THE mini-tasker

DECUS
RT-11 SIG NEWSLETTER

FEBRUARY VOL. 4 NO. 1

Contributions to the newsletter and other correspondence should be sent to:

John T. Rasted
JTR Associates
58 Rasted Lane
Meriden, CT 06450

or to:

RT-11 SIG
c/o DECUS
129 Parker Street, PK-3/E55
Maynard, MA 01754

## FROM THE EDITOR

At the conclusion of the Fall DECUS Symposium I took on the responsibility of RT-11 SIG Chairman. After serving as RT-11 SIG Chairman for three years, Tom Provost regrets that he must step down due to other pressing responsibilities. We wish to thank him for a job well done. He has graciously agreed to act as RT-11 SIG Symposia Coordinator.

I am currently looking for a volunteer to assist in putting together the newsletter, or hopefully willing to take on the job of newsletter coordinator. This individual should be someone who is able to attend both or at least one of the Spring and Fall Symposia.

## SAN DIEGO SYMPOSIUM

RT-11 SIG member presentations in San Diego included interactive sessions such as Hardware Hints and Kinks, and a User Application Panel. Poster paper presentations varied from system extensions such as Extensions to RT-11 Multi-user BASIC, to end user problem solutions such as a BASIC Non-linear Least Squares Curve Fitting Package. Users also gave formal papers on various aspects of their hardware and software applications.

DEC presentations included the RT-11 Product Panel, where V3 of RT-11 was presented and the RT-11 Languages Panel, where the new releases of BASIC-11/RT-11, V2 and FORTRAN IV, V2 were discussed. A copy of the outlines of the above DEC panel presentations will be published in the next newsletter.

## CHICAGO SYMPOSIUM

The Chicago Symposium is shaping up to be one of the most extensive for the RT-11 user. In addition to the usual RT-11 Product and Language Panels, and SIG Sessions, for the first time there will be a 2 hour (or more) RT-11 Technical Session, where those who really want to exploit the RT-11 System capabilities will be able to interact with those DEC people in the know. There will also be a room available for those who wish to discuss RT-11 with other users and DEC RT-11 technical people.

The Hardware Hints and Kinks session and the RT-11 User Application Panel will be continued. No formal presentations are required for these sessions, but users planning to participate should contact myself or Tom Provost.

## STANDARDS ANNOUNCEMENT

A new technical committee of ANSI X3 is being formed to define the character set for optical character recognition. If you would be interested in participating on the committee or on commenting on their "work in progress" please contact:

Patricia M. Caroom
DECUS Standards Coordinator
Box 13047, Capitol Station
Austin, Texas 78711

## REQUESTS

UM

The University of Manitoba

Department of Pharmacology and Therapeutics
Faculty of Medicine
The University of Manitoba
770 Bannatyne Avenue
Winnipeg, Manitoba, Canada
R3E OW3

(204) 786-3553

December 7, 1977

Mr. John T. Rasted
JTR Associates
58 Rasted Lane
Meriden, CT 06450
U.S.A.

Dear John:

We recently needed to push RT-11 to its maximum ADC sampling rate and encountered a surprising delay caused by the RT-11 SJ monitor. It seems that each time the system clock ticks, the monitor services that interrupt at a higher priority than our laboratory peripheral clock (AR11), and takes about 130 microseconds to do its job (11/34).

Our solution was to disable the system clock from interrupts during ADC sampling, thus losing some system account of time, but allowing us to sample considerably below 100 microseconds.

I would be interested in a more elegant solution if anybody out there has one.

Sincerely yours,

Stan Vivian
Lecturer

SV/vmj

THE
BEDFORD
GROUP

25 Industrial Ave., Chelmsford, Massachusetts 01824 (617) 256-99

Anyone have any ideas on reducing memory requirements for large FORTRAN programs?

We have used the following methods thus far:

1) /S switch during compilation
2) /I of module $SHORT during LINKing
3) USR swapping
4) overlays
5) conversion of some subroutines to MACRO-11
6) using subroutine calls rather than function calls when the 'value' of the function is not required
7) controlled shared usage of a large scratch array

Any suggestions would be appreciated!

