



World's first
portable full-service
APL computer.
World wide
distribution/service.

NOV 7 1975

MCM/70 INSTALLATION GUIDE

Incorporating:

MCM/70 Installation Procedures

MCM/70 Printer/Plotter

MCM/APL Extended Features Documentation

MCM/APL and APL/360 Differences

and System Watch-outs

MCM/70 User's Guide Errata and Appendices

MCM/70 INSTALLATION PROCEDURES

MCM/70

Upon receipt of the MCM/70, carefully inspect for any damages which may have been sustained in transit. Notify MCM and file a claim with the carrier if there is any such damage. All equipment is carefully inspected and tested before shipped out.

A check to ensure that all options have been installed within the MCM/70 should be made. The model number on the back plate of the unit describes all options contained within it. The model number consists of three digits followed by two optional letters. The first digit should always be a 7. The second digit describes the amount of memory installed, increments of 2K. The third digit depicts the number of tape drives installed in the unit. In addition, if the letters EF also appear in the model number, it designates that the unit is equipped with the extended I/O features (ie. 782EF is an 8K 2 cassette machine with extended I/O).

INITIAL TURN ON PROCEDURE

The MCM/70 is equipped with integral rechargeable batteries used in the event of power fail. Upon receipt, the unit should be plugged into an AC outlet, the red button on the rear panel depressed and allowed to stay plugged in, unused, for approximately one hour. This will ensure that the batteries are properly charged, in the event the unit was delayed in shipping or receiving. Upon completion of this procedure the MCM/70 is ready for use. Operating instructions are provided in Chapter 2 of the MCM/70 User's Guide. In some instances the depression of "START" will not yield an immediate response. This is due to the fact that the MCM/70 will not turn on until the internal batteries are fully charged. If this occurs, by leaving the MCM/70 plugged several seconds, allowing the batteries to recharge, normal operation should resume. After a lengthy power failure sequence or a prolonged time that the MCM/70 has been unplugged, the battery drain may be more pronounced. For these cases the MCM/70 should be plugged in for a period of five minutes before attempting to restart. If the MCM/70 is left plugged in, no significant delay should be seen in response to a start key, because at any time that the unit is plugged in the batteries are constantly being recharged.

POWER REQUIREMENTS

The MCM/70 has been designed to operate from a nominal line voltage of 115 volts AC. The line frequency may be either 50hz or 60hz. In addition, the MCM/70 may be operated off a DC voltage supply rated at 14.5 volts plus or minus .5 volts at 5 amps. A two pronged male plug accessible at the rear of the unit is provided for this input.

FUSES

The MCM/70 has three fuses, two of which are located at the rear panel. The top fuse is a 5 amp slow blow. It protects the batteries which are mounted internal to the MCM/70. The bottom fuse also rated at 5 amp slow blow protects the external DC input. An internal AC 1 amp slow blow fuse is located in the power supply assembly.

Note: ALWAYS DISCONNECT THE POWER CABLE FROM THE MCM/70 BEFORE CHECKING ANY OF THE FUSES. In addition, factory permission must be obtained in order to remove the cover to inspect or replace the AC fuse. Under normal, operating conditions the AC line fuse should not open.

PRINTER/PLOTTER

The printer is shipped in two separate boxes. One containing the actual printer assembly and the other the base with power supply and interface card. The following is a list of instructions for the assembly and installation of the MCP-132 Printer-Plotter;

- o Visually inspect both cartons for any physical damage and follow the same procedure as outlined for the MCM/70.
- o Remove the printer assembly and the base stand assembly from their respective cartons.
- o You will note that the printer is mounted to a large plywood board. Remove 4 bottom screws from the board which will separate the printer from the bottom board. In doing this be careful where you set the printer down because it has 4 mounting studs protruding from the bottom.
- o Position the printer onto the base, such that the front of the printer is towards the casters that protrude the most, and the four studs go through four mounting holes on the top plate of the base.
- o Fasten the printer to the base using the hardware supplied in an envelope taped to the front of the printer.
- o Remove two back screws of the printer base and open this door.
- o You will note mounted to the lower portion of the base is a power supply, which is used in powering the printer. Remove the right most screw from the power supply.
- o Swing the power supply assembly towards the back and down of the base assembly (power supply is hinged to the base).
- o Locate the power cable from the printer (blue plastic connector with seven grey wires), connect this cable to the left end of the power supply. This connector is keyed and therefore will only seat properly in one direction. Make certain that it is driven all the way in and seated properly. In doing this note that there is a switch in the proximity of this connector and do not activate it (this is an internal on/off switch for the power supply).
- o Locate the ribbon cable from the printer assembly. This connector goes to the printed circuit card which is mounted on the door of the base assembly. In mating the connector make certain that the key on the connector is positioned in the slot of the printed circuit card edge finger connector.
- o Swing the power supply back up to its upright position and fasten with the mounting screw.
- o Carefully dress cables through cutout on left most top of the back door. In addition to the two cables from the printer assembly, another cable (omniport interface) is connected to the

interface card. Unwind and extend out through the same access hole, as the other two cables. Close back door and fasten shut being careful of the position of the cables coming through the back door.

- o Place paper tray through guide holes under the printer and fasten with two #10 - 32 screws as provided.
- o Open top cover of the printer and remove shipping ties from printed head assembly and platen (further instructions for this operation may be found in the Printer Maintenance Manual).
- o Mount print wheel and ribbon to the printer mechanism making certain that the print head is locked back into position.
- o Making certain that the MCM/70 is off, connect the ribbon cable from the back door of the printer base to the back of the MCM/70 on the omniport connector. This connector is keyed and will only seat properly in one direction.
- o If unit supplied is equipped with the optional forms tractor, it should be installed at this time.
- o Making certain that the power switch, located on the upper right hand corner of the printer base panel is in the down (off) position, the AC plug from the printer assembly may be connected to any standard three prong 115 volt AC outlet.
- o Paper should be loaded into the printer at this time.
- o Turn printer on with front panel switch, turn the MCM 70 on (printer should reset at this time).
- o The printer and MCM/70 will now be ready for any user instructions to be performed. For example: to select the printer at the output device simply type `□OU 1` followed by carriage return. A response of "1 66 241" should appear on the display which denotes that the printer has been selected and status is proper for output to begin.

For a complete and detailed description on the operation of the printer/plotter in conjunction of the MCM/70 refer to the Hytype/Qume interface document as well as the omniport system access.

DYADIC FORMAT $R \leftarrow A \nabla B$

Dyadic Format gives the user control of the appearance of numeric output. When output is displayed on the system display, columns of arrays are not aligned but instead, numbers are displayed separated by one space. Integers print without a decimal point and large numbers print in E format. If output is diverted to a printer (see Communications Subsystem), output is formatted using MONADIC FORMAT described below.

Format accomplishes this control by generating a character array from a numeric array in a prescribed (or generated) format. The general form of dyadic format is:

$$R \leftarrow A \nabla B$$

where B is the numeric data (scalar vector or array) to be formatted, A is the control scalar, vector or array describing the format, and R is the resulting character data (vector or array). The elements of A form triplets: the first is to control the field WIDTH, the next to control PRECISION (or signfirst digits), and the last to control EXPONENT INTERVAL.

It is possible to control the appearance of R on a column by column basis, alternatively all columns can be controlled by one control triplet. If all columns are to be treated equally, A must be:

1. A scalar (or one element vector) specifying field WIDTH only.
2. A 2 element vector specifying WIDTH and PRECISION.
3. A 3 element vector specifying WIDTH, PRECISION and EXPONENT INTERVAL.

If each column of B (or element of B if B is a vector) is to be formatted separately, A must be an array where the number of rows of A is equal to the number of columns of B (or the number of elements in B if B is a vector). Each Row of A forms a control field for each column of B. As above the user can specify WIDTH only, WIDTH and PRECISION, or all three control fields for each column of B.

The number of columns in A is:

1. One if WIDTH only is specified for all columns.
2. Two if WIDTH and PRECISION are specified.
3. Three if WIDTH, PRECISION and EXPONENT INTERVAL are specified.

The system automatically inserts default values if the user does not specify all elements of A. Below are listed the default values, along with the resulting action taken by the format operator.

PARAMETER	DEFAULT	ACTION
WIDTH	0	The minimum width is calculated by Format.* One is added to each column (except the first) to ensure one space between columns.
PRECISION	0	Format uses the value in $\square PP$.
EXPONENT INTERVAL	-3	When the exponent interval is negative, the system tries F - format first.* If this fails it defaults to E format where the exponent will be a multiple of three.

