HP RESIDENT

TO RUN:

PUSH *P*
SET STARTING ADDRESS *200*$_8$ IN DISPLAY REGISTER
PUSH INTERNAL, EXTERNAL *PRESET*
PUSH *RUN*

INDICATIONS:

DISPLAY REGISTER should increment until it overflows. then the INTERRUPT SYSTEM light should be on, the pgm listening for interrupts from the 500 and displaying link status as--

bit 7	CHIO has output a char not yet read by HP
bit 15	HP has output a char not yet read by CHIO
bit 0 - 6	last command from CHIO

code	interpretation
2	write tape
4	set tape mode
7	print
10	read
13	tape control
64	skip it
77	clean up
102	header search

bit 8 - 14	last command from HP

61	device error
65	write ok, but unit exception
67	write ok
70	communications error
75	print finished
76	read ok

with LOCK off, push CLEAR DISPLAY, if the DISPLAY REGISTER stays the same then at least the HP resident wait loop is executing.

TO LOAD:

[ON TAPE DECK]
PUT CURRENT RESIDENT DUMP TAPE (S104 OR S105) ON DECK
PUSH LOAD/UNLOAD AND ONLINE

[ON HP]

PUSH *P*
SET START ADDRESS *17200*$_8$ IN DISPLAY REGISTER
MAKE SURE THE *EXTEND* LIGHT IS OFF
PUSH INTERNAL, EXTERNAL *PRESET*
PUSH *RUN*

INDICATIONS:

normally program is loaded, started at *200*$_8$ as above, and the tape deck unloaded and deselected automatically. however if the program halts and tape doesn't rewind, then the controller detected some error. try again from the top if it still doesn't work note the contents of the *A* and *P* registers and consult the listings and Daconics manual.

THE ALOHA SYSTEM

PROJECT:

TITLE:

The HP - BCC 500 Interface Program

Working Paper

October 3, 1972

AUTHOR(s):
Wrenwick Lee

CHECKED:

APPROVED:

ABSTRACT

This document describes the resident program in the HP 2100A. Presently there exists a hard-wired link between the BCC 500 (now commonly referred to as the M1) and the HP 2100A. The role of the HP is that of a channel (a smaller computer connected to a larger one with the smaller computer doing the I/O for the larger one) for the M1. Presently, a 9-track t~~y~~ape unit with a Daconics controller ~~are~~ is connected to the HP. The role of the HP as a channel is more sophisticated than a simple channel commonly found on such computers as the IBM 360. The difference lies in that the HP receives more sophisticated commands as well as performing more sophisticated functions. Most of the present document involves tape unit functions. However the overall structure of the software module is described. Later functions will probably be those involving the addition of a line printer.

PART I PROGRAM DESCRIPTION

DESCRIPTION OF MAJOR ROUTINES IN HP INTERFACE PROGRAM

1. COMND - routine that receives the command from the M1 (BCC 500) and does all checking and initialization for the processing of that command e.g., is it valid, can we allow it, etc. Then it dispatches according to the command to the respective routine such as read from tape, write to tape, rewind the tape, etc.

2. SHPIT - routine that transfers blocks of data between the HP and the M1. Half of it is code to receive data from the M1 and store it in the HP, and the other half is code to send data from the HP to the M1.

   It calls two functions:

   GET - which assists in receiving data from the M1.

   PUT - which assists in sending data to the M1.

3. XWRIT - routine that writes blocks of data onto the tape according to certain parameters passed to it by the M1.

   Calls functions:

   RDTU - get tape unit number.

   RDTW - to get word count.

   SHPIT - to get block of data from M1.

   WRING - to check for write ring.

   DMAIO - to do DMA (direct memory access) I/O; basically the method used to actually write on the tape.

   SNDM1 - to send messages to the M1.

   REDY - check to see if the tape unit is ready.

4. XREAD - this routine has about the inverse function of XWRIT; it proceeds in the opposite direction. Taking data from the tape and sending it to the M1.