Alan MacInnes
The Bedford Group
25 Industrial Ave.
Chelmsford, MA 01824

We have developed several useful RT-11 utilities as software products if anyone is interested please write.

---

HELP

---

**the life support systems group, ltd**

30 November 1977

John T Rasted
JTR Associates
58 Rasted Lane
Meriden CT 06450

Dear John:

I received the October Mini-Tasker today, and noted that you included our memo about modifying the DLV-11 for use with buffered printers. Thank you.

Appearently, however, you did not receive our addendum, and corrected memo. These descibe an additional step necessary to avoid problems when using the mods under F/B or other conditions that can lock out interrupts for extended periods of time.

Essentially what can happen is that if the DSR line goes low after the interrupt request has been made, and before the request is granted, there will occur a bus timeout, and resultant system halt.

The revision corrects this situation by re-latching the status of DSR after each interrupt cycle. This means that up to one extra character can be sent after DSR goes low, well within the limits imposed by the printers.

I hope you will publish the revision in the next newsletter, so that we can avoid a distraught DLV-11 users' group. Copies of both documents are included.

I am also pleased to advise that we can provide similar mods for the MDB serial interface, which is also quite widely used.

Sincerely,

Evan

Evan L Solley
President

ENC.

## ADDENDUM
## to
## DLV-11 MODIFICATIONS

It has been reported from Foreground Background user's that the previously released modification instructions for using DLV-11s with Buffered Printers could cause system crashes.

The problem has been that if interrupts are disabled when the Data Set Ready line changes, by the time they are restored the interrupt request will no longer be remembered by the DLV-11. This causes a bus timeout halt.

The following addendum to the previous instructions corrects this problem, by latching the status of DSR. It has been in use now for the last 3 months, without any further problems.

4. Cut pin 4 of chip E32 (7408) from the circuit board. Wire a jumper from pin 9 of E33 (7474) to this pin. This effectively latches the value of DSR until the interrupt cycle has been completed.

For safety, re recommend making this additional revision to all previously modified boards.

## MDB DLV-11 USERS

Similar modifications are available for use with the DLV-11 replacement manufactured and sold by MDB Systems. These mods have also been in use now for over 3 months, without any problems.

15 October 1977

SIR (GPD) 7

TELEGRAPHIC ADDRESS:
"GEOPHYSICS"
TELEPHONE: 738 208

*In replying, please quote*
*these numbers*

NEW ZEALAND

**DEPARTMENT OF SCIENTIFIC AND INDUSTRIAL RESEARCH**
**GEOPHYSICS DIVISION**

P.O. BOX ~~8005~~, WELLINGTON
1320

17 Nov 1977

Dear John

I have just received my first copy of the RT-11 SIG News letter and would like to tell you how much interest there was in it for members of the Wellington LUG. In particular the Hints & Kinks section was most appreciated.

I wondered if you would be interested in a modification we have made to the DL11W board to enable the Baud Rate to be set by an external switch. The advantage of this particular modification is that if it is desired to revert to normal (ie standard) operation it is easy to do so, simply by unplugging the external switch and resetting the on board switch. External pull-up resistors have been included to minimize problems of noise introduced into the additional cabling.

Yours. Sincerely

Ian Calhaem

## Proposed Modification to DL11W interface board.

Purpose

To provide an external switch so that the baud rate can be changed without removing the board from the backplane.

