, then it is in ODT. If it does not power up into microcode ODT you may be able to force it to enter microcode ODT by running the shutup program in RSX or by unloading the disk drives and forcing a system crash. In microcode ODT you may test a bus address as follows:

@17777160/?

If the system responds with a question mark, as shown above, the bus address is empty. If the system responds with a number, such as 000000, then something is answering that bus address. We have shown the bus address as it would be typed for 22-bit addressing. Systems which have 18 or 16-bit addressing will accept the bus address typed this way. Test the 4 word-addresses used by the D100Q:

@17777160/
@17777162/
@17777164/
@17777166/

If nothing answers the bus then you will know that no conflict exists.

The actual hardware installation process is quite simple, but if you are not sure that you have the technical skill and experience, please seek assistance. Your best source of assistance may be the person who services your computer. The sequence of steps is to 1) Turn off the power to the computer. 2) Open the computer's box to get access to the backplane unit. 3) Install the D100Q

in an empty slot wired for the Q-bus. Be sure that it is seated and lined up with other similar cards. 4) Power the computer up, make sure that it still boots, and continue with the software installation.

Installing the Printed Circuit Card

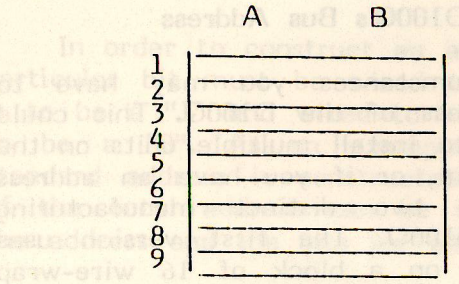
There are many different types of boxes used to hold Q-bus computer systems. Some are manufactured by the Digital Equipment Corporation, others by many different companies. The most common boxes supplied by DEC are the BA11N, BA11M and Micro PDP-11. These boxes are described as follows:

BA11N - Opens at back to gain access to the backplane. Has a quad-high backplane with 9 slots, each slot with connector rows A-B-C-D. Only A-B rows are wired with the Q-bus. Thus the D100Q must be inserted in a free slot in the A-B positions. This box typically holds 11/23 plus systems.

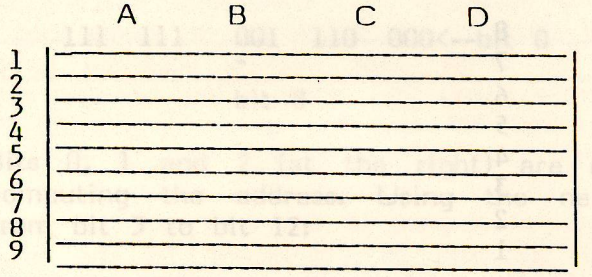
BA11M - Open in the front. Has 4, quad-high slots with connector positions A-B-C-D. Both the A-B and the C-D positions have the Q-bus. The D100Q may be plugged into any slot in this unit.

Micro-PDP-11 - This unit may be an under-table vertical mount unit, a tabletop unit, or a rack mount unit. Usually it contains 5-inch floppy disks and a winchester disk drive. Access to the backplane is by removing a cover plate at the back of the unit. There are 8, quad-height slots. All slots are wired for the Q-bus except for the C-D postions of slots 1, 2 and 3.

In DEC terminology, what would normally be called the width of a printed circuit card is called the height. A dual-high board, such as the D100Q is 5-1/4 inches wide and 8-1/2 inches long. To orient yourself with the numbering of the slots in a backplane unit note that the side of the printed circuit boards on which the components are mounted points toware slot number 1. Be sure to plug in the D100Q so that the component side points in the same direction as the component side on boards already installed in the system. Usually slot number 1 is at the top for a horizontal unit or at the right for a vertical unit, when you look at the top of the cards. Figure 1 shows a schematic layout of the slots in typical backplane units.



Dual Height Backplane Layout



Quad Height Backplane Layout

(Components face toward slot 1.)

FIGURE 1 - TYPICAL Q-BUS BACKPLANE LAYOUTS

How to Change the D100Q's Bus Address

In certain circumstances you may have to change the bus address of the D100Q. This could happen if you need to install multiple units on the same computer system, or if you have an address conflict. There are two distinct manufacturing variations of the D100Q. The first version uses wirewrap connections on a block of 16 wire-wrap pins (J12) located at the lower left of the board. The 12 wire-wrap pins are numbered as follows:

9	8
10	7
11	6
12	5
13	4
14	3
15	2
16	1

Pins 8 and 9 are connected to +5 volts. Pins 1 and 16 are connected to ground. Pins 2 and 10 must always be connected to ground. The remaining pins are assigned to bits in the bus address as follows:

Bit	Pin
3	14
4	6
5	4
6	3
7	7
8	5
9	13
10	11
11	15
12	12

In order to construct an address the pin for a particular bit must be wired to ground if the bit is to be a "1" and wired to +5 volts if the bit is to be a "0". As an example consider how the standard address is constructed. The least 12 bits of the octal address can be computed by writing the address and the bits for each digit as follows:

```
 7    7    1    6    0

111  111  001  110  000<--bit 0
          ^              ^
          bit 8
```

Bits 0, 1 and 2 (at the right) are not used in computing the address. Using the next 10 bits, from bit 3 to bit 12:

Bit	Value	Pin to	
3	0	13	+5
4	1	6	gnd
5	1	4	gnd
6	1	3	gnd
7	0	7	+5
8	0	8	+5
9	1	13	gnd
10	1	11	gnd
11	1	15	gnd
12	1	12	gnd

To change the address of the card from the standard factory setting of 177160 to 177360, the first alternate address, it is only necessary to move the single wire connected to pin 7 from +5 to gnd. To use the next alternate address, 177370 one more wire, the wire connected to pin 13 must be moved from +5 to ground.

Alternate Version of D100Q PC Card

The alternate version of the D100Q can be identified by a small "piggy-back" printed circuit board mounted on the surface of the main printed circuit board. The piggy-back board has has 7 jumpers in the PC etch which are numbered J3 through J9. The standard address of 177160 is built in to the piggy back board by PC traces connecting the jumper positions. Each jumper has 3 circular pads arranged in a trianglular pattern. One pad is the common which is to be connected to either of the other pads with a soldered jumper wire or pre-existing PC trace. One of the other pads is marked "1" and the other is marked "0". Connect the common pad to the "1" pad for a 1-bit in the address. Connect the common pad to the "0" pad for a zero in the address. The jumper positions J3 through J9 correspond to the bits 3 through 9 in the address. A soldering iron and a sharp knife to cut existing traces are necessary to modify the address. The jumper positions for some addresses are shown in the following table:

J4	J6	J5	J7	J3	J2	J1	Bus Address
1	0	0	1	1	1	0	177160
1	0	1	1	1	1	0	177360
1	0	1	1	1	1	1	177370
0	0	1	1	1	0	0	176340

A. Baldwin

SOFTWARE INSTALLATION D100Q-RT-11
OPERATING SYSTEM SPECIFIC GUIDE

3-April-84

Decmation, Inc.
3375 Scott Bl. #422
Santa Clara, CA 95051

(408) 980-1678

This manual contains operating system specific information for installing the D100 to run under RT-11.

Table of Contents

1.0 Introduction

In addition to this document you should consult the **D100 Software Installation Guide.** This document contains information which is specific only to the RT-11 operating system.

The executable control program for RT-11 is named CP.SAV. It will run under the RT-11 single job monitor, versions 3, 4 and 5. A special version which will operate under RT-11 version 2 is available. Although CP.SAV will operate under the forground-background monitor, no forground job which needs to use the system console can operate at the same time. CP.SAV will not run under the XM monitor. No special hardware options, such as extended arithmetic instructions or floating point are required. CP.SAV should run with any processor which supports RT-11 and has a DL type serial port interface for the terminal. 28K words of memory is required, except that 24K may be sufficent under the SJ monitor when there is not a large amount of space taken by device drivers. It makes no difference if the system has 16, 18 or 22 bit addressing.

2.0 Program Patches

For the majority of users no special installation procedure is required. It is only necessary to run CP.SAV to start CP/M, assuming of course that the D100 card has been previously installed. Users who have non-standard hardware configurations may have to make one or more patches to the CP.SAV file. Patches may be made for the following reasons:

1) Non-Standard Address for D100 Card

The standard interface address for the D100 card is 177160 (always expressed as if for 16-bit addressing). Some users may have to change the jumpers on the card due to an address conflict. CP.SAV is patched with the new address, as a single word, at location **IBASEI.**

2) Non-Standard Floppy Disk Name

Some users may have floppy disks with non standard names rather than DY or DX. When disks in standard CP/M format are used only these names are accepted to prevent the program from trying to make the user's system disk into a CP/M floppy. Other names, such as FW, can be patched into the program at location **FNAM.** In the distribution version, FNAM is an 8-byte (4-word) array containing the following byte string: "DX03DY03". This indicates that the legal floppy disks are DX0: through DX3: and DY0: through DY3:. If you wish to include the names FW2: and FW3: at the expense of DX, then change the string to read: "FW23DY03".

3) No Special Function Call Support For Floppy Disks.

DEC handlers for 8-inch floppy disks support special function calls which make it possible to read physical sectors, rather than blocks. Track 0 of a floppy disk can only be accessed via these calls. Setting location **NSPFN** to a 1 will cause the program not to use special function calls. Block reads will be

for 'multi terminal' monitors - patch RMON offsets 304, 306, 310, 312 to desired CP/M console terminal address

used to access physical sectors needed to use CP/M format diskettes. Track 0 then cannot be read. This has the minor consequence that it is not possible to boot CP/M from a CP/M format floppy disk, since the CP/M system partly resides on track 0. A bootable pseudo disk, CPMDSK.CDK, is provided in the distribution kit for this situation. Drivers distributed by the manufacturers of non-DEC floppy disk drives sometimes do not support special function calls properly. If you can boot CP/M using the RT file CPMDSK.CDK, but can't boot from a floppy disk, and you have non-DEC floppy disk, modifying NSPFN may be necessary.

* 4) **Auxiliary Serial Port Support**

A CP/M program can have access to a DL type serial port installed on the host computer. Access is via special CP/M BIOS calls. The control-status register address and the address of the interrupt vector must be patched at location MDCSR in CP.SAV. For example, if your auxiliary serial port has a CSR of 177500 and a vector of 300 then the two words starting at MDCSR should be changed to 177500 and 300 (octal).

5) **Default Special Shift Character**

The caret (^) character is used to prefix commands directed to the control program when under CP/M. A different character can be patched at location CARET. Patch the low byte of the word at that location. The high byte should remain unchanged.

6) **Non-Standard Console Terminal Address**

The standard RT-11 console terminal has a CSR address of 177560 and a vector of 60. An alternate address-vector pair, or several different alternate pairs may be patched at location UVEC. The control program is able to determine the CSR address by examining the monitor, but needs the table to find the vector address, given the CSR address. UVEC is an array of 8-words. This allows 4 pairs of CSR and vector addresses. For example, if your alternate console is at CSR 176500 with vector 300, then place 176500 at location UVEC. Place 300 at UVEC+2. Additional pairs may be placed at UVEC+4, UVEC+10 and UVEC+14.

7) **Printer Driver Problem**

Under some versions of RT,with some printers, partial blocks of print data do not always print correctly. Changing the word at loction ITSX from 0 to 1 will cause the program to send full blocks of data to the printer.

2.1 How to Patch CP.SAV

The patch loction is given as a symbolic name in this document. For example: IBASEI. To find the absolute, octal address of the location print or type the memory map (CP.MAP) and find the symbolic name. These are mostly at the end of the map file. The octal address immediately follows the name in the listing. For example, in one release, the address of IBASEI is 073232. There are numerous ways to patch a program

under RT-11. Utility programs such as SIPP or PATCH may be used. The GET, E, D, and SAVE commands may be used as follows:

```
.GET CP.SAV          (Get program to memory.)
.E 73232             (Examine location 73232.)
177160          (Old contents printed.)
.D 73232=177360      (Change contents.)
.E 73232        (Confirm change.)
177360          (New contents print.)
.SAVE CP.SAV         (Save modified copy.)
```

In the example above the location 73232 is changed from 177160 to 177360. Note that only whole words can be changed, located at even addresses. The "D" commands are not essential and are useful for building confidence. If you need to patch byte by byte either use a utility program, such as SIPP, or manually construct the word made up of 2 bytes. Obviously, you should not apply this procedure to your only copy of the program. For those not familiar with octal numbers a sequence of consecutive word addresses is: 73232, 73234, 73236, 73240, 73242, etc. Only the digits 0-7 are allowed.

If you wanted to put the byte string "FW" in a word, the word value to patch could be computed as follows:

"F"=106 in octal. "W"=127 in octal.

```
high byte="W"              low byte="F"
01 010 111                 01 000 110

     total 16-bit word:

0 101 011 101 000 110  = 053506 in octal
```


The high byte of a word has an address one higher than the low byte's address. The entire word is addressed by the address of the low byte, which is always even.

3.0 Utility Programs for RT-11

PCOPY.SAV

The program PCOPY will turn a floppy disk in CP/M format into a pseudo disk file 502 blocks long. The same operation can be performed under CP/M. The advantage of PCOPY is that you can run it on a system which does not have a D100 processor card installed. The program prompts for the input device and the output file name.

CCOPY.SAV

The program CCOPY copies a CP/M format floppy disk, including track 0 which is missed by the RT-11 device copy operation. CCOPY will run on a system without a D100 processor card installed.

3.1 Rainbow 5-1/4 inch CP/M Disk Support

On PDP-11 systems which are equipped with RX50, 5-inch diskette drives, disks from the DEC Rainbow computer can be read and written in CP/M format. In order to specify that the disk is in rainbow CP/M format specify DU1: or DU2: without a file name in the setup mode. For example:

```
-->B: DU1:
```

Rainbow format CP/M disks cannot be used as disk A: because there is no place in the format to hold the CP/M system tracks. Thus you cannot boot from a Rainbow format diskette. All other operations can be performed. Do not expect to run CP/M-86 programs unless you have a CP/M-86 processor card.

May 86

SOFTWARE INSTALLATION
AND
SYSTEM MANAGERS GUIDE
D100-TSX+

Decmation, Inc.
3375 Scott Blvd. #236
Santa Clara, CA 95054

25-March-85

This manual contains operating system specific information for installing and using the D100 to run CP/M-80 software under TSX+.

DOC-1040-3/25/85

TABLE OF CONTENTS

1.0 INTRODUCTION

The D100 coprocessor comes in 2 version, the D100Q for Q-bus* and the D100U for unibus*. From the software standpoint both versions work the same.

The D100 appears to the host computer as a peripheral device. The D100 hardware interface consists of 4 word registers on the host Q-bus or unibus. The D100 does no DMA transfers and does not interrupt the host computer. (The D100Q is capable of interrupting, but this feature is not currently used.)

Under TSX+ a control program (CPT.SAV) is run to use the D100. The control program accesses the registers of the D100 via the I/O page of the PDP-11*. For this reason CPT.SAV must run with operator privilege. A special account may be established with access rights such that a user can run CPT.SAV with operator privilege but not be able to damage the system.

The control program periodically suspends execution so as not to hog the system. The suspension lasts for 2-ticks (33 milliseconds) by default. When there is no activity, but the CP/M prompt (A>) is present, a small amount of system time is used, since the program checks every 2 ticks for activity on either the user's terminal, or on the D100. This scheme is actually highly efficient, since it avoids considerable system overhead involved with interrupts and calls to drivers.

Except when typing slowly, data transfers between the system and the D100 involve buffers full of data. In particular the terminal output is buffered in a 64-byte circular buffer. When fast output to the terminal is taking place, buffers of up to 64 characters are moved at a time. The TSX+ high efficiency terminal output mode is used.

Multiple D100 cards may be installed on the same system to allow multiple users to run CP/M software at the same time. The number of simultaneous users permitted is equal to the number of D100 units installed on the system.

It is possible to use the D100 in a batch mode, so that many users can share the same card. This requires that the task to be done in the CP/M compatible environment is susceptible to batching. For example assemblies, but not word processing. A detached line can be used, or access from the users terminal can be on a wait for card free basis. TSX+ virtual lines can be used to interrupt CP/M activity on a terminal.

D100 software supports a virtual CP/M serial port under TSX+. This allows CP/M software to perform serial communications using a serial port on the PDP-11. A driver for this purpose may be optionally installed. DL type serial ports are necessary to use the driver. Users lacking DL type serial ports

can still use certain CP/M communications programs via the terminal port.

Floppy disks in standard 8-inch format can be read and written by users who have RX01/02 compatible drives. Rainbow* format CP/M floppy disks can be read and written by users who have RX50 disk drives.

The D100 uses TSX+ files as virtual CP/M disks. These disks can be as large as 8-megabytes each and as may as 8 can be in use at one time by any user. Users can share the same virtual disk on a read-only basis. Virtual disks can be saved and manipulated like any other TSX+ file. Floppies can be used in CP/M format, or to hold TSX+ files which serve as CP/M disks. CP/M format disks in double density are not supported because there is no standard double density format for 8-inch CP/M disks.

File transfer between TSX+ files and CP/M files is supported. Either binary or ASCII files may be exchanged.

Command files, under both TSX+ and CP/M can be used in a variety of ways to set up automatic sequences.

When multiple D100 cards are installed, it will save memory if the software is installed using a shared run time system. This will reduce the memory requirement for each user after the first to 16KB.

2.0 TSX+ SYSTEM GENERATION REQUIREMENTS

Depending on the options already incorporated in your TSX+, it may or may not be necessary to do a sysgen. To change your system it is necessary to edit the file TSGEN.MAC and follow the system generation procedures give in the TSX+ documentation. One feature which is absolutely necessary is the TSX+ real time support feature. To enable it the statement:

```
RTVECT=1.          (or a number larger than 1.)
```

must appear in your TSGEN.MAC file.

It is desirable, but not necessary, that RTVECT be set to a number equal to the maximum number of real time interrupt vectors used already, plus the number of D100 cards installed on the system. This allows for the use of the better of 2 available methods of allocating D100's between users.

Another feature which is necessary if more than one D100 card is installed is support for file locking. This is enabled by the following statements in TSGEN.MAC:

```
MAXSF=n*8
MAXSFC=n*8
MXBLK=2.
```

Where n is the number of D100 cards installed. You should add to the above numbers any existing requirements for file locking. In practice smaller numbers will suffice. The "8" is

related to the maximum of 8 virtual disks in use at one time in the CP/M compatible environment. In most situations 8 disks are rarely in use at one time.

If you intend to use communications programs which will transmit data into the terminal port (rather than the MX driver supplied) then set the terminal input buffer to a large enough size to hold the record sent (300 bytes for the **Move-It** program).

2.1 LIST OF TSX+ SYSGEN FEATURES

This following is a list of relevant entries which may be placed in TSGEN.MAC prior to performing system generation.

RTVECT=n — Set n to the number of D100's plus existing requirements for vectors. Alternately, n may be any non-zero value if a less effective allocation method is to be used.

MAXSF=n*8
MAXSFC=n*8
MXBLK=2. — Add your existing requirements for file locking to the numbers given. "n" is the number of D100's.

DEVDEF <MX>,NODMA — Add this statement to TSGEN.MAC only if you want virtual serial port support using the driver supplied. Additional drivers <MY>, <MZ>, etc. may be added if multiple virtual serial ports are to be installed. The additional drivers are copies with different names and different hardware addresses.

IOABT=1. — Should be set to 1 if you are using the MX driver to assure proper turn off of the driver if the control program aborts.

RTDEF <SY CPTSHRSAV>,R,32 — Include this statement only if you have multiple D100's and want to use a shared run time system to save memory.

2.2 LIST OF PATCHES TO CPT.SAV (CPTS.SAV)

Two copies of the control program are supplied: CPT.SAV and CPTS.SAV. CPTS.SAV is the version for to use if multiple cards are installed and it is desired to use a shared run-time system to conserve memory. The control program CPT.SAV, or CPTS.SAV, may be patched in a variety of ways to obtain modified operation or special effects. Patching a copy of the control program may be done by using the DEC utility SIPP, or

by using the GET, SAVE, and D commands under RT-11. Patch locations are refered to by symbolic names. The octal address for any symbolic name must be read from the CPT.MAP or CPTS.MAP file. These files may be printed or typed.

IBASEI — This is an array with room for the addresses of up to 8-D100's installed on the system. The usual sequence of addresses appearing in the array are the following:

```
IBASEI+0    177160
IBASEI+2    177360
IBASEI+4    177370
IBASEI+6    177400
  etc.
```

As shipped from the factory, IBASEI contains:

```
IBASEI+0    177160
IBASEI+2    000000
IBASEI+4    000000
  etc.
```

The octal numbers given are the lower 16-bits of the 22-bit hardware address of successive D100 units. The control program tests the cards in successive locations of IBASEI until it finds one not busy, or encounters a zero which indicates there are no more cards installed on the system.

RTVECT — This 8-word array is zeroed as shipped from the factory. If the first word is set to a non-zero value which is divisible by 4, then it indicates that an array of "vectors" in low memory is to be used to allocate cards between users. If this location is left zero, then the program creates a file SY:CPM.DAT to maintain allocation information. Alternately, each of the first n locations in the array may be filled in with a separate vector address to be used by the corresponding D100 card defined in **IBASEI.**

INIFLG — If this word location is changed to a non-zero value, then the CPT program will expect additional input from the command line following the "RUN CPT". If there is no additional input, the program will quit. Prepare a copy of CPT with INIFLG set to non-zero for startup from command files.

FNAM

This 8-byte array contains legal names for floppy disks drives capable of reading CP/M format floppies. By default the array contains the string of bytes: "DX03DY03". This indicates that DX0: through DX3: and DY0: through DY3: are legal names. If your system has floppy disks named FW2 and FW3 then you could change the array to read: "FW23DY03". Take care not to allow a non-floppy to be designated as a floppy. It could destroy the data on the device.

NSPFN

If this one-word location is changed from zero to one, then "special function" calls will not be used to communicate with floppy drives holding CP/M format floppies. Instead, standard reads of 512-byte blocks will be performed and the desired 128-byte CP/M sector extracted from the 512-byte RT-11 block. This is slower, but certain non-DEC drives and/or software don't support special functions. It is not possible to boot from such a floppy because track 0 cannot be read.

MDMDRN

A 6-byte array containing the name for the virtual serial port driver. Shipped as "MX ". This patch allows multiple program versions with different serial ports to be used.

CARET

Contains a single byte "^" for the default special shift character. May be changed. Also can be changed while running with the command SETCARET.

CPMINI

This is a 20 byte array which contains the text: "@DK:CPM.INI" followed by a null. This is an initial setup command executed when the program starts to look for a default command file. You may change it to a different name, or see INIFLG below.

ENBRL

If this location is set to -1, then the protection provided against 2 users writing to the same virtual disk will be disabled. Otherwise the location should be zero. Normally 2 users on a multiple card system can use the same virtual disk at the same time only if they both log it on as read only (/RO switch after file name). Note that two

users both writing at the same time can corrupt the directory, especially as write output is buffered and may be held in memory for an indeterminate period before going to the disk. There are circumstances where the user may want to control this himself.

TICKS (2)
TICKS2 (1)
TICKS3 (3)
LOOPCT (5)

These 4 word locations have the integer values shown in parenthesis. They all control how and when the programs suspends execution. TICKS is the number of clock ticks (1/60th second) that the program suspends itself for when no service is required by the Z80 or the users terminal. If this is made less (1) then some CP/M programs will execute faster at the expense of 11 execution time. If it is made larger, the reverse takes place. LOOPCT is the number of times the test loop is traversed before the program suspends when no service is requested. 5 is a good value for an 11/23. It may be desirable to increase the value for more powerful computers. If it is increased too much, the program will become execution intensive, wasting 11 time. TICKS2 is the suspension ticks when more than 4 characters are transferred in a burst between the Z80 and the user's terminal. TICKS3 is the suspension ticks when CP/M is waiting for the user to type a key on the terminal. Generally it is not necessary to change these parameters unless the user is either very concerned about system loading, or very concerned about CP/M performance.

