national accelerator laboratory

TM-526
2330.000

A PRIMER ON THE 032 MODULE -- LINKING BEAM LINE AND EXPERIMENTER COMPUTERS

John Nagy
The Johns Hopkins University

October 1974

I. Introduction

The 032 Serial Memory Buffer is a general purpose CAMAC transmit/receive module. Two computers, each with a branch driver and crate, may exchange information over a distance of the order of a Portakamp length. Specifically, 032 modules allow direct connection of the beam line control computer to an experimenter's on-line computer.

The module was designed and is maintained by the Controls Group (Lau, Burkhart, Lamantia, Schmidt). The MAC-16 beam-line software is also the responsibility of the Controls Group (Hepinstall, Michelsen). Maintenance and software of the experimenter's half of the system is the user's responsibility.

The purpose of this note is to help an experimenter get started. The hardware and MAC-16 software are covered in detail elsewhere (Refs. 1 and 2). Here is collected in cookbook fashion all the information needed to turn on, run, digest input, and debug. The point of view is an experimenter's. In addition, the paper contains some facts that were learned by word of mouth or through experience.

II. Uses

Though the hardware is general, controls group software implements only two types of use.

A. Beam Line Device Reading and Status

The 032 can supply the status and/or value of magnets, SEMs, target manipulators, etc. At present, one calls up a "page" on the beam-line console (the TEC CRT). In this case the experimenter does not record the information on his data tape, or one uses 040 Pulse Train modules which are wasteful of space and have limited application.

Mistuning or dropout of a device may produce misleading physics data. The operator can be warned immediately by his computer and the bad sections of data eliminated off-line easily. Thus the 032 supplements the user's own monitoring which may be relatively insensitive to bad beam-line conditions.

B. Setting Devices

Again this is done at present through the controls console either manually or with a USER page. Bringing beam-line or experimenter elements up to correct values or changing running mode can be done quickly and with great reduction in the possibility of errors using 032 modules. Also incoming information provides feedback for possible trimming of devices automatically. Examples might be beam centering or reduction of losses.

III. Block Diagram

Figure 1 is the block diagram. Rather simple. Note that two 032 modules are required, one in each computer system. This will be the most common configuration. Others are possible. See Ref. 1.

The drawing shows the 032 modules connected together with a pair of cables attached to front panel 4-pin LEMO connectors. An alternative is use of the upper card edges on the rear. The top "ACKNOWLEDGE" cable is the control lines which coordinate the interacting 032's (Request To Send and Clear To Send). The "DATA" cable transfers data in and data out (each unidirectional). The lowest connector labelled "MDC" is discussed in Section V. Do not connect them. Since the 032 modules are identical and what is output from one module is input to the other, the lines in the cable must cross, i.e., Pin 1 goes to Pin 2 at the other end. This is true of both cables. Figure 2 and Appendix I show the two ways of connecting modules, front or rear. Use one or the other, not both.

IV. Hardware -- Barest Facts

The following information is summarized from Ref. 1.

The 032 is a double width module. The left (lower) slot edge contains the station line (N) and the look-at-me line (L) and thus determine station number. The upper right rear card edge contains the connections paralleling the front Lemos.

There is an output ("write") buffer and an input ("read") buffer. These are pipeline type shift memories. To the programmer there appears to be only one register for each.

There is a status register in each module showing some conditions inside (Appendix II). This is not to be confused with the MAC-16 software status word. A given 032's status can only be read by the associated computer. (Then you can pass it to the other 032!)

A LAM is set in a module 20 msec after the end of a transmission to that module from the other or when the read buffer is filled. The LAM is cleared when the dataway reads the first word in the input buffer.

The X response is standard CAMAC. The Q response will be 0 if one reads an empty buffer or attempts to write into a full buffer.

032 to 032 transfers are 8-bit bytes. Though MAC to 032 transmissions contain an even number, a given read of the receiving 032 may contain an odd number of bytes. 24-bit dataway reads should be used and the "Q'" bit utilized.[1]

The CAMAC commands are listed in Appendix III.

V. Transmission Rates

The transmission rate between 032's is currently 200 μsec/16 bit. It will be upgraded to 20 μsec in the future. E95 using about 1 meter of cable between 032's never detected any transmission errors at the higher rate. See Section XVII for testing high capacity transmission lines.

The MAC-16 either writes individual words into the 032 (PDC = programmed data transfer) at a rate of 200 μsec/16-bit word or uses a block transfer (MDC) at 36 μsec/32-bit word.

A PDP-11/20 does block transfers (DMA) using a BD011 branch driver at about 5 μsec per 16-bit word.

Appendix IV calculates the time for a common transfer.

VI. MDC Link

The MDC link is a serial block transfer which by-passes the normal dataway input to the 032. It has nothing to do with the data flow between 032's. The MDC link is not implemented yet. The CAMAC initialize (Z) disables the MDC link inside the 032. Make sure it stays that way in the user crate because enabling the MDC disables dataway writes.

The use of the MDC link makes sense at the control group end. The control computer must transfer large blocks of data to many experimenters. The speed and overhead are considerably reduced relative to individual dataway operations.

The experimenter can interface his UNIBUS or whatever to his MDC input. Thus block transfers <u>into</u> the experiment's 032 are the possible. (Beware though that the initial production run of 032 boards has a strapping error. One MDC line goes to 7R rather than 7. Make sure yours has been corrected.) This, however, is a greal deal of expense and bother to gain absolutely nothing. Editing is done only occasionally and all other transmissions from the user are very short. Time and overhead are not a problem. PDP-11 PDT transfers probably operate faster than the MDC circuitry anyway.

VII. Software -- Barest Facts

The following is covered in detail in Ref. 2.

One talks to the MAC-16 in ASC11 code (8th bit on or off). The only exception is the binary "set to" command. The MAC transmissions are also in ASC11 (8th bit on) except for binary readings and status. ASC11 is a convenient standard and this application is neither time nor space critical.

The important commands that the user issues to the MAC are:

E arg — Places the following device list on EXP file arg (1 to 5) on the MAC disk. In response to the R command the list is used to generate the desired information for the experimenter. The editing list is terminated by "! EOD". See Section X and Appendix VIII.

R arg	Causes EXP file arg to be executed by the MAC. The results are automatically transmitted to the user.
S args	For setting of devices

The MAC processes an input string upon reception of a carriage return (15 or 215 octal). Don't expect any response until then and MAC will appear to hang if it never receives the delimiter.

There is a software status word which can be obtained on the console witl 'STEM E. See Section XIII D and Appendix VI. Among other things the status will show if the MAC has been receiving input but no (CR) yet.

The higher byte of a 16-bit transfer is first. This is opposite the PDP-11 convention.

When setting devices from a TEC console, a word or its truncation are allowed, e.g., SET or S, RESET or R, etc. The 032 excepts only the abbreviations S, ON, OF, T, R, P, U, D. Anything else will produce a *CMMD error. NAME, OWNR, ARGU, etc. errors may be returned as with TEC console SET commands.

Old Ref. 2's do not include four error messages which the MAC may generate. Normally, in response to an R command, the MAC issues GOOD, waits, collects the data, issues a DATA message, and later sends the actual data. If there is an error in the device list, an error message is transmitted immediately and no data at all. (The E command does not look for the following error types. It is simply an editor.)