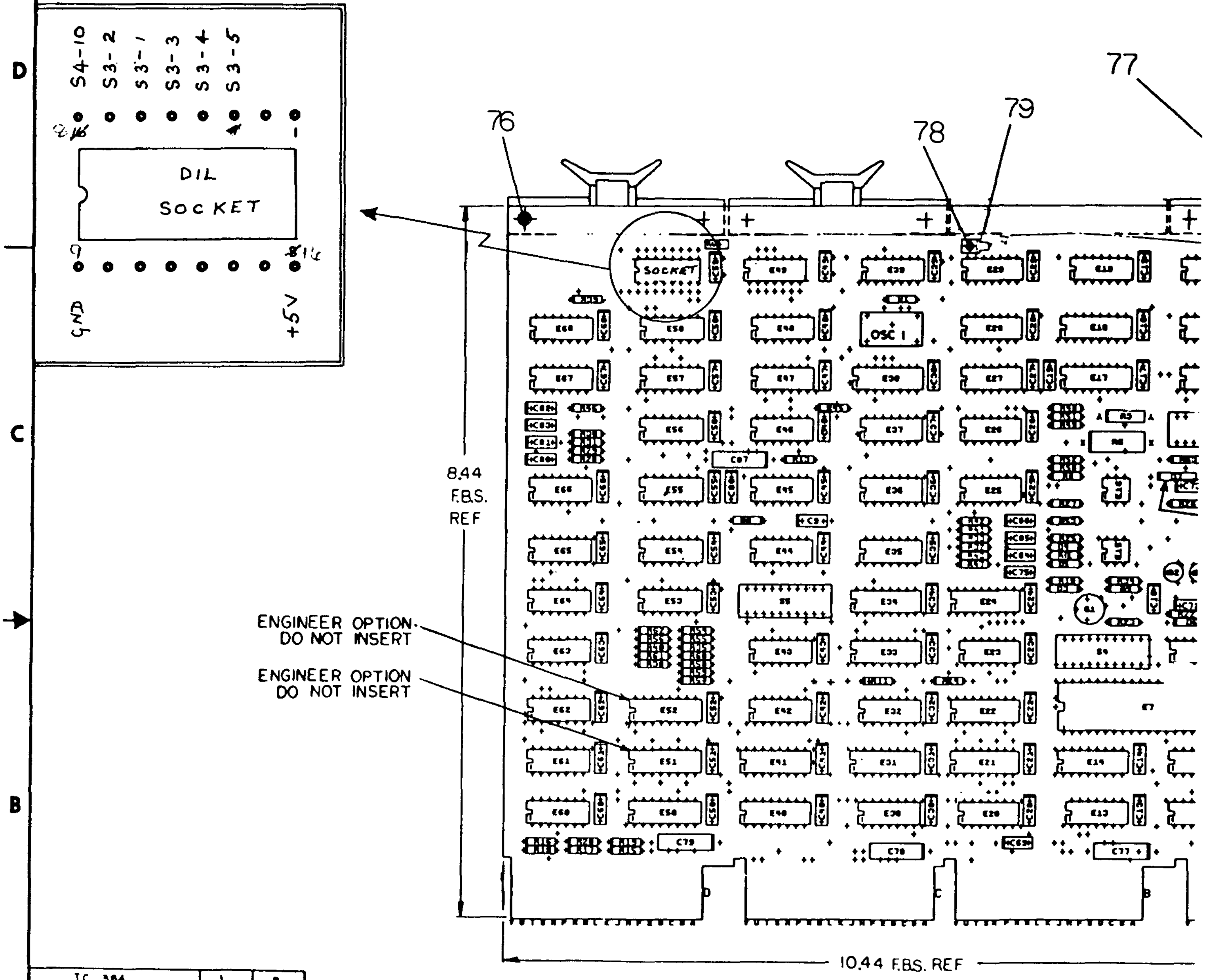
Modification:

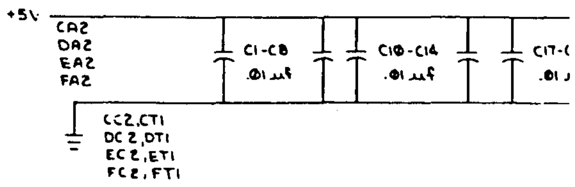
1. Install a low-profile DIL socket above IC E58
2. Wire this using flat multiribbon cabling to the pull-up resistors R48-R53 inclusive and thus parallel the connections on switches S3-1 to S3-5 and S4-10.
3. Add to the socket connections to Ground and the +5 volt rail.
4. Take the connections via a DIL plug and woven ribbon connectors to a diode matrix and an 8 way rotary switch, mounted on the front panel.

Details of these modifications are shown in the accompanying figures.