2.21 HOW CARDS ARE ALLOCATED TO USERS

There are two distinct methods. If the first location of the array RTVECT in CPT.SAV is zero, then the program writes a file SY:CPM.DAT which keeps track of which cards are busy and which are free. Record locking is used on this file. The disadvantage of this method is that if the system crashes with cards in use, then they will still be shown in use when the system is restarted. A false busy indicator will also be established if the user is kicked off for an access violation.

The second and better method is to set the patachable location RTVECT to a free address in low memory, such as

400 (base 8). In this case each successive card address patched into IBASEI will be associated with a successive "vector" starting at RTVECT. This method is the only method we could discover in TSX+ of establishing a reliable semaphore for allocating a resource between tasks. The facility is provided by TSX+ to allow a task to receive interrupts from a vector. The system calls for defining the vector have the property that a user can't get control of a particular vector address if another user has it. Also, he loses control if the task is aborted or terminated for any reason. So, an imaginary vector is associated with each D100 card. No actual interrupts take place. There must not be a conflict with any vector used by the system.

By manipulating the values in IBASEI and RTVECT the system manager can establish preferential access to various D100 cards for various users. The important thing is that the same imaginary vector must always be associated with the same physical D100. Suppose there are 3 cards on the system, one is a 10mhz unit and the other 2 are 6-mhz units. One user is to receive preferential access to the 10mhz unit. In this case prepare 2 copies of CPT.SAV:

Ordinary user copy

```
RTVECT=000400,000404,000410,0,0..
IBASEI=177160,177360,177370,0,0..
```

Preferred user copy:

```
RTVECT=000410,000400,000404,0,0..
IBASEI=177370,177160,177360,0,0...
```

The 10mhz card has the address 177370. The preferred user's copy of CPT.SAV will try to get the fast card first, while the ordinary users will get the fast card last. Suppose that the fast card also had a D201, MS-DOS processor attached. To prepare a copy of CPT.SAV which would gain access to this or nothing, make a copy of CPT.SAV patched as follows:

```
RTVECT=410,0,0...
IBASEI=177370,0,0...
```

The allocation algorithm starts with the first card defined in IBASEI and checks the corresponding RTVECT value to see if the card is free. If not, it goes to the next one, until a zero value (no card) is encountered in IBASEI. Values in RTVECT which are zero and follow a non-zero value are changed to the next sequential vector when CPT starts. Thus 400 in the first location is equivalent to filling in 400, 404, 410, 414, 420, etc. The parameter RTVECT=n in TSGEN.MAC (different than the

patchable array in CPT.SAV) must be set with n large enough to accommodate the number of cards in the system.

2.3 USING COMMAND FILES

Command files can be used in a variety of ways. Several users can use the same card for a batch oriented task by the use of command files. This can be done on the user's terminal, or a detached job can be used. An unsophisticated user can be aided in running a CP/M program. For the best use of command files a copy of CPT with INIFLG patched to 1 should be used. This causes the control program to attempt to read commands from the command line trailer (text following RUN CPT). Once the CP/M compatible environment is booted, the SUBMIT facility can be used to generate commands taken from a CP/M file. As an example the following command line will boot the CP/M compatible environment:

```
RUN CPX A:DL2:SYSDSK/GO/
```

Slashes separate separate commands. The terminal slash is mandatory. To insert a slash in a command, use 2 slashes:

```
.RUN CPX A:DL2:SYSDSK//RO/GO/
```

The /RO is part of the command, so 2 slashes are used to indicate a single slash.

By using substitution in TSX+ command files more complicated functions can be generated. The following single-line TSX+ command file gives a directory listing of a CP/M pseudo disk on the users terminal:

File CDIR.COM

```
RUN CPT A:SYSDSK//RO/B:^1/GO/DIR B:^2/EXIT/
```

This assumes a system disk SYSDSK.CDK file to be shared by many users on a read-only basis. If the user types:

```
.CDIR DL1:SRCDSK *.ASM
```

he will get a listing of all CP/M files of type *.ASM.

Input can be redirected to another file by a command line such as:

```
.CPT @FILE.EXT/
```

Such a secondary command file is limited to 512 bytes in length. Additional lines can be input from the TSX+ command file in use by the command GETCMD n, where n is the number of liines to read from the command file. The following statement takes the next 10 lines from the command file as

typed input to the control program:

```
.CPT GETCMD 10/
```

The amount of text input with the GETCMD should not exceed the internal buffer space of 512 bytes. Successive GETCMD's can be used, provided that the control program is in the setup state when the GETCMD is executed. (GETCMD is meaningless in the CP/M environment.) GETCMD simply reads the number of lines specified into the internal buffer used to substitute for typed input.

Argument substitution can be used in both TSX+ command files and in SUBMIT files. A typical application might be to bring up the **Palantir** word processor with a single command:

```
TSX+ Command file "WP.COM"

CPT A:DISKW/B:DISKWP/LST:LP/GO/SUBMIT WP/

CP/M submit file "WP.SUB"

WPVT100
EXIT
```

When the user types "WP" in response to the TSX+ prompt, execution will go straight to **Palantir.** When the user exits **Palantir,** control will return directly to TSX+. The CP/M submit file is used, because if the "EXIT" were in the TSX+ command file it would be taken as input to the word processor. Commands in a submit file are used only at the A> prompt.

If the system manager wants to set up a command file to allow a user to run CP/M software, but does not want to give the user operator privilege, then a command file such as the following can be used:

Account startup file USERX.TSX:

```
SET ERROR FATAL
ACCESS SY:CPT.SAV/RO,DL1:USRFIL.CDK,LP:,USERX.TSX/RO
R/LOCK CPT A:USRFIL/GO/
```

The account manager will have established a user account with operator privilege, but without virtual lines, or detached jobs. The file above will start CP/M, but log the user off when he leaves CP/M. The user will be logged off if he attempts to access anything except the line printer and his virtual CP/M disk. (R/LOCK does not allow additional input to be taken from a command file.) The file SY:CPM.DAT would have to be included in the access list if RTVECT in the CPT.SAV file is zero. Virtual devices can also be used to restrict the users access more broadly, but still allow him operator privilege.

2.4 Batch Mode

Suppose the system manager wants a number of programmers to be able to run a CP/M assembler, but there are 5 programmers and only one D100 available for their use. A detached line and command file can be used to run the assembler so that each programmer can give a single command and have the results of the assembly dumped in his own virtual device. If all the programmers start an assembly at the same time, then it will take longer to complete for 4 out of 5 of the programmers.

Suppose there are a total of 4-D100 cards installed on the system. The public cards are the first 3 cards and the 4th card is to be dedicated for assemblies. Then the public copy of CPT.SAV would be set up as follows:

IBASEI	RTVECT
177160	400
177360	404
177370	410
000000	--

The special copy would have **RTVECT** = 414, **IBASEI** = 177400 and **INIFLG**=1.

The special copy of CPT.SAV which accesses only the card at 177400 and waits silently for the card to be available might be named CPP.SAV. The command file used could look like:

File DOASM.COM:

```
CPP W/A:SY:COMDSK/GO/SUBMIT CASM ^1/
```

The first command "W/" tells the control program to wait for a the card to become free if it is busy.

The CP/M file CASM.SUB would contain the following:

```
COPY $1.ASM/H A:$1.ASM
ASM $1
COPY $1.PRN $1.PRN/H
COPY $1.HEX $1.HEX/H
ERA A:$1.ASM
ERA A:$1.HEX
ERA A:$1.PRN
EXIT
```

To do an assembly the user would type:

```
.DETACH DOASM MYFILE
```

Where he has a file MYFILE.ASM on DK:. Returned to him on DK: will be the files: MYFILE.PRN and MYFILE.HEX. Note that the name "MYFILE" is substituted for ^1 in the TSX+

command file. Then "MYFILE" is again substituted for $1 in the CP/M submit file. The /H in the copy commands indicate a TSX+ (Host) file. It is not always necessary to switch from the TSX+ command file to a CP/M submit file, but it is a more general method of doing things. Some CP/M programs will not read the TSX+ file because it does not give a TTY character ready indication to the CP/M program. If it did give a character ready indication, then it would bomb some CP/M programs which terminate when a TTY character is typed. Of course, this doesn't have to be run from a detached line. The user can run it on a regular line and watch the progress of the assembly.

The commands QUIET and NOQUIET give in the setup state of the control program determine if commands entered in the setup state are echoed. When INIFLG is non-zero QUIET is the default. Starting the command line with the command NOQUIET will make debugging easier:

```
.CPT NOQUIET/..../..../.../
```

If the program gets stuck in setup mode, usually typing bye will cause an exit. If it is stuck in the interactive dialog for certain commands, usually escape can be effectuated by typing "Y".

Considerably greater sophistication in the use of command files is possible and a separate manual is devoted to this topic.

2.5 Rainbow* Format CP/M Disks

Users with 5-1/4-inch, RX50 disk drives can read and write Rainbow CP/M format disks. A Rainbow disk can be assigned as one of the disks (but never disk A:) by a command such as the following:

```
-->B: DU1:
```

Where DU1: is the logical name of the disk drive with the Rainbow format diskette. There is no provision for reading Rainbow MS-DOS diskettes and it is not possible to read IBM diskettes of any type due to recording format incompatibility (even though this is possible on a Rainbow personal computer). If yyou wish to run programs which run on the Rainbow, only the programs with the file extension .COM can be run with the D100. The files of type .CMD run under CP/M-86 and require the D201 add-on card.

2.6 How to Patch The Control Program

The easiest method is the use the utility program SIPP.

The dialog is:

```
.R SIPP
*CPT.SAV
base?
offset? 1062

base      offset    old       new?
000000    001062    177160    177360
000000    001064    000000    ^Y
*^C
```

Control-Y followed by control-C ends the patching session. Control-Z, instead of control-Y, returns you to the base? question to allow patching in another area. ;A opens a location in ASCII and ;Ax stores the ASCII byte x at the location open. Remember to read the file CPT.MAP or CPTS.MAP to find the actual octal locations of a given symbolic name.

Users who do not have SIPP can use certain RT-11 monitor commands:

```
.GET CPT.SAV
.D 1062=177360
.SAVE CPT.SAV
```

These commands are not supported by TSX+. The E command allows display of a given location.

2.7 MX Serial Communications Handler

This handler is supplied in two different versions: MX.SNG and MX.SHR. Change the name of the version selected for use to MX.TSX and place it on the system disk. A file MX.SYS is a dummy driver which is non-functioning but must be installed under RT-11 for TSX+ to "see" the MX driver. The .SNG version if for a serial port which is not to be shared with a timesharing line, while the .SHR version may be used to drive a serial port which is also a timesharing line. The DEVDEF <MX > statement must be included in TSGEN.MAC. The MX handler allows CP/M programs to communicate via a DL type serial port on the system. The CP/M program must use the driver MOVBIOS3.ASM supplied in the distribution. The distributed communications program MDM3.COM already has this driver installed. The driver works only on DL type serial ports (such as the DLV11J). Only one user can use the driver at a time. An addition user will receive an error message if he attempts input or output. Multiple copies of the driver with different names can be installed and location MDMDRN in the control program patched with different names, such as "MY". The driver is perfectly capable of operating at high baud rates and has a 300-byte input buffer, sufficient for most communications protocols. System overhead caused by

communications should be quite moderate except perhaps at high baud rates. Throughput may be less than implied by the baud rate because of the time required to acknowledge each frame may be extended if the control program is sleeping. Changing the TICKS parameter may speed up communication. Caution must be exercised when sharing with a timesharing line. If the remote station sends unsolicited input and the MX driver is not in control it will excite the TSX+ logon messages. The standard SET CSR=xxxx and SET VECTOR=xxx commands must be applied to the MX driver to set the address of the port used. Reboot TSX+ after these commands. IOABT in TSGEN.MAC should be set to 1 if the MX driver is used. This assures disconnecting the driver from the control program if the control program is aborted. If this is not done MX could crash the system under certain circumstances.

A. Baldwin

D201 CO-POWER PLUS (tm) 8088 PROCESSOR
INSTALLATION AND OPERATION

Decmation is not connected with or approved by the Digital Equipment Corporation of Maynard, Massachusetts which owns the trademark "DEC."

DEC, unibus, Q-bus, PDP are trademarks of the Digital Equipment Corporation.

CP/M is a registered trademark of Digital Research, Inc.

CO-POWER and CO-POWER PLUS are trademarks of SWP, Inc.

The D201 is an adaptation for the DEC environment of the SWP CO-POWER PLUS 8088 coprocessor card.

DOC 1061 2-26-85

TABLE OF CONTENTS

1.0 General Description

The D201 is a DEC, quad-sized board which may be installed in DEC PDP-11 and VAX computers. There are two versions of the D201, the **D201U** and the **D201Q** for unibus and Q-bus computers respectively. These two different versions differ only in a few board jumpers which may be easily changed in the field. (See Appendix A.)

The D201 must be connected to a Decmation D100, Z80 processor card in order to operate. This connection is made by a flexible cable connected between top connectors on each card. The D201 is a peripheral device to the D100 and is controlled by software operating in the CP/M-80 compatible environment of the D100. The combination of the D100 and the D201 allows the user to run both CP/M-80 and MS-DOS software. CP/M-86 is an available option.

When running MS-DOS the D201 emulates an IBM personal computer (IBM-PC). The user must be using a VT-100 or compatible terminal in order to obtain a reasonably good emulation of the IBM-PC. The D201 can be used with a non-VT-100 compatible terminal. In this case it functions as a "vanilla" MS-DOS system, rather than an IBM-PC emulator.

1.1 Distribution Kit

Besides the printed circuit board, the distribution kit contains the necessary software in the form of CP/M files. Normal distribution is on 8-inch diskettes in standard CP/M format, however any DEC compatible media may be used. The major software components are the following:

MS.COM — A CP/M program used to start MS-DOS running.

MSDOSIMG.COM
: A CP/M file which serves as an initial MS-DOS disk.

INDISK.COM
: A CP/M program which initializes CP/M disks to serve as MS-DOS disks. (CP/M disks are usually files under the host operating system.)

RMDSK.COM
: A CP/M program which enables the memory of the D201 to be used as a disk in the CP/M-80 compatible environment.

I88.COM — A CP/M program which resets or initializes the D201.

The following files are MS-DOS files contained within the CP/M file MSDOSIMG.COM. They can only be viewed after MS-DOS is booted.

COMMAND.COM
: This is the MS-DOS command interpreter. It must be resident on MS-DOS disk A: for MS-DOS to boot successfully.

EX.COM — This program returns control to the CP/M-80 compatible environment, exiting MS-DOS.

SEN.COM — Converts (sends) MS-DOS files to CP/M files.

REC.COM — Converts (receives) CP/M files from MS-DOS.

PORT.COM — When executed installs serial communications software which remains resident and allows communications over a serial port on the host DEC system by MS-DOS communications programs.

XPORT.COM
: Used on an IBM-PC to speed up the operation of serial communications. Must be transported to a PC to be used.

VDISK.COM — This program creates a memory disk (disk F:) by reserving part of the D201 memory for this purpose.

RDISK.COM — This program initializes the D201 for use as a memory disk by CP/M-80 programs and returns control to the CP/M-80 compatible environment.

ENHANC.COM
: When executed installs software which implements a more complete IBM-PC emulation.

Many other files are included which are a standard part of MS-DOS. These include a debugger and linker (but not an assembler). Consult the MS-DOS manual included with the distribution kit.

2.1 How MS-DOS Uses Disks and I/O Devices

MS-DOS runs under the CP/M-80-compatible environment created by the D100. The MS-DOS software is not "aware" of what kind of DEC computer it is installed on. The interface

with the DEC computer is carried out at the CP/M-80-compatible level. MS-DOS only talks with the CP/M-80-compatible operating system running on the D100. MS-DOS uses the CP/M terminal, the CP/M list device and specially initialized CP/M disks as MS-DOS disks. The software supplied with the D100 makes the connection between CP/M disks and devices and the host operating system.

DISKS:

1) CP/M disks are files (e.g. DISK.CDK) under the host operating system.

2) MS-DOS disks are CP/M disks which have been specially initialized.

User Terminal:

1) The CP/M terminal is simply the user's terminal on the host operating system.

2) MS-DOS sends all terminal I/O to the CP/M terminal. Options which affect the operation of the terminal for CP/M programs will have a similar effect under MS-DOS.

Print Device:

1) The CP/M list output can generally be directed to a file or device, and sometimes to the system spool, on the host operating system.

2) The MS-DOS list output goes to the CP/M list output.

Serial Communications Port:

1) CP/M serial communications programs, under most supported DEC operating systems, can use a virtual serial port which is physically realized by a serial port on the host DEC system. This is a different serial port than the user's terminal.

2) When the serial communications support (PORT.COM) is used under MS-DOS the serial communications takes place over the CP/M communications port.

2.2 Disk Assignments for MS-DOS

Under most supported DEC operating systems up to 8 CP/M disks can be in use at one time. These disks are named

A:, B:, C:, D:, E:, F:, G:, H:. Each is actually a host file. When MS-DOS is booted, CP/M disk B: is used as MS-DOS disk A:, CP/M disk C: is MS-DOS disk B:, etc. CP/M disk A: is not used by MS-DOS except as the transfer disk when files are exchanged between CP/M and MS-DOS formats. A maximum of 4 disks (not counting the memory disk) is allowed under MS-DOS.

2.3 How to Run MS-DOS

The user must first copy the MS-DOS distribution disk from the distribution media to an **empty** CP/M disk. This can be done with a CP/M command such as:

```
A>PIP B:=C:MSDOSIMG.COM
```

Copy the other CP/M-80 files to CP/M disk A:. CP/M disk B: must contain the file MSDOSIMG.COM as the first file in the directory. Usually this will be the only file on disk B:. To boot MS-DOS execute the following commands:

```
A>I88                    (Reset thee D201.)

A>MS                     (Boot MS-DOS)
```

(For other than a VT-100 compatible terminal a slightly different command is required.)

This should boot MS-DOS and set the MS-DOS system date. The MS-DOS prompt:

```
A>
```

will be displayed. If MS-DOS does not boot investigate the following:

1) Is the D201 installed and connected to the D100 card you are using.

2) Does CP/M disk B: contain the file MSDOSIMG.COM as the first file in the directory. CP/M disk B: must not be a floppy in standard CP/M format, or a Rainbow CP/M floppy disk. It must be a host file used as a CP/M disk.

2.32 Booting and Exiting MS-DOS - Command Syntax

```
A>I88        (Resets the D201, use to reboot MS-DOS
              from a cold start.)
```

A>MS [A] [Sxxxx]

The CP/M program MS.COM boots or restarts MS-DOS. There are 2 possible optional arguments. The letter "A" causes MS-DOS to start in the vanilla MS-DOS mode, rather than in the IBM-PC emulation mode. Non VT-100 users should always use the "A" argument. It is optional for VT-100 users. The Sxxxx specifies the size of the memory in the D201 used by MS-DOS. If the Sxxxx is not included, then the memory is automatically sized. xxxx is a 4-digit hex number specifying the number of paragraphs of memory. A paragraph is 16 bytes. This is useful for testing software which is to run in a more restricted memory environment. The principal use is to limit the memory used by MS-DOS to 704k (SB000) which is necessary to use the enhanced IBM-PC emulation (program ENHANC.COM).

Examples:

A>MS (Start or restart MS.DOS with PC emulation and automatic memory sizing.)

A>MS A (Start or restart MS-DOS without PC emulation.)

A>MS SB000 (Start MS-DOS with memory set at B000 paragraphs (704K).)

2.34 Difference Between a Cold Start and a Restart

When MS-DOS is started after the program I88 (initialize) has been executed, a complete initialization is performed. MS-DOS is loaded to the D201, the MS-DOS system date and time is set and the system command files CONFIG.SYS and AUTOEXEC.BAT are executed if present.

If MS-DOS is exited with the EX command, and then restarted with the MS command, MS-DOS simply continues where it left off. the Sxxxx command has no effect since the size of system memory is determined when the MS-DOS system is initialized. If the EX command is executed as part of command file, execution will continue with the next command after the EX command. The "A" option is effective for cold start or restart.

2.36 Transporting Files Between MS-DOS and CP/M Formats

To transfer a file from CP/M disk A: to MS-DOS use the following command under MS-DOS:

A>REC FILE.EXT

To send an MS-DOS file to CP/M disk A: use the following command:

A>SEN FILE.EXT

The file name will be the same under MS-DOS and CP/M. Wild cards are allowed. For example:

A>SEN FOR?????.*

will send all the files starting with FOR from MS-DOS to CP/M.

Similar commands in the CP/M-80 environment allow exchanging files with the host operating system.

CP/M-80 files are always a multiple of 128 bytes. End of text in ASCII files is marked with a control-Z if the length of the file is not exactly a multiple of 128 bytes. Under MS-DOS files can have any length in bytes. A control-z for end of text is usually, but not always, recognized. The following command will truncate a file to the length implied by the first control-Z encountered in the text:

A>COPY FILE.EXT/A FILE2.EXT

This is needed if a CONFIG.SYS file is created as a CP/M-80 file and transported to MS-DOS.

2.38 Using the MS-DOS Memory Disk

The file VDISK.COM is a disk driver which must be installed at MS-DOS startup time with a command in the CONFIG.SYS file. Whenever MS-DOS starts up from a cold start it searches disk A: for a file named CONFIG.SYS. If this file is present the commands it contains are executed to configure the MS-DOS system. The command:

DEVICE=VDISK.COM

in CONFIG.SYS will install the memory disk as MS-DOS disk F:. The size of the memory disk is 64K bytes by default. By patching VDISK.COM following instructions in the file VDISK.DOC the size of the memory disk can be made larger, up to 800K bytes or more. The files stored on the memory disk are killed when a cold start of MS-DOS is made.

2.40 Creating New MS-DOS Disks

An MS-DOS disk is created by the CP/M-80 program INDISK.COM. This program initializes an empty CP/M disk to be an MS-DOS disk by writing a file named MSDOSIMG.DOS on the empty CP/M disk. INDISK is self-prompting and will

not allow you to make a mistake or destroy existing data. MS-DOS disks can have any of the following sizes:

MS-DOS disk size	CP/M disk size
160k	338 blocks
320K	676 blocks (664)(660)
616k	1276 blocks (1262)
1232k	2524 blocks

This table gives the approximate minimum CP/M disk size, in DEC blocks of 512-bytes, needed to accommodate a given MS-DOS disk size.