	Error	Example
DLOP	device list operation	RE(SE309) should be RD
DLFM	device list format	blank page, a DE not first
DLTM	device list time	DE(X) should be I or a number
DLNM	device list name	RD(SM309) misspelled or not named

One need not "own" a device to read it but must to set it. A share in ownership is obtained with the console command ASSIGN which requires unlocked console. (See crew chief.)

VIII. Experimenter Console Setup

This section concerns only the MAC end of the system. One must prime the MAC software, i.e. define location of your 032. In lieu of the following, the controls group can and should make the definition of your experimental console a permanent part of their software.

A. Console Number

First determine your beam-line console number. This is not to be confused with your crate number. On console type P 1 2, a page will appear with PAGE 1, Cn on the lower right. Your console number is n. Say it's 3.

B. Crate Number

On console enter P2 and DEFINE 2. A list will appear which looks like

C1 = 150, 19	E1 = 150, 7	L1 = C1, C7, C8
C2 = 50, 16	E2 = NONE	L2 = NONE
C3 = 95, 16	E3 = NONE	L3 = C1, C7

This shows that console #1's CTY console controller is in crate 150, slot 19. It also has an experimenter console controller (i.e., an 032) in slot 7. Console #3 (you) has no 032 board defined. Your crate number is 95.

C. Definition of 032

This requires an unlocked console. The beam-line crew can define your experimenter console.

First though plug the 032 into the beam-line crate. The station number will be the left slot. Say it is 1. Type on console DEFINE E3 AS95,1. Note agreement of console and crate numbers. Problems will occur if one tries to mix different beam line vs experimenter console numbers. The message *LOCK indicates a locked console. A correct definition causes no response.

D. Check

Type DEFINE ? again, response should be

C3 = 95,19 E3 = 95,1 L3 = C1,C7

and SYSTEM EXP , response looks like

E1 = 150,19
E2 = 0,0
E3 = 95,1 #32 EM = 66CA STATE = 0100 0000 0000

Experimenter #1 must have some sort of problem. The MAC either cannot find crate 150 or the 032. The software has found an 032 in the proper place for experimenter #3. MAC location 66CA (hex) is the beginning of the core input/output buffer. See Appendix VII. The MAC state shows that the software is idling. See Appendix VI.

E. Undefining

Type DEFINE En AS 0. (0,0 doesn't work).

IX. Normal Sequence of Events

At this point assume that the user software is running, both 032 modules are OK and connected together. The discussion below describes what happens when during 032 operation. In the following nothing bad occurs.

A. Program Start

At this point one will (1) edit experimeter device list pages and (2) turn on devices and set values. Automatic editing is less tedious and less error prone than manual editing from the console. However, when debugging user software it is easier to forget editing and manually enter a short list on the TEC CRT console. (See Appendix V.)

If the MAC must be rebooted or loaded from tape, it is possible, but not probable, the definition of the experimenter's console 032 may be lost or obsolete experimenter files put onto the MAC disk. Check after any occurrence.

B. Change Running Mode

The experimenter will wish to (1) change device settings such as magnet currents and spectrometer angles and (2) possibly changing the device list to be read. One cannot edit pages or change file number requested every accelerator cycle because of device list setup time.

If one must change a list of readings, it can be done by either re-editing the page being used or reading a different page. Neither can be done within one accelerator cycle. When the first R n command is received, disk device file n is setup in core. It remains there as long as an R n or K n command comes every 4 or less cycles.

If a page is re-edited during running, only the disk copy changes. To get into core one must request another page and then the first page again. Or the change can be forced manually by putting the beam-line crate off-line for a cycle.

When an R 2 command is issued and R 1 had been used, a few seconds are required for the switch. The MAC will respond "SOON" to R 2 and no data will be transferred. The change may not occur in time.

C. Every Pulse

After each spill the experimenter will (a) read the device list, (b) digest readings, (c) possibly issue warnings, and (d) adjust device settings.

The following three sections detail the procedures above.

X. Editing

Appendix VIII is a PDP-11 example of the editing procedure for EXP 1 starting at EDIT32: and ending with .WORD 0. The latter is an end sentinel not transmitted by the user software. The remainder is transmitted. The "E 1(CR)" is the "operation" and following is the "data" to be put on the experimenter page. The "!EOD" is the MAC software's end sentinel which (as well as E 1) does not appear on the edited page.

The MAC will respond with "GOOD (CR)" for each line sent to the editor. It will answer "CMMD (CR)" for a command error [e.g., "R1(CR)" or "ARGU(CR)" for a page out of range (1 to 5).

This list will be used as an example later. Section XII describes the actual transfer software from user to MAC.

An experimenter file can be inspected on the console with H EXP, n 2.

XI. Requesting Device List Reading

The character string beginning at KICK32: (Appendix VIII) is required to initiate a read. The second blank is filled in by the software depending on which page is desired.

Below is the typical sequence of events using the experimenter page above as an example.

(a) Transmit "R 1 (CR)" at T2. Issuing more than one R command before the first is satisfied will result in an ACTV error for the extra commands and the MAC will ignore them.

(b) Normal response is "GOOD (CR)." Others might be "CMMD" for bad syntax, "ARGU" if page out of range, or "SOON" if some other page besides 1 is in core. The device list values are not read at this time.

(c) At T6 some devices are read. This is the purpose of the DE(I) command which must be the first in the list. It means delay reading until the MAC interrupt. This time is determined by a pre-det in the MAC room under the cross gallery and is used for other purposes. It is usually set to $T6+\epsilon$. The timing method of interrupts and delays will be made more versatile in the future.

(d) At T6 + 0.5 sec the devices below DE(30) are read. DE(30) means delay an additional 30-line voltage cycles (i.e., 0.5 sec).

(e) Now the software has collected everything and issues the message "DATA 1, 9(CR)" indicating that binary data is coming from page 1 and consists of 9 devices. This includes dummy "readings" from the DE commands.

(f) One-quarter second later the binary data flows. The delay is in the MAC software to allow the user to clean out the 032 modules. The controls group assures us that no LAM'S will occur between that due to the "DATA" message and the LAM from the binary data.

The read buffer is 128 bytes long. At 32 bits per device this is 32 devices. If the user does not process the incoming data faster than it comes a LAM will occur when the read buffer is full. When the remainder of the data is transferred, another LAM will be raised.

XII. User Output (To MAC-16)

As shown in Appendix VIII generation of a string of command characters with PDP-11 .ASC11, .BYTE, and WORD directives is easy.

A list cannot be transferred to the 032 as a DMA (direct memory access = NPR = block transfer) with a BD011 even with BB = 1. This is because the 032 recognizes only F(16)A(0) and not F(16)A(X) or just F(16). Thus transmission into the 032 will halt after one transfer since F(16)A(1) will not respond with Q = 1 and one will write into modules further down the crate. Thus a PDT loop (programmed data transfer) must be set up.

Another reason for a PDT loop is byte order incompatibility between the MAC16 and PDP11. The MAC wants first character in higher order byte

1st word	R	;P)
2nd word	1	(CR)

whereas the .ASC11 directive will code

1st word	(SP)	R
2nd word	(CR)	1

since the PDP11 convention is lower location is first byte. Thus a byte swap is required before transmission.