NOTE;* This is on a column Basis.

MONADIC FORMAT $R \leftarrow \nabla B$

Monadic Format is identical to Dyadic Format except that the system inserts all three default values above for A. If B is alphabetic, monadic format is equivalent to $R \leftarrow B$. Therefore monadic format can be used to ensure a value is alphabetic.

ERRORS

The table below describes the errors issued by format, the causes of these errors and the corrective action required.

ERROR	CAUSES	CORRECTIVE ACTION
LENGTH ERROR	1. Number of Rows of A not equal to number of columns of B.	Correct shape of A.
	2. Sum of field widths >255.	Correct Width field(s)
	3. Number of columns of A is not 1, 2 or 3.	Correct shape of A.
RANGE ERROR	1. Field <u>Width</u> too small.	Increase Width or decrease $\square PP$.
	2. F formatting requested (Exponent interval $\neq 0$) but E format necessary.	If Exponent interval is set to $\bar{3}$ instead of 0, then format will use F format where possible.
	* 3. More significant digits are requested than the system can produce	Decrease precision or significant digits.
DOMAIN ERROR **	1. A is not numeric.	Correct A.
	2. Exponent interval is greater than significant digits (or $\square PP$ if significant digits $\neq 0$).	Decrease exponent interval or increase precision.
	3. B is not numeric (Dyadic format only).	Correct B.
	4. Width W not in the range $0 \leq W \leq 255$.	Correct Width.
	5. Exponent interval E not in the range $\bar{16} \leq E \leq 16$.	Correct Exponent Interval.

NOTE: * If significant digits is less than zero, trailing zero suppression is performed on a column basis.

** If the exponent interval E is negative and the absolute value of E is greater than significant digits, the system sets E to $\bar{1}$ rather than issuing a DOMAIN ERROR.

EXAMPLE 1:

B+3 4p1.006 2 3 4 5 6.3333 7 8 9 10 11.44 12

B•'DIRECT OUTPUT THROUGH COMMUNICATIONS SUBSYSTEM.'

1.006	2.0000	3.00	4
5.000	6.3333	7.00	8
9.000	10.0000	11.44	12

▼B•'MONADIC FORMAT. (SAME AS ABOVE)'

1.006	2.0000	3.00	4
5.000	6.3333	7.00	8
9.000	10.0000	11.44	12

10▼B•'WIDTH OF 10 FOR ALL COLUMNS.'

1.006	2.0000	3.00	4
5.000	6.3333	7.00	8
9.000	10.0000	11.44	12

(4 1p10 0 8 3)▼B•'WIDTH DIFFERENT FOR EACH COLUMN.'

1.006	2.0000	3.00	4
5.000	6.3333	7.00	8
9.000	10.0000	11.44	12

10⁻⁴▼B•'WIDTH OF 10 AND AT LEAST 4 SIGNIFICANT DIGITS.'

1.006	2.000E0	3.00	4
5.000	6.333E0	7.00	8
9.000	10.00E0	11.44	12

10 3 3▼B•'FORCE E-FORMAT WITH EXPONENT INTERVAL 3.'

1.01E0	2.00E0	3.00E0	4.00E0
5.00E0	6.33E0	7.00E0	8.00E0
9.00E0	10.0E0	11.4E0	12.0E0

(4 3p10 0⁻³ 9 3 3 8 0⁻³ 0 0⁻³)▼B•'FORCE E-FORMAT IN COLUMN 2.'

1.006	2.00E0	3.00	4
5.000	6.33E0	7.00	8
9.000	10.0E0	11.44	12

EXAMPLE 2.

Print a table consisting of Names, ages and salaries for 5 employees:

NAME←5 6p'SMITH JONES SAXBY WOODS MOORE '

AGE←5 1p27 35 19 31 50

SALARY←5 1p15000 17291 8342 11291 30000

NAME,(7 0 0▽AGE),10 2 0▽SALARY

SMITH	27	15000.00
JONES	35	17291.00
SAXBY	19	8342.00
WOODS	31	11291.00
MOORE	50	30000.00

- OR -

NAME,(2 3p7 0 0 10 2 0)▽AGE,SALARY

SMITH	27	15000.00
JONES	35	17291.00
SAXBY	19	8342.00
WOODS	31	11291.00
MOORE	50	30000.00

EXAMPLE 3.

E - FORMAT

B←3 4p1 2 3E⁻² 4E20.5 6E⁻⁷ 7 8E⁻⁹ 9 10.33E10 11E⁻⁵ 12

▼B

1	2.000E00	30E ⁻³	400E18
5	600.0E ⁻⁹	7E00	8E ⁻⁹
9	103.3E09	110E ⁻⁶	12E00

Since the decimal point has less significance than the exponent E, vertical alignment is keyed on 'E' making columns much easier to read. Also note the exponents are all multiples of 3, (the default value for exponent interval is -3) making conversion to familiar units (miliamps, microfarads, thousands, etc.) unnecessary.

0 0 ⁻¹▼B

1	2.000E00	3.0E ⁻²	4.0E20
5	6.000E ⁻⁷	7.0E00	8.0E ⁻⁹
9	1.033E11	1.1E ⁻⁴	1.2E01

When exponent interval is set to ⁻¹, decimal points and 'E' are vertically aligned.

FORMAL DEFINITION

FORMAT:

- Syntax:** $R \leftarrow \nabla B$ (Monadic)
 $R \leftarrow A \nabla B$ (Dyadic)
- Domain:** No Restriction (Monadic)
B must be numeric (Dyadic)
A must be integer
- Conformability:** $1 \leq \rho B$
 $(\rho A) \in 1 \ 2$
 $(\bar{1} \uparrow \rho A) \leq 3$
- If $2 = \rho A$, then: $(1 \uparrow \rho A) = \bar{1} \uparrow \rho B$
If $(1 = \rho A)$, then
A is treated as if it were:
 $A \leftarrow ((\bar{1} \uparrow \rho B), \rho A) \rho A$
If $(\bar{1} \uparrow \rho A) < 3$, then A is extended as follows:
 $M \leftarrow 3 - \bar{1} \uparrow \rho A$
 $A \leftarrow A, ((\bar{1} \uparrow \rho A), M) \rho (-M) \uparrow 0 \ 0 \ \bar{3}$
- Result:** R is the character representation of B,
formatted according to the specification A.
If B is character, then for Monadic format:
 $R \leftarrow B$
If B is numeric, then:
 $(\rho R) = (\bar{1} \uparrow \rho B), W$
where W is the width required to format B.
- Operation:** The left argument to the format function is
used as a format specification for the
result. For the purposes of the formatting
operation, the right argument is created as
if it were:
 $B1 \leftarrow ((\times / \bar{1} \uparrow \rho B), \bar{1} \uparrow \rho B) \rho B$
All formatting is done on the basis of the
columns of B1.
- Each row of A is a format specification
for a Column of B1. All members within a
Column of B1 are formatted uniformly to
form a matrix. The result is formed by
catenating these matrices together (along the
last dimension), and giving it the appropriate
shape.

Operation:
(cont.)

Each column format specification (i.e., row of A) has three elements, defined as follows:

$A[I;] \leftrightarrow W, P, E$

Width:

Range $0 \leq W \leq 255$

W is the width requested for Column I .
The formatted $B1[;I]$ will be right justified in a blank field of width W .

If any member of $B1[;I]$ cannot be formatted in the width W , a RANGE ERROR is issued.

If $W=0$, the column width will be exactly 1 + that required by P, E , and $B1[;I]$ (except for the first column, for which the exact width is used).

Precision:

Range $-127 \leq P \leq 127$

P is the precision requested for Column I .
If F - format is requested (see E below), P is the number of decimal places requested. Otherwise, it is the maximum number of digits requested.

If $P < 0$, trailing zero suppression in the fraction part is performed, if it can be done uniformly throughout the column.

If $P=0$, and F - format is not requested, the column is formatted as if P were $-\square PP$.

Exponent
Interval:

Range $-16 \leq E \leq 16$

E is the exponent interval requested for column I . For columns printed in E -format, the exponent following the 'E' symbol is calculated as follows:

$XI \leftarrow [10 \otimes B[;I]$

$EXPI \leftarrow XI - (|EI|) | XI$

Thus the exponent is always a multiple of E .