3.1 EXAMPLE OF GETTING STARTED WITH MS-DOS

In this example the user sets up his CP/M disks as follows:

```
-->A: DISK          (users CP/M system disk)
-->B: DIST          (Empty disk, 338 blocks in size)
-->C: MSYS          (Empty disk, 676 blocks in size)
-->D: DY0:          (CP/M Floppy with MS-DOS on it.)
-->GO               (Boot CP/M)

A>DIR B:

No file
A>DIR C:

No file
A>DIR D:        (Directory of distribution floppy disk.)

  MSDOSIMG DOS   64 : INDISK    COM    5
  RMDSK    COM    4 : MS        COM   64
  I88      COM    1
A>STAT B:

Bytes Remaining On B: 160k

A>STAT C:

Bytes Remaining On C: 328k

A>PIP A:=D:*.COM    (Get executable files from (floppy.)
    (file names print)
```



```
A>PIP B:=D:MSDOSIMG.DOS   (Get MS-DOS disk off
                           distribution media.)
A>STAT B:MSDOSIMG.DOS

 Recs  Bytes  Ext Acc
 1280   160k   11 R/W B:MSDOSIMG.DOS
Bytes Remaining On B: 0k

A>I88       (Reset D201)

A>MS A      (Start MS-DOS)

     (boot messages follow)
Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue  1-01-1980
Enter new date: 02-27-85

Current time is  0:00:00.00
Enter new time: 14:16:08


A>DIR       (Get an MS-DOS directory)

 Volume in drive A has no label
 Directory of  A:\

COMMAND  COM     15957   11-10-83   12:03p
EX       COM       361    7-31-84
SEN      COM       740    9-11-84    3:26a
REC      COM       682    9-09-84    4:53p
PORT     COM       145   12-26-84    1:32a
CHKDSK   COM      6468   10-19-83    7:51p
RECOVER  COM      2295   10-19-83    7:51p
MORE     COM      4364   10-19-83    7:51p
EDLIN    COM      8080   10-19-83    7:51p
DEBUG    COM     12223   10-19-83    7:52p
PRINT    COM      3808    2-28-84   11:00a
RDISK    COM       507    1-27-84    2:17a
ENHANC   COM      8188    2-25-85    1:39a
FIND     EXE      6331   10-19-83    7:51p
FC       EXE      2585   10-19-83    7:51p
LINK     EXE     42330   10-19-83    7:51p
SORT     EXE      1632    1-19-84    3:00p
XPORT    COM      1021    1-04-85   11:42a
        18 File(s)      38400 bytes free

A>EX        (exit MS-DOS)
```

```
(now the CP/M prompt prints)

A>INDISK          (Initialize MS-DOS disk with CP/M program
                   INDISK.)


ENTER DISK (B, C, D, ETC.) TO INIT TO MSDOS: C
Chose choice 1, 2, 3 or 4 below by typing the digit

1 --160K-byte disk, 64 directory entries
2 --320K-byte disk, 112 directory entries
3 --616K-byte disk, 192 directory entries
4 --1232K-byte disk, 192 directory entries
5 -- **** EXIT PROGRAM ****

entry?--> 2

A>I88      (initialize the D201)


A>MS A  (start MS-DOS)

Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue  1-01-1980
Enter new date: 02-27-85

Current time is  0:00:00.00
Enter new time: 14:16:59


A>DIR B:          (Directory of disk just initialized.)


 Volume in drive B has no label
 Directory of  B:\

File not found

A>COPY A:*.* B:              (Copy everything to B:)

A:COMMAND.COM
A:EX.COM
A:SEN.COM
A:REC.COM
A:PORT.COM
A:CHKDSK.COM
A:RECOVER.COM
A:MORE.COM
```



```
A:EDLIN.COM
A:DEBUG.COM
A:PRINT.COM
A:RDISK.COM
A:ENHANC.COM
A:FIND.EXE
A:FC.EXE
A:LINK.EXE
A:SORT.EXE
A:XPORT.COM
       18 File(s) copied

A>CHKDSK B:    (Shows disk status and memory.)


   322560 bytes total disk space
   125952 bytes in 18 user files
   196608 bytes available on disk

  1015808 bytes total memory
   989120 bytes free


A>EX      (Exit MS-DOS.)

A>
```

3.2 EXAMPLE - SETUP OF THE MS-DOS MEMORY DISK

The following example shows how to use the MS-DOS memory disk. It also illustrates the use of the MS-DOS CONFIG.SYS file. Many programs run faster when a memory disk is used.

```
A>I88             (Reset the D201)

A>MS A            (Start MS-DOS)

Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue  1-01-1980
Enter new date: 02-27-85

Current time is  0:00:00.00
Enter new time: 16:44:38


A>DIR CONFIG.SYS          (No CONFIG.SYS file is present.)
```

```
 Volume in drive A has no label
 Directory of  A:\

File not found
```

```
A>REC CONFIG.DAT
```

(Get file from CP/M created with CP/M word processor.)

```
CONFIG.DAT

A>DIR CONFIG.DAT

 Volume in drive A has no label
 Directory of  A:\

CONFIG     DAT        128    2-27-85    4:44p
        1 File(s)     66560 bytes free
```

```
A>COPY CONFIG.DAT/A CONFIG.SYS
```

(Convert file to actual length of contained text. Change name.)

```
        1 File(s) copied

A>TYPE CONFIG.SYS

DEVICE=VDISK.COM
```

```
A>CHKDSK
```

(Note available memory size without memory disk present.)

```
  322560 bytes total disk space
  257024 bytes in 30 user files
   65536 bytes available on disk

 1015808 bytes total memory
  989120 bytes free
```

```
A>EX
```

(Exit and restart MS-DOS so that it executes the CONFIG.SYS file and installs the memory disk.)

```
A>I88
```



```
A>MS A

Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue  1-01-1980
Enter new date: 02-27-85

Current time is  0:00:00.00
Enter new time: 16:45:33
```

```
A>CHKDSK
```

(Note that the size of available memory is decreased by approximately 64K bytes due to installation of the memory disk.)

```
  322560 bytes total disk space
  257024 bytes in 30 user files
   65536 bytes available on disk

 1015808 bytes total memory
  922752 bytes free
```

```
A>CHKDSK F:
```

(Now display parameters of the memory disk (F:) just installed.)

```
   61952 bytes total disk space
   61952 bytes available on disk

 1015808 bytes total memory
  922752 bytes free

A>DIR F:

 Volume in drive F has no label
 Directory of  F:\

File not found
```

(A 64K-bytes memory disk is too small to be useful. The program VDISK.COM is patched using the MS-DOS DEBUG program. Detailed instructions are given in the file VDISK.DOC. VDISK.COM is being patched to provide a 512K-byte memory disk.)

```
A>DEBUG

-N VDISK.COM
-L                          (Load VDISK.COM)
-A129
19D8:0129 DW 200,8000       (Sector count, paragraph count,
                             in hex.)
19D8:012D
-W
Writing 02E0 bytes
-Q

A>EX                        (Exit MS-DOS and restart so as
                             to reload the modified
                             VDISK.COM.)

A>I88

A>MS A

Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue  1-01-1980
Enter new date: 02-27-85

Current time is  0:00:00.00
Enter new time: 16:46:48


A>CHKDSK          (Most of the D201 memory is now used for
                   memory disk.)

   322560 bytes total disk space
   257024 bytes in 30 user files
    65536 bytes available on disk

  1015808 bytes total memory
   464000 bytes free


A>CHKDSK F:       (512K-bytes is now allocated to disk F:, the
                   memory disk.)

   518656 bytes total disk space
   518656 bytes available on disk

  1015808 bytes total memory
```



```
   464000 bytes free
```

3.3 Example - Using The Memory Disk and MS-DOS Batch Files

This example shows how the memory disk can be used to perform assemblies more rapidly than would otherwise be possible. A batch (command) file is set up which reads the source from CP/M, assembles it and then sends the list file back to CP/M. (The IBM macroassembler is an extra cost option.)

```
A>COPY MASM.EXE F:          (First copy the needed program
                             files to the memory disk.)

        1 File(s) copied

A>COPY SEN.COM F:

        1 File(s) copied

A>COPY REC.COM F:

        1 File(s) copied

A>REC DOASM.BAT             (Get the command file to be
                             used from CP/M.)

DOASM.BAT

A>TYPE DOASM.BAT            (Type out the command file to
                             be used.)

REC %1.SRC
MASM %1.SRC,%1,%1;
SEN %1.LST
EX
        (The symbol "%1" indicates that argument
        substitution will take place. The first argument in
        the command line will be substituted for %1 when
        the batch file is executed.)

A>COPY DOASM.BAT F:         (Copy the file to the memory
                             disk.)

        1 File(s) copied
```

```
A>F:          (Log on the memory disk as the default
               disk.)

F>EX

Bad command or file name          (Whoops. Forgot to
                                   copy EX.COM to
                                   the memory disk.)

F>A:

A>COPY EX.COM F:          (Fix the omission.)

        1 File(s) copied

A>F:      (Log back on to F:)

F>DIR

 Volume in drive F has no label
 Directory of  F:\

MASM       EXE      67584   12-08-81
SEN        COM        740    9-11-84    3:26a
REC        COM        682    9-09-84    4:53p
DOASM      BAT        128    2-27-85    4:46p
EX         COM        361    7-31-84
        5 File(s)     448000 bytes free

F>EX

    (The user could edit his source file for assembly under
    CP/M here.)

A>MS A

CONTINUE MS-DOS           (The continue message is output
                           when MS-DOS continues, no
                           D201 reset.)

F>DOASM VDISK             (When any file name is typed, if
                           it is of type .BAT, it is taken
                           as a batch file. "VDISK" is the
                           argument to be substituted.)

F>REC VDISK.SRC           (What was %1.SRC in the file
                           DOASM.BAT is expanded into
                           VDISK.SRC.  This command
```



```
                           moves a file from CP/M disk A:
                           to MS-DOS disk F:.)

VDISK.SRC

F>MASM VDISK.SRC,VDISK,VDISK;          (Assembler command
                                        line)
The IBM Personal Computer MACRO Assembler
Version 1.00 (C)Copyright IBM Corp 1981


Warning Severe
Errors  Errors
0       0

F>SEN VDISK.LST           (Send the output of the
                           assembler back to CP/M disk
                           A:)

VDISK.LST

F>EX                      (Quite MS-DOS, return to the
                           CP/M-compatible environment.)

A>XDIR VDISK.LST          (Use the CP/M utility XDIR to
                           examine the file transferred.)

**XDIR v3.5 - April 9, 1981**

VDISK    .LST   32k

Specified files:  1 files occupy    32K bytes
Disk A:  4K blocks, Size= 1488K, Used= 1408K, Space=    80K

A>MS A          (Go back to MS-DOS)

CONTINUE MS-DOS

F>
F>DIR F:

 Volume in drive F has no label
 Directory of  F:\

MASM       EXE      67584   12-08-81
SEN        COM        740    9-11-84    3:26a
REC        COM        682    9-09-84    4:53p
DOASM      BAT        128    2-27-85    4:46p
EX         COM        361    7-31-84
VDISK      SRC      16384    2-27-85    4:46p
VDISK      OBJ       1410    2-27-85    4:46p
VDISK      LST      30115    2-27-85    4:46p
```

```
        8 File(s)     399872 bytes free

F>DOASM REC                (Assemble a different file this
                            time.)

F>REC REC.SRC

REC.SRC

F>MASM REC.SRC,REC,REC;
The IBM Personal Computer MACRO Assembler
Version 1.00 (C)Copyright IBM Corp 1981

Warning Severe
Errors  Errors
0       0

F>SEN REC.LST
REC.LST

F>EX
```

4.1 IBM-PC Emulation

For most applications programs it is required that the D201 emulate an IBM-PC computer. There are several aspects to this emulation.

Monochrome display emulation - There are various display units available for the IBM-PC. The monochrome display is the closest to a VT-100 terminal. It has 25 lines of 80 characters. The VT-100 has only 24 lines. The extra line is hidden at the top or bottom of the user's display and may be toggled into view with a "toggle" key on the VT-100 auxiliary keypad (the period key). The IBM monochrome display unit has more displayable characters than the VT-100, so substitutes are made for obscure characters.

Keyboard emulation - The IBM-PC has a quite different arrangement of keys compared to the VT-100. Keys on the auxiliary keypad are used to emulate the IBM keys. For example the keys 1-9, 0 are used to emulate the IBM function keys F1-F10.

ROM-BIOS firmware emulation - The PC contains several thousand bytes of code in a permanent ROM. Applications programs can and do call this ROM by means of program interrupts. The most important services emulated are calls for writing on the display and reading the keyboard. Also emulated are ROM calls for reading the diskette. These are not normally used by MS-DOS, but certain programs make direct calls to the diskette.

ANSI.SYS terminal driver - IBM supplies a driver which causes the IBM display to appear to be an ANSI-type terminal. This driver is emulated by the software associated with the D201. Many programs use this mode of communicating with the display and keyboard because it is less hardware dependent, making the software transportable between different types of personal computers.

Memory Mapped Display - An IBM-PC contains a 4K-byte memory at address B0000 (704K). Anything stored in this memory will appear on the display as a character. The display screen can be read by reading cells in this memory. The D201 supports an enhanced emulation mode which uses system memory to emulate the memory mapped display. The disadvantage of the memory mapped display is that MS-DOS memory is then limited to 704K bytes. The D201 must have memory at address B0000 for the enhanced emulation mode to work. This can be achieved by installing 3 banks of 256K-bytes each, or by adding a 64K-byte memory at address B0000.

4.2 How the Display Emulation Works

The D100, Z80 processor receives messages from the D201 to read and write the display, position the cursor, etc. The D100 keeps an image in its memory of how the display should look. It also keeps a record of what is actually on the screen of the VT-100 at a given instant. Part of the Z80's time is devoted to constantly updating the VT-100 screen to agree with the image of the IBM screen in memory. Data and cursor positioning commands are only sent to the VT-100 when a difference needs to be corrected. This keeps traffic on the serial port to a minimum. Unnecessary cursor positioning commands are avoided. It is also possible for the program on the D201 to run faster than the display, since writing to the memory image is much faster than writing to a VT-100 over a comparatively slow serial port. This allows, for example, skipping to the next menu in a sequence before the previous menu is completely up on the screen. The D201 is forced to wait for the terminal only when data being listed on the screen might fly by faster than it can be viewed.

4.3 Startup in the PC Emulation Mode

The user should be at a VT-100 or compatible terminal. The VT-100 should be in ANSI mode. If the VT-100 is equipped with the advanced video option, then character attributes, such as blink, bright, reverse and underline will be used in the same manner as on the PC. If advanced video is not installed, then all attributes will be translated into reverse video. The command to boot MS-DOS is:

```
A>MS
```

or, if the enhanced emulation mode is to be used and the D201 has 768K or more memory:

```
A>MS S8000
```

The screen will be erased as MS-DOS starts.

4.4 D201 Memory Options

The D201 has provision for 4 banks of memory. Each bank consists of 8-256K dynamic random access memory integrated circuits. The memory addresses associated with each bank are as follows:

bank A	0-256k (0-3FFFFh)
bank B	256k-512K (40000h-7FFFFh)
bank C	512K-768K (80000h-BFFFFh)
bank D	768K-992K (C0000h-F7FFFh)

For D201's with 256K or 512K of memory, an additional 64K of memory can be installed in bank C for the purpose of display emulation. This memory will answer with the same memory at 4 different address within the 256K bank. If the memory configuration is 256K in bank A and 64K in bank C, then only 256K will be available to MS-DOS. If 512K is in banks A and B and 64K in bank C, then MS-DOS can use 512+64=576K of memory and will automatically size the memory at 576K. If the 64K bank is to be used for display emulation, then MS-DOS memory usage should be limited to 512K by the following startup command:

```
A>MS S8000
```

If 768K or 992K of memory is available, then MS-DOS memory must be limited to 704K by the following startup command:

```
A>MS S8000
```


if the enhanced emulation mode is to be used. The program file MS.COM may be permanently altered to automatically boot with different memory options. Follow the instructions in the file MSPAT.ASM.

4.5 Enabling the Enhanced Emulation

The command:

```
A>ENHANC
```

executed under MS-DOS loads the enhanced emulation software which remains resident until MS-DOS is cold started. If this command is placed in a file named AUTOEXEC.BAT it will automatically be executed on a cold start of MS-DOS.

4.6 The Emulated Keyboard

The keys of the IBM-PC keyboard are emulated, in a logical manner, by the keys of the auxiliary keypad on a VT-100 terminal as follows:

Function Keys:
Keys 1, 2, 3, ... 9, 0 emulate the 10 function keys: F1, F2, F3.. F10.

Esc - Enter key (or the VT-100 ESC key twice)

Home - Minus (-)

End - PF2 key

Arrow Keys - Arrow Keys

—>| - Tab key

Two keystrokes are required to emulate the following IBM keys:

|<— - PF1 then left arrow (<--)

PG UP - PF1 then up arrow

PG DN - PF1 then down arrow

PRT SCRN - PF1 then P

The VT-100 has 2 shift keys: SHIFT and CTRL. The IBM has these keys plus the additional shift key ALT. To perform alternate shift prefix the key with PF3. Type PF3, then type the key. Do not hold both keys at the same time.

ALT - PF3 (as a prefix key)

In certain instances, the normal shift or control key of the VT-100 cannot be used, for example with the function keys, or in the case of control keys preempted by the DEC operating system. Alternate prefix keys are provided:

SHIFT - PF4 (as a prefix key)

CTRL - comma (,) on the auxiliary keypad as a prefix

Special, non-IBM, Key Combinations

Re-draw Display - PF1 then D

Toggle Hidden Line - period (.) key on aux keypad.

ABORT - Caret then C (^C)

Flush Print Buffer - caret-Z (^Z)

CTRL-U - translates to ctrl-X for delete line.

CTRL-C - aborts system utility programs

Comments - To abort a running program try control-C first. If that does not work, then use caret-C to kill MS-DOS and CP/M. PF1-D redraws the display. This is useful when extraneous information gets on the display. This can happen if the operating system, the CP/M control program, or noise injects characters on the screen which the IBM emulation does not know about. Under some DEC operating systems, output to the printer is buffered. The buffer is flushed and the print file (if any) closed when either of the following happens: 1) CP/M warm boots on exit from MS-DOS. 2) A control-z character is sent to the print stream. 3) The user manually types caret-Z on the keyboard. You may have to type caret-Z before taking the paper out of the printer depending on the situation.

4.7 IBM-PC Serial Communications Emulation

The program PORT.COM when executed, loads and remains resident until the next MS-DOS cold start. It emulates the ROM-BIOS interrupt 14h for serial port communications. The physical serial communications link is realized by a serial port on the host DEC computer. Under RT-11 or TSX+ this must be a DLV type serial port. Under RSX or VMS the serial port is any terminal line supported by the operating system. This feature is not currently supported under RSTS.

The functionality is the same as on the IBM-PC except that commands to set the baud rate are ignored. There is no carrier detection support. Input and output is buffered, rather than polled, which may allow faster communication than on an IBM-PC. The serial port must be enabled before it can be used. The detailed procedure is contained in the D100 software installation manual for the applicable DEC operating system. In summary the procedure for each system is as follows:

RT-11	Patch the .SAV file with the address and vector of the DLV serial port card.
TSX+	Install the driver MX.TSX. The serial port may or may not be shared with a timeshare line.
RSX	Allocate the TT port. Give the command MODEM at the double arrow prompt and respond with the line number of the TT in **decimal.**
VMS	Allocate the TT, assign it to a special name and give the command MODEM at the double arrow prompt.

The maximum communications speed varies with the type of computer and the type of serial port. If a problem is encountered it will be losing input characters. RT-11 or TSX+ will easily handle 9600 baud. Under RSX or VMS with DZ or better serial ports 4800 or 9600 will usually be obtained. With DLV serial ports on the least powerful RSX computer, the 11/23, the baud rate on input must be limited to 2400 baud.

4.8 The XPORT Program

This is a program which works only on an IBM-PC or compatible. When executed it loads and remains resident until MS-DOS is reloaded. (Does not work under PC-DOS, but it is simple to install MS-DOS on a PC.) It enhances the ability of the PC to communicate at high baud rates by providing an input buffer to hold input characters from the serial port. When XPORT is used communications programs which were limited to 2400 baud will work at 9600 baud. It enhances the operation of interrupt 14h in the PC.

4.9 Example of the Use of the Move-It (tm) Communications Program.

Move-It is a communications program sold by Wolf Software. It is available in both CP/M and MS-DOS versions. In the example here the MS-DOS version is run on both an IBM-PC and on the D201. A 17,000 byte file is transferred in 51 seconds over a 9600 baud connection. **Move-It** performs full error checking and retransmission of noisy records. For this example, the D201 is installed on a PDP-11/23 running TSX+. The PC is a basic model with dual floppy disk drives and an asyncronous communications card. The PC is connected to the 11/23 by a 30 foot RS-232 cable. The serial port on the DLV-11J card in the 11/23 is connected to the asyncronous port on the back of the IBM-PC.

Move-It can be operated from one end of the link, once started on both systems. There is also a mode of operation which can be entirely single-ended, as when the user calls up the PDP-11 system or VAX from a remote location. For this mode it is necessary to run the CP/M version of Move-It at the DEC computer end. The MS-DOS version and the CP/M version can talk to each other over a serial line.

A>I88 (Start MS-DOS)

A>MS A

Microsoft MS-DOS version 2.11
Copyright 1981,82,83 Microsoft Corp.

Command v. 2.11
Current date is Tue 1-01-1980
Enter new date: 02-28-85

Current time is 0:00:00.00
Enter new time: 11:03:20

A>PORT (Install serial communications support.)

A>MOVEIT (Start the Move_It program.)

* * MOVE-IT (tm) * *
Inter-computer communications utility
Version 3.0
Copyright (c) 1982
Woolf Software Systems

(For the I.B.M. Personal Computer)

*RDIR (This command shows the remote system directory.)

Directory for remote disk B:

MOV96 .COM XPORT .COM

*? (Ask for the help menu.)

The proper commands are:
Send FILNAME.TYP [AS DESTNAME.TYP]
(Send file to remote computer)
Get FILNAME.TYP [AS DESTNAME.TYP]
(Get file from remote computer)
LDir FILNAME.TYP
(Get local directory)
RDir FILNAME.TYP
(Get remote directory)
LUser [USER NUMBER]
(Get/Set local user number)
RUser [USER NUMBER]
(Get/Set remote user number)
Call [NUMBER] [NAME]
(Call number or name)
Hangup
Hang up the phone)
ANswer
(Answer the phone)
NOConsole
(Inhibit console output)
Binary
(Set 8 bit transmission mode)
AScii
(Set 7 bit transmission mode)
Message [MESSAGE TO SEND]
(Send message to remote console)
TRies
(Display retries)
TAlk [FULL] [HALF] [EVEN] [ODD] [NOPARITY]
(Enter smart terminal mode)
Exit

*GET MOV96.COM AS JUNK.COM

Receiving A:JUNK .COM..... Working ....
Complete
(The working message flashes while date is being transferred.)

*EX

A>

5.1 Using the D201 as a CP/M Memory Disk

An MS-DOS program RDISK.COM and a CP/M program RMDSK.COM are used to initialize the D201 for use as a CP/M memory disk. Execute RDISK under MS-DOS:

```
A>RDISK

A>
```

This will initialize the D201 for use as a memory disk and return control to the CP/M-80 compatible environment. Then execute the CP/M program:

```
A>RMDSK
```

This will attach the memory disk as the next unassigned CP/M disk. For example, if you are currently working with disks A:, B: and C:, then the memory disk will be disk D:.

If you cold boot CP/M the memory disk will become detached. It can be re-attached, without loss of data, by executing RMDSK again. The only way to reset the D201 and destroy the data is to use the program I88.COM. The memory disk is automatically set up to use the available memory of the D201. A checksum check is made whenever data is read from the memory disk. A "BDOS error" when reading the memory disk indicates a checksum error. This is an indication of a malfunctioning D201, or, very rarely, a soft error in a dynamic ram chip caused by comsic rays.

APPENDIX A - HARDWARE INSTALLATION

The D201 is always cabled to a D100, Z80 processor card. Figure 1 and figure 3 show the two different ways of connecting the cable. In figure 1 the cable is connected to connector J1 on the D201. In figure 3 connector J2 is used. The cable is twisted differently, otherwise the electrical effect is identical.

Figure 4 partially shows the jumpers for passing DEC system interrupt grants. When a loop of wire is soldered in place the jumper is connected. When the wire is absent or cut, the jumper is unconnected. The jumpers are connected as follows:

D201Q (Q-bus version)

Connected: W1, W2, W7, W8 (others unconnected)

D201U (unibus version)

Connected: W3, W4, W5, W6, W9 (others unconnected)

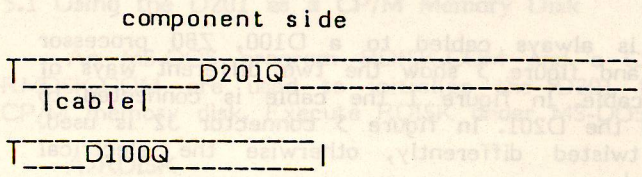
These jumpers are the only difference between the D201Q and the D201U. A third version of the D201 would be one with all jumpers removed. This could be installed in any previously empty slot with the correct pin assignments for +5 volt power and ground.