During transmission if a no-Q is received then (a) you've backed up the output memory, (b) 032 hardware problems, or (c) MAC software problems.

Figure 3 is a block diagram of a working program. R5 carries the first address of the string. A word of all zeroes terminates the transmission of a string. The write routine is entered by the Task Complete Ready from the previous write.

XIII. User Input (From MAC-16)

Figure 4 shows the flow chart for a simple handler for information received from the beam line computer.

The F(4) command is a read which ignores A. The data are read and then the next 16-bit word in the pipeline is shifted into the register. Thus

BD011 with BB = 1 transfers can be used. This example uses it only for the binary readings. The more elegant approach is to have an input buffer in the PDP 11 itself which is then analyzed after block transfer. The 032 LAM must be enabled again after use of F(4).

The example routine does not take into account the possible back up of the read buffer since handling this situation has never been needed. Checking the number of the device read stated in the DATA message with the actual amount of the binary data received would be helpful.

The ability to run in a mode in which transmissions from and/or to the MAC-16 are dumped is helpful in debugging.

When DATA message is encountered a flag is set indicating that the routine will handle the binary data at the next LAM.

MODES:	1 -	"GOOD (CR)"	Ignored
		"DATA..."	Set flag Calculate length
		"XXXX (CR)"	PRESUMED ERROR TYPED OUT
		binary	DMA'd Into Core
	2 -	non-binary	Typed as ASC11
		binary	Flushed
	3 -	all	Typed in ASC11 and octal

XIV. Interpretation of Binary Data

Appendix IX shows the format of transmissions from the MAC-16. The link status byte should be 40_{16}, 41_{16}, 50_{16} 51_{16}. Otherwise, the reading in the following word is meaningless.

Most binary readings (magnet currents, SEM's, etc.) are ADC outputs scaled 2^{15} counts = 10.24 volts. The user must know the conversions to physical units such as amperes or protons, as well as fine corrections or possible nonlinearities. The integrating capacitor of SEM sample and holds are usually selected so that 1 millivolt corresponds to some order of magnitude of protons (e.g., 10^8). Magnet current monitoring resistors are likewise selected so that 1 count is some reasonable fraction of an ampere (e.g., 1/500).

If 2^{15} counts = 10.24 volts then 2^5 counts = 0.01 volts or 1 count = 1/3200 volt. The number of counts must be multiplied by 1/3.2 = 5/16 to obtain the reading in millivolts. Figure 5 shows the flow chart of the MAC-16 routine which converts the binary input for output to the 040 pulse train generators. It is these numbers which the user is familiar with as scalar readings. One may wish to simulate the conversion and display the number using his own computer just so one is dealing with the same units as before. No adjustment in thinking or analysis software is then required.

Now that the experimenter has the raw data available himself, he is not limited to the above. He might use readings such as spectrometer magnet current and spectrometer arm angles to display and record the central momentum, or calculate secondary quantities such as split ratios, targeting efficiency, loss factors, etc.

XV. Display of 032 Information

The 032 is a blind system. The experimenter will wish to display certain values either on a pulse by pulse basis or integrated. With the 032 and a versatile computer display, one can be more sophisticated then the 040-to-visual scaler or 040-to-blind scaler-to 72A. There are many options.

Reference 6 describes one method which makes use of a CAMAC output latch and the ADDS MRD-200 Raster Generator.

XVI. Stability of Device Readings

The question arises concerning the dependence of a reading on when it is read. To test this three proton lab devices, a SEM and two loss monitors were read as the second and sixteenth device in the list at T6 and T6 + 0.5 sec for the same spill. The average and standard deviation of the differences were calculated. The average discrepancy was never greater than one millivolt (maximum 10240 mV). The standard deviation (pulse to pulse variation) was never greater than 2.5 mv./(1 A/D bit) These numbers seem to be independent of the magnitude of the reading. Thus the percentage discrepancy decreases with magnitude. A magnet reading was rock solid. For greatest accuracy calibrate devices under the same conditions as you read them when running.

XVII. Debugging Aids

(a) SYSTEM E
- (i) Correct Definition
- (ii) Response of 032
- (iii) Software Status Word
- (iv) MAC core dump (See Appendix VII.)

(b) 032 Status Word (A0, F1)
- (i) Beam Line (Link Test)
- (ii) User

(c) Link Test (See Help 40ff or Consult) and Equivalent User Routine

(d) Routine to Dump Transmissions to User on TTY or CRT

(e) Shorting Stubs for Isolation of Computers (Rear Off! ?!)

(f) Test Transmission Cable by Shorting Ends. Program write/read loop and check.

(g) Disabling User Branch Demand

(h) Beam Crate Online/Offline Switch. MAC 032 and Software Initialized. INIT Sent to User.

(i) 032 and MAC-16 Exerciser Program With Complete Diagnostics Requiring 24K Core and Running Under Dos (/45 Model PDP 11 Only)

XVIII. Shorting Stubs

It is very useful during debugging of software or hardware to isolate the two 032 systems, beam line and user. The task is simplified tremendously because one does not have the other system competing with the experimenter by interacting with the first system. In one instance isolation allowed debugging PDP 11 software on a computer department machine before either the control console or on-line PDP 11 were available to the programmer.

Isolation is achieved by merely constructing two shorting LEMO connectors by wiring pins 1 and 2 together (3 and 4 are chassis grounded internally). This causes ACK and DAT output to become input to the same 032 module (hermaphrodite mode).

An example of their use is checking the output character string. By routing input to the teletype, the actual transmission is displayed. Common problems picked up this way are byte reversal, encoding errors, and character string beginning or end problems. The sections on hardware testing an 032 give other examples.

XIX. Testing of User Crate 032

The following procedure checks the 032 module hardware. It is assumed that the user has a CAMAC debugging routine with which he can initiate dataway operations from the teletype (e.g., Ref. 7). The readings are presented in the form OCTAL = DECIMAL = HEX in the following.

(a) Disconnect cables to control crate 032.

(b) Read module number C, N, A0, F6. Should be 40 = 32 = 20. This checks the CAMAC system and shows you have correct C and N.

(c) Read status A0, F1. See Appendix II.
   (i) Normal reading is 225 = 149 = 95. The write and read buffers are empty and "clear to send" not ready since other 032 absent.
   (ii) If needed clear read and write buffers A0, F9, and A1, F9.
   (iii) If needed disable MDC A0, F28.
   (iv) Recheck status.

(d) Disable branch demands N30, A10, F24.

(e) Write a 16-bit word into 032 A0, F16, data.

(f) Read status A0, F1. Normal reading is 224 = 148 = 94 indicating write buffer is not empty.

(g) Put in DAT shorting stub and then ACK shorting stub. Branch demand light should come on.

(h) Do graded-L N30, A0, F0 and check slot demanding service.

(i) Read status A0, F1.
   (i) Normal reading will be 205! CTS now satisfied and write buffer emptied. Read buffer also shows empty. The one word we have written is in the output register and there is nothing in the shift memory. Apparently only the latter is tested to set the status bit.
   (ii) 224 = 148 = 94. Something is wrong with hardware or shorting stub.
   (iii) 204 = 132 = 84. Hardware problem.