If $E=0$, the exponent is always zero, so F -format is implied.

If $E < 0$, F - format is attempted, and if it fails (for reasons discussed below), E - format is used. In either case, P is the maximum number of digits requested.

If $E \neq 0$, then P must be $\geq E$.

If $E < 0$, and $P < E$, E is assumed to have the value -1.

FORMATTING RULES:

A number formatted in F - format has the following form:

$\bar{\quad}$ <Integer Part> . <Fraction Part>

where fields in braces are present only if required.

If the number is negative, the neg sign is present.

<Integer Part> and <Fraction Part> each consist of a string of one or more decimal digits.

<Integer Part> never contains leading zeros, unless it consists of exactly one zero.

If <Fraction Part> is absent throughout a column, the decimal point is deleted.

In normal F - format ($E=0$, or $E<0$ and $P\leq 0$), the decimal points of all members of a column are vertically aligned.

A number formatted in E - format has the following form:

$\bar{\quad}$ <Integer Part> . <Fraction Part> E $\bar{\quad}$ <Exponent>
where fields to the left of the 'E' are described above.

A neg sign follows the 'E' if the exponent is negative.

The 'E' symbols of all members of a column are vertically aligned.

<Exponent> is a string of one or more decimal digits right justified in a field of zeroes. The least significant digit of the exponent of all members of a column are vertically aligned.

The sum of the number of digits in <Integer Part> and <Fraction Part> is the same for all members of a column (note that this means that in general the decimal points are not aligned).

If <Fraction Part> is absent throughout a column, the decimal point is deleted.

In compact F - Format ($E<0$ and $P>0$), the above rules are used, and if the exponent is zero throughout the column, the entire E - field is deleted.

FORMATTING RULES:
(cont.)

If $E < 0$ and $P < 0$, a normal F - format is attempted, using P as a digit count, and trailing zero suppression is performed. If any member, for alignment reasons, requires more than $|P|$ digits, the entire column is formatted in E - format. Similarly, E - format is used if any member is less than $10^{-3}|EI|$.

Rounding is done as follows:

F - format: $NI \leftarrow 10^{|PI|}$
 $BI \leftarrow (\lfloor 0.5 + B[;I] \times NI \rfloor) \div NI$

E - format: $XI \leftarrow \lfloor 10^{\oplus B[;I]} \rfloor$
 $EXPI \leftarrow XI - (|EI| | XI$
 $NI \leftarrow 10^*(|PI|) - EXPI$
 $BI \leftarrow (\lfloor 0.5 + B[;I] \times NI \rfloor) \div NI$

MONADIC FORMAT:

B is numeric, the following format specification is used:

$A \leftarrow 0 \ 0 \ -3$

CANONICAL REPRESENTATION ($\square CR$)

Canonical Representation converts a function into a character array. This is used when a listing of the functions is required ($\square + \square CR$ 'NAME') or if a large function is to be broken into smaller functions.

In $R + \square CR A$, A is a character vector (or Scalar) containing the name of the function to be converted. If A is not character, a DOMAIN ERROR is issued, if A is not a Vector (or Scalar) a RANK ERROR is issued. The result R is a character array containing one row for each line of the function including the header line, and sufficient columns to contain the largest function line. The bracketed line numbers do not appear in the result. If the item named in A is not a function, R is a zero character array.

EXAMPLE:

```
V Z+MAGICSQ X;M
[1] +(0≠2TX)/3
[2] →0•□+'NOT ODD ORDER'
[3] M+(X,X)ρ1X×X
[4] Z+(⌈X÷2)φ(⌈1+1X)θ(⌈1+1X)φM
V
```

```
□+M+□CR'MAGICSQ'
VZ+MAGICSQ X;M
→(0≠2TX)/3
→0•□+'NOT ODD ORDER'
M+(X,X)ρ1X×X
Z+(⌈X÷2)φ(⌈1+1X)θ(⌈1+1X)φM
```

```
□+ρM
5 27
```

```
□+MAGICSQ 3
8 1 6
3 5 7
4 9 2
```

FUNCTION ESTABLISHMENT (□FX)

This system function attempts to transform a character matrix into an executable function. Therefore, □FX can be considered to be the converse of □CR. The format is:

R←□FX X

where 'X' is the name of the character array to be converted, and 'R' is the name of the function established. A RANGE ERROR is issued if a function cannot be established from the data X, or the name of the function to be established exists and is not a non suspended function.

EXAMPLE:

7000 LINES

```
      □←ASORT
VR←N SORT A;ALP
ASORT ON FIRST N COLUMNS
ALP←' ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
R←A
K←R[(AALP;R[;N]);]
→(0≠N←N-1)/3
```

```
      □←pASORT
6 44
```

```
      □←□FX ASORT
SOR
```

```
      □←p□←AA
SMITH
ABLE
JONES
PEARCE
4 8
```

```
      □←6 SORT AA
ABLE
JONES
PEARCE
SMITH
```

NAME CLASSIFICATION ($\square NC$)

The monadic function $\square NC$ accepts a character matrix of names, and returns a numerical indication of the class of the name represented by each row. In $\square NC X$, the argument X may also be a single name presented as a vector or scalar. A DOMAIN ERROR is issued if X is not character. The table below shows the meaning of the numerical result.

<u>RESULT</u>	<u>NAME USAGE</u>
0	No Value
1	Label
2	Variable
3	Function
4	System or invalid Names
5	Other ($\square Z2^{FW}$ OR TRANSLATE TABLE FOR (OHR))

EXECUTE SYSTEM FUNCTION (R←X □ZZ S)

Special functions can be implemented in the MCM/70 to perform tasks which are impractical or impossible to implement as standard APL functions. These special functions (S) can be read from tape(□XR), written on to tape (□XW), expunged (□EX) but any attempt to access these through APL other than via the system function □ZZ results in a DOMAIN ERROR.

□NC 'S' returns a 5. A SYNTAX ERROR is issued if S is not executable data.

In the above statement S is the Executable System Programme Data, X (if required) is an argument, and R (if produced) is the Result from executing S. The executable data S is somewhat analogous to a locked function, except that these can only be created by MCM. This facility allows MCM to create custom software (special functions and input-output drivers) for the MCM/70.

DIFFERENCES BETWEEN MCM/APL AND APL/360

PRIMITIVE OPERATOR DIFFERENCES

1. QUAD QUOTE INPUT (\square) IS IMPLEMENTED AS A MONADIC OR DYADIC FUNCTION. THE RIGHT ARGUMENT IS A PROMPT DISPLAYED WHEN THE KEYBOARD IS UNLOCKED. THE LEFT ARGUMENT IS USED TO POSITION THE CURSOR. IF IT IS ABSENT, THE CURSOR IS POSITIONED AFTER THE PROMPT. \square ' IS EQUIVALENT TO APL\360 \square .

EXAMPLE: $3\square'1+ =5'$ THE KEYBOARD UNLOCKS WITH THE CURSOR IN POSITION 3, THE BLANK. THE RESULT IS THE ENTIRE CHARACTER STRING.

2. DYADIC FORMAT IS EXTENDED. IN $A\vee B$, THE ELEMENTS OF A FORM TRIPLETS REPRESENTING WIDTH, DIGITS AND EXPONENT INTERVAL (E) FOR EACH COLUMN OF B. IF $E=0$, F - FORMAT IS IMPLIED. IF $E>0$, THE EXPONENTS FOR THIS COLUMN WILL BE MULTIPLES OF E. IF $E<0$, F - FORMATTING IS ATTEMPTED. IF THIS FAILS, E FORMATTING IS APPLIED AS ABOVE.
3. EXECUTE ($R\leftarrow C$) IS IMPLEMENTED. BRANCH (\rightarrow) IS NOT ALLOWED WITHIN C.
4. THE NULL OPERATOR ($R\leftarrow A\circ B$) IS EQUIVALENT TO $R\leftarrow A$. THEREFORE NULL CAN BE USED TO GROUP INDEPENDENT CALCULATIONS ON A SINGLE LINE.
- ✓ 5. SCAN EXECUTES FROM RIGHT TO LEFT. FOR EXAMPLE, $+\backslash 1\ 2\ 3$ PRODUCES 6 5 3. $\phi+\backslash\phi N$ (GENERAL CASE $\phi[I]+\backslash[I]\phi[I]N$) IS EQUIVALENT TO APL.SV $+\backslash N$.
6. MATRIX DIVISION / INVERSE (\oplus) IS NOT IMPLEMENTED AS A PRIMITIVE OPERATOR.
- ✓ 7. TAKE(\dagger) WITH A RESULT LARGER THAN THE ORIGINAL ARGUMENT IS NOT IMPLEMENTED
8. IN FACTORIAL ($!B$) AND COMBINATIONS ($A!B$), A AND B MUST BE INTEGERS.
9. THE MAXIMUM RANGE OF QUERY ($?A$) IS 32767.
10. $0\div 0$ AND $0\ast 0$ PRODUCE DOMAIN ERRORS.
- ✓ 11. NEGATIVE VALUES CANNOT BE RAISED TO NON INTEGER POWERS.
12. LAMINATE ($A,[1.1]B$) IS NOT IMPLEMENTED.
13. ONE ELEMENT LITERAL CONSTANTS AND \square RETURN ONE ELEMENT VECTORS.