Board Orientation and Insertation

Although it is difficult to plug the D201 in backwards, care should be taken to understand which side of the board is which. The D201 has unconventional construction in that it consists of a DEC sized quad board with an SWP CO-POWER-88 PLUS board mounted on it. The side with the CO-POWER board is the component side. The side with the access holes which allow access to the components (figure 2) is the solder side. All PC boards have a component side and a solder side. The component side is normally used to mount the components and the solder side is the side passed through the soldering machine. All the component sides point in the same direction in the backplane. A plastic insulating sheet is affixed to the back of the CO-POWER board to prevent any possibility of shorting to an adjacent board. Be careful not to catch the plastic on an adjacent board. If necessary, remove the adjacent board and plug it and the D201 in simultaneously to avoid any relative scraping between the boards.

Normal Q-bus Installation

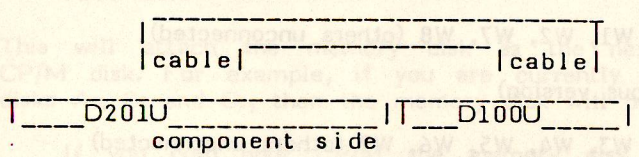
Normally the D201Q will be installed in the slot adjacent to the D100Q and a short (1/2 inch) connector cable will be used. This is shown schematically as follows:



The order in the backplane is not important.

Normal Unibus Installation

Normally the D201U is placed in a quad SPC slot and connected to the D100U which must be in a MUD (Modified unibus device) slot in a DD11-CK or DK backplane unit. This is shown schematically as follows:



The cable is connected to J2 on the D201U and twisted as hown in figure 3. It is not necessary that the D201U and D100U share the same hex slot.

General Installation

The D201 will function provided it is cabled to a functioning D100 and is connected to power and ground. The cable can be connected to J1 with no twist, or to J2 with a 180 degree twist.

Dual Mounting Option

Two D201's may be assembled as a single unit with the connector cut off of one D201. This assembly may be used in slot #1 of a DD11-CK or DK backplane unit, taking advantage of the gap between backplane units which provides extra clearance. The assembly must be connected to 2-D100's by 2 cables.

D201 Memory

Figure 2 shows the rear view of the D201. Access holes allow access to the dynamic memory integrated circuits without disassembling the unit. There ae 4 banks, each with 8 dynamic ram circuits. Bank A is at the bottom, or connector side of the board. Industry standard 256K dynamic rams may be added by plugging them in. An access time of 200 ns is

required. The banks should be filled in order: A, B, C, D. Industry standard 64K rams may be placed in bank C if bank D is empty to provide a 64K display memory only. It is easy to bend the pins on the integrated circuits.

Power Consumption

The D201 requires .6 A of +5 Volt power. This is nearly independent of the amount of memory installed. The D201 creates no bus loading, since it does not connect to the bus.

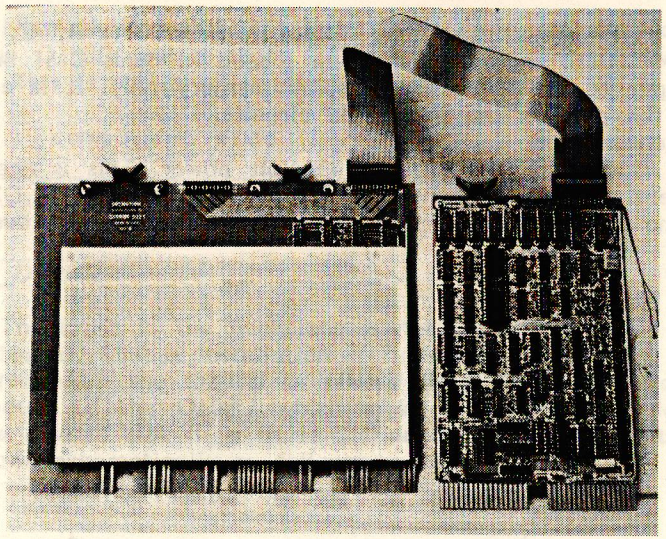


FIGURE #1

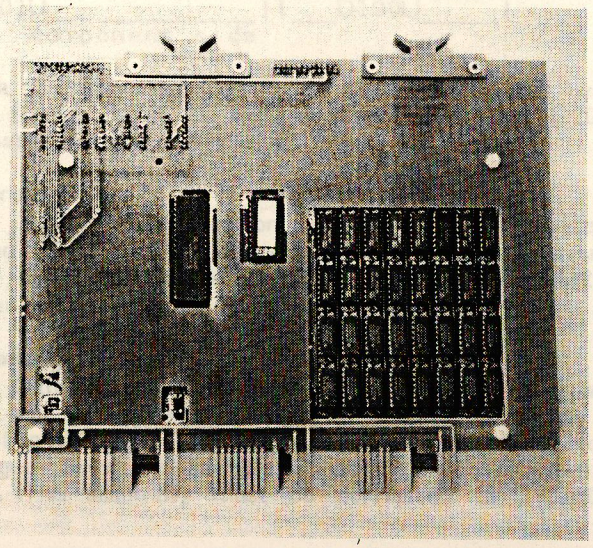


FIGURE #2

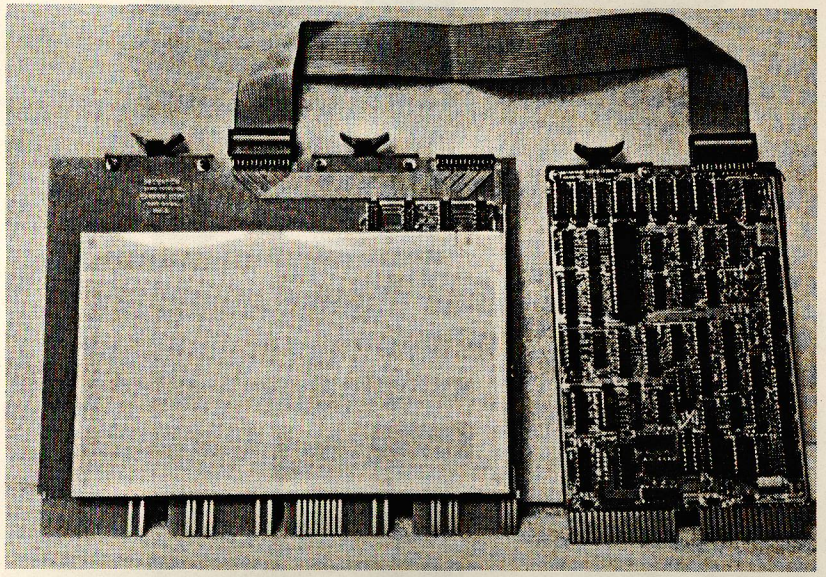


FIGURE #3

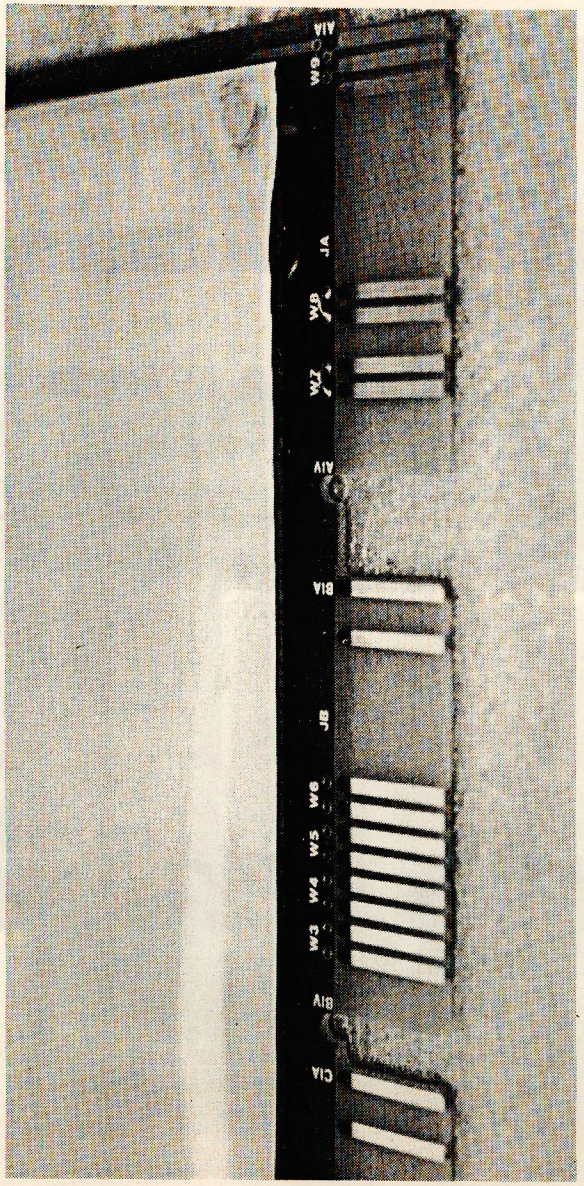


FIGURE #4

A. Baldwin

D100 SOFTWARE INSTALLATION GUIDE

FOR ALL DEC OPERATING SYSTEMS
FOR DECMATION D100 PROCESSOR CARD

1-MARCH-84

Decmation, Inc.
3375 Scott Blvd. #422
Santa Clara, CA 95051

(408) 980-1678

This manual contains non-operating-system-specific information for the software installation of the Decmation D100 processor card.

TABLE OF CONTENTS

INTRODUCTION

This document contains information about the software the installation of the D100Q and D100U, Z80 processor cards for DEC computers. The information here is not specific to any particular DEC operating system. Information which is specific to your particular operating system or hardware is contained in a separate manual which you should use in conjunction with this manual.

The Decmation D100U (unibus) or D100Q (Q-bus) processor card has a Z80 processor, 64K bytes of on-board memory and an interface to the DEC commputer bus. To the host computer, the card appears as 4, 16-bit registers starting at the address for which the card has been set by means of wire-wrap jumpers on the card. All data communications with the card is carried on by a programmed transfer performed by the host computer.

When CP/M is running on the D100 card, the control program is executing in the host processor, at the same time. The control program handles all I/O between CP/M and peripheral devices supported by the host's operating system.

1.0 DISTRIBUTION KIT

The standard distribution media is single-density, 8-inch floppy disks. Since many DEC computer systems cannot read this media, alternate distribution may be made on magnetic tape, disk pack, RX50 floppy or TU58 tape. When alternate distribution is made, the file format on the alternate media may be different than the native file format of the operating system of the host computer. For example, the software may be distributed on an RL02 disk pack in RT-11 format for a system running RSX-11M+. Whenever file formats are mixed in this fashion, there is a way for the host to read the foreign file structure. Under RSX, for example, the program FLX may be used to read RT-11 disks.

When distribution is made on 8-inch floppy disks, all of the disks, except disk #1, are in "standard CP/M format." This is a format which is native to none of the possible host computers, but which can be read by all of them using an RX01 or RX02 floppy drive, a D100 card, and the supplied software.

When an alternate media is used, a file is substituted for each CP/M format disk. For example, the file DISK2.CDK is substituted for disk #2 in the distribution kit. This file can be used in a similar manner to the actual CP/M diskette. The basic software kit contains 4, 8-inch, single-density diskettes:

Disk #1	Decmation control program and associated files. This disk is in the host file format unless marked otherwise. If distribution is made on an alternate media, then each file on this disk will be a

separate file on the alternate media.

The following disks are in standard, 8-inch, CP/M format

Disk #2	Digital Research CP/M operating system
	Digital Research CP/M utilities
	DECMATION CP/M utilities
	Palantir word processor
	Public Domain CP/M utilities
Disk #3	Additional Palantir files
	Additional Public Domain CP/M utilities
ScratchPad disk	Contains the ScratchPad spreadsheet program and associated files.

There may be a considerable number of additional disks in the kit if you ordered additional CP/M software. If your distribution is on an alternate media you will receive both the original floppy disk and a file named in such a manner that you can easily associate the file with the floppy disk. For example, if you purchased the program MultiPlan, you might receive a file named MUL_PLN.CDK, in addition to the manufacturer's distribution diskette. The original diskettes have the software manufacturer's label and are your evidence of license. Possessing the original diskette entitles you to low-cost updates in many cases.

Optional Decmation Diskettes

You may also receive a "VT-100 applications diskette." This contains a number of programs which improve the performance of CP/M programs with the VT-100 terminal. If your distribution is not on diskette you may receive a file VT100.CDK, without the diskette.

Note: You may receive diskettes recorded on both sides of a "flippy" diskette. To read the second side, insert it upsidedown in your disk drive. Flippy diskettes have 2 index holes, instead of 1, and usually are labeled on both sides.

Below is a partial list of the files contained on the distribution media.

CONTENTS OF DISK #1

The following files are supplied by Decmation

1) An executable program for running CP/M. May be called CP.SAV, CPT.SAV, CPX.TSK, CPM.BAS, CPM.EXE, etc.

2) Under most operating system, a program ("PCOPY") for converting a CP/M format diskette to a file.

3) Various other files specific to your particular operating system.

CONTENTS OF DISK #2

The following files are supplied by Digital Research

PIP.COM	File copy utility program
STAT.COM	File status utility
ED.COM	A CP/M line editor
ASM.COM	An 8080 assembler
LOAD.COM	Utility to convert hex files to load files
DUMP.COM	File dump utility
SUBMIT.COM	Program to initiate a CP/M command file
XSUB.COM	Provides an additional facility to SUBMIT
DDT.COM	CP/M Debugger
MOVCPM.COM	CP/M system utility used in special circumstances.

The following are files supplied by DECMATION

GENSYS.COM	Utility program for transferring the CP/M system from one CP/M disk to another; Similar in function and operation to the Digital Research program SYSGEN
REC.COM	CP/M program used to receive a file from the PDP-11 operating system
SEN.COM	CP/M program used to send a file to the PDP-11 operating system.
INITFL.COM	CP/M program used to initialize a single-density floppy for use with the standard CP/M format
BIOS64.ASM	Source code for the "BIOS" used by DECMATION's D100QU
MOVTERM.ASM	Used to adapt the MOVE-IT communications program to any D100 system
MOVBIOS2.ASM	Used to adapt MOVE-IT to a D100QU under operating systems where an auxiliary serial port is supported.
VT100AV.ASM	Palantir terminal driver for a VT-100 with the advanced video option

The following files are supplied by Palantir

WP.COM	The Palantir word processor, pre-configured for the standard VT-100 terminal
WPVT100.COM	Palantir word processor patched to enhance performance on the VT-100 terminal. This version cannot be repatched for a different terminal.
*.WPO	Several overlay files required by WP.COM
HELP.WPH	Help message file used by WP.COM
TYPICAL.ASM	Palantir printer driver for a typical minimum printer
STD10.ASM	Palantir Font definition file for a standard 10 pitch font
PRINTER.SUB	Submit file for configuring Palantir for multiple printers

These Public Domain programs are supplied free & unsupported

XDIR.COM	A substitute for the CP/M DIR command. Produces alphabetized and expanded directory listings of CP/M disks
DIF.COM	The command: DIF FILE1 FILE2 will do a binary compare of the two files and list any differences

encountered

UNLOAD.COM	This program performs the reverse operation of LOAD

CONTENTS OF DISK #3

The following files are supplied by Palantir

TERLIB.HET	A library of Palantir Terminal drivers. See installation notes for instructions.
*.ASM	Other Terminal and Printer drivers and Font definitions
*.HEX	Other Terminal and Printer drivers and Font definitions

These Public Domain programs are supplied free & unsupported

DEL.COM	A substitute for the CP/M ERA command. You will be prompted for each file to be deleted.
TY.COM	A substitute for the CP/M TYPE command. After each screen full of text, you hit any key to continue.
DUMP22.COM	A substitute for the CP/M DUMP command. The file is dumped in both binary and ascii forms.
SWEEP.COM	A substitute for the CP/M PIP command for file transfers. Much more convenient for mass transfers.
SWEEP.DOC	An explanation of how to use the sweep program.

1.1 BACKING UP THE DISTRIBUTION MEDIA

You should make permanent backup copies of the distribution media as soon as possible in the installation procedure. Anything supplied as a file may be easily backed up in whatever manner you normally use. Backing up CP/M format floppy disks may be more difficult until you have CP/M running on your system. It is acceptable to back them up as files, which may be created using the PCOPY program or by copying the CP/M files and system to a pseudo disk file. A device copy program under your host operating system will probably not make a complete copy of a CP/M format disk. DEC doesn't use track 0 of floppy disks, but track 0 contains the CP/M operating system on a standard CP/M format diskette.

1.2 Summary of Features and Installation for Various Operating Systems

The following information is included for users who may be contemplating installing a D100 card on a different operating system or who do not have the individual, operating system specific software manual. Consult the manual specific to the particular DEC operating system for detailed information. This information is subject to change as new features are added to the software.

RT-11 Operating System

The RT-11 control program, CP.SAV, will run under versions 3, 4 and 5 of RT-11. A version for RT-11 version 2 is available on special request. Normally it is run under the single job monitor. A special version for the FB monitor is available, but it runs only in the background. At least 48K-bytes of memory are required. Any type of 11 processor

which runs RT-11 will serve. Provision is made for non-standard floppy disk drives which will not support special function calls. 5-1/4 -inch, RX50 diskette drives may be used to read and write the CP/M format diskettes of DEC's Rainbow computer system. A DL type serial interface on the DEC computer may be accessed from CP/M as an auxiliary serial port.

TSX+ Operating System

Users ordering TSX+ automatically receive support for RT-11 also. Multiple D100 cards may be installed on a TSX+ system. The number of permitted simultaneous users is equal to the number of CP/M cards installed. Real time program support is a required feature needed in the TSX+ system. It is simple to add this feature to TSX+ if not installed. The record locking feature should also be installed in TSX+. The control program, CPT.SAV, requires 40K-bytes for each user running CP/M. It will run on TSX+ versions 3 and 4 and in most cases with version 2. An auxiliary serial port, accessable to CP/M programs, may be implemented with a special driver which is supplied. This feature can utilize a DL type serial port installed on the DEC system. Rainbow format CP/M disks are supported for users who have RX50 disk drives.

RSX-11M, RSX-11M+, Micro RSX

No sysgen is required to install the D100 card under RSX. A few privileged commands must be added to the system startup file. Multiple cards may be installed on the same system. Pseudo disk files may be shared among users on a read-only basis. An auxiliary serial port, accessable to CP/M programs, is implemented using the RSX terminal driver. Any serial port supported by the terminal

driver may be assigned to be an auxiliary serial port. Support for Rainbow CP/M disks under Micro RSX is under development.

RSTS/E Operating System

The RSTS/E control program is implemented in BASIC+. Full source is supplied. A special driver must be added to the system by performing a shortened sysgen. Multiple CP/M cards may be installed. RSTS/E version 7.0 or later is required.

VMS Operating System

Multiple cards may be installed on a VAX. Software installation is performed by an automated process by a privileged user. CP/M disks may be shared if they are set read only by the control program. The console floppy on an 11/780 may be used to read and write CP/M floppy disks. An RX01 or RX02 may be used for this purpose on any VAX.

DECMATION SERIAL COMMUNICATIONS PROGRAMS OPERATING INSTRUCTIONS

*Decmation is not connected with or approved by the Digital Equipment Corporation who is the owner of the trademark DEC and the trademarks: RSTS, RT-11, RSX, VMS, VAX. CP/M is a registered trademark of Digital Research.

Decmation, Inc.
3375 Scott Blvd. Suite 236
Santa Clara, CA 95054

(408) 980-1678

Contents

1.0 Introduction

This manual covers Decmation serial communications programs for CP/M* MS-DOS* (IBM-PC) and Intel development systems, including the MDS-800, series II, series III and the iPDS. The different versions of the communications programs all have nearly identical operation and can all talk to each other. The "modem-7" or "xmodem" protocol is used. This is a communications protocol used in many public domain communications programs. Records are sent in 128-byte blocks with a cyclic redundancy check. Bad records are sent again. Decmation has expanded the protocol with a "mainframe" protocol. The mainframe protocol translates the byte stream before it is sent to eliminate all characters disagreeable to mainframe computers. The receiving program reverses the translation. This allows communications to go through a terminal port on all DEC systems, without sending 8-bit characters, or control characters, which might upset the terminal driver of the DEC operating system. Use of the mainframe protocol is optional.

2.0 Basic Operating Modes

The two computers which will communicate must be connected by a serial line. This may be a simple 3-wire physical connection, or it may be a duplex connection via any sort of telephone or telecommunications network. It is important to understand the difference between a terminal serial port and an auxiliary serial port. A terminal serial port is the serial line to which a standard computer terminal is normally attached. An auxiliary serial port is a separate serial port which a program using the correct drivers and protocol can talk through.

Small personal computers may not have any such thing as a terminal serial port because the terminal is an integral part of the computer, not a separate unit connected to the rest of the computer by a serial communications line. The IBM-PC is an example of such a computer. However the IBM-PC may be equipped with an auxiliary serial port. It is also possible to use a separate terminal with an IBM-PC.

On some multiuser DEC computers the serial line used as an auxiliary serial port is exactly the same as some other users terminal port. By the proper operating system commands it can be temporarily taken over to be used as an auxiliary serial port under CP/M or MS-DOS running on a Decmation co-processor card.

If each of the communicating computers has both a terminal port and an auxiliary port, then there are 4 possible ways to connect them:

System #1	System #2	
AUX	AUX	(1)
AUX	TERM	(2)
TERM	AUX	(3)
TERM	TERM	(4)

Method (4) makes little sense and can only be made to work by dynamic switching of connector cables. All the other methods can be used for communications. Binary files can be transferred with full error checking and re-transmission of bad data. When using method (1) the user must type commands on the terminals of both computers, or two persons must cooperate. This method is easiest to understand and often results in the fastest transmission of files.

When using methods (2) or (3) all commands are entered from one end of the communications link. The user will pass commands to the remote system by using his local system as a terminal on the remote system. His local system can emulate a terminal on the remote system by using the communications program as a terminal emulator. When used in this mode, each character typed is sent to the remote system, each character received from the remote system is output to the screen of the local terminal.

The sequence of operations for control from one end of the link is the following:

1) The user enters terminal emulation mode. Now his terminal is a terminal on the remote system.

2) The types a command telling the remote system to either send or receive a file via the terminal port. Nothing happens immediately because the communication doesn't start until certain special characters are sent. The program will wait up to 100 seconds for these characters before it "times out."

3) The user escapes terminal emulation mode (by typing control-E for example). Now his terminal is

a terminal on his local system. He types a command telling his local communications program to receive or send a file on the auxiliary port. Within about 10 seconds the two programs will synchronize and the file will be transmitted. A running log on the user's screen counts the 128-byte sectors transmitted.

2.1 7-Bit versus 8-Bit Communications

An asynchronous communications line (the only type we normally deal with) may be able to send 8-bit bytes or only 7-bit bytes. For the most direct communication of binary files 8-bits is desirable. The limitation to 7-bit bytes is usually a consequence of system software. Usually the full 128 byte character set associated with a 7-bit byte cannot be used. Terminal lines on computers may have certain characters reserved by the system software or special purposes. These characters cannot be freely transmitted though the terminal port. Decmation communications programs can operate under these restrictions and still transmit binary files. This is done by the "mainframe" communications protocol option. Under the mainframe communications protocol only the ASCII characters space through tilde (32-126) are ever sent over the communications line. The caret character is also prohibited and never sent over the line. All other characters which must be sent are translated into a sequence of 2 allowed characters and then re-translated back to the original character at the receiving station. This slows communication slightly for English text. For typical binary files the transmission time is increased by a factor of about 1.6. The mainframe option is always invoked when communicating through a terminal port on a DEC system. When both systems are using auxiliary ports it is optional. It can't be used when communicating with other communications programs which support the xmodem communications protocol. To communicate with these programs the mainframe protocol should not be invoked.