(j) Write another data word (see e).

(k) Read status. Normal reading 201 = 129 = 81. The shift memory is no longer empty.

(l) Read one data word (A0, F2/F4).
   (i) Should get Q = 1 and correct data.
   (ii) Branch demand goes off.

(m) Keep reading until no-Q.

(n) Read status 205 = 133 = 85.

(o) Cram >320 bytes in. (edit page a few times)

(p) Read status 232 = 178 = E8. Buffers full, CTS not ready.

(q) Try to write again. Should get no-Q.

One can check the 032 hardware using just the shorting stubs and the normally running program. If the program runs in a mode in which all transmissions are dumped onto the teletype or unrecognizable messages are dumped, just start the program running. The editing list will be printed at initialization. If a pre-det triggers a read request, "R 1" will appear every cycle. This is just the converse of using the 032 hardware to check your user 032 software.

XX. Testing of Beam-Line 032

The easiest thing to do is switch modules and repeat (XIX). In principle LINK TEST could be used but the MAC-16 software interferes with the testing. The 032 could be undefined or moved to a different slot.

Debugging

1 Check console. Does it respond normally? If not. . .
- (a) MAC down - is whole line down?
- (b) Hardware problem upstream of your repeater - only part of line down (check with TEST and TC command on working console)
- (c) Your crate off-line
- (d) Crate power supply problems (check all ±6, ±24 volt)

2. SYSTEM EXP