ONLY A SCALAR OR ANOTHER MATRIX CAN CATEWATE TO A MATRIX

SYSTEM DIFFERENCES

1. SYSTEM COMMANDS ARE REPLACED BY SYSTEM VARIABLES AND FUNCTIONS SIMILAR TO THOSE IN APL.SV. NOTE: THE DEFAULT VALUE FOR \square PP IS 5.
2. A FILE SUBSYSTEM, EXTERNAL ALLOCATION SYSTEM (EASY) ALLOWS THE USER TO READ AND WRITE DATA AND FUNCTIONS ON THE FILES.
3. A VIRTUAL SYSTEM (AVS) EXTENDS THE AVAILABLE MEMORY BY AUTOMATICALLY ROLLING DATA AND FUNCTIONS IN AND OUT OF MEMORY. IT ALSO ENABLES AUTOMATIC POWER FAIL AND RESTART.
4. A COMMUNICATIONS SUBSYSTEM IS IMPLEMENTED TO HANDLE ALL EXPLICIT INPUT AND OUTPUT (\square AND \square) FOR THE APL SYSTEM. SIMPLE COMMANDS ARE USED TO DIVERT THE INPUT AND/OR OUTPUT TO ANY DEVICE EQUIPPED WITH AN RS-232C INTERFACE. OUTPUT CAN ALSO BE DIVERTED TO THE MCP132 PRINTER OR OTHER MCM DIRECTLY SUPPORTED DEVICES.
- ✓ 5. EACH DIMENSION IS LIMITED TO 255 ELEMENTS. RANK IS LIMITED TO 32. BECAUSE OF THIS, THE DOMAIN OF INDEXING PRIMITIVES (\downarrow , \uparrow ETC) IS FROM 0 TO 255.
- ✓ 6. THE SYSTEM WILL GENERATE VARIABLES AND FUNCTIONS WHICH CONTAIN MORE THAN 3968 BYTES. HOWEVER AVS AND EASY CANNOT WRITE ITEMS CONTAINING MORE THAN 3968 BYTES.
7. FUNCTION EDITING IS ENHANCED BY THE USE OF INSERT (CONTROL SPACE) AND DELETE (CONTROL BACKSPACE). FUNCTION DEFINITION CAN BE OPENED IN QUAD INPUT. FUNCTION LINES CAN BE DELETED BY INTERRUPT (CONTROL SHIFT ARROW) NOTE: CONTROL (AND SHIFT) KEYS MUST BE PRESSED SIMULTANEOUSLY WITH THE SPACE, BACKSPACE OR ARROW (\rightarrow) KEY.
- ✓ 8. NAMES CAN BE OF ANY LENGTH BUT THE SYSTEM RECOGNIZES ONLY THE FIRST THREE CHARACTERS.

SYSTEM WATCH-OUTS

MCM/APL RELEASE 1.1

- ✓ 1. IF AN ARRAY A IS CREATED SUCH THAT $(x/\rho A) > 8191$, A DAMAGED WORKSPACE MAY RESULT.
2. IF A FUNCTION IS INTERRUPTED AT THE END OF A LINE, THE LINE NUMBER DISPLAYED IS THE NUMBER OF THE FOLLOWING LINE.
3. VALUES REPLACED INTO SYSTEM VARIABLES (IE. $\square PP$, $\square PT$) ARE NOT CHECKED CORRECTLY WITH RESPECT TO SHAPE AND DOMAIN. *OR TYPE*
- ✓ 4. WHEN A REQUEST FOR QUAD PRIME INPUT (\square) IS INTERRUPTED, IF THE '+' KEY IS RELEASED BEFORE THE CONTROL KEY, THE SYSTEM WILL WAIT FOR ANOTHER KEY TO BE PRESSED BEFORE CONTINUING. RETURN MUST BE PRESSED TWICE IN RESPONSE TO THIS INTERRUPT.
5. FUNCTION LINE LABELS MAY NOT HAVE SPACES BETWEEN THE NAME AND THE COLON. DUPLICATE LINE LABELS DO NOT PRODUCE A DEFINITION ERROR.
- ✓ 6. IF THE RESULT OF A FLOATING POINT CALCULATION IS MUCH GREATER THEN $7E75$, A RANGE ERROR IS NOT ISSUED AND THE RESULT IS ZERO. $7E75 \neq 7E75 \leftarrow \rightarrow 1$
- ✓ 7. IN INDEXED REPLACE ($A[B]+C$), AVAILABLE WORKSPACE IS REDUCED IF: 1. A AND B ARE VECTORS AND C IS A SCALAR. 2. AN INTERRUPT OR ERROR OCCUR DURING EXECUTION. *IN ARRAY*
- ✓ 8. WHEN AVS IS ACTIVE AND A FUNCTION F IS CURRENTLY NOT IN THE WORKSPACE, $F+X$ DOES NOT PRODUCE A SYNTAX ERROR. THE FUNCTION F IS REPLACED BY X.
- ✓ 9. IF AVS IS ACTIVE AND AN ITEM WHOSE NAME IS IN THE ACTIVE GROUP IS READ EXPLICITLY FROM ANOTHER GROUP, THE AVAILABLE WORKSPACE IS REDUCED.
- ✓ 10. THE STATEMENT $N \square XR'N'$ PRODUCES UNPREDICTABLE RESULTS. THE WORKSPACE COULD BE DAMAGED.
- PLS STAYS PROBS* 11. IF A WORKSPACE FULL ERROR OCCURS DURING STARTUP OF FUNCTION EXECUTION, A ZERO MAY BE INSERTED IN THE $\square LC$ LIST.
- 7? 12. IN THE SEQUENCE $A F B$, IN EVALUATING A, IF AVS ROLLS F OUT OR F IS MOVED BY GARBAGE COLLECT, THE RESULT IS UNPREDICTABLE AND WORKSPACE DAMAGE MAY OCCUR.
13. WHEN A WORKSPACE FULL ERROR OCCURS, TRY RESUMING EXECUTION ($\rightarrow \square LC$) BEFORE OTHER CORRECTIVE ACTION.
14. IF A COMMENT IS EXECUTED ($\square 'Axxxxx'$), THE KEYBOARD IS UNLOCKED FOR INPUT
15. WHEN EDITING A SUSPENDED FUNCTION, THE HEADER LINE CAN BE DISPLAYED BUT CANNOT BE ALTERED IN ANY WAY. TO ESCAPE FROM THE LINE 0 DISPLAY, ENTER TO DISPLAY ANY OTHER LINE.
16. DO NOT LOCALIZE AN OBJECT WITH THE SAME NAME AS A FUNCTION. ^{??}
17. DON'T TRY TO TERMINATE FN TOO SOON AFTER $\rightarrow \square LC$ (WITH CTRL)
18. PLACE IN FILE CHECK TO ADDRESS D.E. (10/1/04)

MCM/70 USER'S GUIDE

ERRATA

- Pg. 8 . 9 (Defined Functions: Line Modification)
- Insert at the bottom of the page:

"Review the section on Character Insertion and Deletion, pg. 2.9. The same procedure works in function line modification."