2.2 The Modem-7 Protocol

As used by the programs described here data is sent in records formatted as follows. The first byte is an SOH (1). Byte 2 is the record number 0-255. Byte 3 is byte 2 complemented. The record number wraps from 255 back to 0 and any length files may be sent. These first 3 bytes are followed by 128 bytes of data. Following the data is a 2-byte CRC (cyclic redundancy check). When a

record is received it is acknowledged with an ACK (6) if it is ok (good CRC) or a NAK (15H) if it is bad. In the mainframe mode the entire record is subject to a conversion on a byte by byte basis. All forbidden characters are sent as two characters consisting of a special quote character followed by the forbidden character transformed into a non forbidden character. The forbidden characters are control characters, 8 bit characters, the caret character, the delete character and the quote characters. Regular text has few characters which must be expanded into two characters, but binary files are typically increased in length by about a factor of 1.6. The EOF character (ctrl-D or 4) is sent for end of file. The program may be aborted by sending a ctrl-D or a ctrl-X.

3.0 Distribution Kits

Some of the files and programs associated with the various communications programs are listed below:
shown below:

3.1 CP/M Communications

MDM3.COM	Public domain general purpose communications program, set up to run on the D100.
MDM.DOC	Documentation for MDM3
COM.COM	Decmation communications program to run on the D100.
DRVCPM.ASM	General purpose driver adaptable for various CP/M computers for COM.
DRVDEC.ASM	Driver for COM for use on D100.
COMCF.ASM	Source code for COM. Supplied when specially ordered.
CCOM.DOC	Additional technical documentation.
TALK.COM	Program which may be used to automate communications.
MAKSUB.COM	Program which generates command files substituting file names expanded from a wild card specification.

3.2 ISIS Communications

COMISI.OBJ	Object modules for the ISIS communications program. Must be linked with a driver.
DRV800.MAC	Driver source MDS800

DRV230.MAC	Driver source MDS230 and similar models.
DRVPDS.MAC	Driver source iPDS system.
DRVISI.MAC	Driver for Decmation serial port under the isis emulator. Supplied for test and debug. Use COM.COM for production work.
COMIFU.MAC	Abbreviated source code for ISIS communications program. (No comments.)
COMIF.MAC	Full source code for the ISIS communications program. Supplied only when specially ordered.
COMPDS.HEX	A complete communications program for the Intel iPDS development system. Set up to operate at 9600 baud. Convert to executable file with HEXOBJ utility before using.
ICOM.DOC	Additional technical documentation.

3.3 IBM-PC Communications

COMIBM.EXE	Runable copy of the communications program.
XPORT.COM	Load and stay resident serial port driver for IBM-PC communications port. Buffered and interrupt driven.
PORT.COM	Load and stay resident serial port driver for MS-DOS on Decmation D201 only.
COMIBM.DEF	Sample configuration program which holds definition of communications parameters for COMIBM.EXE.
DNLD.COM	A utility program to aid in downloading COMIBM.EXE to a PC.

Distribution media may be 8-inch standard CP/M diskettes, 8-inch, single density ISIS diskettes, 5-1/4 inch single-density, single sided, 8-sector IBM-PC diskettes, or any of a variety of other DEC compatible formats. Generally it is possible to download a communications program to a remote system which currently has no communications program. This is difficult to do over the telephone, since an error free line is needed until the communications program can be brought up on the remote

system. For systems with difficult-to-supply media, such as iPDS, PC-AT and ISIS double density, the easiest approach is to download the program.

4.0 Operating Instructions

The following instructions apply to any of the supported computers, unless otherwise noted, except that the name of the communications program may differ. For example, COM800 instead of COM.

Both computers use an auxiliary port:

Enter the command at terminal #1 to send a file:

```
A>COM S FILE.EXT
```

You will now have about 100 seconds to enter the command on the terminal of system #2 to receive the file.

```
A>COM R FILE.EXT
```

"S" is the command to send. "R" is the command to receive. You do not have to use the same file name on both systems. The above example is for 2 CP/M systems. "COM" is the name of the CP/M version of the communications program. The suggested names for the versions of COM are as follows:

COM	CP/M version for Decmation D100 processor installed on a DEC computer.
COMIBM	IBM-PC, Decmation D201
COM800	Intel MDS-800
COM230	Intel MDS-230
COMPDS	Intel Personal Development System

Example, computer #2 uses the terminal port:

At the terminal of computer #1 start com in terminal mode:

A>COM T

(Now your terminal is a terminal on computer #2. You may have to log on and start CP/M if it is a DEC system.)

A>COM SQ MYFILE.EXT

(Send a file in "quiet" mode over the terminal line.)

(Now type control-E to escape terminal mode.)

COMMAND: RM NEWFILE.EXT

(You must specify the mainframe "M" mode because the quiet mode always enables mainframe mode.)

Remember that the above examples are for CP/M. On different systems the commands will be modified for the conventions of the system. For example, on an ISIS-II system the above command might be: RM :F1:MYFILE.ABC. In this case the ISIS-II disk name (:F1:) is used.

COM Commands - General Form

The general form of the command is 1-3 option letters, a space and the file name. The command may be entered on the command line when COM is run, or in response to the message "COMMAND:" when escaping from terminal mode.

The first option letter must be one of the following:

T -enter terminal mode.
I -enter special iPDS terminal mode (COM.COM and COMIBM.EXE programs only)
S -send a file.
R -receive a file.

The second option letter may be absent or any of the following:

M -use mainframe communications protocol.

Q -Use the quiet mode. The terminal port is used for communications. Messages which would interfere with communications are suppressed. Mainframe protocol is also enabled by this option. Use on COM.COM version only.

X -Use the aux port for communications and messages. Suppress messages which would interfere with communications. (ISIS and PC versions only).

0-9 -A digit from 0 to 9 may be entered to slow the transmission rate. This puts a time delay between each transmitted character. When receiving, the program transmits acknowledgement messages. The delay is also inserted in these messages. The delay is processor clock dependent and varies between about 1 millisecond for 0 and 50 milliseconds for 9.

A digit to indicate delay may be inserted as a third option letter if the second option letter is used for "M" or "Q".

The following two control characters may be entered while in terminal emulation mode:

Ctrl-E Control E, causes an escape from terminal mode, at which point a command line may be entered.

CTRL-T (CP/M and IBM versions only) When Ctrl-T is typed in terminal mode, it will allow specification of an ASCII file to be downloaded using the wait for echo protocol. This is used to bootstrap a communications program to a remote system.

In response to ctrl-T (not availabe on ISIS versions) the following download commands may be entered:

L FILE.EXT Send all ASCII characters in the file, waiting for echo.

N FILE.EXT Send all except linefeeds. When linefeed is encountered flush the incomming echos.

NT FILE.EXT Same as "N" except change all tabs to spaces while sending.

The following commands may be entered at the prompt COMMAND:

E — The letter E entered as a command line causes an exit back to the host operating system.

? — Print a help message.

5.0 Bootstrapping a Remote System

Frequently the situation is encountered where it is desired to install a communications program on a remote system which has incompatible media. (That is why the communications program is desired in the first place.) This is particularly true of the Intel iPDS system which has incompatible diskettes.

Decmation communications programs are usually supplied, or may be easily converted to, source or Intel "hex" format. Source code or hex formats are pure ASCII text, without binary data or control characters except for carriage return, line feed and tab. Pure ASCII text is easy to download to a remote system by improvised means. For systems which cannot accept the tab character, an option is available to convert all tabs to spaces before sending, or this can be done with an editor, such as the **Palantir** word processor. This does not affect the integrity of assembler source code.

It is necessary to have an excellent communications link to the remote system because downloading is a very simplified form of communications which does not include correction of transmission errors. Hex code includes checksums, which are checked when the hex code is converted to object or binary form. It may be possible to correct transmission errors manually with a text editor on the remote system. Modems connections are subject to transmission errors. Local, direct wire, connections are usually error-free.

General Methods for Downloading

If the remote system already has a communications program capable of capturing data input to a serial port, then it is only necessary to spit out the file to the serial port. This can be done in many cases by disconnecting the serial data cable to the printer and connecting it to the

serial port of the remote system. The remote system is set up to capture data, then the file is "printed." The main problem is sending data too fast. This can be controlled by slowing down the baud rate. Rather than printing the file on the local system, a communications program which has the capability of sending a file via a serial port can be used.

Wait for Echo Method of Downloading

One of the best ways to download is for the local system to send the file byte by byte to the terminal port of the remote system, waiting for the byte to be echoed before the next is sent. This has the advantage of controlling the speed of transmission and checking the validity by comparing the echo against the character sent. It requires that the remote system have a separate terminal connected by a serial line. Personal computers often have no separate terminal and thus no terminal serial port. Intel development systems allow for the use of a separate terminal, even when they have a built-in terminal.

Using the Program COM or COMIBM to Download By the Wait for Echo Method

The Decmation communications programs COM.COM and COMIBM.EXE have built-in provisions for downloading by the wait for echo method. The following example assumes the remote system is a CP/M system:

A>COM T (Enter terminal mode, becoming a terminal on the remote system.)

A>
A>PIP FILE.EXT=CON: (Start copy from terminal to file.)

A>**ctrl-T** (Enter download mode)

COMMAND: L FILE.EXT

(File will be sent as if typed on remote system's terminal. It will also echo on the local terminal.)

ctrl-Z (Type a ctrl-z to tell PIP on the remote system that the file is ended.)
A> (The download is complete. You can continue as a terminal on the remote system if desired.)

COM has two modes of download. In the "L" modde line feeds are sent when in the text. The remote station is expected to exactly echo the file sent. In the "N" mode, line feeds are not sent. When a linefeed is encountered in the file the program "gobbles" up all characters sent by the remote station until it has no more characters to send for 2 tenths of a second. This permits the remote station to echo cr-lf in response to cr. The remote station may also send a prompt at the start of each line, such as is done by some editors. The NT option is the same as the N option, except that tabs are all changed to spaces. COM will print an error message and beep if the echo does not match the character sent. The "N" mode also accepts a tilde as an echo for a tab to match the Intel iPDS system.

Converting Hex Files to Program Files

Under CP/M the utility program LOAD.COM or PATCH.COM can be used to convert a HEX file into a runable program (COM file). For example:

```
A>LOAD FILE      (Generates FILE.COM from
                  FILE.HEX)

A>ERA A:FILE.COM
A>PATCH FILE=FILE    (Same function as LOAD.)
```

Another utility, UNLOAD, is supplied by Decmation to perform the reverse function:

```
A>UNLOAD FILE  (Generates FILE.HEX from
                FILE.COM.)
```

On Intel development systems the programs HEXOBJ and OBJHEX perform a similar function. Intel executable files may be converted back and forth to hex.

Examples:

```
-HEXOBJ COMPDS.HEX TO COMPDS    (executable)
-OBJHEX COMPDS TO COMPDX.HEX
```

On the IBM-PC the standard program DEBUG can be used to convert hex files to executable files. For example to convert COMIBM.HEX to COMIBM.EXE:


```
A>DEBUG COMIBM.HEX
-N COMIBM.COM          (Specify out file name.)
-W                     (Write the out file.)
-Q
A>RENAME COMIBM.COM COMIBM.EXE
```

The last step is performed only when the original file was an "EXE" type of executable file.

Notes For Other CP/M Systems

The CP/M communications program COM can be adapted to work on any CP/M system by installing an appropriate driver. COM comes set up to work under Decmation's CP/M on DEC systems. The driver for this configuration is DRVDEC.ASM. A prototype driver, DRVCPM.ASM is included which may be adapted to work on most other CP/M systems. This is a job which requires programming experience. Users who are not programmers can purchase a commercial communications program which supports the XMODEM protocol, or they can use the MOVE-IT communications program which is supported for nearly all CP/M computers and which can run under Decmation's CP/M environment.

6.0 Notes For Intel Development Systems

All Intel development systems use the same basic communications program which is linked with a different serial port driver for each system. The drivers are supplied in source form and must be edited to set the baud rate and the processor clock speed. To generate a runable copy of the communications program the following steps are required. These steps may be performed on any ISIS system or on the D100 using the ISIS emulator.

```
-ASM80 COMIFU.MAC    (Assemble communication program.)
-ASM80 DRV800.MAC    (Assemble driver.)
-LINK DRV800.OBJ,COMIFU.OBJ TO COM800.OBJ
-LOCATE COM800.OBJ TO COM800
```

Of course, the names will depend upon the versions used. The example is for an MDS-800.

On the MDS-800 the baud rate can be controlled by software only in a limited way. It can be adjusted to either of 2 values by editing the driver, using the X64 or X16 option for the UART. Other adjustments must be made by means of jumpers. On the Series II, Series III and the

iPDS, a programmable baud rate generator allows complete adjustment of baud rate by editing the driver. The driver is very simple and non-interrupt driven.

If the Intel system is to be used in conjunction with a VAX or PDP-11 various working configurations can be used:

1) A separate terminal is used for each system. This is convenient only if the two terminals are located close together. The user must type on both terminals to transfer files. In this case an auxiliary serial port from the DEC system is connected to an auxiliary serial port on the Intel system. The DEC terminal is connected to another user serial line on the DEC system. The Intel system terminal is connected directly to the DEC system. This requires 2 terminal lines to be run from the DEC system.

2) The auxiliary serial port on the Intel system is connected to a logon line or user line from the DEC system. The user uses the terminal on the Intel system as a terminal on both the DEC and Intel systems. Unless the terminal on the Intel system is a VT-100 compatible unit the user will not be able to use the DEC text editors or CP/M text editors on the DEC system without some special modifications. The base iPDS has only one serial port, so without special work the user must use the built-in terminal of the iPDS as a DEC system terminal. This terminal is not VT-100 compatible. Of course the user could have a separate VT-100 terminal and a manual switch to switch the VT-100 to the line from the DEC system. There is also the danger that the DEC system will send data to the built-in terminal faster than it can accept it. This results in lost characters. The solution is to slow the baud rate or set the DEC system to insert nulls after carriage return. The iPDS built-in terminal is fast enough to run at 9600 baud.

3) Both the Intel terminal function and communications function are assigned to the same serial port which is connected to an auxiliary serial port on the DEC system. This allows any user on the DEC system to use his terminal as a terminal on the Intel system and to exchange files with the Intel system. The main disadvantage is that the user may be remote from the Intel

system so he cannot push the reset button.

6.1 Notes for the iPDS System

ASCII text (such as hex files) may be downloaded to the iPDS system as illustrated in the following example. This downloads the file COMPDS.HEX which is a complete 9600 baud communications program and then the hex file is converted to an executable file, thereby installing the communications program.

Starting at the iPDS keyboard:

```
A0>SERIAL A S=1 B=9600      (Set the serial port to
                             asynchronous, 1 stop
                             bit, 9600 baud)

A0>ASSIGN :CO: TO :SO:      (Assign console function
                             to serial port.)
```

(At this point the terminal goes dead, but the keyboard is still alive. Type the next command blind. You can also use a submit file.)

```
A0>ASSIGN :CI: to :SI:
```

Then, with the DEC or IBM-PC, auxiliary serial port connected to the iPDS serial port use the following commands with the communications program from a terminal on the DEC system.

```
A>COM I          (Start in terminal emulation mode, "I"
                  for special iPDS mode.)
```

(The DEC terminal or PC is now the terminal on the iPDS.)

```
A0>COPY :CI: TO COMPDS.HEX  (Copy from the
                             serial port to a
                             file.)

ctrl-T           (Enable the download mode.)

COMMAND: NT COMPDS.HEX
```

(The file will be sent waiting for echo on each character. Any wrong echoes are flagged. If you have a clean line and an ASCII file there should be no wrong echoes.)

```
ctrl-Z          (At end of transmission, type ctrl-z to
                signify eof to iPDS.)
ctrl-E          (Exit
A>COMMAND: T    (Return to terminal mode)
A0>HEXOBJ COMPDS.HEX TO COMPDS
A0>
```

If your DEC system is not capable of 9600 baud communication (certain RSX systems) then you must download the file COMISI.HEX and DRVPDS.MAC and link them to make a new communications program working at a different baud rate. Edit DRVPDS.MAC to change the baud rate. Just because you can download at 9600 baud does not indicate that you can perform 2-way communication at 9600 baud. The critical parameter for the DEC systems is the input transmission speed. This is always half the baud rate or slower on a download because the next character is not sent until the echo from the previous character is received.

6.2 Examples of Use With the iPDS System

Working From the Remote System Terminal

An auxiliary serial port from the DEC or PC system is connected to the iPDS serial port connector. The following file is created using the JOB command or with an editor:

File ABOOT.CSD

```
SERIAL A S=1 B=9600
ASSIGN :CO: TO :SO:
ASSIGN :CI: TO :SI:
ENDJOB
```

If this file is present, whenever you press the reset button to boot the iPDS it will automatically switch the terminal function to the serial port.

With the communications program from the remote system terminal you can exchange files and operate the iPDS:

```
A>COM I         (Start the terminal emulation mode in
                the special Intel mode. The special Intel
                mode allows the delete and backspace
                keys to work properly.)

(At this point your terminal is a terminal on the iPDS
 system. CREDIT will not work because it communi-
```



```
cates directly with the terminal and the intel
terminal is not VT-100 or ANSI compatible.)
```

To send a file to the iPDS use the following commands:

```
A0>COMPDS RX FILE.EXT
(bblah, blah)
ctrl-E
COMMAND: S FILE.EXT
(blah, blah)
COMMAND: T      (or EX to return to CP/M
                environment)
A0>
```

To receive a file from the iPDS use the following commands:

```
A0>COMPDS SX FILE.EXT
(blah, blah)
ctrl-E
COMMAND: R FILE.EXT
(blah, blah)
COMMAND: T
A0>
```

Note that you can use different file names on each end. The communications program will not overwrite an existing file. Delete the pre-existing file first.

When using a TSX+ system (11/23) as the DEC host communications throughput in this mode was about 780 bytes per second with a 9600 baud rate.

The input buffer on the DEC line used should be at least 132 characters long on systems where this can be adjusted. At 1200 baud an under-sized input buffer will probably still work because it will be emptied faster than if fills under most system loading. The iPDS obeys Xon-Xoff protocol when the terminal is remoted to the serial port. Decmation's serial driver sends an "emergency" xoff when its local buffer (not the system's buffer) is nearly full. This will keep the iPDS from overrunning the DEC system input when typing a long file. On VMS the auxiliary serial port used must be set up for 8-bit and NOTTSYNC modes of operation.

Working From the iPDS Terminal

In this case you will use the iPDS as the terminal on the DEC system. To start up:

A0>COMPDS T

(Now the iPDS is a DEC system terminal. Log on to the DEC system if necessary and start up the CP/M compatible system.)

A> (the CP/M prompt)

To send a file to the DEC system:

A>COM RQ FILE.EXT
(blah,blah)
ctrl-E
COMMAND: SM FILE.EXT
(blah, blah)
COMMAND: T (or EX to return to iPDS mode)

To get a file from the DEC system:

A>COM SQ FILE.EXT
(blah, blah)
ctrl-E
COMMAND: RM FILE.EXT
(blah, blah)
COMMAND: T
A>

In this mode of operation the DEC system can be remote, even at the end of a telephone call. File transfer will be slower, especially for binary files. The iPDS does not have a terminal Xoff-Xon function, but the iPDS seems to be fast enough to accept data to the display at 9600 baud without losing characters. At higher speeds the DEC terminal input buffers should be at least 270 characters long to hold the incoming message packets. At 1200 baud it may be possible to get away with an insufficient sized input buffer.

You can also talk to a DEC system which runs XMODEM directly. (XMODEM is a public domain communications program available from DECUS.) In this case use the commands: **COMPDS S FILE.EXT** or **COMPDS R FILE.EXT.** Under VMS or RSX the type of files created by the XMODEM utility should be known. Only ASCII files and binary files of 128-byte record length are easily moved from the VMS or RSX file structure to the CP/M or MS-DOS file structure. The record size of binary files can be changed by translating them to a foreign device and back (e.g. RSX FLX) or by using a program CVT512 which changes between 128 and 512 byte record

sizes. Neither VMS or RSX has an easy way to change record sizes. TSX+ and RT-11 have no record size structure as a part of their file structure, so the problem does not exist.

Working From Both Terminals

This mode of operation is useful if a terminal on the DEC system and the iPDS are in the same location, or if 2 people work together. Note that 2 serial lines must run from the DEC system. One for the terminal and one for the communications line. If modems are used then the person at the DEC terminal will call out to the iPDS. A modem will be connected to the serial port on the iPDS. To transfer a file the commands used are:

COMPDS R[S] FILE.EXT

COM R[S] FILE.EXT

They should be typed at the same time on both systems with a 100 second tolerance. If both systems are in terminal mode, then messages can be typed back and forth.

7.0 Notes for the IBM-PC

The communications program COMIBM.EXE runs on an IBM-PC or any MS-DOS computer which follows the PC conventions regarding serial port communication via 8088 interrupt number 14h. An asynchronous communications interface must be installed on the PC.

COMIBM.EXE will run on Decmation's D201 processor using a virtual serial port on the host.

Default communications parameters (including baud rate) are established by a file named COMIBM.DEF which should be on the default disk. Whenever COMIBM is invoked it tries to read this file from the default disk. The file has a format as follows:

```
BAUD=9600
COMPORT=1
TIMING=10
DELAY=0
CRC=1
BUFFER=0
```

If COMIBM.DEF cannot be found, then the parameters shown above are used. The allowable values for the parameters are as follows:

BAUD= [9600] [4800] [2400] [1200] [600] [300] [150] [110]. Ignored on D201 where Baud rate is determined by DEC computer software or hardware.

COMPORT=[1] [2] Which serial port to use on PC. Ignored on D201.

TIMING=[nn] A number between 1 and 99. Speed relative to IBM-PC which has the value of 10. Calibrates timing loops. If your computer is 3 times as fast as a PC, then use 30.

DELAY=[n] A digit between 0 and 9 to indicate a time delay loop to be executed before sending any character to remote. Use for full speed. An alternative to lowering baud rate. Data rate slows 50% for each digit step.

CRC= [0] [1] 0 for checksum, 1 for crc checking of records. Use 1 unless talking to system which requires checksum.

BUFFER= [0] [1] 0 for no receiver buffer, 1 for buffering. If the system has a buffer, filled by interrupts and long enough to hold the incomming record, computations can be performed while receiving the next record, resulting in faster operation. Always set to 1 for the D201. With a PC set to 1 only if the XPORT.COM driver is installed.

The file COMIBM.DEF may be prepared on a PC by using the copy command, as for example:

```
A>COPY CON: COMIBM.DEF
BAUD=4800
ctrl-Z                    (Type a control-Z, then CR.)
A>
```

Parameters not changed from the default need not be included.

COMMAND STRUCTURE

COMIBM.EXE follows the same command structure as other Decmation communications programs which run under

CP/M or ISIS. The program is started with the initial command in the same line:

```
A>COMIBM S FILE.EXT      (Send a file with
                          XMODEM protocol.)

A>COMIBM R FILE.EXT (Receive a file with XMODEM
                     protocol.)

A>COMIBM T               (Enter terminal emulator
                          mode, making the PC a dumb
                          terminal on the remote
                          system.)
A>COMIBM I               (Same as T but with special
                          features for the intel iPDS
                          system.)
```

If the program is started in T or I mode, the control-E will cause an escape from terminal emulation and allow a command to be entered to send or receive a file:

```
ctrl-E
COMMAND: S FILE.EXT (Escape terminal emulation
                     mode and send a file using
                     XMODEM protocol.)
```

After the file transfer takes place a return to terminal emulation mode can be made by the command T or an exit by the command Exit. Intel terminal emulation mode remains in effect if the program was started by COMIBM I.