- (a) Does #32 appear? Bad definition - check C and N (even if #32 there, it is conceivable you defined somebody elses 032)
- (b) Software status word

Problems

"Clear to send" flag down
- (a) Receiving buffer is full (status = 270 octal)
  - (i) The software is not reading out transmissions to it
  - (ii) LAM may be disabled or the LAM not processed for some reason

(b) Cabling bad - If the buffers are empty, replace cable by shorting stub or another cable which is known to be good (try the DAT cable)

(c) 032 bad

032 MODULE REFERENCES

[1]Norm Lau, Dave Burkhart, Sandy Lamantia, Karl Schmidt, CAMAC Module 032 - Serial Memory Buffer, 15 November 1973 (hardware specs).

[2]Larry Hepinstall, John Michelsen, Control System Computer Users Manual, Section 4, 13 May 1974 (MAC-16 software. A later version is current.)

[3]Drawings: 0815.032-CD-34459
0812.032-CD-34554

(Schematics of boards A and B of 032 module). The documents above are available from Fermilab Controls Group, CL-14E.

[4]HELP Files on MAC-16 console (H 1, etc.)

[5]Ron Truxton, Experiment 45 Beam Line Data Collection Program, 20 August 1974, (describes a system which uses MiniBison software, 8K of core, and a KS011 to handle 032 module).

[6]Interfacing CAMAC System to MRD-200 Raster Generator, Fermilab TM-525, October 1974.

[7]Jeff Mack, CAMAC Diagnostic Program, (to be a computer group PN).

FIGURES

APPENDICES

ENGINEERING NOTE

SUBJECT

Figure 1

BLOCK DIAGRAM OF 032 SYSTEM

NAME

DATE

REVISION DATE

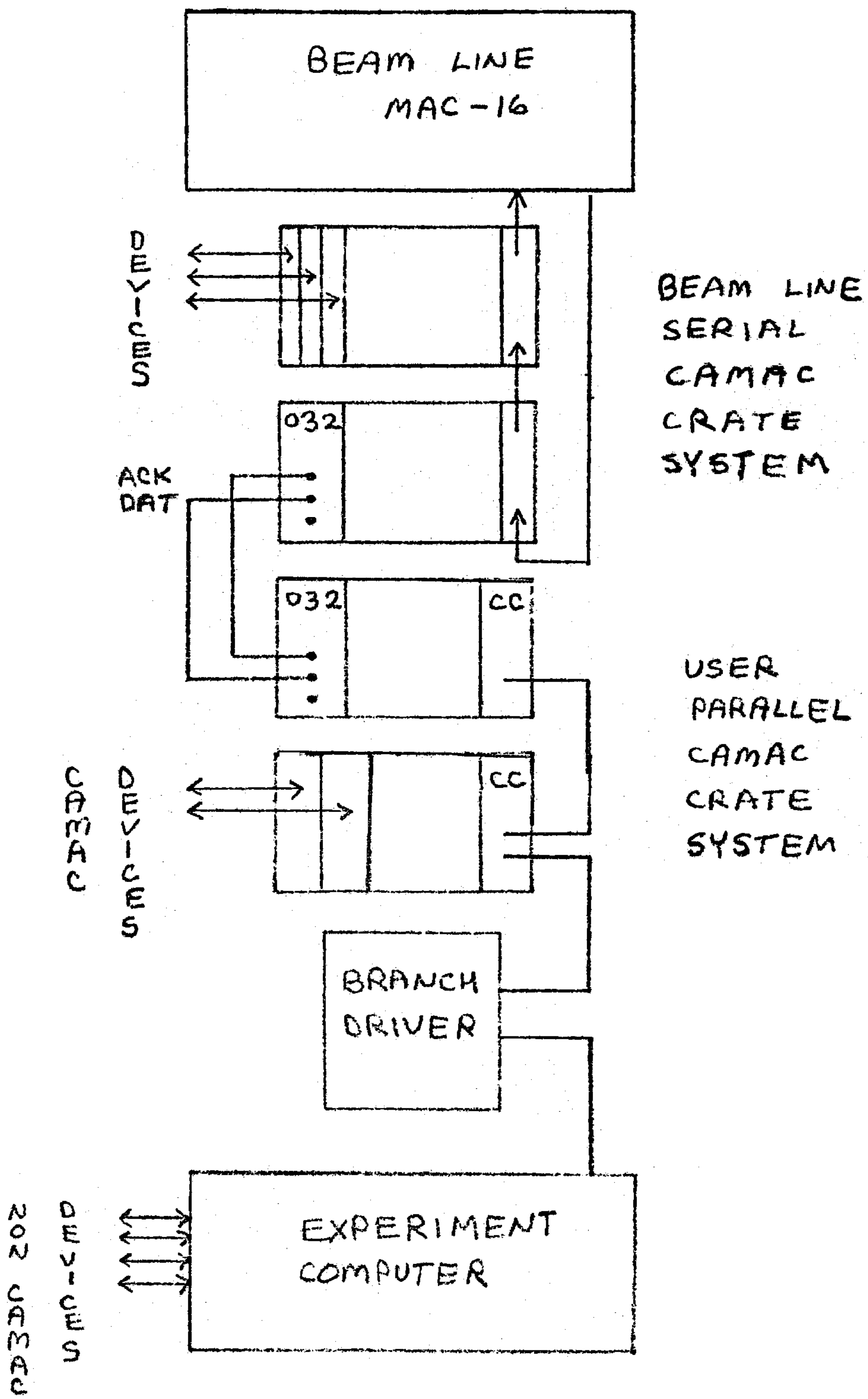
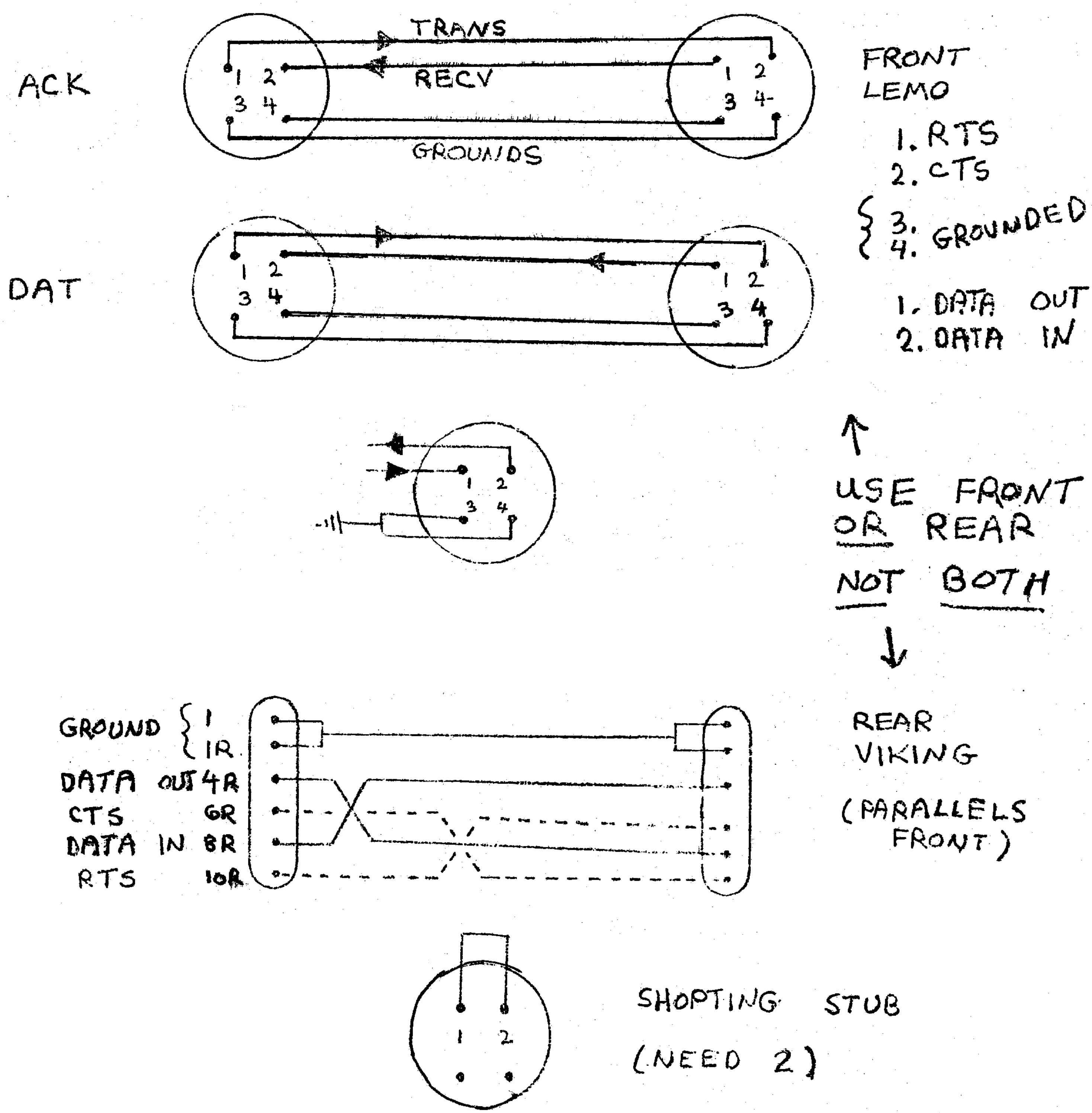