   Calls functions:

   RDTU

   RDTW

   REDY

   DMAIO - only this time reads from the tape, instead of write onto the tape which was how XWRIT used this function.

   SHPIT - to send the data read from the tape to the M1.

   SNDM1 - to send messages to the M1. 2

   SNDDA - to send messages to the M1.

5. XHDRS - this routine scans a tape for a record with a certain beginning header. The header itself is sent by the M1, as well as the length of the header. As soon as the routine finds the header, it responds with a positive acknowledgement, by sending back a variable number of bytes read from the tape to the M1. If no header is found that matches (in effect we hit the end of file mark) then a negative acknowledgement is sent back.

   Calls functions:

   RDTU

   RDTW - read twelve bit data.

   RDSX - read six bit data.

   SHPIT - to get header that must be searched for on tape (sent from M1 to HP).

   DMAIO - to actually read a record from the tape.

SNDM1 - to send messages to M1.

SHPIT - to send data to M1. ?

Actually the header search routine jumps to entry points in the read routine since some of the functions performed are similar. The exit (or completion) of the header search function is via the read routine.

6. XCNTR - routine that handles all the tape control functions that relate to tape positioning. It is actually another level of dispatching (the first level being the jump to this routine itself) to jump to one of several tape control routines. These are:

REWIND and keep tape unit on line.

REWIND and put tape unit in off line status. lower case?

GAP WRITE 3" blank tape

WRITE FILE MARK

FORWARD SPACE FILE

BACKSPACE FILE

FORWARD SPACE RECORD

BACKSPACE RECORD

The last four commands may be done a multiple number of times, and is a parameter sent by the M1. ER?

Much of the logic in this routine does checking to see if the command should be allowed to begin with. For example, we do not want to backspace if we are already at the bottom of the tape. Similarly with rewind. Nor do we want to forward space if we are already at the end of tape reflective marker.

Functions called are:

RDTU

RDSX

REDY

WRING - check for write ring.

EOT - check if end of tape.

TPSTT - get status of tape unit.

OUTF - output a tape unit command to the Daconics tape unit controller.

7. XSETM - routine that should set modes for tape such as density, parity, and other parameters. Presently mode bits are ignored by HP Interface program. It assumes the whole world is a 9TRK 800 bpi tape unit and doesn't really care what you send it.

8. XRST - this is the routine used to get the HP in an initialized state from whatever state it may have been in before. It is entered whenever a Restart interrupt is sent. Its final exit is to the HP initialization routine.

DESCRIPTION OF FUNCTIONS AND MINOR ROUTINES IN HP INTERFACE PROGRAM

1. SNDM1 - send data to M1.

2. RCM1 - receive data from M1.

3. PUT - utility used by SHPIT routine to send data to M1 (calls SNDM1).

4. GET - utility used by SHPIT routine to receive data from M1 (calls RCM1).

5. RDSX - receive data from M1. Check to see if two high order bits are '1's.

Uses the last six bits. Generally used for short parameters such as tape unit number, control functions, etc.

6. RDTW - gets sixteen bits of data from M1 (one HP word) by calling RDSX three times and discarding extra high order bits. Used to pass parameters such as word count. Generally for parameters requiring more than six bits.

7. RDTU - gets tape unit number. Logic exists to switch from 7-track to 9-track but presently being ignored (sigh).

8. SNDDA - sends a variable number (<6) of six bit bytes to the M1. Similar function to RDTW except we are going from the HP to the M1.

9. OUTC - routine that sends a command to the Daconics Tape Controller. Once the command is accepted a quick exit is made to the next routine, EXIT.

10. EXIT - sets up all the messy stuff like interface flag bits and sending control pulses to start up the tape I/O. Also prepares for an interrupt from the tape controller which signals tape I/O complete.

Sounds horrid.

11. TPINT - entry point for receipt of a tape I/O completion interrupt. When the tape unit has successfully ? completed its operation, an interrupt occurs, and this routine is entered. This routine then causes a jump indirect through the EXIT routine (which was the last routine entered before our wait for the tape I/O to be completed). This gets us back to the calling routine.