Geophysics Division.
18th Oct.77

**NOTES:** MODIFICATION FOR EXTERNAL SWITCHING OF BAUD RATE.





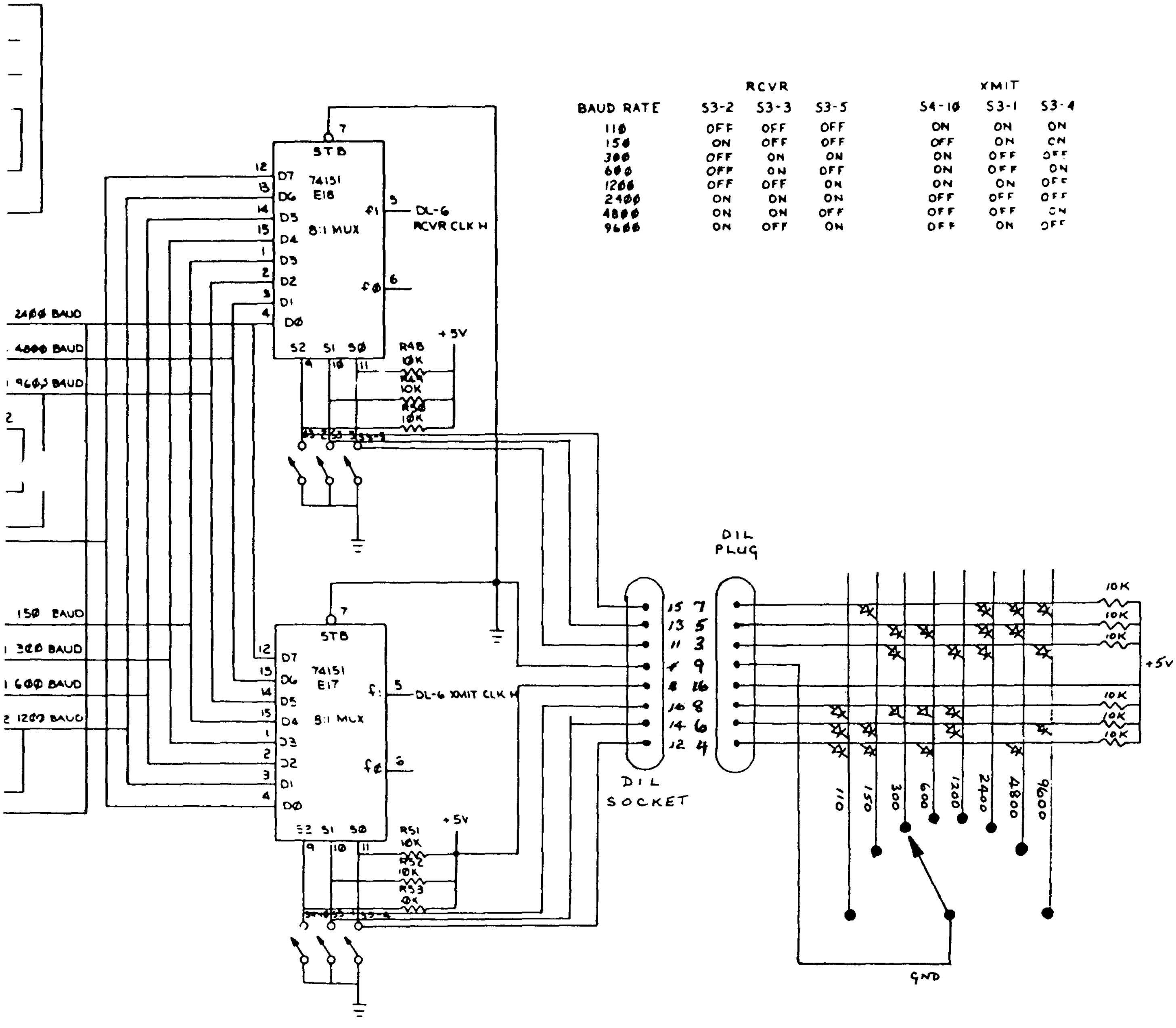
| IC TYPE | GND | +5V |
|---|---|---|
| IC 384 | 1 | 8 |
| IC 8837 | 8 | 16 |
| IC 314A | 1 | 8 |
| IC 74123 | 8 | 16 |
| IC 74157 | 8 | 16 |
| IC 74153 | 8 | 16 |
| IC 7493 | 10 | 5 |
| IC 7492 | 10 | 5 |
| IC 74151 | 8 | 16 |
| IC 74175 | 8 | 16 |
| IC UART | 1 | 3 |
| IC 8097 | 8 | 16 |
| IC 8641 | 8 | 16 |