Figure 2

032/032 CONNECTION

CROSSING CABLES



PDP-11 OUTPUT ROUTINE

MOV #EDIT32, R5
JSR PC, WRIT32

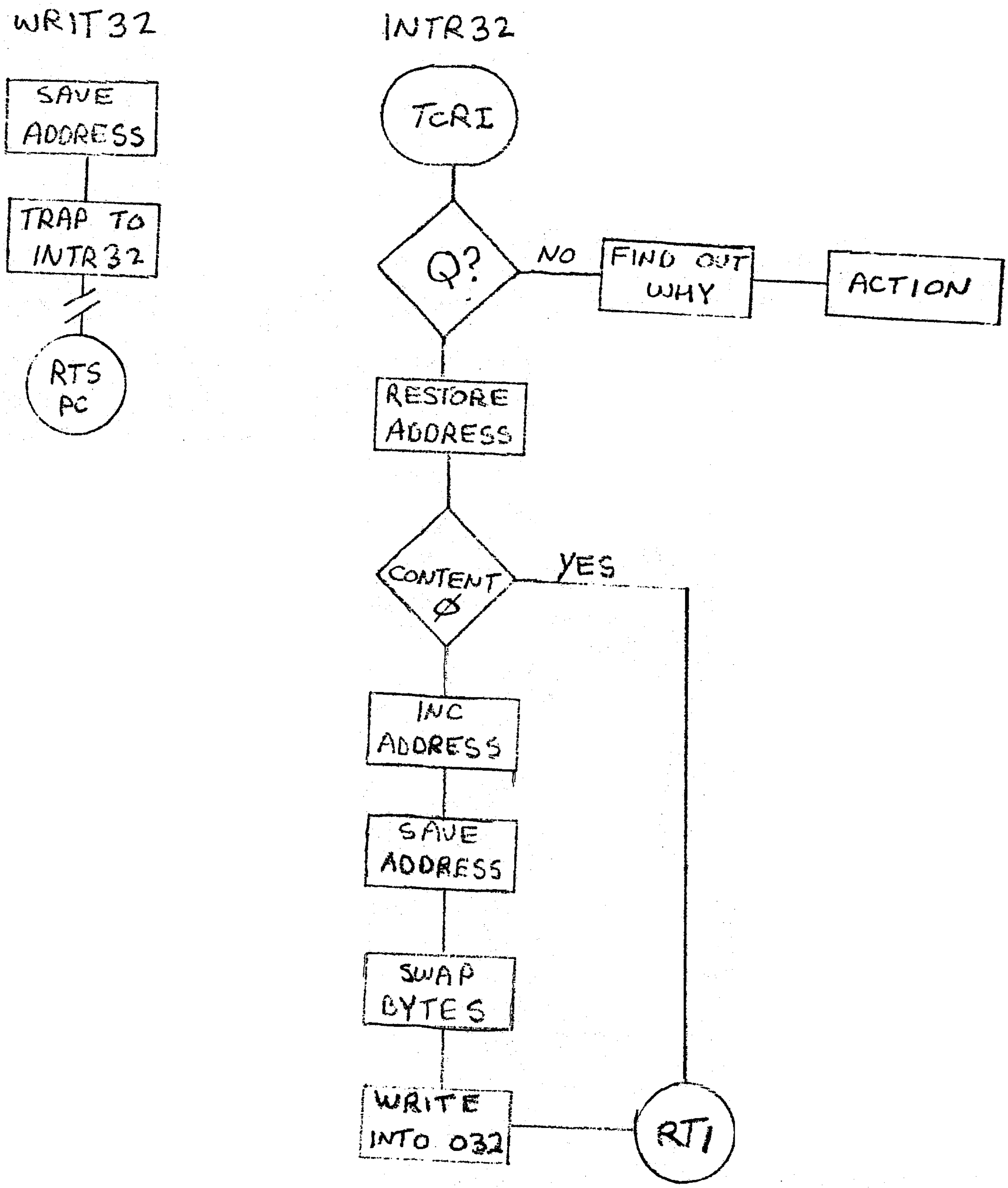
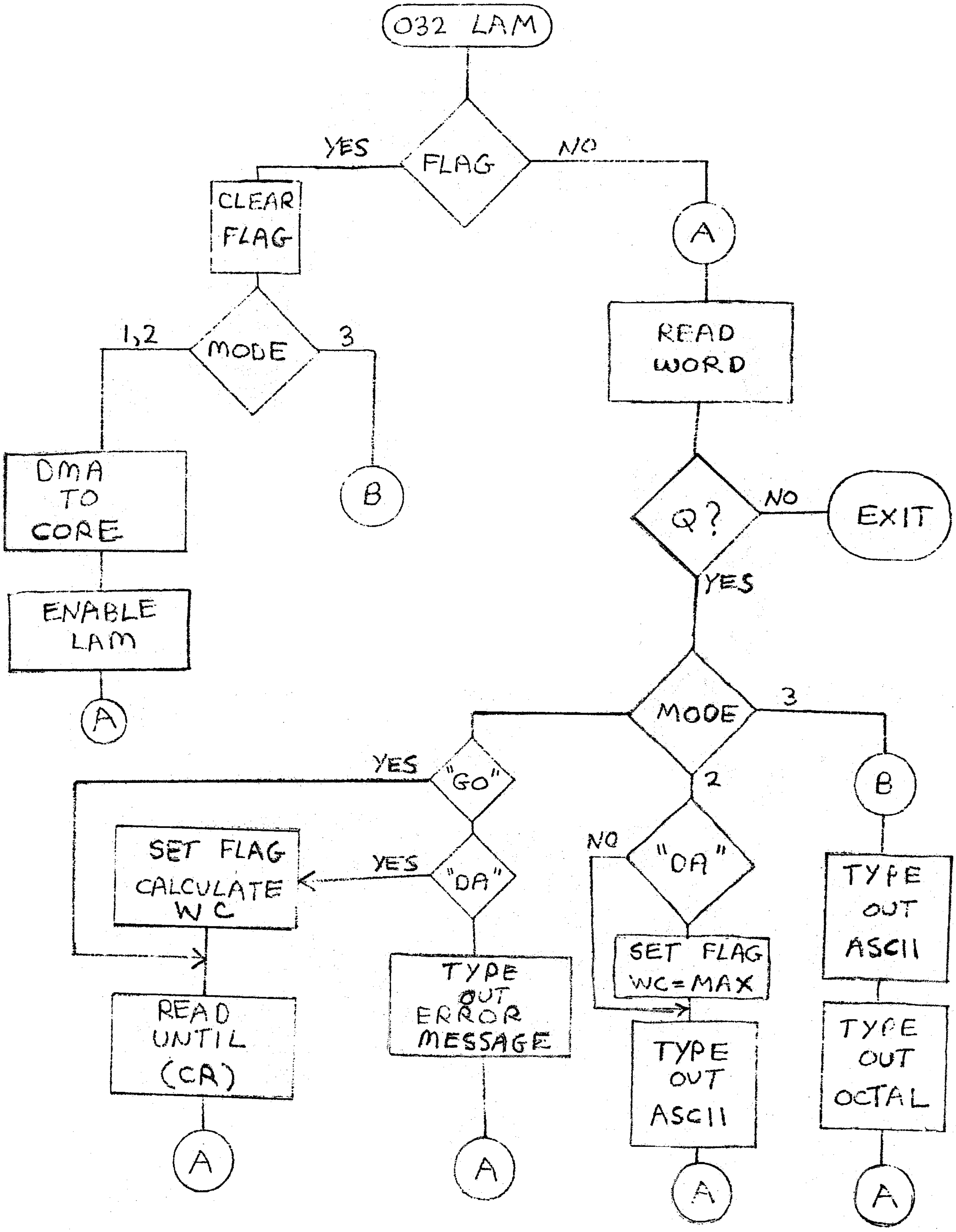