12. WRING - check for write ring. If none present when some kind of write operation is called for, then signal that a device error has occurred.

13. EOT - check for End of tape.

14. LOOP - place to wait for interrupts.

15. DVER - routine to tell M1 some abnormal condition (error condition) with the tape unit has occurred.

16. TPS~~I~~TT - reads status word from Daconics tape controller.

MAJOR DATA STRUCTURES IN HP INTERFACE PROGRAM

1. At Location 30B

   a. Tape and print buffer locations.

   b. Locations of various interrupt handling routines.

2. At Location 50B

   Locations of various major routines.

3. At Location 71B

   Locations of various functions and minor routines.

4. OPTAB

   Command Dispatch Table to cause jumps to correct routines, given a command from the M1.

5. CMDTB

   Command table for commands to tape controller.

6. HP => M1 response codes.

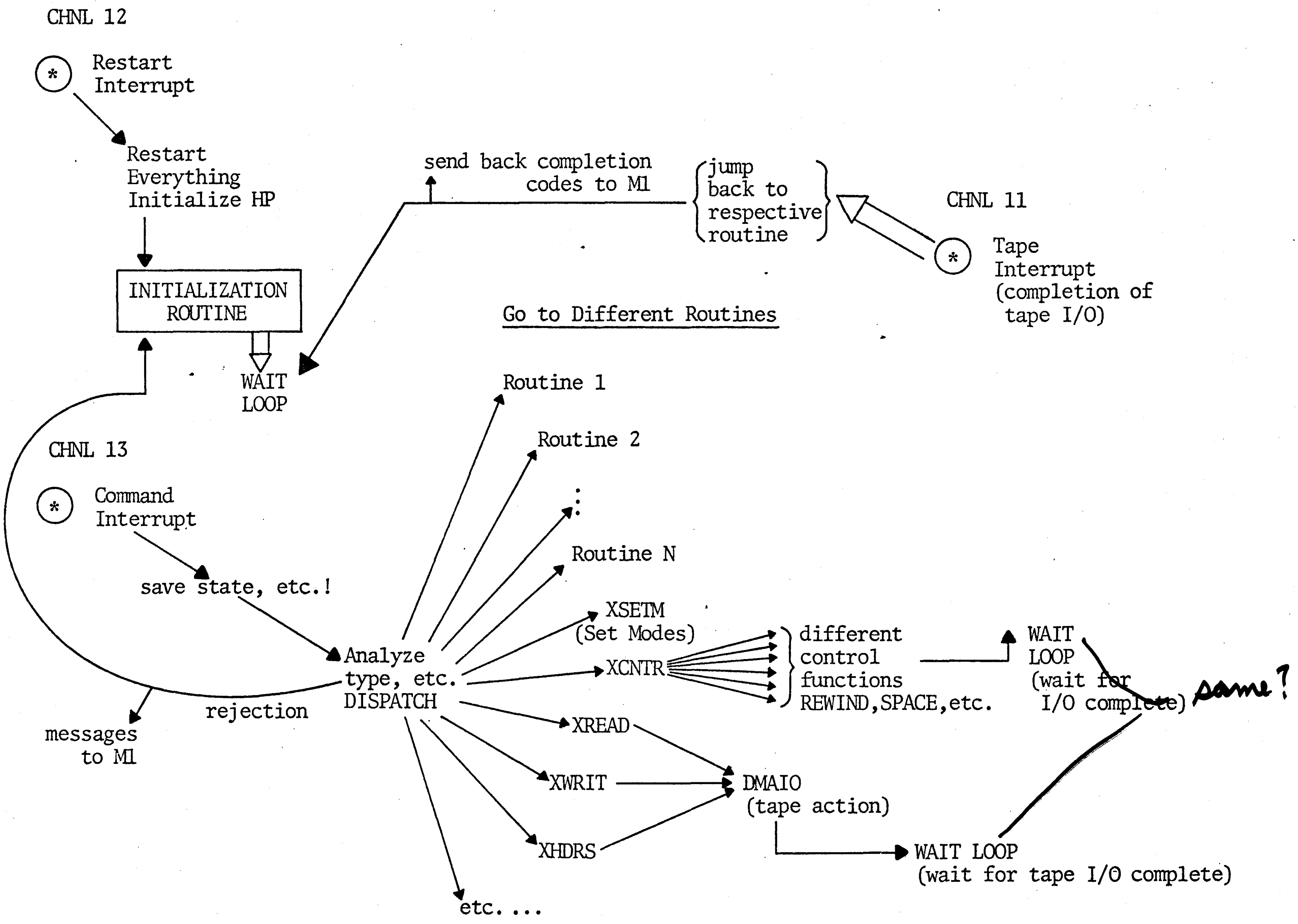
7. I/O channel declarations.