- Pg. 8 .11 (Defined Functions: Line Deletion)
- Change lines 1 - 7 to read as follows:

"hold the CTRL key and SHIFT key down and press the "X" key. (N.B. This must be done as the first action in response to the line display. At any other time, the CTRL key will be ignored, causing a "X" to be entered in the line.) The system will then respond by displaying the next line:

```
[3](+/X):+/X-X
```

The function may then be closed by typing a del followed by a RETURN as described previously. After this is done and the function is displayed, you will see it looks like this: "

- Pg. 8 .10 (Defined Functions: Line Modification)
- Insert the following at the end of the section:

"Note that any modification which results in a redirection (line no.) or a request to close definition (del) at the beginning of the line, will result in the indicated action taking place without any modification to the current line. For example, if we had inserted a line number in front of the line as follows:

```
[3][3](+/X):+/X-X
```

The system will simply redisplay the line unmodified:

```
[3](+/X)#N
```

"

MCM/70 USER'S GUIDE

Appendix B Supplement

TAPE ERROR RECOVERY PROCEDURE

In the event of a "Tape Error" message in the MCM/70, several alternative actions may be taken in order to fully recover or minimize the amount of data which may be lost. Whenever a tape error message is displayed, the user must first depress the return key in order to determine whether the error occurred during a Read or a Write operation. The recovery procedure in the event of a Write error is simply to try the same operation over again. The second try should lead to a successful write operation because the tape operating system maintains a list of blocks which it has had difficulty in writing into. In subsequent tries, it does not write into any blocks that have previously resulted in tape errors.

Read errors are much more difficult to recover from, since the data is obviously not available to do anything else with. Whenever a tape error occurs during a read operation the following steps should be taken:

1. Retry reading explicitly the objects of interest several times.
2. If step one is not successful and no current backup tape is available read as much as possible from the tape with the read error into memory and explicitly write out onto the other tape drive.
3. Close out the tape that was just written into (ie. XF[2] 10).
4. Try closing out the tape which gave you the read error by using XF 10, WC, OR OFF.
5. If, and only if, the above also yields a tape error message, remove the two tapes, then depress the red button on the back of the MCM/70.

At this point the tape with the read error may be reinitialized and reused but make a note of the error and use cautiously. The other tape which was written on may be used but do not forget that back up tapes should always be made. ✓



MICRO COMPUTER
MACHINES INC

BOX 310
133 DALTON STREET
KINGSTON
ONTARIO CANADA
K7L 4W2
TEL 613/544 9860

NOV 7 1975

OMNIPOINT SYSTEM ACCESS

Preliminary System Documentation

July 7, 1975.

Copyright 1975

By

Micro Computer Machines Inc.

All Rights Reserved

The information and/or drawings set forth in this document are copyrighted material and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein are the exclusive property of Micro Computer Machines Inc.

No disclosure of the information or drawings shall be made to any other person or organization without the prior written consent of Micro Computer Machines Inc.

Reproduction of the information or drawings or use in any manner not expressly intended by this release is specifically forbidden without the prior written permission of Micro Computer Machines Inc.

Micro Computer Machines reserves the right to make changes at any time to improve design and supply the best product possible.

Preliminary information. Supplementary information may be issued at a later date.

OMNIPOINT SYSTEM ACCESS

The Omniport is an 8-bit parallel interface between the main system and all peripheral devices. In order to understand the nature of the interface itself, the reader should refer to the document on the Standard Omniport Interface.

I - Direct Device handling

Basically, six operations are available:

- 1) Address device
- 2) Read device code
- 3) Send command to device
- 4) Read device status
- 5) Send data to device
- 6) Read device data

The function of these operations is achieved with four APL system functions - [OU], [IN], [BC], [BT].

[OU], [IN] - These functions are identical in operation except that [OU] (short for [OUT]) sets the system SINK, while [IN] sets the system SOURCE. The purpose of the SINK and SOURCE are described further on in this document.

Syntax: $R \leftarrow \left\{ \begin{array}{l} [IN] \\ [OU] \end{array} \right\} B$

Conformability: $1 \leftrightarrow \rho\rho B$
 $2 \geq \rho B$

Domain: $0 \leq B \leq 255$

Result: $1 \leftrightarrow \rho\rho R$
 $3 = \rho R$

Operation: The current SOURCE/SINK is set to the device whose interface address is $B[1]$. If B is empty, the SOURCE/SINK device is left unchanged. The device is selected, and the answer-back code from it is returned in $R[2]^*$. $R[1]$ contains the device address. $R[3]$ contains the current device status. If $B[2]$ is present, it is output as a command to the device.

* In communicating with the omniport, the 8 bits of the omniport are represented in the system by their base 2 value. Bit 7 on the omniport is the most significant bit.

WRITE - These functions are used to output and input data via the omniport.

Syntax: $R \rightarrow [WB] B$

Conformability: $0 \leq B \leq 255$

Domain: $0 \leq B \leq 255$

Result: $R \rightarrow B$

Operation: B is output as data to the current SINK device.

Syntax: $R \leftarrow [W]$

Result: $0 \leq R \leq 255$

Operation: R is data from the current SOURCE

Device Addresses -

Peripheral devices can be assigned any address in the range 1 - 199. The address 0 is reserved for the MASTER SOURCE (namely, the system keyboard) and the MASTER SINK (the system display).

Addresses in the range 200 - 255 are reserved for other system devices. Currently, the only ones used are 200 - 207, which are the system tape drives. These drives are used by EASY. The left hand system drive, which is drive 1 (in origin 1) is address 200. This drive is used by AVS. The other system drive has address 201. External tape drives may be attached to the Omniport and assigned addresses 202 - 207.

Any attempt to output a command to a system device which is in use by AVS or EASY will result in an I/O ERROR.

Device Answer-Back Codes -

The answer-back code of a device is made up as follows - let ABC be the code received, and let:

$$C = \begin{matrix} 7 & 6 & 5 & 3-0 \end{matrix} \text{ ABC}$$

Then the elements of C are interpreted as follows:

Element	ABC bit	Meaning
C[1]	7	1 = Input Device
C[2]	6	1 = Output Device
C[3]	5	1 = Interactive (i.e. a terminal)
C[4]	3 - 0	Device Type

The following device types have so far been assigned:

Type	Device
0	EIA/ASCII terminal
1	MCM Cassette drive
2	MCM/Hytype printer
3	MCM/True Data Card reader
4	MCM Dot Matrix printer
5	EIA/APL terminal

Command and Status Codes -

These codes are peculiar to the device being accessed. Refer to the documentation for the device in question.

If no device is present, the answer-back code, status, and data will all be zero.

II - Line I/O

Most program I/O in APL is line I/O; that is, I/O is done with lines of APL characters. This type of I/O occurs in two forms - implicit and explicit.

Whenever the system has nothing to do, it waits for input from the user. This is an implicit input request. Implicit input is also requested by the system when it is in open function definition mode. Further, if the system has the result of a calculation and has not been told what to do with it, it outputs it. This is an implicit output operation. Implicit output also occurs when an error message is issued.

The functions [] and [!] are explicit requests for input. The function [] is an explicit output command.

All implicit I/O is done through the system MASTER SOURCE and MASTER SINK, namely, the keyboard and display unit. Explicit I/O, however, is done through the current SOURCE and SINK devices. These are initially set to be the MASTER SOURCE and SINK, but may be diverted to other devices using []IN and []OU as described under those functions.

If the SOURCE or SINK has been diverted, the system will attempt to perform the requested I/O using the requested device. The following device types are currently supported.

- Type 2: MCM/Hytype printer (output only)
- Type 3: MCM/True Data Card reader (input only)

Error Messages -

If the system cannot perform the requested I/O, it will issue an I/O ERROR, followed by a code. The meaning of the codes is as follows:

- 01 - No device found with the specified address
- 02 - Output requested to non-output device
- 03 - Device not supported for line I/O.
- 04 - Input requested from non-input device.
- 05 - I/O requested for device in use by AVS/EASY

In addition, each device handler may issue its own error. Refer to documentation for the handler in question.