Figure 3

Figure 4

PDP-11 INPUT ROUTINE



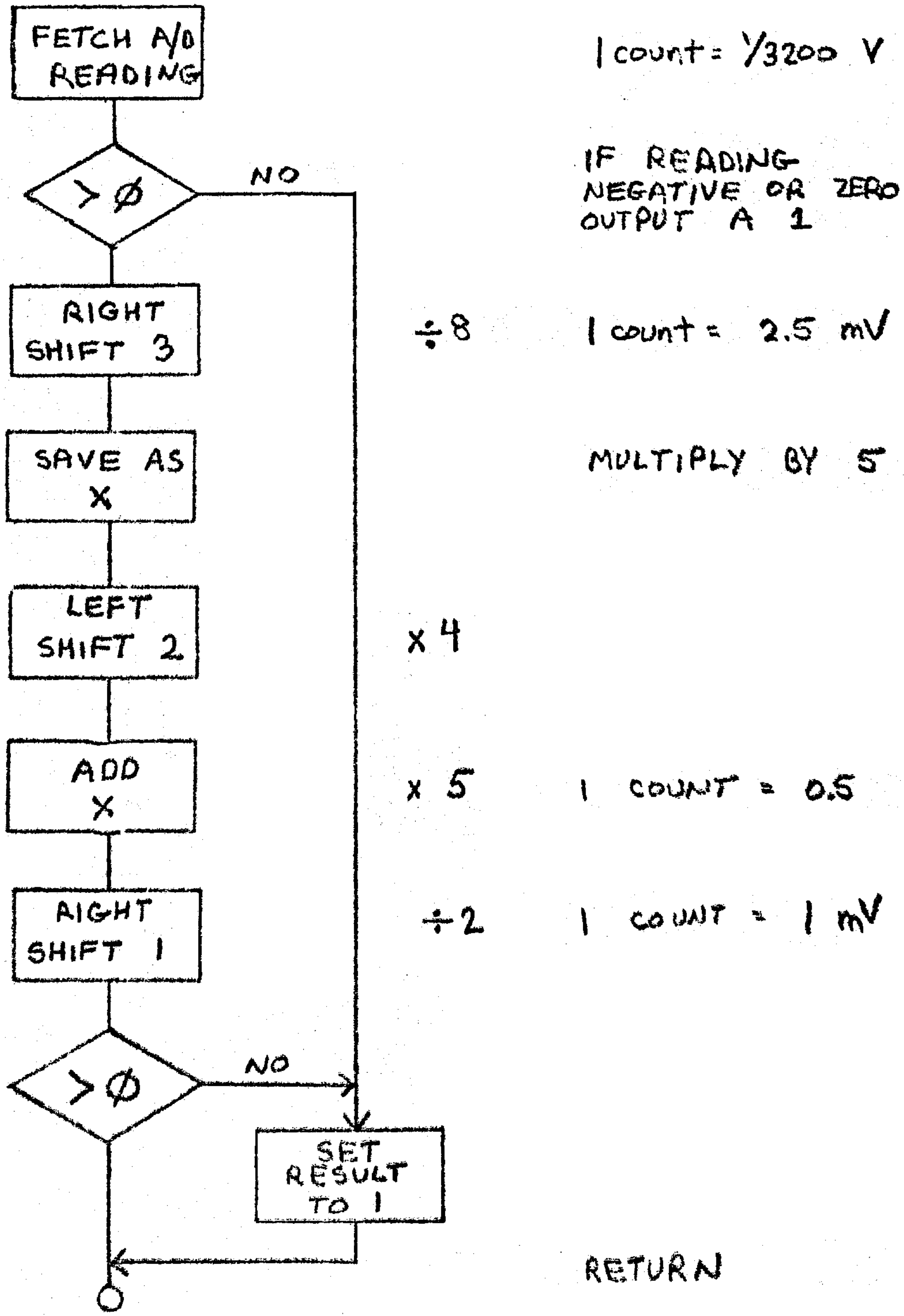
ENGINEERING NOTE

SUBJECT: Figure 5

MAC-16 040 PULSE TRAIN SOFTWARE

NAME

DATE | REVISION DATE



APPENDIX 1

032 FRONT PIN AND REAR CARD EDGE CONNECTIONS

(Front and rear are paralleled and eqivalent)

(Rear connections on upper right card as viewed from front.)

FUNCTION	LEMO FRONT PIN NUMBER	REAR CARD EDGE SCHEMATIC NUMBER	VIKING 2x18 EDGE CONNECTOR
Request to Send	ACK 1	10R	10A
Clear to Send	ACK 2	6R	6A
Chassis Ground	ACK 3,4	1,1R	1B,1A
Transmit Data	DATA 1	4R	4A
Receive Data	DATA 2	8R	8A
Chassis Ground	DATA 3,4	1,1R	1B,1A
MDC	MDC 1	5	5B
Differential inputs	MDC 2	3	3B
2 Pairs	MDC 3	9	9B
	MDC 4	7	7B

Looking from front card edge connections look like	TOP	1 1R	1B 1A
		2 2R	2B 2A
		etc.	etc.

APPENDIX 2.

HARDWARE STATUS WORD

A0, F1

IF BIT = 1 MEANS

UNUSED 0 / ALWAYS ON	MDC LINK ENB	REQUEST TO SEND N.R.	CLEAR TO SEND N.R.	READ BUFFER FULL	READ BUFFER EMPTY	WRITE BUFFER FULL	WRITE BUFFER EMPTY
7	6	5	4	3	2	1	0

TYPICAL READINGS

octal	hex	decimal	
205	85	133	buffers empty
204	84	132	W.B. not empty
201	81	129	R.B. not empty
225	95	149	CTS not ready
226	96	150	W.B. full
270	B8	184	R.B. full
305	C5	197	MDC enabled

APPENDIX 3. CAMAC COMMANDS

F(1)·A(0)·N	read status word
F(2)·A(0)·N	read data and clear at S2 LAM cleared data buffer shifted up one
F(4)·N	read data, MDC mode. Used for BD011 DMA.
F(6)·A(0)·N	read module number 32(10) = 40(8) = 20(16)
F(8)·A(0)·N·S1	test LAM
F(9)·A(0)·N·S1	clear WRITE buffer
F(9)·A(1)·N·S1	clear READ buffer
F(10)·A(0)·N·S1	clear LAM
F(12)·A(0)·N·S1	transmit first bytes in WRITE buffer (?) (strap option on board)
F(16)·A(0)·N	write data into WRITE buffer
F(24)·A(0)·N·S1	disable LAM
F(26)·A(0)·N·S1	enable LAM
F(28)·A(0)·N·S1	MDC input link OFF
F(30)·A(0)·N·S2	MDC input link ON
C, I	not used
Z	clears both buffers clears LAM MDC link OFF LAM enable/disable flip-flop not affected may power up in undesired state use software

APPENDIX 4

TRANSMISSION RATES

DEVICE	BASIC RATE	40 DEVICE READINGS[2]
1. (a) MAC PDC	200 usec/16-bit	16 msec
(b) MAC MDC	36 usec/32-bit	1.5 msec
2. (a) 032 @1MHz	10 usec/byte[1]	1.6 msec[3]
(b) 032 @100kHz	100 usec/byte[1]	16 msec[3]
3. PDP11/20 DMA	c. 5 usec/16-bit	1 msec [2,4]

[1] A byte transfer involves two extra bits for a total of 10 bits.

[2] Total of 160 bytes. Thus two reads of read buffer required.

[3] In addition there is a delay following eah filling of read buffer before LAM is generated.

[4] Added 20% for LAM handling and setup.
Used 32-bit reads of 032 register to obtain Q1 bit so PDP-11 handles twice as much information (320 bytes).

MANUAL EDITING OF EXPERIMENTER PAGE

NOT SUGGESTED FOR FINAL RUNNING PROGRAM
(LABORIOUS, ERROR PRONE)

SUGGESTED WHEN FIRST DEBUGGING SOFTWARE

P ↲

EDIT EXP, 1 ↲ (NOW IN EDIT MODE)

DE(I)
ST (CLOCK)

ESC KEY Y ↲ Y ↲ (CLOSE FILE)

INSPECT PAGE WITH " H EXP, 1 ↲ "

ST (CLOCK) RETURNS THE DATE IN BINARY CODED DECIMAL. IT IS EASY TO INTERPRET AND EVERYONE KNOWS THE ANSWER.

MANUAL	AUTOMATIC
EDIT EXP, n ↲	E n ↲
character string	character string
ESC, YES ↲, YES ↲	EOD ! ↲

MAC-16 SOFTWARE STATE [REF 2]

The meanings of the state bits are as follows. The bits and groups are numbered left to right here:

Group 1:

Bit 1 = Initialize

Bit 2 = Input ready (no characters read)

Bit 3 = Input active (1 or more characters read)

Bit 4 = Processing a command

Group 2:

Bit 1 = In EDIT mode

Bit 2 = In EDIT mode; file full

Bit 3 = Device command table being build

Bit 4 = Waiting for binary data of a SET - binary command

Group 3:

Bit 1 = Message being output

Bit 2 = Data block output mode

Bit 3 = Waiting for empty output buffer

Bit 4 = Data block being output

COMMON READINGS:

0100	0000	0000	NORMAL QUIESCENT
0000	0000	0110	WRITE BUFFER FULL
0010	0000	0000	DATA READ BUT NO CR YET

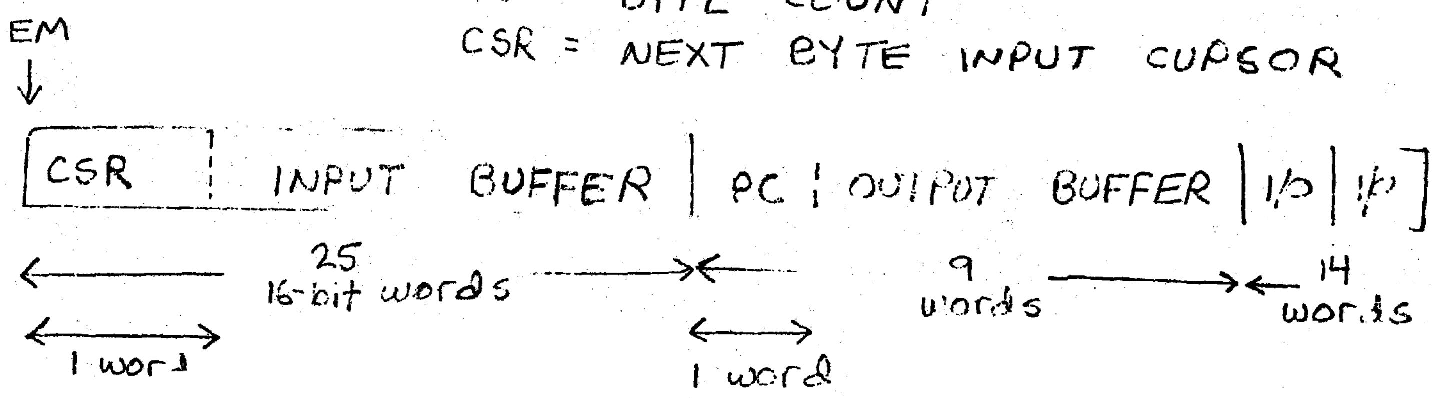
APPENDIX 7

MAC CORE INPUT/OUTPUT BUFFERS

(1.) SYSTEM EXP ↵ on CRT corsole gives beginning address in hex, e.g. EM = 66F3

(2.) Structure of Buffer

BC = BYTE COUNT
CSR = NEXT BYTE INPUT CURSOR



(3.) R is CRT corsole command to dump core

a) R 66F3,+25 — dumps input region
this region emptied upon receipt of (CR)
therefore usually empty but will have data if you haven't sent (CR) yet (debugging aid)

example: 66F[illegible]-0005 66F4-D2A0 66F5-B5A0 66F6-A0AC etc.
R ½ 5 ½ ↑ cursor

b) R ,+9 — dump next 9 words
last output buffer sent

example: 6700-000C 670D-C401 670E-D4C1 670F-A[illegible]
6710-A0B5 6711-A[illegible]0 6712-B1B6 6713-[illegible]
6714-A0A0

APPENDIX 8.

EDITING AND OP TABLES

```
EDIT32:   .ASCII   /E  1/<15>
          .ASCII   /DE(I)/<15>
          .ASCII   /ST (CLOCK)/<15>
          .ASCII   /RD (CLOCK)/<15>
          .ASCII   /RD (PULSE)/<15>
          .ASCII   /RD (SE309)/<15>
          .ASCII   /RD (MVT310)  /<15>
          .ASCII   /DE(30)/<15>
          .ASCII   /MN (TEST)  /<15>
          .ASCII   /RD (SE309) /<15>
          .ASCII   /!EOD/
          .WORD    Ø
KICK32:   .ASCII   /Rb̸b̸/<15>
          .WORD    Ø

MAG32:    .ASCII   /S  QH600  R/<15>
          .ASCII   /S  QH600  ON /<15>
          .ASCII   /S  QH600  T        /<15>
          .WORD    Ø
```

.ASCII /DE/ ASSEMBLES

E	D

MAC-16 WANTS

D	E

APPENDIX 9. FORMAT OF MAC-16 TO 032 TRANSMISSIONS

1.

ALL MESSAGES (anything not a binary reading or status) is 8-bit ASCII. The M.S.B. is always on. The highest order byte of the 16-bit word is first. Appendix 10 lists the code in hexadecimal. The latter is also useful when interpreting core images of your MAC-16 buffer area. Below is an example of a message (Ref. 2, page 4.3-3).

octal	hexadecimal	character
142301	C4 C1	D A
152301	D4 C1	T A
120240	A0 A0	
120263	A0 B3	3
126240	AC A0	,
120666	B1 B6	1 6
120215	A0 8D	(CR)

2.

BINARY DATA [except DE(), PULSE, and CLOCK]

(a) 2 16-bit words

LINK	TOP BYTE
BINARY DATA	

(b) LINK BYTE

R9	bit 0	unused
R10	1	1= Q time out
R11	2	1= transmit error
R12	3	1= device in LOCAL

R13	4	1= CAMAC X response
R14	5	1= demand present
R15	6	1= CAMAC Q response
R16	7	1= receive error

40 or 50 (hexadecimal) are normal values. Other values indicate device module or CAMAC system problem. (Bit 0 might also be up, i.e. 41 or 51.)

(c) TOP BYTE - Most significant byte of a 24-bit device reading, e.g., module 211 scalar. Zero otherwise. Can also be non-zero if there is a link problem [see (b)].

(d) BINARY DATA

Readings are sign plus 15-bit magnitude. Negative values are two's complement. Most devices read via MADC are scaled 2^{15} = 10.24 volts. Not all devices have 15 bits of precision. The significant bits are still left justified in 15-bit field and the rest zero. For example, a 12-bit readout would look like SXXXXXXXXXXXX0000.

If the status of a device is requested rather then the reading, the interpretation of the resulting bit pattern depends on the device and the MAC software.

If the link status is bad, the binary data are meaningless.

3. "PULSE"

Obtained with RD(PULSE). Same as BINARY DATA above except that second word is unsigned 16-bit word. The pulse counter is reset at midnight.

4. "CLOCK"

Information returned in binary data string but it is 4-bit decimal code. Link status is 40 (hex).

Date	ST (CLOCK)	040020	0100/0000/0001/0000	10	10/14/74
		012164	0001/0100/0111/0100	14 74	
Time	RD (CLOCK)	040030	0100/0000/0001/1000	18	18:22:31
		021061	0010/0010/0011/0001	22 31	

		MM	MM
DD	DD	YY	YY
		HH	HH
MM	MM	SS	SS

5. "DE( )"

Two sixteen bit words are returned for each of these operations in the position they occupy on the experimenter's page. The values have always been 40 00 00 00 (hex)

MAC-16

Appendix 10. USACII Character Set and Hexadecimal Codes

HEX	CHARACTER	HEX	CHARACTER
A0	space	C1	A
A1	!	C2	B
A2	"	C3	C
A3	#	C4	D
A4	$	C5	E
A5	%	C6	F
A6	&	C7	G
A7	' (apostrophe)	C8	H
A8	(	C9	I
A9	)	CA	J
AA	*	CB	K
AB	+	CC	L
AC	, (comma)	CD	M
AD		CE	N
AE	. (period)	CF	O
AF	/	D0	P
B0	0	D1	Q
B1	1	D2	R
B2	2	D3	S
B3	3	D4	T
B4	4	D5	U
B5	5	D6	V
B6	6	D7	W
B7	7	D8	X
B8	8	D9	Y
B9	9	DA	Z
BA	:	DB	] left bracket
BB	;	DC	/ back slash
BC	< less than	DD	[right bracket
BD	=	DE	↑ up arrow
BE	> greater than	DF	← left arrow
BF	?		
C0	@	87	bell
		8A	line feed
		8D	carriage return