8. Return flags fields - flags sent to M1 under various conditions.

9. CMNDS - list of addresses of all instructions that send commands to the tape controller. Used so that can change the instruction itself to reflect which controller is wanted.

ugh!

HP INTERFACE PROGRAM FLOW


CHNL 12
*
Restart Interrupt
Restart Everything Initialize HP
INITIALIZATION ROUTINE
WAIT LOOP
send back completion codes to M1
jump back to respective routine
CHNL 11
*
Tape Interrupt (completion of tape I/O)
Go to Different Routines
Routine 1
Routine 2
:
Routine N
CHNL 13
*
Command Interrupt
save state, etc.!
Analyze type, etc. DISPATCH
rejection
messages to M1
XSETM (Set Modes)
XCNTR
different control functions REWIND, SPACE, etc.
WAIT LOOP (wait for I/O complete)
same?
XREAD
XWRIT
XHDRS
DMAIO (tape action)
WAIT LOOP (wait for tape I/O complete)
etc. ...

PART II HP - M1 COMMAND INTERFACE PROTOCOL

Abstract: This is a description of the current implementation of the software communication between the HP 2100 (which now replaces the IBM Model 30 in the old BCC configuration) and the M1 (Model 1; meaning the BCC 500 system).

The IBM Model 30 was primarily used for tape input/output controlled remotely via the M1. A printer and card reader were also available. The HP 2100 now replaces the function of the IBM Model 30. Presently only a 9TRK tape drive is connected but the later addition of a printer is envisioned.

Currently the HP Resident program communicates with IPL (ITP Program Library Facility) and all of the functions available under IPL can be used. These are described in ITPPL, a previously published BCC manual.

COMMANDS IMPLEMENTED

The following are the currently implemented commands in the HP Resident. They are followed by an 8-bit octal code that signifies the actual code for the command.

1.	READ	210B	read a record from tape.
2.	WRITE	202B	write a record to tape.
3.	CONTROL	213B	tape motion control done a variable number of times.
4.	SET MODES	204B	normally sets density, parity, etc., for M30. Because of possible future use, kept in. However, presently ignored and everything defaults to 9TRK, 800 bpi.

5. HEADER SEARCH 102B check headers on a tape to see if there is one that matches that sent by the M1.

6. SKIPIT 264B after HP notifies that some error has occurred; ignore it and get back to inactive state. This error may be leaving out the write ring, not having the tape unit on-line and some others.

7. RESTART this routine is entered not with a command but by a special interrupt. It puts the HP into an initialized state.

COMMANDS UNIMPLEMENTED

1. Read Backwards - will probably never be implemented.

2. Read card image, read card EBCDIC - will only be implemented if a card reader is obtained.

3. Print - the current design has taken into consideration the eventuality of the addition of a printer. With nominal effort, no major obstacle should present itself in the implementation of this command.

4. Mount chain - no comment.

5. Sense - to be implemented.

6. Set for retries - to be implemented.
   Set for no retries - to be implemented.

At present, retries (up to 10) are made when attempts to read or write tape fail.