GND AND 5V ARE USUALLY PIN 7 AND 14 RESPECTIVELY EXCEPTIONS ARE STATED ABOVE

IC PIN LOCATIONS

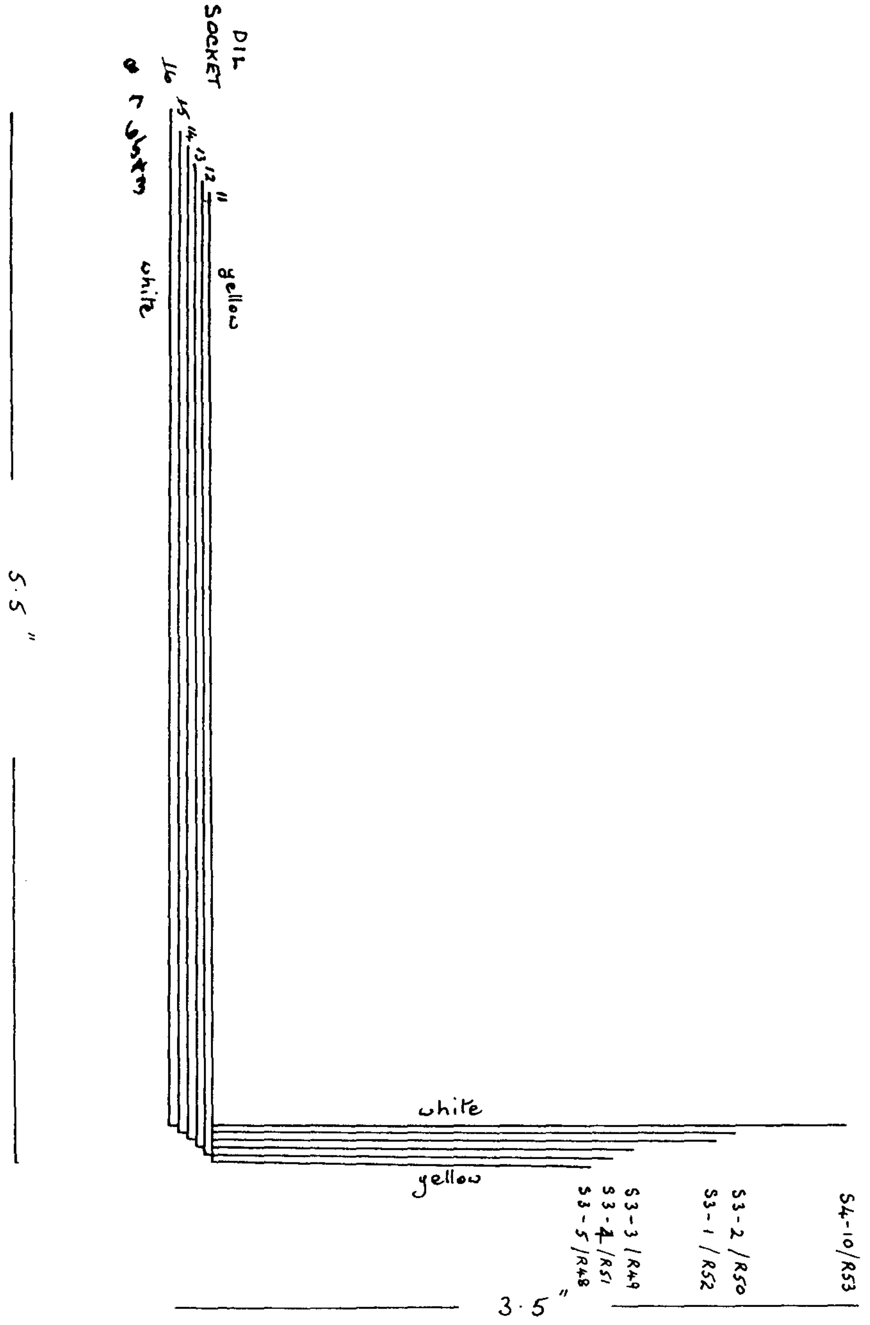
## MODIFICATION TO PERMIT EXTERNAL SWITCHING OF BAUD RATE

| | RCVR | | | XMIT | | |
|---|---|---|---|---|---|---|
| BAUD RATE | S3-2 | S3-3 | S3-5 | S4-1Ø | S3-1 | S3-4 |
| 11Ø | OFF | OFF | OFF | ON | ON | ON |
| 15Ø | ON | OFF | OFF | OFF | ON | ON |
| 3ØØ | OFF | ON | ON | ON | OFF | OFF |
| 6ØØ | OFF | ON | OFF | ON | OFF | ON |
| 12ØØ | OFF | OFF | ON | ON | ON | OFF |
| 24ØØ | ON | ON | ON | OFF | OFF | OFF |
| 48ØØ | ON | ON | OFF | OFF | OFF | ON |
| 96ØØ | ON | OFF | ON | OFF | ON | OFF |



| TITLE | SIZE | CODE | NUMBER | REV. |
|---|---|---|---|---|
| SLU/RTC OPTION (DL-6) | D | CS | M785-Ø-1 | E |
| SCALE | SHEET 8 OF 9 | DIST. | | |

# CABLE DETAILS FOR MODIFICATION TO DL11W



## RT-11 MARKETPLACE

The following is a brief summary of software available from the University of Waterloo, which was compiled by Jerry C. Forshee of Indiana University. Thank you Jerry.

### Waterloo Software

The Computer Systems Group at the University of Waterloo has been developing educational computing systems for the past fifteen years. Many of these systems are available for the PDP-11 family of computers. Brief summaries of some of the available systems are presented below. Readers who are interested should, for more information, contact:

K. Ian Mc Phee  
Editor, WATNEWS  
Computer Systems Group  
University of Waterloo  
Waterloo, Ontario  
Canada  N2L 3G1  
(519) 885-1211, Ext. 3419

The WIDJET System is a job-preparation and debugging stand alone system, specifically designed to help the student or novice programmer prepare and run programs on the computer. It consists of an editor servicing multiple terminals and a scheduler which accepts programs, with a simplified command language, and notifies the terminal user when jobs are completed.