67 □
40 ⊕
42 ⊖
39 ⊗
53 △
54 ▽
68 □
54 ⊕
65 ⊕
68 ⊕
66 ⊖
41 ×
44 ×
32 ×
33 ×
30 |
45 ⊗

CONVERSION TABLE

DECIMAL/HEX/APL10/

INTERNAL
HEX
BINARY

00	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
20	22	24	26	28	2A	2C	2E	30	32	34	36	38	3A	3C	3E
32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
40	42	44	46	48	4A	4C	4E	50	52	54	56	58	5A	5C	5E
54	66	68	70	72	74	76	78	80	82	84	86	88	90	92	94
.)	<	≤	=	>]	v.	^	≠	÷	,	+	.	/	
64	58	2A	2B	2C	2E	56	30	31	2F	37	48	34	28	42	
60	62	64	66	68	6A	6C	6E	70	72	74	76	78	7A	7C	7E
96	98	100	102	104	106	108	110	112	114	116	118	120	122	124	126
0	1	2	3	4	5	6	7	8	9	([;	x	:	\
00	01	02	03	04	05	06	07	08	09	57	55	51	36	5C	43
80	82	84	86	88	8A	8C	8E	90	92	94	96	98	9A	9C	9E
128	130	132	134	136	138	140	142	144	146	148	150	152	154	156	158
	α	ι	η	λ	ε	-	∇	Δ	ι	ο	'	□		τ	ο
29	5E	47	60	3A	49	0A	5D	25	4A	59	58	26	3C	48	3E
A0	A2	A4	A6	A8	AA	AC	AE	B0	B2	B4	B6	B8	BA	BC	BE
160	162	164	166	168	170	172	174	176	178	180	182	184	186	188	190
*	?	p	l	~	+	u	w	∃	↑	c	←	┌	→	≥	-
38	40	4C	38	3F	46	61	5F	62	45	63	5D		52	2D	35
C0	C2	C4	C6	C8	CA	CC	CE	D0	D2	D4	D6	D8	DA	DC	DE
192	194	196	198	200	202	204	206	208	210	212	214	216	218	220	222
◇	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
E0	E2	E4	E6	E8	EA	EC	EE	F0	F2	F4	F6	F8	FA	FC	FE
224	226	228	230	232	234	236	238	240	242	244	246	248	250	252	254
P	Q	R	S	T	U	V	W	X	Y	Z	{	-	}	\$	-
1A	1B	1C	1D	1E	1F	20	21	22	23	24					
														6A	27

BLANK



MICRO COMPUTER
MACHINES INC

BOX 310
133 DALTON STREET
KINGSTON
ONTARIO CANADA
K7L 4W2
TEL 613/544 9860

NOV 7 1975

STANDARD OMNIPOINT INTERFACE

Preliminary System Documentation

July 7, 1975.

Copyright 1975

By

Micro Computer Machines Inc.

All Rights Reserved

The information and/or drawings set forth in this document are copyrighted material and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein are the exclusive property of Micro Computer Machines Inc.

No disclosure of the information or drawings shall be made to any other person or organization without the prior written consent of Micro Computer Machines Inc.

Reproduction of the information or drawings or use in any manner not expressly intended by this release is specifically forbidden without the prior written permission of Micro Computer Machines Inc.

Micro Computer Machines reserves the right to make changes at any time to improve design and supply the best product possible.

Preliminary information. Supplementary information may be issued at a later date.

STANDARD OMNIPORT INTERFACE

Purpose:

The omniport interface is the basic building block for all omniport peripheral interface boards. It decodes omniport signals and makes them available to the device interface and encodes signals for the omniport generated by the device interface.

Signal Description:

To Device Interface:

RESET
SEL
SCOS
SGSI
SDOS
SGDI
D0-7

CLK

SYNC2
+5V
-12V

From Device Interface:

DATA0-7
STATUS0-7
CSEN
START

RESET:

True during system reset time. This signal should be used to do any initialization appropriate for the device. This signal is also true during power-up.

SEL:

True when the interface has been addressed. This signal is reset when another interface is addressed.

SCOS:

Device command strobe ($SEL \wedge COS$). The command is present on data lines D0-7.

SGSI:

Device status strobe ($SEL \wedge$ ^{GATE STATUS IN} GSI). The status should be present on data lines STATUS 0-7.

SDOS:

Data output strobe ($SEL \wedge DOS$). The data to be output to the device is present on data lines D0-7.

SGDI:

Data input Strobe ($SEL \wedge GDI$). The data from the device should be present on data lines DATA0-7.

Operation:

When the system is started, or the interface is powered on, a RESET is generated and the interface is deselected.

In order to activate the interface, it must be addressed by presenting the Device Address on the omniport data lines concurrent with an Address Out Strobe. This must be followed by a Gate Status In, which returns the Answer-Back code for the device. At termination of the first Gate Status In after the interface has been addressed, SEL comes true, and the interface is then active.

Any subsequent RESET or Address Out Strobe deselects the interface. If the interface is re-addressed, it will become reselected after the next Gate Status In.

Device Address:

The device address is set by two jumper plugs in the interface. It can be wired in directly or directed to an external device such as a thumbwheel switch.

Device Answer-Back Code:

This code is wired into the interface (refer to the schematic for the standard omniport interface). Its purpose is to inform the system as to the nature of the device being addressed.

Clock Signals:

Two clock signals are available from the system: a 1.25 μ s clock called SYNC 2, and a 5 μ s clock called CLK. These signals are enabled by grounding CSENB. CSENB is normally high, so that the clock signals do not cause the interface to draw any power.

Start Signal:

This is a 12V CMOS level which, when grounded, starts the main system. ✓

Power:

Two voltages are available from the interface for use by other equipment on the board: +5V and -12V. Both are fused at 250 ma and Zener protected.

Interface:

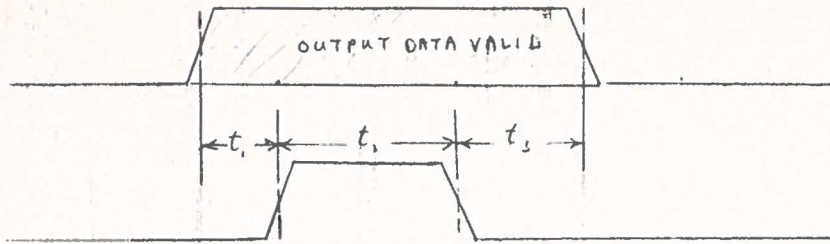
All interface lines are 5-volt CMOS levels. These are TTL compatible. All output lines will drive one standard TTL load. Timings are given on the accompanying diagram.

Protection:

All lines between the interface and the Omniport connector are protected against over voltage, so that a fault on the interface board will not propagate destructively either to the main system or to other interfaces on the Omniport.

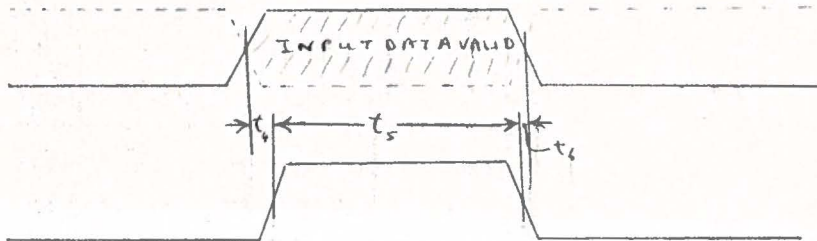
OMNIPORT TIMINGS

DO-7



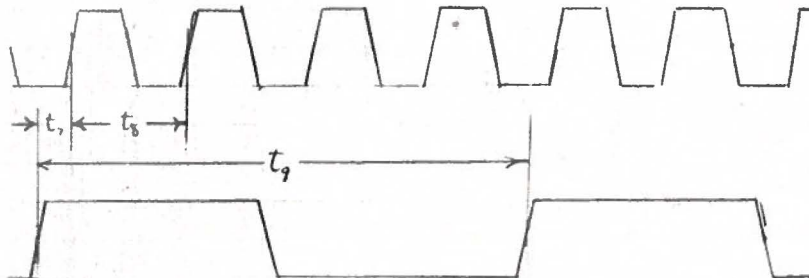
	min	nom	max
t ₁		0.8 μ s	
t ₂		1.8 μ s	
t ₃		1.3 μ s	

DATA 0-7
STATUS 0-7



	min	nom	max
t ₄			
t ₅		2.6 μ s	
t ₆	0	-	-

$\overline{\text{SYNC2}}$



	min	nom	max
t ₇		0.3 μ s	
t ₈		1.2 μ s	
t ₉		5 μ s	

$\overline{\text{CLK}}$



MICRO COMPUTER
MACHINES INC

BOX 310
133 DALTON STREET
KINGSTON
ONTARIO CANADA
K7L 4W2
TEL 613/544 9860

NOV 7 1975

MCP-132 (HyType) PRINTER

OPERATING GUIDE

July, 1975.

Copyright 1975

By

Micro Computer Machines Inc.

All Rights Reserved

The information and/or drawings set forth in this document are copyrighted material and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein are the exclusive property of Micro Computer Machines Inc.