7. Echo - probably will never be implemented.

* Note 1: Special logic in the HP Resident is available but not used presently to allow the addition of a 7TRK tape unit. This is contingent on the following conditions:

1. An interface to the tape unit is identical to the present Daconics Controller for the 9TRK unit in use.
2. The same tape commands are used.

If the two conditions are met, then adding a 7TRK unit is somewhat trivial. Otherwise it may mean a reasonable amount of coding.

* Note 2: The restart command is used to initialize the HP. Another of its functions is to initialize the BCC 500 from tape. This second function has not yet been implemented.

COMMAND PROTOCOL

The following is a detailed description of the chain of events that occur with the sending of various commands. Each command is described in detail.

Important Convention: Both commands and data are sent across to the HP as 8-bit quantitites. However for data there are two high-order '1's and thus only the remaining six bits really count. In octal notation,

3DD

the 3 represents the two high-order '1's and the DD represents the data (actually 6 bits of it). Notice that no commands have two high-order '1's. Anything with high-order '1's must be data.

1. READ

a. M1 sends 210B: this causes a jump to the read from tape routine.

b. M1 sends 300B or 301B: either unit 00 or unit 01 (ignore the octal 3 which signifies high-order '1's meaning the next six bits are data) unit 00 is the 7TRK unit, and unit 01 is the 9TRK unit. Presently both are defaulted to 9TRK by the Resident program.

If an error occurs (i.e., unit number is wrong), we send 270B (communications error) followed by 10B signifying that a bad unit number was given.

c. M1 sends: 3XX 1st
3YY 2nd
3ZZ 3rd

which when put together is the byte count of the number of 8-bit bytes we want to read off of the tape. ZZ is the last 6 bits of the ~~word~~ byte count. YY is the next six bits, and XX is the first 4 bits of the word count, e.g., if the M1 sent

300
300
305

it means it wants to read 5 8-bit bytes from the tape.

d. At this point, the program goes to check if the tape unit is ready. If it is busy, for example, the tape is rewinding, it sends back 273B (Device not ready). However if it is something more permanent such as the tape unit being off-line it sends back 261B (Device error) and followed by ØXB where X is the tape unit number. Usually the

?

program resident in the M1 will take the appropriate action. The HP will be waiting for a SKIPIT command, but a RESTART will also purge the system clearly.

e. When the program tries to read the tape, if it continually gets parity errors, then after ten tries, it will send back 261B (Device error) followed by ØXB where X is the tape unit number. It will then wait for a SKIPIT command. ?

f. Assuming d and e do not occur, it means that the tape has been read properly. The program then sends 6-bit data across the link according to the byte count sent to it. Note that the byte count sent to it was in terms of 8-bit bytes. So if it read 3 8-bit bytes from the tape, it sends 4 6-bit data bytes across the link.

g. Then a 276B (Read OK) is sent across the link, followed by three 6-bit bytes which contain the byte count (8-bit bytes) of bytes sent to the M1. This is the inverse operation of step c.

h. Finally a data item consisting of flags is sent across. With the Read command, two in particular are of interest.

40B (UNXCCF) unit exception flag

20B (ILF) incorrect length flag

the unit exception flag bit if set, means that the end-of-file mark was read; the incorrect length flag if set means that the record just read either has more or less than the number of bytes requested by the M1.

2. WRITE

a. M1 sends 202B: this causes a jump to the write to tape routine.

b. M1 sends 300B or 301B: 7TRK or 9TRK tape unit.

c. M1 sends: 3XX 1st
3YY 2nd
3ZZ 3rd

which is the byte count of the number of 8-bit bytes that we want to write onto the tape. ZZ is the last six bits, YY are the next six bits, and XX is the first four bits.

d. M1 sends to the HP the data that it wants to write onto the tape, 6 bits at a time. As in our convention, each 6-bit data item is preceded by two high-order ones.

e. Then the program goes to check if the tape unit is ready. The same possible kinds of errors may occur as in step d of the READ command.

f. Since this is a WRITE command, we need a write ring. If no write ring is present, then a 261B (Device error) is sent, followed by OXB where X is the tape unit number.

g. While trying to write, if a parity error keeps occurring, then
? 261B (Device error) is sent, followed by ØXB where X is the tape unit number.

h. Having passed through steps e, f and g safely, we then send a 267B (Write OK) back to the M1. However, if we have hit the end of tape mark, we send back a 264B.