The Monash Educational Computer System, a stand alone system, developed by the Monash University Computing Center for the economical and rapid processing of large numbers of small programs written by students. Students prepare their programs on cards and submit them to the MONECS processor. A few seconds later their results are available. No computer operator is needed; students put their cards into the reader and take their own results from the printer. The system consists of: a batch operating system, a FORTRAN compiler, a COBOL compiler, and a BASIC compiler.

WATBOL is another COBOL compiler. This version has been developed to satisfy an important requirement of education. WATBOL expedites the debugging process and thus significantly reduces the students programming time. This objective is accomplished through fast and efficient compilation and excellent error diagnostics.

WATFOR is a compiler for the FORTRAN IV language. This compiler is intended to expedite the debugging process. WATFOR has two main design objectives which contribute to its efficienty in debugging; speed and ease of compilation and excellent error diagnostics.

Structured WATFOR-11 has encorporated a set of control statements which facilitate Structured Programming. These added control statements are: IF-THEN-ELSE, WHILE-DO, DO-CASE, AT END, EXECUTE-REMOTE BLOCK, WHILE-EXECUTE. The major objective in providing these extensions to FORTRAN was to encourage Structured Programming, modularity and top-down design.

WATPAK/C is a collection of materials used for instructional purposes. These materials include the WATBOL compiler, a textbook, various utility programs, files and an instructor's manual.