No disclosure of the information or drawings shall be made to any other person or organization without the prior written consent of Micro Computer Machines Inc.

Reproduction of the information or drawings or use in any manner not expressly intended by this release is specifically forbidden without the prior written permission of Micro Computer Machines Inc.

Micro Computer Machines reserves the right to make changes at any time to improve design and supply the best product possible.

Preliminary information. Supplementary information may be issued at a later date.

MCP-132 (HyType) PRINTER

OPERATING GUIDE

Also any Diablo HyType; Qume Q30, Q45 printer connected via a GPI-2 interface card.

The MCP-132 printer incorporates a Diablo HyType I printer mechanism and a GPI-2 interface card, together with power supply, pedestal, cables etc. This GPI-2 interface card may also be used to connect a Qume Q30 or Q45 printer to the MCM/70. Therefore, this guide also encompasses the operation of these printers and any differences or options.

There are two modes of operation for the MCP-132 printer. One, AUTOMATIC PRINTING MODE, requires little knowledge on the part of the user and is suited for printing line output. The other, DIRECT ACCESS MODE, gives the user total control. He may issue any valid command to printer - ie. forward or backward space carriage in increments of 1/60th inch, move paper up or down in increments of 1/48th inch, print special characters, etc.

This guide describes both modes and incorporates a description of the Omniport System Access.

AUTOMATIC PRINTING MODE

□OU D

This command selects device D as output device for the MCM/70 APL System. All quad output commands (□+) issued after the □OU D command will be directed to device D. (□OU 0 resets output to the self scan on the MCM/70). For the MCP-132 the device address D is set to 1 at the factory. This may be changed by rotating the HEX switches on the GPI-2 card to any number between 01 and C7 HEX (1 to 199 decimal).

□OU returns a result R which is a three element vector. The meaning of this result is described below:

- R[1] This is the device address selected (Same as D in □OU D)
- R[2] Answer back code from the device. For the MCP-132 this has the value 66 decimal or 01000010 Binary.
- R[3] The current status of the device. This is described under Direct Access.

□PW=X (PRINT WIDTH)

The print width desired on the printer is set by this command ($30 \leq X \leq 138$). If the number of characters (ρC) presented in one output statement ($\square+C$) is greater than X, the line is broken into lines each containing X characters. (Note: Since the width of the MCP-132 is 132 characters, $\square PW$ should not be set to more than 132).

□+C (PRINT C)

The characters in the argument C are printed on the device selected by the $\square OU D$ command. If C is an array, each row of C is printed on a new line. If C is numeric, it is formatted by the system before output.

ERROR MESSAGES

The following printer error codes are issued by the system if abnormal circumstances occur, preceded by the message "HYTYPE ERROR".

Error No.	Error	Corrective Action
00	Printer Check	<ol style="list-style-type: none"> 1. Check for proper margin settings 2. If 1 fails, check electrical connections 3. If 1 and 2 fail, call service man
01	Device not HyType or QUME	<ol style="list-style-type: none"> 1. Check the device address on the Printer Interface
02	Printer not Ready	<ol style="list-style-type: none"> 1. Check electrical connections. Be sure power is turned on
03	Line too long	<ol style="list-style-type: none"> 1. Reset $\square PW$ so that $\square PW \leq 132$

DIRECT ACCESS MODE:

It is possible to drive these printers directly using the omniport access functions $\square OU$ $\square BO$.

NOTE: A DETAILED DESCRIPTION OF THE SYNTAX OF THESE FUNCTIONS IS CONTAINED IN THE ATTACHED, "OMNIPORT SYSTEM ACCESS" DOCUMENTATION.

The printer interface requires a 3 bit command, and thirteen bits of data. The least significant data Bit is called Data $\frac{1}{2}$ (2^0-1) and the most significant bit is called Data 2048 ($2^{12}-1$). The 3 command bits, followed by the 13 data bits form a 16-bit field. The most significant 8 bits of this field are output as an omniport command byte using $\square OU$, and the least significant 8 bits are the data byte output using the preceding $\square BO$.

The 3 bit command forms a binary word whose value is 0 through 7. The meaning of these commands is as follows:

- 0 - Set Printer Flag
- 1 - Print Character
- 2 - Move Carriage
- 3 - Toggle Ribbon States
- 4 - Feed Paper
- 5 - Enable Platens
- 6 - Form Feed
- 7 - Restore Printer

The operation of these commands will be described using the following functions:

```

∇ C COMMAND DATA;ADDR
[1] DATA+32 256τ|2×DATA
[2] ADDR+1↑□OU10
[3] °□OU ADDR,8 32∟C,1↑DATA°□BO 1↑DATA
∇

∇ R←STATUS
[1] R+(8ρ2)τ'|ρ2↑□OU10
∇

```

0 - Set Printer Flag:

Set the printer flag to the state indicated by Data 1048. All other data bits are ignored. The function of the flag is determined by the printer. It is intended to be used to turn the printer on and off.

Thus for a printer which is equipped with this option, the following will turn it on:

1 - Print Character:

The character whose code is represented by the binary word in Data 1 through Data 64 is printed. If the printer is equipped with variable ribbon step, the ribbon is stepped according to the binary word in Data 1-28 through 512. All other data bits are ignored. No carriage motion occurs.

Thus to print an 'A', whose code is 97 on an APL print wheel:

1 COMMAND 97

2 - Move Carriage:

The carriage is moved the number of increments (60ths of an inch) represented by Data $\frac{1}{2}$ through Data 512. If Data 1024 is 0, the carriage is moved to the right. If Data 1024 is 1, the carriage is moved to the left. Data 2048 is ignored. Any attempt to move the carriage past either end stop will result in a check condition.

Thus to move a distance N inches (left if N is negative);

2 COMMAND $(1024 \times N < 0) + 60 \times |N$

NOTE: HYTYPES IGNORE DATA $\frac{1}{2}$.

3 - Toggle Ribbon States:

There are two state bits which affect ribbon position: UP/DOWN, and RED/BLACK. If Data 2048 is 1, the UP/DOWN state is toggled. If Data 1024 is 1, the RED/BLACK state is toggled. All other data bits are ignored.

The ribbon state is normally UP and BLACK. Thus, in order to drop the ribbon:

3 COMMAND 2048

Repeating this command will raise it again. (LOW NEXT PRINT)

NOTE: ALL HYTYPES, AND QUMES WHICH DO NOT HAVE A RED RIBBON OPTION, WILL DROP THE RIBBON IF THE STATE IS SET TO RED.

4 - Feed Paper:

The paper is moved the number of increments (48th's of an inch) specified by Data 1 through Data 512. If Data 1024 is 0, the paper is fed up (normal motion). If Data 1024 is 1, the paper is fed down (reverse motion). Data $\frac{1}{2}$ and Data 2048 are ignored. Thus to feed the paper N inches (down if N is negative):

4 COMMAND $(1024 \times N < 0) + 48 \times |N$

5 - Enable Platens:

Printers with split platens may have paper feed for each platen enabled independently. If a platen is disabled, it will ignore paper feed commands. If Data 2048 is 1, the left platen is enabled. If Data 1024 is 1, the right platen is enabled. Printers with-only one platen are considered to have only a right platen.

6 - Form Feed:

Printers with the top-of-form option will feed the paper to the top of the next form. Others will ignore this command. All data bits are ignored.

7 - Restore Printer:

If a check condition occurs, the printer must be restored to remove the condition. The carriage is restored to the left margin. No other functions are affected. All data bits are ignored.

The status returned from the printer is an 8 bit binary word, defined as follows:

- Bit 7 - Paper feed ready
- Bit 6 - Carriage ready
- Bit 5 - Character print ready
- Bit 4 - Ribbon up
- Bit 3 - Ribbon red
- Bit 2 - Paper Out
- Bit 1 - Check condition (see carriage motion)
- Bit 1 - Printer powered and ready

Bits 0, 1, and 2 should be checked prior to any operation. Bits 5, 6, and 7 should be checked prior to each operation for the corresponding bit (ex: Bit 5 before printing a character). Bits 3 and 4 are used for setting a desired ribbon state. To set the ribbon unconditionallyUP, for example:

```
3 COMMAND 2048x~STATUS[[]IO+7-4]
```

NOTE: THAT THE BITS ARE NUMBERED IN REVERSE ORDER TO THAT IN WHICH STATUS PRESENTS THEM.