Either S or R may be followed by an M for Decmation mainframe mode, or X for remote 8-bit communications. Optionally a digit from 1-10 may also follow either S, R, M or X to indicate a delay loop to limit sending speed. (On receive the acknowledge characters are sent delayed.) A typical communications sequence with a remote system might be as follows:

```
A>COMIBM T
(Log on to remote system as a terminal.)
A>COM RQ MYFILE.EXT (Start remote communications
                     program in quiet mode.)
(blah,blah message from remote)
ctrl-E
COMMAND: SM MYFILE.EXT (Send to remote system)
(sending log)
COMMAND: T
A> (now log off remote system)
ctrl-E
COMMAND: E (all done)
```

XPORT.COM Driver

XPORT is a PC program which when executed becomes permanently (until re-boot) resident in memory. It intercepts all interrupt 14h's and provides a replacement for the standard IBM serial port driver. It has the advantage that serial input is collected in a 500 byte buffer by an interrupt driven routine. This avoids the loss of input characters during rapid input. To install XPORT.COM use a command as shown in the following examples:

```
A>XPORT       (Install for com port #1)

A>XPORT 2     (Install for com port #2)

A>XPORT 1 6 (Install for port #1, 19200 baud)
```

The first optional argument is either the digit 1 or 2 for com port 1 or 2. The second optional argument is a substitute baud rate divisor. If not present the driver responds to the standard calls to set the baud rate. If present, and less then C, then the driver ignores the standard calls and uses the digit to determine the baud rate. (In which case the BAUD= statement in COMIBM.DEF is ignored.) The digit 6 is for 19,200, 3 for 38,400. Other hexidecimal digits will give intermediate values. The port will not necessarily function correctly at the higher rates, especially if a long cable is used. If XPORT is installed for com port #1, then calls for com port #2 will be sent to the pre-existing serial port driver. It may be installed for both ports by installing it twice:

```
A>XPORT 1
A>XPORT 2
```

The last port installed will operate the fastest. Each time XPORT is installed it uses up about 1500 bytes of memory. If a call is made to the driver for a different serial port than the one for which t is installed, the call is relayed to the previous driver installed. If you keep installing it, besides using up memory, the relay to the other port driver will become slower and slower.

The IBM-PC supports 2 asynchronous communications cards. Normally, the first unit has an I/O bus address of 3F8h and and uses interrupt vector 12. The second has an address of 2F8h and and uses vector 11. If, for some reason, you have only one card, but have the I/O address set to 2F8, then XPORT will be confused because it will try to use the address 3F8h, while the IBM will use this as com port #1. On an IBM-PC (but perhaps not on other

models) the addresses of the COM ports can be viewed by dumping address 40:0 with DEBUG. With one com port, debug will show: F8 03 00 00. If you see: F8 02 00 00, then the jumper on the card is set incorrectly. The jumper is an 8-pin dual in-line package in a socket near the top rear of the card. It is rotated 180 degrees to change the address. When the small dimple is toward the top the address if 3F8h. With 2 com ports DEBUG will show: F8 03 F8 02 at address 40:0.

XPORT.COM sends an XOFF when its input buffer is nearly full. It sends XON when the buffer later becomes less full. This never happens during normal file transfer because the threshold is larger than the record size used. During terminal emulation the XOFF will be sent when the remote system sends characters faster than the IBM display can write.

XPORT.COM always sets parity disabled, stop bits to 1 and 8 data bits. No other settings are possible when XPORT is installed. The baud rate can be changed by a standard call provided that it has not been set to a value larger than 9600 at installation, which locks the baud rate.

Communication is usually much faster if the BUFFERS=1 statement appears in the COMIBM.DEF file. XPORT makes it possible to use this mode of operation.

Using A PC as a Remote System

It is possible to set up the PC so that both its terminal and communications functions are done over the serial port. This depends on using the MS-DOS command:

```
A>CTTY COM1
```

or:

```
A>CTTY COM2
```

Once you give this command the PC terminal and keyboard go dead and it expects to perform its terminal function via the serial port. The serial port must be enabled and set to the correct baud rate. If XPORT is not installed it may be necessary to send one character on the serial line to get it going. (See the section on downloading below.)

```
[A>XPORT 1]
A>MODE COM1:9600,n,8,1
A>CTTY COM1
```

Substitute COM2 if com port #2 is to be used.

Automatic Remote PC

The PC has 3 drivers for the serial port: AUX, COM1 and COM2. COM1 and COM2 obey the Xon-Xoff protocol and use ctrl-C for break. (This information may be quite specific to an IBM-PC running DOS 2.1.) AUX does none of these. The following AUTOEXEC.BAT file will cause the PC to boot up automatically with a remote terminal.

AUTOEXEC.BAT

```
XPORT 1
MODE COM1:9600,N,8,1        (set your baud rate)
CTTY COM1
DATE            (optional)
TIME            (optional)
```

If the PC is remote, then the X parameter should be used in commands:

A>COMIBM SX FILE.EXT

or:

A>COMIBM RX FILE.EXT

The X parameter prevents typing of messages which would interfere with communications and prevents testing of the user terminal for abort signals. When the PC is remote ctrl-C will cause a break and ESC is used to cancel a line partly typed. If a program hangs up, usually a remote re-boot can be performed by typing ~~~~ (4 tildes) slowly (1-second gap between each). This invokes a trap in the XPORT driver and causes a jump to the 8088 reset address. This will work unless interrupts have been disabled or if memory holding XPORT has been destroyed.

If you are working with a remote PC from a DEC system it is usually better to use the CP/M-80 communications program since key translation for the local MS-DOS may confuse the issue in some modes of operation.

Commands which make an explicit reference to the CON: device will still work with the PC keyboard and display. Thus use:

TYPE file

rather than:

COPY file con:

DOWNLOADING

COMIBM.EXE can be used to download to remote systems by the wait for echo method. The download menu can be entered by typing ctrl-T while in terminal emulation mode. The commands are as follows:

L FILE.EXT (Send a file including CR, LF and wait for the remote system to echo the same character back. Beep and put up an error message if any echo does not agree.)

N FILE.EXT
NT FILE.EXT (Send a file except when linefeeds are encountered don't send them and gobble up and echo all characters echoed in response to the linefeed until nothing is echoed for 1/10th second. In the NT mode change all tabs in the file to spaces before sending them.)

For all downloading modes only 7 bit characters are sent and any control characters except for CR, LF and TAB are tossed out. The main reason for downloading is to load a communications program to a remote system with no communications program.

DOWNLOADING COMIBM.EXE

If you do not have COMIBM.EXE on PC compatible media it may be downloaded. None of this will probably work on DOS versions less than 2.0. Downloading is difficult to do over a modem connection because a 70K byte file must be sent with no errors (or errors must be repaired manually). Before attempting downloading make sure that you have no communications program which is XMODEM compatible, which you can use instead. The description here assumes that you have the CP/M-80 program COM.COM running on the Decmation D100 on a system which supports an auxiliary serial port. The method is to first download a small program DNLD.COM which, in turn, is used to download a hex version of the file COMIBM.EXE.

Make sure that the connector on the IBM serial port has pins 4 and 5 connected together and pins 6, 8 and 20 connected together. The serial line from the DEC system is connected to the COM port on the PC. Boot the PC and enter the command:

Convert the file DNLD.COM to hex format under CP/M-80

with the following command:

```
A>UNLOAD DNLD
```

This will create a file DNLD.HEX.

Create a hex version of COMIBM by:

```
A>REN COMIBM.COM=COMIBM.EXE
A>UNLOAD COMIBM
```

Then on the PC:

```
A>MODE COM1:9600,n,8        (9600 baud, no parity, 8
                             data bits.)
A>COPY CON: COM1:
XXXctrl-Z
                            (May be needed to fully
                            enable port by enabling the
                            signal data terminal ready.)
A>

A>CTTY COM1                 (Make the PC terminal be
                            remote.)
```

Then, under the CP/M-compatible environment on the DEC system, run:

```
A>COM T      (Enter terminal mode.)
```

If you have a connection you will get the PC prompt and you DEC terminal will be the PC terminal. On some system you may have to use a lower baud rate, especially RSX system with DLV serial ports, which are limited to 1200 to 2400 baud.

```
A>          (PC prompt)
A>DELETE DNLD.HEX
A>EDLIN DNLD.HEX
*I                          (Editor in insert mode.)
1*ctrl-T                    (Start download)
COMMAND: N DNLD.HEX
```

(The entire file will type slowly and be downloaded.)
returning to terminal mode

```
ctrl-Z                      (exit editor insert mode)
*E                          (exit editor, saving file)
A>                          (PC prompt)
```


Still working from the CP/M side:

```
A>DEBUG DNLD.HEX
-NDNLD.COM
-W
(writes file)
-Q
A>CTTY CON
```

Now go to the PC keyboard, which should now be alive and:

```
A>
A>DNLD COMIBM.HEX
```

Now go back to the CP/M keyboard:

```
ctrl-T
COMMAND: L COMIBM.HEX
```

(The file will take several minutes to download, watch for errors signaled by beeps.)

```
Download complete, returning to terminal mode.
ctrl-Z          (Type ctrl-Z)
```

Now go to the IBM keyboard and:

```
A>DEBUG COMIBM.HEX
-NCOMIBM.COM
-W
-Q
A>RENAME COMIBM.COM COMIBM.EXE
```

The program is now ready to use (create a COMIBM.DEF file).

Note that EDLIN cannot be used to download the main file because it is too long. Also the "L" method is faster than the "N" method. Make sure that COM files end up as COM files and EXE files end up as EXE files. They are different formats. Make sure to create a COMIBM.DEF file, especially if you are not working at 9600 baud.

8.0 Automatic Batch Transfers

By driving the computers on both sides of the line with command files a long sequence of file transfers may

be performed automatically. For example:

```
Computer #1 CP/M        Computer #2 IBM-PC

COM S FILE1.EXT         COMIBM R FILE1.EXT
COM S FILE2.EXT         COMIBM R FILE2.EXT
COM S FILE3.EXT         COMBIM R FILE3.EXT
etc.                    etc.
```

To aid in creating such command files a utility program called MAKSUB.COM under CP/M-80 and MAKSUB.EXE under MS-DOS is available. This program allows you to create a command file using either a file containing a list of file names or by using a wild card file specification which implicitly defines a list of file names. The command line to use this utility is illustrated below.

General form:

A>MAKSUB outfil=wildcard prototype_command_str

outfil- the file to be created. Default extension is .SUB under CP/M and .BAT under MS-DOS. If outfil= is omitted, output will be to the user's terminal.

wildcard- A standard wildcard file specification containing *'s and ?'s. For example: *.?s?, or abc??.*. Instead of a wild card specification, a file containing a list of file names can be specified by @FILE.LST.

prototype- The command line with *.* or *. or .* to indicate substitution of file names. A slash (/) can be used for new line.

Examples:

A>MAKSUB TEST=*.ASM COM S *.*

creates a file TEST.SUB containing:

```
COM S FILE1.ASM
COM S FILE2.ASM
COM S FILE3.ASM
etc.
```

A>MAKBAT COMD=@FILES.LST COMIBM R *.*

If the file FILES.LST contains:

```
FILE1
B:FILE2.OV1
C:FILE4
```

Then a file named COMD.BAT is created as follows:

```
COMIBM R FILE1
COMIBM R B:FILE2.OV1
COMIBM R C:FILE4
```

The MAKSUB/MAKBAT utility can handle file names with an 8-character name and 3-character extension. CP/M style disk names (A:, B:, etc) are handled. It can be used to contstruct a command file for Intel development systems by first creating a list of file names using the Intel directory utility program. ISIS disk names (:F0:, :F1:, etc.) are not specially accounted for, but they can be included in the fixed text of the prototype string.

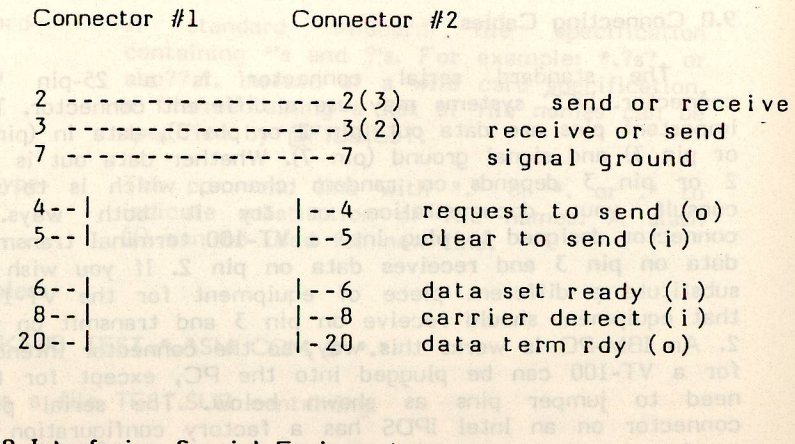
9.0 Connecting Cables

The standard serial connector is a 25-pin "D" connector. DEC systems may use a different connector. The important pins are data out (pin 2 or pin 3), data in (pin 2 or pin 3) and signal ground (pin 7). Whether data out is pin 2 or pin 3 depends on random chance, which is to say consult your documentation or try it both ways. **A connector designed to plug into a VT-100 terminal transmits data on pin 3 and receives data on pin 2.** If you wish to substitute a different piece of equipment for the VT-100, that equipment should receive on pin 3 and transmit on pin 2. An IBM-PC is works this way, so the connector intended for a VT-100 can be plugged into the PC, except for the need to jumper pins as shown below. The serial port connector on an Intel iPDS has a factory configuration of receive on pin 2 and send on pin 3. It is a female connector. If you wish to use a cable designed to connect to a VT-100 you will have to interchange pins 2 and 3 in the cable (null modem cable) and you will have to change the sex, since a VT-100 is male, rather than female. A sex changing cable has female or male connectors on both ends. It is useful to have 3 short cables. A male to male, a female to female and a male to female null modem. With these 3 cables you can connect

anything to anything. Connect them as shown in the figure below.

It does not hurt the equipment to short the signal lines together or to ground in most systems.

In addition to the lines already mentioned, certain other signals may be on the connector. These are things such as clear to send, data set ready, etc. These signals are mainly used for modems. However, in some systems the serial port will not work if these signals are not in the correct state. The jumpers shown below will usually ensure that the port will work. They fool the interface. The interface enables request to send but cannot send until it receives clear to send. The jumper between pins 4 and 5 feeds the request to send back to the clear to send. The interface enables data terminal ready but cannot send until it receives data set ready and carrier detect. The jumper between pins 6, 8 and 20 assures that these signals are received. On the IBM-PC at least one character must be sent before any characters can be received, because data terminal ready and request to send are not enabled until the first character is sent. They remain enabled after that, unless the port is reset. Wiring the modem control lines from one system to the other will not work in most cases.



10.0 Interfacing Special Equipment

Special equipment includes such items as PROM burners or in-circuit emulators. An RS-232 serial interface is the most universal computer interface, so much special equipment uses it. Problems are encountered when the special equipment uses an uncontrolled communications protocol. That is to say that it assumes that the system it

talks to can accept unlimited data at full speed. Some terminals can do this, but no DEC system can accept uncontrolled input except when the baud rate is made very slow. The amount of high speed input that the DEC system can accept is determined first by the size of the system input buffer and secondly by the size of the buffer in the Decmation control program. If the buffer in the Decmation control program is near full an "emergency" Xoff character is sent. An Xon is sent if the buffer is then emptied below a certain level by the CP/M or MS-DOS program. This will control systems which respond to Xon-Xoff protocol provided that the baud rate is not too high or the system buffer is large enough. The buffer in the control program is typically about 400 bytes with the Xoff threshold at about 350 bytes. A regular communications protocol which sends fixed length records of less than 300 bytes will never trigger the Xoff.

Another problem occurs when the special equipment uses the CTS (clear to send) and the DTR (data terminal ready) lines to control data flow. This is a method generally incompatible with DEC equipment which has the additional disadvantage of requiring the use of 2 more signal lines in the cable. The solution is to run very slow, modify the software, or to use a protocol converter box to change the communications protocol not to use these lines.

The best protocol for a serial port situation is one where the maximum length of all messages a known in advance. Either fixed length packets are exchanged or variable length packets not longer than the size of the input buffer. Another packet is not sent until the previous one has been acknowledged. Very short packets, such as one byte packets, will be excessively slow, especially on a timesharing system which takes a while to wake up and send the acknowledgement. A good maximum packet size is about 80 bytes in most situations. 200 bytes is more efficient, while some systems can only handle 20 bytes at the highest speed.

11.0 Baud Rate, Serial Port Setup

The baud rate which may safely be used with a given DEC host system depends on the type of hardware, loading and operating system used. The following rules will be a guide, but not a substitute for experimentation. It is suggested starting at a slow baud rate, such as 1200, and working up to higher rates. Over a modem you will be usually be limited to 1200 baud.

RT-11* operating system

Only DLV-11 (DL-11 on unibus) type serial ports may be used as auxiliary communications ports under CP/M. No problems at 19200 baud with any processor should be encountered.

TSX+ operating system

Only DLV-11 type serial ports may be used as auxiliary communications ports. 9600 baud and beyond should work ok. If problems are encountered on heavily loaded systems, try raising the interrupt priority of the serial port or lowering the baud rate. When using the terminal port sysgen the input buffer about 280 bytes long. Maximum baud rate depends upon TSX+ version and may be limited to 1200 or 2400.

RSX* operating system

Main problem areas are DLV-11 serial ports on slow processors, such as 11/23. In this case limit baud rate to 2400 for receive or have transmitter insert delay to limit input rate. Will output at rates near 9600. DZ or DH serial ports should support rates of 9600 on input or output, possibly limited to 4800 baud on the slowest processors such as the 11/23 or 11/24. If you can control the input buffer size (M+ only) set it to the maximum up to 300 bytes.

RSTS* operating system

Auxiliary serial ports under CP/M are not currently supported under RSTS. Determine the maximum communications speed through the terminal port by experimentation. It may be necessary to boost the task priority on loaded systems to keep up with the serial input.

VMS* operating system

9600 baud should work on most VAX systems, through an auxiliary port or the terminal port. The system manager should adjust the input buffer size for the line to 150 or 300 bytes in length, depending on the packet size to be used.

IBM PC, Intel systems and other CP/M systems

If the communications port driver does polling, then the maximum baud rate is restricted by the length of the pollling loop. Usually 4800-19,200 baud is achievable.

The problem which may be encountered is loss of characters when using terminal emulation. This can happen if the systems display cannot display data as fast as it is sent. The IBM-PC display is very slow, so when a polling mode is used characters are lost, especially at the start of lines after the screen scrolls up. The XPORT interrupt-driven driver will correct the situation for systems which are PC-compatible. Directing the remote operating system to insert nulls after each line may also help.

If the local system's terminal is buffered and supports Xon-Xoff protocol, then this will effectively prevent loss of characters on the screen up to the baud rate which the polling loop can support.

Baud Rate Adjustment

DEC systems which use the DLV-11 type of interface can have the baud rate adjusted by changing wire-wrap jumpers on the interface board. The suggested setting is as follows:

1-stop bit, parity disabled
8-data bits

Other DEC systems which use DZ or DH type interfaces can have the baud rate and other settings changed by operating system commands.

On Intel systems and the IBM-PC the baud rate is initialized by the COM program. By default this is 9600 baud, 1-stop bit, 8-data bits, parity disabled. To change it you can patch the executable image, or assemble the driver with new values and do a new link. Under CP/M the process is always a patching process. Use either DDT or PATCH.COM. On the IBM-PC use DEBUG to patch. Unless you have the IBM macro assembler, patching is the only method available.

The source listing for the driver is always included. It contains the documentation needed to create a runable program.

A. Baldwin

USING COMMAND FILES TO [INVISIBLY] RUN CP/M SOFTWARE WITH THE DECMATION D100

DOC 1062 15-APR-85

TABLE OF CONTENTS

1.0 INTRODUCTION

Many users of the D100, Z80 processor card may wish to use it in an "invisible" or "dumb" fashion. This means that the end user types a command under his native operating system, such as RSX*, TSX+ or VMS* to start an automatic sequence which calls upon the D100 to perform some task. The end user may have no idea how the task is performed. He needs only to learn the command syntax, which may be made very simple for a complex sequence.

An example would be to use the D100 to perform a Z80 assembly. The command syntax might be:

```
>@DOZ80ASM file [hex]
```

If the user has a source file named ABCD.ASM then he would type:

```
>@DOZ80ASM ABCD
```

The assembly would run and he would receive the file ABCD.PRN back in his account. An option could be that if he types:

```
>@DOZ80ASM ABCD HEX
```

Then he will also receive the file ABCD.HEX (an additional output of some assemblers) back in his account.

The example above is only one of many possible ways to handle the commands. We have show the operating system prompt as ">" which is the prompt for RSX. Obviously TSX+ and VMS have different prompts ("." and "$").

The usual sequence of execution will be as follows:

1) The user types a command line with parameters which define the action to be taken. These parameters are substituted in a call to the control program. The control program waits for a free D100 card. When the CP/M compatible system boots, the appropriate parameters are passed to a CP/M submit file, which is the CP/M version of a command file.

2) A sequence of CP/M commands are executed. The user's source files are copied over to CP/M. The desired operation is performed and any result files are copied back to the user's account.

2.0 CONTROL PROGRAM STARTUP

This manual attempts to be independent of the host operating system. The three operating systems TSX+, RSX and

VMS are each different and the reader will have to make mental adjustments for the characteristics of his own operating system. We use CP? to indicate the name of the control program which is actually named CPT.SAV under TSX+, CPX.TSK under RSX and CPM.EXE under VMS. In the CP/M compatible environment, the host operating system differences are mostly invisible.

The "normal" or interactive mode of operation is for the user to run the control program and then specify the files to be used as CP/M disks during an interactive setup stage. Then the user types "GO" to boot the CP/M compatible operating system. For example:

```
.RUN CP?
blah, blah
-->A: DL0:DISK1
-->B: DL2:DISKX
-->GO
A>SUBMIT SFILE PARM1
```

would be an interactive sequence for starting the CP/M compatible environment and then starting a CP/M submit command sequence with one substitutable parameter "PARM1." In the non interactive mode, the same commands are entered, but they are entered as a part of the command line, or from a separate, special command file. A single command line with the same effect is as follows:

```
>CP? W/A:DL0:DISK1/B:DL2:DISKX/GO/SUBMIT SFILE PARM1/
```

The "/"'s are used to indicate a carriage return, or end of command line. (Use two consecutive slashes to insert a slash in the text.) The first command W/ is a special command which means wait until a D100 card is free for use. If the commands needed won't fit on the command line, then an auxiliary command file may be used:

```
>CP? W/@FILE.CMD/
```

If an auxiliary command file (FILE.CMD above) is used, then slashes should not be used to separate the commands. Each command should be on a separate line.

Certain operating systems requirements may have to be met before additional text can appear on the same host operating system line which runs the control program. For example, under RSX, the control program must be installed as ...CPX. Under TSX+ the "iniflg" must be patched non-zero to use the above mode of operation.

The most common startup command structure will have a series of parameters given in the initial command line which starts an indirect command file running under the DEC

operating system. The parameters then are passed to the command line in the DEC command file which starts execution of the control program. The parameters are then passed again to the command line of a CP/M SUBMIT command. A single-line
host command file might be as follows:

```
CP? W/A:SY:SYSDSK/B:WRKDSK/GO/SUBMIT DOASM ^1 ^2/
```

The ^1 and ^2 (TSX+ convention) indicate arguments to be substituted from the command line the user will type. Under other operating systems the convention is different. For example, "P1" and "P2" are used to indicate substitution from the command line under RSX.

CP/M File: DOASM.SUB

```
SETFATAL
COPY $1.ASM/H TO $1.ASM
ASM $1
COPY $1.PRN TO $1.PRN/H
IF $2=HEX THEN COPY $1.HEX TO $1.PRN/H
EXIT
```

If the user types the following command:

```
.@DOZ80ASM MYFILE
```

Then the following sequence will take place automatically (TSX+):

```
.CPT /W/A:SY:SYDSK/B:WRKDSK/GO/SUBMIT DOASM MYFILE/
A>SUBMIT DOASM MYFILE
A>SETFATAL
A>COPY MYFILE.ASM/H TO MYFILE.ASM
A>ASM MYFILE
blah, blah
A>COPY MYFILE.PRN TO MYFILE.PRN/H
A>IF =HEX THEN COPY MYFILE.HEX TO MYFILE.PRN/H
A>EXIT
.
```

The user would then have the file MYFILE.PRN as a TSX+ file. He could view any error messages output to the terminal by the assembler while the assembler is running. The statements in the submit file are each explained as follows:

SETFATAL - This is a CP/M program which patches the BIOS (basic input output system) so that any request for terminal input by a program will cause an immediate exit back to the host operating system. This prevents control

from stopping in the CP/M environment if a programs attempts to communicate interactively with the user.