3. CONTROL

a. M1 sends 213B: this causes a jump to the tape motion control routine.

b. M1 sends 300B or 301B: 7TRK or 9TRK tape unit.

c. M1 sends 3XXB where XX is the number of times to repeatedly perform the same command. For example, if the command was forward space record, and XX was 05, then we forward space 5 records.

d. M1 sends 3XXB where XX is the command itself. 00≤XX≤07 otherwise a 270B (bad communications) is sent followed by a 4B (bad control operation flag) both sent to the M1. The legal operations are:

XX	Operation
00	Rewind and standby
01	Rewind and deselect
02	Write 3" blank tape
03	Write file mark
04	Backspace 1 record
05	Backspace 1 file
06	Forward space 1 record
07	Forward space 1 file

Note: (a) If the control operation is a rewind, then the number of times it is performed is defaulted to 1.

(b) If forward or backspacing hits the end-of-tape marker, or bottom-of-tape marker, then forward or backspacing is ceased regardless of the want of number of times it should be done.

(c) The write blank tape and write file mark commands involve a check for the write ring. If it is not present, a 261B (Device error) followed by ØXB, where X is the tape unit number is sent to the M1. ?

e. Similar to the READ and WRITE commands a check is made to see if the tape unit is ready. The error replies (if any) are the same as in step d of the READ command.

f. At the completion of the tape control routine, the HP sends a 267B (Write OK) back to the M1. This is for all the tape control commands, whether rewind or positioning. For multiple performance of the same control command, only one write OK is sent back, at the completion of all the repetitions of the same command.

4. SET MODES

a. M1 sends 204B: this causes a jump to the set modes routine.

b. M1 sends 300B or 301B: 7TRK or 9TRK tape unit.

c. M1 sends 3XXB where XX is the mode bits used by the Model 30. This is now obsolete.

d. HP sends 267B (Write OK) back to the M1.

5. HEADER SEARCH

a. M1 sends 102B: this causes a jump to the header search routine.

b. M1 sends 300B or 301B: 7TRK or 9TRK tape unit.

c. M1 sends: 3XX 1st
3YY 2nd
3ZZ 3rd

which is the byte count of the number of 8-bit bytes we want read to the M1 from the tape if the header search succeeds.

d. M1 sends 3XX where XX is the compare count. The XX is the length of the record header (in 8-bit bytes) that we want to search for. For example for IPL, XX is 03.

e. M1 sends the header that it is looking for. The length of the header depends on the compare count just sent in step d. For example, if as in the example in d the compare count was 03, then the M1 sends

3XX

3YY

3ZZ

3AA

where each of XX, YY, ZZ, AA is 6-bits. Therefore 24 bits are sent which is the same as 3 8-bit bytes (the compare count).

f. The program then checks to see if the tape unit is ready. The same kind of errors can occur as in step d of the READ command.

g. Similarly, errors might occur when attempting to read the tape. These are of the same nature as in step e of the READ command.

h. If the header search was successful (i.e., a correct match was made between the header sent over from the M1 and a header on a tape) then the following happens:

   (1) A variable number of 6-bit bytes is sent to the M1. This number is dependent on the byte count sent previously from the M1 in step b.

   (2) 276B (Read OK) is sent.

   (3) Followed by 3 6-bit bytes which contain the byte count (8-bit bytes) of bytes sent to the M1.

   (4) Followed by a byte of flags. The only one that could be set in this circumstance is 20B (ILF; incorrect length flag) where the number of bytes just read from tape differ from that requested by the M1.

i. If the header search was unsuccessful (i.e. hitting the end of file mark without matching any of the headers on the tape successfully) then the following occurs:

   (1) 276B (Read OK) is sent.