NOTE: ALSO THE DIFFERENCE BETWEEN THIS METHOD AND THE ONE DISCUSSED UNDER RIBBON STATES, WHICH TOGGLES THE RIBBON STATE.

HYTYPE APL10 PRINT WHEEL CODES

0:	32:•	64:~	96;◇
1:	33:¨	65:α	97:A
2:	34:)	66:⊥	98:B
3:	35:<	67:∩	99:C
4:	36:≤	68:⊂	100:D
5:	37:=	69:ε	101:E
6:	38:>	70:⊃	102:F
7:	39:⊂	71:∇	103:G
8:	40:∨	72:Δ	104:H
9:	41:∧	73:ι	105:I
10:	42:≠	74:∘	106:J
11:	43:‡	75:′	107:K
12:	44:,	76:□	108:L
13:	45:+	77:	109:M
14:	46:.	78:⊤	110:N
15:	47:/	79:○	111:O
16:	48:0	80:*	112:P
17:	49:1	81:?	113:Q
18:	50:2	82:p	114:R
19:	51:3	83:⌈	115:S
20:	52:4	84:~	116:T
21:	53:5	85:‡	117:U
22:	54:6	86:⊆	118:V
23:	55:7	87:ω	119:W
24:	56:8	88:⊃	120:X
25:	57:9	89:†	121:Y
26:	58:(90:c	122:Z
27:	59:[91:←	123:{
28:	60:;	92:⊢	124:¬
29:	61:x	93:→	125:}
30:	62::	94:≥	126:\$
31:	63:\	95:-	127:¬

Note: When used with $\square BO$, these codes should be multiplied by 2.

OMNIPOINT SYSTEM ACCESS

The Omniport is an 8-bit parallel interface between the main system and all peripheral devices. In order to understand the nature of the interface itself, the reader should refer to the document on the Standard Omniport Interface.

I - Direct Device handling

Basically, six operations are available:

- 1) Address device
- 2) Read device code
- 3) Send command to device
- 4) Read device status
- 5) Send data to device
- 6) Read device data

The function of these operations is achieved with four APL system functions - $\square OU$, $\square IN$, $\square BO$, $\square BI$.

$\square OU$, $\square IN$ - These functions are identical in operation except that $\square OU$ (short for $\square OUT$) sets the system SINK, while $\square IN$ sets the system SOURCE. The purpose of the SINK and SOURCE are described further on in this document.

Syntax: $R \leftarrow \left\{ \begin{array}{l} \square IN \\ \square OU \end{array} \right\} B$

Conformability: $1 \leftrightarrow \rho\rho B$
 $2 \geq \rho B$

Domain: $0 \leq B \leq 255$

Result: $1 \leftrightarrow \rho\rho R$
 $3 = \rho R$

Operation: The current SOURCE/SINK is set to the device whose interface address is $B[1]$. If B is empty, the SOURCE/SINK device is left unchanged. The device is selected, and the answer-back code from it is returned in $R[2]^*$. $R[1]$ contains the device address. $R[3]$ contains the current device status. If $B[2]$ is present, it is output as a command to the device.

* In communicating with the omniport, the 8 bits of the omniport are represented in the system by their base 2 value. Bit 7 on the omniport is the most significant bit.

$\square BO, \square BI$ - These functions are used to output and input data via the omniport.

Syntax: $R \leftarrow \square BO B$

Conformability: $0 \leftrightarrow \rho \rho B$

Domain: $0 \leq B \leq 255$

Result: $R \leftrightarrow \dot{B}$

Operation B is output as data to the current SINK device.

Syntax: $R \leftarrow \square BI$

Result: $0 \leftrightarrow \rho \rho R$

Operation: R is data from the current SOURCE

Device Addresses -

Peripheral devices can be assigned any address in the range 1 - 199. The address 0 is reserved for the MASTER SOURCE (namely, the system keyboard) and the MASTER SINK (the system display).

Addresses in the range 200 - 255 are reserved for other system devices. Currently, the only ones used are 200 - 207, which are the system tape drives. These drives are used by EASY. The left hand system drive, which is drive 1 (in origin 1) is address 200. This drive is used by AVS. The other system drive has address 201. External tape drives may be attached to the Omniport and assigned addresses 202 - 207.

Any attempt to output a command to a system device which is in use by AVS or EASY will result in an I/O ERROR.

Device Answer-Back Codes -

The answer-back code of a device is made up as follows - let ABC be the code received, and let:

$$C \leftarrow 2 \ 2 \ 2 \ 32 \ \tau \ ABC$$

Then the elements of C are interpreted as follows:

Element	ABC bit	Meaning
C[1]	7	1 = Input Device
C[2]	6	1 = Output Device
C[3]	5	1 = Interactive (i.e. a terminal) ^{PROMPTED}
C[4]	3 - 0	Device Type

The following device types have so far been assigned:

Type	Device
0	EIA/ASCII terminal
1	MCM Cassette drive
2	MCM/Hytype printer
3	MCM/True Data Card reader
4	MCM Dot Matrix printer
5	EIA/APL terminal

Command and Status Codes -

These codes are peculiar to the device being accessed. Refer to the documentation for the device in question.

If no device is present, the answer-back code, status, and data will all be zero.

II - Line I/O

Most program I/O in APL is line I/O; that is, I/O is done with lines of APL characters. This type of I/O occurs in two forms - implicit and explicit.

Whenever the system has nothing to do, it waits for input from the user. This is an implicit input request. Implicit input is also requested by the system when it is in open function definition mode. Further, if the system has the result of a calculation and has not been told what to do with it, it outputs it. This is an implicit output operation. Implicit output also occurs when an error message is issued.

The functions `⎕` and `⎕←` are explicit requests for input. The function `⎕←` is an explicit output command.

All implicit I/O is done through the system MASTER SOURCE and MASTER SINK, namely, the keyboard and display unit. Explicit I/O, however, is done through the current SOURCE and SINK devices. These are initially set to be the MASTER SOURCE and SINK, but may be diverted to other devices using `⎕IN` and `⎕OU` as described under those functions.

If the SOURCE or SINK has been diverted, the system will attempt to perform the requested I/O using the requested device. The following device types are currently supported.

- Type 2: MCM/Hytype printer (output only)
- Type 3: MCM/True Data Card reader (input only)

Error Messages -

If the system cannot perform the requested I/O, it will issue an I/O ERROR, followed by a code. The meaning of the codes is as follows:

- 01 - No device found with the specified address
- 02 - Output requested to non-output device
- 03 - Device not supported for line I/O.
- 04 - Input requested from non-input device.
- 05 - I/O requested for device in use by AVS/EASY

In addition, each device handler may issue its own error. Refer to documentation for the handler in question.

CONVERSION TABLE DECIMAL/HEX/APL10

00	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
20	22	24	26	28	2A	2C	2E	30	32	34	36	38	3A	3C	3E
32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
40	42	44	46	48	4A	4C	4E	50	52	54	56	58	5A	5C	5E
54	56	58	60	62	64	66	68	70	72	74	76	78	80	82	84
.	..)	<	≤	=	>]	v.	.	#	÷	,	+	.	/
60	62	64	66	68	6A	6C	6E	70	72	74	76	78	7A	7C	7E
96	98	100	102	104	106	108	110	112	114	116	118	120	122	124	126
0	1	2	3	4	5	6	7	8	9	([;	x	:	\
80	82	84	86	88	8A	8C	8E	90	92	94	96	98	9A	9C	9E
128	130	132	134	136	138	140	142	144	146	148	150	152	154	156	158
	α	1	n	l	ε	-	v	Δ	l	ρ	.]		T	o
A0	A2	A4	A6	A8	AA	AC	AE	B0	B2	B4	B6	B8	BA	BC	BE
160	162	164	166	168	170	172	174	176	178	180	182	184	186	188	190
*	?	p		~	+	u	ω	ρ	↑	=	+	Γ	→	≥	-
C0	C2	C4	C6	C8	CA	CC	CE	D0	D2	D4	D6	D8	DA	DC	DE
192	194	196	198	200	202	204	206	208	210	212	214	216	218	220	222
◇	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
E0	E2	E4	E6	E8	EA	EC	EE	F0	F2	F4	F6	F8	FA	FC	FE
224	226	228	230	232	234	236	238	240	242	244	246	248	250	252	254
P	Q	R	S	T	U	V	W	X	Y	Z	{	→	}	3	-