COPY - This copies a file from one place to another. The /H switch indicates the file is a host (TSX+, RSX, VMS) file.

IF - This conditionally executes the statement following the "THEN" keyword. In the example above the 2nd substitutable parameter ($2) is compared to the character string "HEX". Since the parameter is null, nothing shows up on the left of the equal sign.

EXIT - This causes an immediate exit back to the host operating system.

ASM - This invokes the default assembler. The source is taken as MYFILE.ASM. The output is MYFILE.PRN and MYFILE.HEX. Other assemblers have different command structures.

2.1 THE SUBMIT PROGRAM

The submit program included with the D100 software is compatible with the standard Digital Research program of the same name, but has additional features. The format of the command line is as follows:

```
A>SUBMIT SUBFIL PARM1 PARM2 PARM3 .... parm15
```

SUBFILE is a file specification for a file with an assumed type of ".SUB". PARMn are character strings separated by spaces. The file SUBFIL.SUB contains command lines, such as can be input at the A> prompt. Certain special characters have special meaning in the text of a .SUB file or in the arguments. Whenever $n appears in the text, the character string for PARMn is substituted. "$$" is changed into "$". "^X" is changed into ctrl-X. "^^" is changed into "^". A $U in the text is transformed into the default disk, A, B, etc. (without the colon). If the character "~" (tilde) appears in an argument it is changed into a space. "~~" is changed into "~". The parameters numbered 10-15 are represented by $A-$F in the text.

When submit is executed, the text, with substitutions, is copied in reverse order to a file named "$$$.SUB" on the default disk. This file contains each command line in a fixed-length 128-byte record. The next command to be executed is at the end of the file. The CP/M compatible operating system, each time it warm boots (comes back with the A> prompt), takes another command from the $$$.SUB file, shortens the file by one record, and executes the command. This works as a sort of a stack kept in a file. If submit is executed from a

submit file, the additional commands are added to the existing $$$.SUB file. Thus, unlimited nesting is implemented, except that the $$$.SUB file is limited to 127 lines at any time.

An extension of the Digital Research submit is an IF statement and statement labels. The IF statement (Actually a program IF.COM) can be used in several different ways. To check if files(s) or disk(s) exist. The syntax is:

```
A>IF [ALL] [ANY] [NOT] [EXIST] file1 file2 etc. THEN
        statement [ELSE statement]
```

for example:

```
A>IF EXIST MYFILE.ASM THEN ASM MYFILE ELSE EXIT
```

If the file MYFILE.ASM exists on the default disk then the command line: A>ASM MYFILE will be executed, otherwise the command line: A>EXIT will be executed. By using the keywords ALL, ANY and NOT several files can be checked at one time:

```
A>IF NOT ALL EXIST FILE1.ASM FILE2.ASM THEN EXIT
```

The IF statement can also be used to compare character strings, usually a substitutable string against a fixed string. For example:

```
IF ANY $4=LIST $5=HEX THEN GOTO ABC
```

In this case if either of the character strings on the two sides of the equal sign are identical the statement GOTO ABC will be executed, otherwise the next statement in the file. A pound sign (#) may be used to indicate not equal. In the example above ABC must be a label on a statement after the statement containing the GOTO. For example:

```
IF $4#LIST THEN GOTO ABC
COPY $2.PRN TO $2.PRN/H
:ABC: ERA $U:$2.PRN
EXIT
```

is the code which may appear in a .SUB file. Assuming that $2 is "FILE", $4 is "LIST" and the default disk is "A", then the submit program will change this code to the following code which is actually executed:

```
A>IF LIST#LIST THEN SKIP 1
A>COPY FILE.PRN TO FILE.PRN/H
A>ERA A:FILE.PRN
A>EXIT
```

Labels must start and end with colons. All GOTO's are changed

to SKIP nn statements, where nn is the number of statements which must be skipped to reach the labeled statement. The labels are removed from the executable code. The program SKIP.COM simply removes off the number of statements specified from the $$$.SUB file.

To see if a substitutable parameter exists use a statement such as the following:

```
IF $3= THEN ....
```

This will be true if the $3 parameter is undefined.

The command:

```
ERA $U:$$$$$$.SUB
```

in a submit file will erase the $$$.SUB file on the default disk ($U:) and thus terminate automatic execution. Note that $$ is reduced to $, so 6 $'s must be used to represent "$$$". The commands EXIT and QUIT will also erase the $$$.SUB file. Other programs may erase the $$$.SUB file if they encounter an error. Typing control-C will also erase the $$$.SUB file.

2.2 COMMAND SUMMARY FOR SUBMIT FILES

IF [NOT] [ANY] [ALL] EXIST f1 f2 .. THEN s1 [ELSE s2]

Checks a list of files or disks to see if they exist then executes statement s1 if the conditional is true, else s2 or the next statement.

Examples:

IF NOT EXIST B: THEN QUIT THERE IS NO DISK B:

If disk B: is undefined, then the QUIT program is executed. It stops execution of the submit file and prints the message following "QUIT".

```
IF NOT ANY EXIST A:MYPROG.COM MYPROG.COM THEN
                        QUIT NEED MYPROG.COM
```

One of the files must exist or the QUIT is executed. This is the normal syntax to check if an executable program exists, since A: is searched if a program does not exist on the default disk.

IF [ANY] [ALL] st1=st2 st3=st4 .. THEN s1 [ELSE s2]

Compares a series of strings. Either equal (=) or not equal (#) operator may be used between the strings.

Examples:

IF ANY $1= $2= $3= THEN QUIT MISSING ARGUMENTS

Checks the substitutable parameters $1, $2 $3 to see if any is missing. If so, quits execution.

IF $3#8080 THEN GOTO USEZ80

If the 3rd parameter is not the character string "8080" then execution continues at the statement labeled :USEZ80:.

GOTO xxx

GOTO xxx is an executable statement only when it appears in a .SUB file. The program SKIP.COM must be present on the default disk or disk A:. The submit program changes each GOTO xxx into a SKIP nnn command in the executed commands. xxx is a character string. nnn is a decimal number. Example:

```
IF $1=HEX THEN GOTO USEHEX
...
...
:USEHEX: ....
```

becomes:

```
IF HEX=HEX THEN SKIP 2
```

in the executable code. ("HEX" is assumed to be the resolved value of $1 for the example.) The skip program removes the next nnn executable lines from the $$$.SUB file. The labels (:USEHEX:) do not appear in the $$$.SUB file. Backward (to a previous statement) GOTO's are not allowed.

QUIT [message]

Stops submit execution (erases the $$$.SUB file) and prints the optional message. If QUIET (see below) is in effect, console output is re-enabled.

EXIT [message]

Prints the optional message (overriding quiet mode) and exits to the host operating system.

QUIET

Disables all terminal output except for special routines

which override the quiet mode. Stays disabled until NOQUIET is executed.

NOQUIET

Re-enables terminal output disabled by QUIET above.

EMESG message

Prints the message given overriding quiet mode.

! comment

Any line in a .SUB file starting with (!) is a comment and is ignored. Comment lines can be used to document a .SUB file.

XSUBP PROG [command line]

Executes the program PROG with the command line if any. Any terminal keyboard input requests made via the CP/M call 10 will be satisfied with data from the submit file. A substitute for the Digital Research program XSUB.

SETFATAL

When executed sets a state such that any request for terminal input will cause an immediate exit to the host operating system.

HCMD c1/c2/c3/.../CON/

Causes a series of setup mode commands to be executed. The last command should always be either CON (continue) or GO. As an example, the following command re-assigns disk B:

HCMD B:/B:DISKY/GO/

RDONLY d:

Sets the disk specified A:, B:, etc. to read only. Any attempt to write on the disk will cause a bdos error.

2.3 UNDERSTANDING CP/M COMMAND INPUT

When the A> (or B>, etc) prompt is present the CP/M compatible operating system expects a line of input consisting of a program name - a space - then a command string. How the command string is analyzed depends on the program executed. The entire command line when executed from a

submit file cannot exceed 126 characters in length. All lower-case is converted to upper case.

Certain programs may input additional commands from the terminal. If it is desired that these additional commands be taken from the $$$.SUB file then the program name should be preceded by XSUBP:

```
A>XSUBP PROG COMMAND LINE
```

instead of:

```
A>PROG COMMAND LINE
```

Provided that the program does line-oriented input using the operating system call number 10, any additional lines of input will be taken from a $$$.SUB file. If the $$$.SUB file is empty, then the console will be read. XSUBP is Decmation's substitute for the Digital Research XSUB program. The usage is different.

Host Command Files

The other type of command file is a host file used to provide console input to the CP/M program. For example the command:

```
.CP? @FILE.CMD/
```

or:

```
.CP? A:DISK/GO/SUBMIT PROG PARM1 PARM2/
```

causes input which otherwise would be typed to be taken from a host file or the command line following the program name. A problem with using a host command file is that some CP/M programs will not read a character from the console without first testing if a character is ready. The character ready flag is never turned on by a host command file. If it were, then many programs would be aborted, because they abort when the user types on the terminal. Generally the best method is to use the host command file is in such a way that the last statement it contains is a SUBMIT. The submit file also allows a sequence of commands to be interrupted by interactive input from the user's terminal. For example:

File WORD.COM (host file)

```
CP? A:SYSDSK/B:WORK/GO/SUBMIT WORD/
```


File WORD.SUB (CP/M)

```
WPVT100
EXIT
```

These files allow the user to type "@WORD" at the host prompt and have the **Palantir** word processor start immediately. When the user exits **Palantir** he is immediately kicked back to the host system by the "EXIT" command.

2.4 HANDLING ERROR CONDITIONS

It is often important that a command sequence complete regardless of any error condition encountered. This is especially true if no one is watching the terminal, for example with a TSX+ detached line, where all terminal output is discarded.

Many CP/M programs will query the operator for input if they encounter an error condition. The SETFATAL command will alter the CP/M compatible operating system so that any console input request (but not a test console input character ready) will cause an immediate exit to the host operating system. Under some host operating systems it may be possible to place a time limit after which the control program will be aborted.

Another error condition can occur if a file to be used as a CP/M disk is inaccessible because another user is using it. If disks are used in read-only mode (/RO switch) then several users can jointly use the same disk. Important restrictions on shared disks are that they cannot serve as the default disk when SUBMIT is executed (an attempt is made to write $$$.SUB file) and some programs may create work files on the default disk. It is possible to have a bank of host files which serve as disks and dynamically allocate them to different D100 cards. (If the system only has one D100 card none of these questions come up.) Dynamically changing disks depends on the use of the HCMD program. This allows a command to be issued from a submit file which changes the configuration of the CP/M disks, or executes any other command normally executed from the setup state of the control program. As an example:

```
A>HCMD B:/B:[2,4]FILE/GO/
```

will undefine disk B:, redefine it as [2,4]FILE.CDK and re-boot the CP/M compatible operating system. The last command in the string must always be either "GOcpm" or "CONtinue". CON should not be used when disk assignments are changed, unless the new disk assignment replaces a previous disk of exactly the same size and characteristics. In this case a control-C (^C in a submit file) must be executed to signal the system that the disk has been changed. If an attempt is made to define a pseudo disk which doesn't exist or is in use by another D100

card in non-shared mode, then the assignment will not be made. A sequence such as the following can try a series of disks until one is found which is usable.

```
IF NOT EXIST B: THEN HCMD B:/B:DISK1/GO/
IF NOT EXIST B: THEN HCMD B:/B:DISK2/GO/
IF NOT EXIST B: THEN HCMD B:/B:DISK3/GO/
IF NOT EXIST B: THEN EXIT ALL DISKS ARE IN USE
```

An ELSE clause could be added to skip the tests after the first successful one.

If a disk is shared between users by means of the /RO switch after the file name, an attempt to write on it will result in possibly voluminous error messages, one for each attempt to write a sector. The following command will set a disk read only under the CP/M compatible operating system and cause a "bdos error" which will ask for console input and thus abort the control program if SETFATAL is in effect:

```
A>RDONLY B:
```

This only remains in effect until the next cold ("GO") or warm boot (every command), so it must be the command directly before any program which may write the disk erroneously. However, if the NOBOOT (see below) option is used, then the read-only status will remain in effect until a boot of the system is forced with the GO command.

2.5 NOBOOT OPTION

The CP/M compatible system (technically called RP/M and a product of MicroMethods, Inc.) has two levels of reset. A cold boot means that the entire system is reloaded, including the Basic Input-Output System or BIOS. This affects Z80 memory from E400h to the end of memory, as well as some locationss below 100h. A warm boot reloads the command interpreter and the disk operating system (normally located at addresses E400 - F9FF. When a warm boot takes place the default disk is normally re-logged in. This means that the entire directory must be scanned in order to construct an allocation bitmap which is stored in the BIOS. On disks with large directories the warm boot can take a second or more. When the noboot option is used no part of the operating system is reloaded and no disk logging is done. Successive steps when a submit file is executed take place much faster if the noboot option is used. The only delay is the time required to read the next command from the $$$.SUB file. The disadvantages of NOBOOT option are that transient programs will have available 2.5K bytes less memory to work with. This is a problem for very few programs. Certain programs may overlay the entire operating system. These programs will disable the NOBOOT option, or even cause a CP/M environment crash.

Decmation's ISIS-X emulator program disables the NOBOOT option. Follow its execution with a NOBOOT command to re-enable the noboot option.

The noboot option must not be used with removable floppy disks. A disk change will not be detected.

The main advantage of the noboot option is that submit files are executed much faster.

Installing the Noboot Option

A command file, NOBOOT.SUB, will patch the RP/M operating system and install the noboot option. The command file MAKESYS.SUB may be used to remove the patches and create the original version of the operating system. After a cold boot with a patched operating system the command NOBOOT must be executed to disable warm booting. If a control-C is typed the A> prompt comes back instantly when noboot is in effect. Otherwise there is a slight delay of about 1 second.

A. Baldwin

USING THE DECMATION D100 CARD TO RUN CP/M PROGRAMS

FOR ALL DEC OPERATING SYSTEMS

(RT-11, TSX+, RSX-11M, RSTS/E, VMS)

21-March-1984

Decmation, Inc.
3375 Scott blvd. #422
Santa Clara, CA 95051

(408) 980-1678

CONTENTS

INTRODUCTION

The Decmation D100, Z80 processor card comes in two hardware versions, the D100Q for Q-bus computers (LSI-11, PDP-11/23, PDP-11/73, Micro VAX), and the D100U for unibus computers (PDP-11, VAX). The D100 can be used under a number of different operating systems. These operating systems include: RT-11, TSX+, RSX-11M, RSTS/E and VMS.

This manual gives basic operating instructions for using the D100 to run CP/M software. Separate hardware and software instaallaion manuals are also provided for the particular hardware configuration and operating system in use at your installation. Differences between the various operating systems are noted in this manual and in the software installation manual for your operating system.

The number of different models, versions and variations of DEC computers supplied by the Digital Equipment Corporation and 3rd parties is very great. The same can be said for operating systems. The D100 will work on the most popular versions of these operating systems. At this time support is not available for some of the less popular operating sytems, but we are constantly expanding our base of support.

Many hundreds of D100 cards have been installed in the U.S. and foreign countries. The majority of these installations were performed by the end users' personally. In some cases these users were accountants, lawyers or other non-technical persons who had never looked inside a computer before. To our knowledge, none of these installations has ever resulted in damage to either the computer or the D100 card.

Hardware installation or software installation is not difficult, but it is the hardest first step for most users. If you encounter a problem in performing the installation, or in using the D100 card, you should not hesitate to contact Decmation or the dealer you purchased the card from. The problem invariably has a solution.

CHAPTER 1. RUNNING UNDER CP/M

Assuming that you have completed the hardware installation and the software installation, you are now ready to start running CP/M programs. CP/M is very similiar to your DEC operating system in the concepts and the type of things you can do. Just as your DEC system has names for disks, such as DL0: or DY0:, disks under CP/M have names. The disks under CP/M are always named: A:, B:, C:, etc. Unlike the DEC operating system, the CP/M disk name does not indicate the nature of the physical device. With the Decmation D100 card you will generally use files, under your DEC operating system, as CP/M disks. For example, you may have a file named MYFILE.CDK which is 1,000,000 bytes in size defined on one of the disks attached to your DEC computer. This file can be used as CP/M disk A:. CP/M will treat it the same as it would a separate disk drive which contained 1,000,000 bytes. A DEC operating system file used as a disk for CP/M is called a pseudo disk or a virtual disk. Hundreds of distinct CP/M files may be inside a single pseudo disk.

A stand-alone CP/M computer usually has a rigid corespondence between the disk names A:, B:, etc. and the physical devices. With the D100 card you are free to specify and respecify which file will serve as CP/M disk A:, B:, etc.

When you are using CP/M on your DEC computer, CP/M is running on the D100 card. At the same time a program is running on your DEC computer which handles input/output for CP/M.

This is called the control program. When you run the control program it first enters a setup mode which allows you to specify which files will serve as CP/M disk A:, B:, etc. You specify which files to use as the CP/M disks, then you type "GO" to boot CP/M. At that point you are for practical purposes working at a terminal connected to a powerful CP/M computer. One difference is that a special sequence (caret-C) can be typed on the terminal to escape from CP/M and go back to the setup mode of the control program. Typing caret-C is equivalent to hitting the reset button on a stand-alone CP/M computer. In the setup mode of the control program you can change the CP/M disk assignments and reboot by typing GO, or you can exit back to your DEC operating system by typing "BYE". The following diagram should make this sequence clear:

```
(DEC system)  --->  (Control setup)  ---> (Running CP/M)
              "RUN CP"               "GO"

(Dec system)  <---  (Control setup)  <--- (Running CP/M)
              "BYE"                  "^C"
```

1.1 Setup Mode

To run CP/M first run the Decmation control program. This is an executable program under your DEC operating system. Generally it will be named CP.SAV, CPT.SAV, CPX.TSK, CPM.BAS, CPM.EXE, or some similiar name which is consistent with the conventions of your operating system. The control program will type a log-on message and then an

arrow prompt:

```
-->
```

The arrow prompt indicates that the control program is in setup mode. It is waiting for you to type commands to specify how CP/M will operate.

Note: The message "no CPM.INI file present" is not an error.

An Example of Starting Up CP/M

This example strictly applies to RT-11, but the only change for a different operating system would be in the first line of the example, since the name of the executable program and the operating system prompt vary.

```
.RUN CP      (Run control program.)

(A log-on message prints)

-->A: DISK2.CDK      (Define A: to be a
                      file.)

-->B: DY0:           (Define B: to be a floppy in
                      CP/M format.)
-->GO                (Start CP/M running.)
A>                   (The CP/M prompt.)
(You are now running CP/M)
```

To run CP/M, you have to tell the control program what files to use as CP/M disks and where to put the CP/M print output. You can automate this setup stage by creating a file named CPM.INI with the necessary commands. (See section 1.8.)

To be able to boot CP/M you need, at least,

to specify the location of CP/M disk A:. CP/M disks can be realized in one of two distinct ways. Usually, a file is used as a CP/M disk. This is called a "pseudo disk." As far as your host operating system is concerned it is a file just like any other file. The other type of CP/M disk is an actual floppy disk in CP/M format. Usually this is an 8-inch floppy disk in standard CP/M 8-inch format. The following prototype commands show how disk A: is specified while in the setup mode of the control program.

```
-->A: DY1:
```

or:

```
-->A: DISK2.CDK
```

The first command (A: DY1:) declares CP/M disk A: to be a floppy disk in drive #1. When a device name is specified without a file name, the lack of the file name tells the control program to treat the device as a CP/M format floppy disk. Only floppy disks can be in CP/M format. (If your floppy drives do not have the name "DY" or "DX" see your software installation manual for instructions on installing other names.) The second command above, declares that CP/M disk A: is to be realized by the file specified. The file used must have been previously created by using the control program "OPEN" command. (The "OPEN" command is used only 1-time to create a pseudo disk.) The file specification may include all the elements of a file specification acceptable to your operating system, such as the account number and version number under those DEC operating systems which support file names with account numbers and version numbers.

When you type the command specifying CP/M disk A:, an entry is made in an internal table. If the CP/M disk A: is a pseudo disk, the control

program will read the first block of the file when you type the command.

When you specify a CP/M disk to be a floppy, in CP/M format, the floppy disk does not have to be physically inserted in the drive until CP/M attempts to access it. If your operating system software requires that disks should be software "mounted" you should not try to software mount CP/M format diskettes. They cannot be software mounted, since they do not have your operating system's file structure. Under some operating systems you can mount a CP/M format disk as "foreign" to keep other users on the system from accessing it. You may also have to "allocate" the diskette drive under some operating systems.

You can change CP/M format diskettes in the drive while running under CP/M provided that they are in CP/M format and provided that you type control-C to notify CP/M that you have changed the disk. This is in keeping with standard CP/M practice for removable diskettes. If you have pseudo disk files on a removable media, such as a floppy disk, you should not change the media without redefining the CP/M disk in the setup mode of the control program. This is in accordance with standard CP/M practice for non-removable, hard disks.

To review: Most users, most of the time, will use files defined under their DEC operating system as CP/M disks. These are called pseudo CP/M disks. However, users who have floppy disk drives can also use single density, 8-inch, diskettes in standard CP/M format. (Under certain operating systems, 5-1/4-inch, Rainbow format CP/M disks can also be used.) **The only reason to use CP/M format diskettes is media compatibility.** For this reason, Decmation distributes software on standard, 8-inch CP/M diskettes. They operate slowly and do not hold very much data, but you can exchange

them with nearly all CP/M computers which have 8-inch diskette drives. You can easily purchase CP/M software in standard, 8-inch format.

To answer a commonly asked question: There is no such thing as standard, 8-inch, CP/M double-density format diskettes. There are many private double density formats. To use a double density diskette, put a pseudo disk file on it. (Rainbow, 5-1/4 inch CP/M diskettes are not a "standard" outside of DEC.)

Once you have defined disk A:, you can boot CP/M by typing "GO".

```
-->GO
A>
```

When CP/M boots it will type its prompt: "A>". If instead you receive an error message, probably your CP/M disk A: does not have the CP/M system installed on it. Start out using a copy of the distribution disk #2 to be sure that you have a valid CP/M disk. Users who have floppy disk drives on their system receive Decmation's software on floppy disks. Disk #2 is a bootable CP/M format floppy. If you receive Decmation's software on a different media, such as magnetic tape, then you will have a pseudo disk file named DISK2.CDK which can be initially used as disk A: to boot CP/M.

> **Note:** Booting CP/M from a floppy disk is quite slow (approximately 5 seconds). This is of little concern since once you have created a bootable pseudo disk, you will have little need to use CP/M format floppies. Under some operating systems you may get a false error on the first attempt to boot from a floppy disk.

After you have booted CP/M, you can print

the directory of the CP/M disk as follows:

A>DIR

(The directory will print on the terminal.)

For detailed information on the operation of CP/M consult the CP/M documentation supplied in the distribution kit.

It is important to know how to exit from CP/M. Everything you can type at your terminal potentially means something to CP/M, or to a CP/M application program. To provide a means of entering a few commands, which are not passed to CP/M, one key on your terminal is considered a special shift key. By default, this is the caret key (^). If you don't want the caret key to be the special shift key, then you can change it to something else (with the "setcaret" command). To escape from CP/M and return to the setup mode of the control program, type: ^C. This is caret-C, not control-C. We will always write cntrl-C when we mean control-C, in this manual. When you type ^C the result will be:

A>^C
-->

You will be kicked out of CP/M and the arrow prompt of the setup mode of the control program will return. Although we have printed the ^C above, it may not echo on your terminal. After typing caret-C you are in the setup mode of the control program. You can change the assignments of the CP/M disks and then type GO again to re-boot CP/M. If you want to exit back to your DEC operating system, type:

-->BYE

(DEC operating system prompt prints here.)