   (2) Followed by 3 6-bit bytes which count the byte count of bytes sent to the M1. In this circumstance, it is zero so the following are sent over.

300B

300B

300B

(3) Two bits are set in the byte of flags that are sent over.

These are:

	4B	bad compare flag
and	10B	unit error flag

j. Would you believe that we are now finished? We are.

6. SKIPIT

a. M1 sends 264B. This causes the program to return to an initialized state and sent back all falgs as though there were no error. E.g. ROK if a device error had occurred on a READ command, etc. ?

7. RESTART

a. M1 sends an Interrupt via I/O Channel 12 on the HP.

b. In order to clear the commands that might be stuck in command channel (I/O Channel 13) the command channel is purged. All commands are ignored until:

c. The M1 sends a 276B (Disk Restart) which the causes the HP Resident to initialize itself.

* Note 3: There are some other error conditions which may occur and result in some less known (less probable to occur) error messages to be sent to the M1 from the HP Resident. Some of these are:

1. 270B (communications error)

   followed by

   20B (bad communications flag)

   which occurs if a new command is sent to the HP while one is still being processed. (Do not confuse this with RESTART which may be sent anytime.)

2. If when trying to read data (with high order '1's); it is not the case

that there are two high order 1's before the six bits of data, the HP will HALT with 102001 in the switch register. This will be fixed shortly so that RESTART will be able to reinitialize the system and a valid error message sent back.

* Note 4: In sending blocks of data across from the M1 to the HP, the M1 always sends across a multiple of 6 eight-bit bytes (or 2 BCC words). If it does not really intend to have all of these stored on tape it pads the ending with zeroes. One can always tell how many are really intended to be sent because the byte count sent by the M1 always tells the true number. For example, if the byte count says 4, though the M1 sends 6, only the first 4 will be used by the HP (a~~s~~d nauseum).

* Note 5: Plans are to compute checksums at both the HP and M1 sides of the link. This is to ensure correct transmission when large blocks of data (6144 bytes for IPL files) are sent across.

* Note 6: Of all the functions to be completed, the highest priority should be that of getting the RESTART function fully implemented. Basically what is not yet implemented is the code that initializes the BCC 500 (full system) from tape. This code should be ready by the time the system is brought up.

Not true

~~IPL: ITP PROGRAM LIBRARY FACILITY~~

~~IPL is a program which resides in low core in the ITP. It provides access to the IBM 36Ø/3Ø at two levels - low level direct access routines, and a high level executive for accessing a program library tape. This manual describes how to get IPL into the ITP core, and how to use it.~~

~~LOADING IPL~~

~~If M3Ø Power is off, start at step 1.~~

~~If Dump Tape* is not mounted, start at step 3.~~

~~If you are not sure the tape is positioned properly, or if you think the M3Ø program is dead, start at step 4.~~

~~If printer is ready, and you are reasonably confident that the M3Ø is in a reasonable state, start at step 8.~~

~~Otherwise, start at step 6.~~

1. ~~Press POWER ON on M3Ø.~~
2. ~~Go to step 4 if Dump tape* is mounted on unit Ø81 (right hand unit).~~
3. ~~Dismount tape on unit Ø81 (if any) and mount Dump tape*. (See M3ØT/M - 18 if not familiar with tape procedures).~~
4. ~~Press RESET, LOAD-REWIND, and START on unit Ø81.~~
5. ~~Set LOAD UNIT switches on M3Ø to Ø81 and press LOAD. Tape should twitch, and M3Ø should stop with WAIT light lit and A-REGISTER and B-REGISTER showing Ø .~~
6. ~~Ready the printer by pressing START.~~
7. ~~Press SYSTEM-RESET, START, and START again on M3Ø. Console lights should be as in step 5.~~
8. ~~Place appropriate paper tape in reader (there are 3, all labelled "BOOT-STRAP LOADER" with the date, subtitled either "IPL ONLY", "IPL & DDT ONLY", or "ALL", for loading IPL only, IPL and DDT (not including symbol table) or all of core. There is a sign on the tape reader giving the date of the correct tapes). Turn on reader.~~
9. ~~If paper tape runs away, press RESET on ITP, type 74Ø, and go to step 8.~~
10. ~~Press RESET on ITP, and type 715G.~~
11. ~~Paper tape should read in, and roughly 15 seconds later, IPL should type its greeting. If not, go back to step 4.~~

~~* Dump tape says on it "Dump Tape" and there is a sign on tape unit Ø81 giving tape number of current tape.~~

APPENDIX I

EXECUTIVE LEVEL FACILITIES

The IPL executive is patterned after a 94Ø subsystem, using the 94Ø command recognition scheme. The commands are all recognized by their first letter. The balance of the command is echoed and a confirming period or space is expected. IPL types GO$GO when entered, and its herald - #, which is typed at the completion of each command. The available commands are listed below. All the commands except "GO TO" expect an IPL program library tape [redacted] to be mounted and ready on the tape unit [redacted]. All numbers input or output by IPL are octal. For addresses in main memory don't forget 4B7.

#INIT{␣}

Initializes the Model 3Ø and the link. Should be done each time IPL is entered, and after any error messages.

#READ{␣& GO{␣}}

FILENAME: NNNN.

Reads a file from the library file and puts it in core. Filename is any four characters except ?

terminated by a period. ? aborts at any time. The load address, word count, and transfer address of the file are typed. Control is transferred to the transfer address if the "READ & GO" option was used.

#WRITE{␣}
 FROM firstw.
 TO lastw.
 TRANSFER ADDRESS ta.
 TO FILE NNNN.

Writes core from firstw to lastw onto a library file, with transfer address ta. A given filename may only be written once, due to the simple minded nature of the tape format. ~~Note: This command should be used only if a tape saying "Write on M3Ø only" is mounted.~~

#GO TO{␣} ta.

Transfers control to ta.

#&CREATE TAPE{␣}

Creates a new IPL program library tape. Any files which were on the tape are lost .

#DIRECTORY{␣}

Lists the names of all the files on the tape.

~~#CRUNCH TAPE{␣}~~
~~DELETE FILES: 1111,2222,...NNNN.~~

~~Removes the listed files from the tape and~~

~~compacts it. This represents the only way to get rid of a file, or to change it (by deleting and replacing it). THIS COMMAND MAY CLOBBER ANY PART OF CORE. In fact, it probably will, so don't expect anything to survive afterwards, except of course IPL and the new tape.~~

#&DUMP{␣}

Dumps all of first 8 pages of core in which IPL resides onto tape ~~on unit Ø81~~ in format readable by bootstrap paper tapes. No checking or tape positioning is done. For system use only by VPCs.

REFERENCES

1.	HP Manual	HEWLETT PACKARD 2100A Reference Manual, June 1971.
2.	Daconics Manual	Applications Note Daconics 2914-01 Nine-Track Magnetic Tape Controller (Packard Binary Format)
3.	--	M1-360 Communication - Codes and Messages sent between Processors Memo to W. Lichtenberger from W. Lee (March 14, 1972).
4.	360 ASM	360 Assembly Manual
5.	--	ITPPL/M ITP Program Loader Users Guide Henry Thompson
6.		Communications with the IBM Model 30 M30/S-7 Henry Thompson
7.	--	M30 - IBM 360 Interface Program (360 Assembly program) Henry Thompson
8.	Roger's memo	Preliminary Memo Interface of HP with M1. April 3, 1972, Roger Bissonnette.
9.	--	G-4 Directions for Using HP Assembler Lee Castonguay, July 19,1972.
10.	--	D.53 - BCS Driver for 9TRK MT(DMA) Daconics (driver for 9 track tape driver controller)
11.		8K SIO MAG TAPE DRIVER DACONICS, December 1971.
12.		A22-6866 IBM Tape drive manual,IBM Publications.