The command BYE causes the control program to terminate. This returns you to the DEC operating system prompt.

We have described the commands you must know to run CP/M. These are: GO, ^C, and BYE. If you forget commands, you can type HELP, while in the setup mode, and a list of all the commands will be printed on the terminal. In the next section we will describe the commands to setup the CP/M system in more detail.

1.2 DEFINING THE CP/M DISKS AND DEVICES

The control program can assign up to 8 (4 under RSTS) possible CP/M disks:

A: B: C: D: E: F: G: H:

You can assign five possible devices:

LST: PTP: RDR: SEN: REC:

A CP/M device is an I/O port which transmits data sequentially. The devices LST:, PTP:, and RDR: are the standard CP/M devices for printer, punch and reader. These devices transmit data one, 7-bit character at a time. The devices SEN: and REC: are devices added to the CP/M system by Decmation. They transmit data sequentially, in 128-byte sectors. SEN: transmits data from CP/M to your DEC file structure. REC: receives data from your DEC file structure.

The CP/M devices may be connected to files under your DEC operating system, or to physical devices supported under your DEC operating system. The most important device is the LST: device. This is used by CP/M programs to output

to the printer. Usually you will connect this to the DEC printer or to a file under your DEC operating system for later printing. Connect a CP/M device to your DEC system by an assignment such as one of the following:

```
-->LST: LP:          (Connect CP/M list
                      device to printer
                      LP:)

-->LST: PRNFIL.LST   (Connect CP/M list
                      device to file
                      PRNFIL.LST)
```

The assignments for disks and devices currently in effect may be displayed on the terminal by using the SHOW command in the setup mode of the control program.

1.2.1 Rules for Disk and Device Assignments

To make a disk assignment, type the CP/M disk name followed by the DEC operating system file specification. For example:

```
-->A: DY1:
-->B: DL:MYFILE.CDK
-->C: DL1:[5,120]AFILE.CBA
```

To delete an assignment, type the disk name without the file specification. For example:

```
-->C:
```

The following rules apply:

* Disks must be assigned consecutively, without any gaps. If you delete B:, then C:, D:, etc. will also be deleted. Delete the assignment for A: to clear the disk table.

* Once a CP/M disk is defined, you cannot redefine it unless you delete the previous assignment first. (Exception: RT-11 and TSX+.)

* If no file name is present, then the disk is considered to be in CP/M format. Only device names DX0:, DX1:, DX2:, DX3:, DY0:, DY1:, DY2:, and DY3: will be accepted as CP/M format disks. Systems which support Rainbow format CP/M disks will accept DU1: and DU2:. (The program may be patched to accept other device names under some operating systems. See your software installation manual.) Note that in this situation, DY: or DX: is not an acceptable abbreviation for DY0: or DX0: and an error message will be given if this abbreviation is used.

* The device containing the pseudo disk file must be mounted when the pseudo disk assignment is made if your operating system requires software mounting of devices (RSTS, RSX, VMS).

To define the DEC operating system equivalent of a CP/M device, type the device name, followed by the DEC operating system file specification or device specification. For example:

```
-->LST: LP:
-->PTP: DY1:[77,77]PNCH.DAT
-->SEN: YOURFI.ABC
```

To delete an entry from the device table, type the device name without a DEC operating system file

specification. For example:

-->LST:

The following rules apply to the CP/M devices and their DEC operating system counterpart devices and files:

* A CP/M program may access any of the CP/M devices even if they have not been defined in the setup mode. In this case, CP/M will be momentarily interrupted, and the control program will prompt you, on the terminal, for the definition of the CP/M device. This prompt will occur every time the CP/M device is opened unless it was previously defined in the setup mode of the control program.

* The devices LST: and PUN: treat cntrl-Z in the data stream as an end of file. CP/M ASCII files use cntrl-Z as an end of file marker. The reader device transfers data from the DEC operating system file structure to CP/M. End of file on the RDR: device is the physical end of the DEC file or the first null encountered in the DEC file.

* Under some operating systems (RT-11 and TSX+) output to the physical printer is buffered in 512-byte blocks. This can result in some print output left in the buffer at the end of a report. The print buffer is automatically flushed if a CP/M program terminates ("warm boot") or if the CP/M program sends a cntrl-Z to the printer. If neither of these takes place, the user can force the buffer to be flushed by typing caret-Z on his terminal.

Decmation **Application Note #17** gives a way around this problem under certain operating systems.

* CP/M print output can be sent to the system print spool in most cases, if desired. Consult your operating system specific documentation for further information.

The devices SEN: and REC: are supported by CP/M programs SEN.COM and REC.COM. These are used to pass data between the DEC operating system file structure and CP/M. SEN: and REC: transmit data efficiently in 128-byte sectors. The other devices transmit one byte at a time. (See section 1.7.) You will probably have little occasion to use the devices PTP: and RDR:.

1.2.2 Technical Description of CP/M Disks

To CP/M, a disk is a mass storage device divided into 128-byte sectors. CP/M allocates space on the disk by units called logical blocks. A logical block is fixed for a particular disk between 1024 and 16384 bytes. A disk is partitioned into three functional areas: system tracks, directory and data area. The system tracks hold the CP/M operating system and require a total of 52, 128-byte sectors. The directory area contains a list of the files on the disk. Thirty-two bytes are required for each potential directory entry. The directory area is fixed in size. On standard, CP/M, 8-inch floppy disks, the directory is 2048 bytes. This allows up to 64 named files. On pseudo disks the size of the disk is specified when the pseudo disk is created. Pseudo disks can be up to eight megabytes in size and hold hundreds of directory entries. The remaining storage, in the data area of the CP/M disk, is allocated to files stored on the

disk. A file always occupies a whole number of logical blocks. The logical blocks assigned to a file may be physically scattered on the disk. There is no operation in CP/M corresponding to the squeeze operation found under RT-11.

A CP/M program reads or writes disks or devices by making calls to the CP/M operating system. The higher-level (BDOS) part of the CP/M operating system calls upon the lower level I/O routines (BIOS) to actually send the data to the disk or device. The BIOS, which is supplied by DECMATION, communicates I/O requests to the control program running on the DEC computer. The control program has to know what file or device to send or receive data from. If a CP/M program tries to read or write an undefined disk, an error message results (BDOS error). If a CP/M program tries to read or write an undefined device, the control program prompts you to supply a DEC operating system file or device name.

1.3 Creating CP/M Pseudo Disks

A pseudo disk is a DEC operating system file which is formatted internally as a CP/M disk. It is created by the OPEN command given in the setup mode of the control program. **The OPEN command is only invoked one time, when the pseudo disk is created, not every time you use the pseudo disk.** An example:

```
-->OPEN DL1:MYFILE.CDK
```

The OPEN command will ask you to enter the size (in 512 byte blocks) of the pseudo disk desired. The file size you specify may be rounded down, by the control program, to a number which will hold an integral number of CP/M logical blocks. The OPEN command will allow you to set the size of the CP/M directory to between 64 and 512 entries.

The choices you will be given depend on the size of the pseudo disk created. The reason for not making the directory very large, other than the disk space used, is that it may have to be searched from time to time. If the directory is excessively large, the operation of CP/M may be slowed.

The OPEN command assigns a CP/M logical block size which is a function of the size of the pseudo disk created. Generally a CP/M logical block size of 1024 bytes will be assigned for pseudo disks smaller than about 500 blocks. For larger pseudo disks the logical block size will be 2048 bytes unless you specify very large pseudo disks, in which case the CP/M block size will be even larger, up to a maximum of 16384 bytes. The CP/M logical block size will be of little interest to most users. The CP/M program STAT or XDIR can be used to determine the logical block size of a given pseudo disk. If the CP/M logical block size is 2K (2048) all files will use a multiple of 2K bytes. Thus, if you copy a small file from a disk with a logical block size of 1K to to a disk with a logical block size of 2K, the file will appear to increase in size from 1K to 2K. The file's size has not changed, only the minimum amount of space which can be allocated for it.

For the best utilization of disk space a small logical block is favorable. For execution speed a large logical block is favorable. Each logical block on a disk requires 1 bit of storage space from a table in the CP/M BIOS. This limits the total, on-line storage under CP/M to about 25 megabytes. If you exceed that amount of allowed on-line storage, you will get the error message "BIOS too large." However this message is also common when an attempt is made to boot from a disk which does not contain the CP/M system on the system tracks. A single CP/M disk cannot be larger than eight megabytes.

The OPEN command creates the DEC operating system file and initializes the CP/M disk directory to empty. After the OPEN command, you will usually use the CP/M utility GENSYS to copy the CP/M system to the system tracks of the new disk created. Files may be copied to the new disk by using CP/M PIP.

1.4 Console Keyboard

Your DEC operating system attaches special meaning to certain console keyboard characters. CP/M has its own special assignments, which differ. Remember, this discussion refers to the CP/M operating sytem, not necessarily, or even usually, to applications programs which run under CP/M. Some of the similarities and differences are discussed below:

CNTRL-U — Same function in DEC operating systems and CP/M. Deletes current input line.

CNTRL-C — Causes a "warm boot" in CP/M. Under DEC operating sytems control-C kills or interrupts an executing program. Under CP/M the main use of control-C is to inform CP/M that you have swapped a CP/M format diskette in a floppy disk drive. The control-C function under your DEC operating system is disabled and cannot be used to exit from CP/M. Use caret-C to exit from CP/M, then "BYE" to exit to your DEC operating system.

DEL — (Rubout on some keyboards). This deletes the last character typed in CP/M and under DEC operating systems. **Under CPP/M it is better to use the backspace key or control-H.** The delete key echoes the character deleted. Some application programs support the use of the delete key under CP/M, in a manner identical to its use under DEC operating systems.

CNTRL-S — Under DEC operating systems cntrl-S stops output to the terminal. The same thing happens under the CP/M monitor, but not necessarily under CP/M application programs. This subject is discussed in further detail below.

CNTRL-Q — Under DEC operating systems cntrl-Q causes output stopped by cntrl-S to be resumed. Under CP/M, any character including cntrl-Q, will resume the stopped output. This applies in general only to the CP/M monitor. See the further discussion of control-S/Q below.

Most terminals used on PDP-11, or VAX, systems implement the Xon/Xoff communications protocol. This is also known as DC1/DC3 or cntrl-S/cntrl-Q protocol. When the terminal's input buffer is nearly full, it will, of its own accord, send a cntrl-S to the computer to stop further output.

When it has printed enough characters from its buffer to make some space, the terminal will send cntrl-Q to tell the computer to resume output. If the terminal does not follow this protocol, there is the possibility of losing data which is sent from the computer faster than the terminal can absorb it. This would happen if the terminal was in smooth scroll mode, a mode in which it can accept new data very slowly. The problem created by the Xon/Xoff protocol under CP/M is that the cntrl-S and cntrl-Q keys on the terminal are pre-empted for the protocol. Some application programs under CP/M use these keys for data entry. WordStar is a well-known program that does this.

There are two approaches for avoiding a problem with the cntrl-S and cntrl-Q keys. The first is to patch the CP/M application programs which use the cntrl-S or cntrl-Q keys to use other keys. In the case of WordStar, the auxiliary keypad on the VT100 can be used to control WordStar. It is actually much easier to use WordStar from the auxiliary keypad than from the control keys. A number of Decmation Application notes describe how to modify the CP/M BIOS so that the VT100 special function keys (PF1-PF4) or the other auxiliary keys on the VT100 may be used to send any control characters or other sequences to executing CP/M programs.

A second approach is useful if it is only occasionally necessary to send a cntrl-S or cntrl-Q to a CP/M program. The sequence ^S or ^Q can be typed. This will send the appropriate control character to CP/M, without the DEC operating system seeing a cntrl-S or cntrl-Q. In general, the sequence ^x where "x" represents any character (except Z or P) will send a cntrl-x to the CP/M program.

Under most DEC operating systems it is possible to disable the cntrl-S, cntrl-Q protocol. If you do this you should use caution. The terminal

must never be used in smooth scroll mode. You may have to reduce the baud rate to the terminal. VT100's usually work at 4800 baud, but may not work at 9600 baud. Some of the operating system commands used are: 1) Under RT-11 the command PAGE or NOPAGE may be given in the setup mode of the control program. 2) Under TSX+ the command SET TT (NO)PAGE may be used. 3) Under RSX MCR the command SET /[NO]RPA=TTn: may be used. Under RSTS the commands STALL and NOSTALL are used in the control program setup mode.

If you are doing an operation which involves long print lines, either input or output to the terminal, without carriage returns, you may have to set other parameters on your terminal driver to prevent it from inserting carriage returns on its own because it thinks the line is too long. This could happen if the CP/M program uses cursor positioning, rather than carriage returns. A simiiliar situation can happen with the printer on some operating systems, if very long print lines are sent, such as for daisy wheel printers which require an escape sequence for each character transmitted when doing proportional spacing. These problems are not encountered by most users.

1.5 Using the CP/M LST: Device

You will generally assign the LST: device in the setup mode to LP:, LS: or TTnn:, depending upon the device used for printing on your operating system. For example, if your printer device is called LS:, then the following command, given in the setup mode, will direct print output from CP/M programs to your printer:

-->LST: LS:

If your operating system uses print spooling and

the print spooling is automatically implemented when a program writes to the printer device, then the print output should go directly to the spool. If spooling on your system requires a special command to direct a file to the spool, then you should send your CP/M print output to a file and later submit that file to the spooler under your DEC operating system. If the LST: device is undefined to CP/M and a CP/M program starts outputting to the printer, you will be prompted at that time for the LST: device assignment. The list output can be assigned to any legal file or device supported by your operating system.

CP/M transmits only 7-bit bytes to the LST: device. If a ctrl-Z character is encountered in the character stream to the print device, the control program will close the print device or file. In the case of a device (e.g. LS:) the program will not forget the assignment. In the case of a file, once the file is closed there will be no assignment for the LST: device. If output continues to LST:, then the user will be prompted for a new file or device name.

1.6 Copying CP/M Disks

Pseudo disks are just files under your DEC operating system. They may be copied and moved around in the same manner as any other file. Floppy disks in CP/M format my be difficult to copy under your DEC operating system. Under some supported operating systems Decmation provides a program named CCOPY which will make a copy of a CP/M format floppy using an RX01 or RX02 disk drive. Another program, c alled PCOPY, is supported under some operating systems. It converts a CP/M format floppy disk into a pseudo disk file. This is especially useful if the system where the D100 card is installed does not have floppy disks, but you have access to another

computer system with floppy disks and some other removable media compatible with the first system.

It is possible to exchange pseudo disks between different operating systems. This is done by using an RT-11 formatted media, or a magnetic tape in DOS or ANSII format. All DEC operating systems can read tape. Under RSTS/E the program FIT can be used to read or write RT-11 disks. Under VMS or RSX the program FLX can be used to read or write RT-11 disks.

8-inch, CP/M format floppy disks can be freely exchanged between all DEC systems which support floppy drives, as well as with most CP/M computers equipped with 8-inch disk drives.

If it is desired to copy files from 5-1/4 inch diskettes on CP/M computers or the IBM personal computer, this can be easily done by using the Move-It communications program. A serial data link is used to transfer the data, which may be binary files. (See Decmation **Application Note #8**)

1.6.1 Copying Disks Under CP/M

To copy all the files from one CP/M disk to another use the following command:

```
A>PIP B:=C:*.*
```

Of course, substitute the proper letters for B and C in the example above, depending on which disks you are using.

To copy the CP/M operating system from one disk to another use the following command:

```
A>GENSYS
A
B
```

The example above will take the system from disk

A: and put it on disk B:. The program GENSYS will prompt you in the same way the standard CP/M program SYSGEN does.

1.7 DEC Operating System and CP/M File Interchange

If you want to transfer a file between CP/M and DEC operating system formats, use the CP/M programs SEN.COM and REC.COM. For example, if you have a CP/M file named ADDRESS.FIL and you want to copy it to a DEC operating system file to be called ADR.LST, use the following command:

```
A>SEN ADDRESS.FIL[B]
```

After you type the above command, the system will query you for the DEC operating system file name, unless you have pre-defined a DEC operating system equivalence for the SEN: device. Answer with the name of the DEC operating system file: ADR.LST. The file will then be copied. The [B] indicates that the transfer is to be in binary mode. If you omit the [B], then the transfer will be in ASCII mode. (Under VMS and RSX the [B] should also be appended to the DEC file specification.)

To transmit a file from the DEC operating system to CP/M, use the CP/M program REC in a similar manner:

```
A>REC THISFIL.ABC[B]
```

The above command will cause a CP/M file named THISFIL.ABC to be created. The system will query you for the name of the DEC operating system file, unless you have pre-defined a DEC operating

system equivalence for the device REC:. If the [B] is not present, then the transfer will be in ASCII mode. (Under VMS or RSX the [B] should also be appended to the DEC file specification.)

The difference between an ASCII and binary file in CP/M is that an ASCII file is considered to be terminated by a cntrl-Z, while a binary file always has an integral number of 128-byte sectors and ends at the end of the last sector. **If you fail to use [B] for a binary file, you will lose all data after any cntrl-Z which may happen to be in the file.** If you use [B] for an ASCII file, the DEC operating system file created will contain the cntrl-Z terminating the CP/M file and any random data which may follow the cntrl-Z.

The REC function copies a DEC operating system file to a CP/M file. In ASCII mode the physical end of file, or the first null, encountered in the DEC operating system file is considered to be an end of file and is transmitted to CP/M as a cntrl-Z.

It is also possible, but slower, to use the devices LST:, RDR: and PTP: to interchange data between the DEC operating system and CP/M. The main reason for using this method is to take advantage of the formatting switches offered by CP/M PIP. The following commands are examples of this method of copying files:

```
A>PIP LST:=CPMFIL.ABC
A>PIP CPMFIL.ABC=RDR:
A>PIP PRN:=CPMFIL.ABC          (inserts page breaks and line numbers)
```

If the DEC operating system equivalent file name has not been defined, the program will prompt you for the information.

1.8 Command Files

The DECMATION control program allows switching keyboard input to a DEC operating system file. Commands are read from the file and treated exactly as if they had been typed at the keyboard by the operator. To switch the input to a command file, give the command:

-->@ filnam

A command file can also contain "@" commands to transfer control to another command file, but nesting is not implemented. That is to say that control will never return to the statement following @ filnam in a command file. Execution of a command file continues until end of file is reached, or another @ command is encountered, unless a key is struck on the keyboard, which immediately cancels the command file. Under RT-11 or TSX+, command files are limited to 1-block of 512 bytes.

To use a command file to send a control character to CP/M put the combination of a caret and the letter in the file. For example, ^G will cause a cntrl-G to be sent to CP/M. To execute the equivalent of caret-C place ^^C in the command file.

When the control program is started, it looks for a command file with the name CPM.INI on the default device (and account number). If it finds this command file, it starts executing the file without further prompting from the operator. This allows a command file to be prepared which will immediately start CP/M. For example, if CPM.INI

contained the following:

```
A: DL0:CPMDSK
LST: LP:
GO
```

then CP/M would automatically start after the control program is started.

Not all CP/M applications will operate in the manner desired when using a command file. The problem is rooted in the console character ready test. The CP/M operating system provides a service to applications programs of testing to see if a typed character is ready. Many applications programs (PIP for example) periodically test to see if a character has been typed and then take action if a character is ready. PIP aborts if a character is typed. When characters are input from a command file, instead of the keyboard, the console character ready test is not set true. If an application program reads a keyboard character, it will receive the next character in the command file. But, if it does not read the character until it gets a console character ready signal, then it will hang up, waiting. If this problem is encountered, it can be circumvented by a short CP/M program which will patch the CP/M BIOS to always return character ready.

By combining the command file facility of your DEC operating system with the command file facility of the control program, complicated sequences can be executed automatically. A key trick is to temporarily create a file named CPM.INI using commands embedded in the command file facility provided by your DEC operating system. For example, suppose that you wish to type one command to bring up CP/M word processing. This could be done as shown below. This example is for RT-11, but can be adapted to any of the operating systems.

Contents of file "WORDP.COM":

```
COPY WORD.CMD CPM.INI     (Create a file CPM.INI)
RUN CP                    (Start the CP/M control program.)
DELETE/NOQ CPM.INI        (Delete the file CPM.INI after the user has finished with CP/M)
```

Contents of file "WORD.CMD":

```
A: WDISK.CDK              (Use pseudo disk WDISK as A:)
LST: LS:                  (Use LS: as printer device.)
GO                        (Start CP/M)
WP                        (Start the word processing.)
```

With these files on the user's default device, he only needs to type:

```
.@WORDP
```

to go straight to the CP/M word processing program. The file WORDP.COM is an RT-11 command file which creates the file CPM.INI by making a copy of the file WORD.CMD. When the control program CP is run it sees the file CPM.INI and starts executing the commands which start word processing. When the user finishes and exits from the control program the file CPM.INI is deleted. Be careful with this approach if several

users on a multi-user system share the same default device and account number.

1.9 Miscellaneous Commands

The following provides descriptions of several additional commands.

The CONtinue command allows you to re-enter CP/M after escaping with a ^C command. CP/M is not re-booted and execution continues where it left off. You cannot change the disk assignments and correctly continue execution.

The SETcaret command allows you to assign a different keyboard character to have the special role of the caret. For example, to use the backslash instead:

```
-->SET \
```

This assignment can also be changed by patching the control program.

CHAPTER 2

ERROR MESSAGES
COMMON PROBLEMS

CAN'T BOOT CP/M

When you type GO you get an error message **CP/M DISK?or BIOS too large.** The problems is that the disk A: you are using does not have a CP/M system on it. Use a bootable disk as A:. You can move to system to a different disk with the CP/M utility GENSYS. If you copy CP/M format diskettes with the COPY/DEV command under RT-11 you will lose the CP/M system, because track 0 is not copied. Only DISK2 in the software distribution is bootable.

BDOS error

This message is a catch-all for any error on a disk. Usually you are trying to access a disk not defined in the setup mode of the control program. Carriage return and continue. If you get an actual hardware disk error you will be kicked back to the setup mode of the control program with additional error messages.

Other Disk Errors

The most common causes of errors are no space left on the disk or no directory space left on the disk. Application programs may not clearly tell you what the problem is. If your disk is nearly full, or the directory is nearly full, that is probably the problem. Use XDIR or STAT to see how much space is left.

Errors On Defining Pseudo Disks

When you define a pseudo disk it is opened as a read/write file under your operating system and the first block is read to check the header format. You may mistype the name, the account number, or not have proper privileges under the operating sytem to write that file.

Errors On the OPEN Command

There may be insufficent space on your mass storage device. There may be sufficent space, but not sufficent contiguous space under some operating systems. There may be no more room in the directory to create another file.

Timeout trap, Odd address trap, Bus trap, etc.

This indicates that the D100 card did not answer the bus. The card could have failed, but more likely the control program or the card is not correctly installed. Someone may have unplugged the card and taken it away.

All CP/M Cards in Use

On a multiuser system no D100 cards are available because other users are using all the cards installed. Under TSX+ if you get this message and know it is not true you should delete the file CPT.DAT when none of the cards are being used.

Terminal Lock Up

Type cntrl-Q. If that doesn't work see if the Keyboard Locked light on a VT-100 is on. Turn it off by pushing the SET-UP key twice. If that doesn't work your CP/M program may have

crashed. Try ^C to exit CP/M. If that doesn't work and you are under RT-11 it may be trying to print and the printer is turned off. Under some circumstances terminals with very fast repeat keys may cause RSX to hang up and send beeps. Try typing cntrl-C a few times.

System Crash

Under RT-11 you may crash if you attempt to do RT-11 file operations on a CP/M format diskette. This is a weakness of RT-11.

Gibberish On Terminal

You may be trying to run a program which requires the VT100 to be in ANSI mode in VT52 mode, or vice-versa. You may have the cntrl-S, cntrl-Q turned off and have the baud rate too high.

Spurious Carriage Returns Inserted

This can happen on the terminal or printer if your operating system has a maximum line width specification. Use the proper operating system command to change the specification.