WATPCF - Portable Computing Facility, a demonstration of what could be done with self contained portable systems for instructional use featuring a PDP 11/10, RK05 disk, a Documentation OD150 card reader, a Centronics 101A printer, and a Volker Craig terminal. This system is capable of running the MONECS and WATPAK compilers.

Debugging Aids for Assembly Language Programmers. A set of programming tools for debugging and general software development consisting of:

1. ZAP, an interactive program that provides the capability to modify linked executable modules.
2. SUPER-TRACE, an instruction and data trace for executing programs.
3. FLOW-TRACE, a flow of control or execution sequence trace for executing programs.
4. SUPERDUMP, a program monitor with the power to stop, resume, examine and modify executing programs.

St Peter's Lutheran College
HARTS ROAD, INDOOROOPILLY, Q. 4068
P.O. BOX 111, INDOOROOPILLY

TELEPHONE 370 7141

21st December, 1977

Mr John Rasted,
JTR Associates,
58 Rasted Lane,
MERIDEN, CT 06450
UNITED STATES OF AMERICA

Dear Sir,

Enclosed ; a brief description of a SORT program which may be of interest t some of your readers.

The need ıor a more efficient ASCll SORT for mini-systems running RT-11 was seen in the production of school timetables, cummulative records, data base etc. on a PDP 11/10 with 16K words of memory and flexible disk storage.

Details of the program may be obtained from -

Mr D.R. WOODROW,
Senior Master,
St Peter's Lutheran College,
P.O. Box 111,
INDOOROOPILLY,
QUEENSLAND, 4068,
AUSTRALIA,

Yours sincerely,

Woodrow

D.R.WOODROW

# SORTING UTILITY FOR RT-11

SORT is a general purpose MACRO-11 sorting utility which sorts an input file of unordered records, and produces an ordered file of output records, or tags used to access the input file in order. It is designed to operate under RT-11, (and COS-350 on PDP-11's). SORT provides the following advantages over the DIBOL sorting routines:

-EASY TO USE

SORT accepts simple runtime commands from the keyboard, or a command file, so there is no sort generation phase necessary, nor the necessity for a program for every different sorting task to be done. SORT will also handle variable length records without difficulty, and will handle both RT-11 and DIBOL format end-of-file markers.

-EXTREMELY RAPID

SORT is conservatively 10 to 20 times faster than the DIBOL sorts on floppy disk systems, but has been measured at 60-100 times faster where splitting of files was necessary for the DIBOL sorts to operate. The largest file sorted to date is a file of 17,300 27-character records on RK05, sorted by a PDP-11/40 with 28K words in less than 1 1/2 minutes. (That's about the total on-line storage of a dual floppy disk drive).

-EXTREMELY ECONOMICAL

SORT requires only one scratch file the size of the input file for most disk sorts, with no loss in speed. Should work space be critical, SORT can perform record sorts with only space for the input and output files. This requires backup of critical input data, as the input file is destroyed in the process. Tag sorting may also be used to alleviate the problem. SORT requires at most two scratch files.

-CHAINING CAPABILITY

SORT may be chained to from FORTRAN, or MACRO-11, (and perhaps in future DIBOL, depending on requests), and can be made to chain back, transparent to users of the calling program.

-ALGORITHM

SORT uses a REPLACEMENT/SELECTION sort for the initial run generation phase, followed by optimized N-way merging. I/O and computation is overlapped during all phases of the sort taking advantage of the real-time capability of RT-11. SORT dynamically expands memory to achieve efficient use of resources.

## RT-11 SOFTWARE

The following patch and formatter have been submitted by:

Carl D. Lowenstein
Marine Physical Lab.
Bldg. 106, NOSC
San Diego, CA 92152
(714) 452-2308

```
;       PATCH TO RT-11 V02C TO USE KW-11P FOR SYSTEM CLOCK
;       RKMNFB.SYS ONLY!
.R PATCH
*MONITR.SYS/M
*20000;0R
*100/54042      0
*102/341        0
*104/0          54042
*106/0          341
*660/177546     172540
*1566/100       115
*1574/100       0
*1576/177546    172542
*1624/100       104
*0,17332\303    63
*E
.R PIP
*A<MONITR.SYS/U
*SY:/0

;
;       C. D. LOWENSTEIN
;       MARINE PHYSICAL LAB.
;       BLDG. 106, NOSC
;       SAN DIEGO, CA 92152
;       (714) 452-2308
;
```

```
                     .TITLE  RKFMT
                     .IDENT  /V05/
             ;
             ;       FORMATTER-SURFACE VERIFIER
             ;       STOLEN LIBERALLY FROM MAINDEC-11-DZRKI-B
             ;               LINES 1229 - 1483
             ;
                     .MCALL  .REGDEF,.PRINT,.TTYIN,.EXIT
000000               .REGDEF
             ;       DATA BASE
      100000         BIT15=  100000
      040000         BIT14=  40000
      020000         BIT13=  20000
      010000         BIT12=  10000
      004000         BIT11=  4000
      002000         BIT10=  2000
      001000         BIT9=   1000
      000400         BIT8=   400
      000200         BIT7=   200
      000100         BIT6=   100
      000040         BIT5=   40
      000020         BIT4=   20
      000010         BIT3=   10
      000004         BIT2=   4
      000002         BIT1=   2
      000001         BIT0=   1

00000 177400 RKDS:   177400
00002 177402 RKER:   177402
00004 177404 RKCS:   177404
00006 177406 RKWC:   177406
00010 177410 RKBA:   177410
00012 177412 RKDA:   177412

00014 000000 BA:     0
00016 000000 DA:     0
00020 000000 DSKTMP: 0
00022 000000 ERRRF:  0
00024 000000 ERRRFC: 0
00026 000000 ERRWCH: 0
00030 000000 ERRWCS: 0
00032 000000 ERRWF:  0
00034 001664'RBA:    RBUFF
00036 000000 RWC:    0
00040 000000 WC:     0

00042 000403         BR FORMAT        ;RE-ENTRY POINT
00044 052737 START:  BIS #BIT13,@#44  ;MAKE RE-ENTERABLE
      020000
      000044
00052 012777 FORMAT: MOV #1,@RKCS     ;CONTROL RESET
      000001
      177724
00060                .PRINT #1$
00066 000424         BR 2$
                     .NLIST BIN
00070        1$:     .ASCII <15><12>/FORMATTER - SURFACE VERIFIER/
```

```
00126                 .ASCII  /, DRIVE #/<200>
                      .EVEN
                      .LIST BIN
00140          2$:    .TTYIN R1
00146          21$:   .TTYIN
00152 020027          CMP R0,#12      ;FLUSH OUT CRLF
      000012
00156 001373          BNE 21$
00160 162701          SUB #'7,R1      ;CHECK FOR DRIVE #
      000067
00164 100332          BPL FORMAT      ;TOO BIG
00166 062701          ADD #'7-'0,R1
      000007
00172 100727          BMI FORMAT      ;TOO SMALL
00174 001030          BNE 5$          ;JUST RIGHT
00176                 .PRINT #3$
00204 000411          BR 4$   ;DON'T CASUALLY WIPE OUT DISK 0
                      .NLIST BIN
00206          3$:    .ASCII <15><12>/ARE YOU SURE? /<200>
                      .EVEN
                      .LIST BIN
00230          4$:    .TTYIN R1
00236          41$:   .TTYIN
00242 020027          CMP R0,#12
      000012
00246 001373          BNE 41$
00250 162701          SUB #'Y,R1
      000131
00254 001276          BNE FORMAT      ;NOT SURE ABOUT IT!
00256          5$:    .PRINT #6$
00264 000401          BR 7$
00266 000000 6$:      0               ;CRLF
00270 006301 7$:      ASL R1
00272 006301          ASL R1
00274 006301          ASL R1
00276 006301          ASL R1
00300 006301          ASL R1
00302 000301          SWAB R1 ;MAKE UNIT NUMBER FOR RKDA
00304 010167          MOV R1,DSKTMP
      177510
00310 005067          CLR ERRWF       ;CLEAR ALL ERROR FLAGS
      177516
00314 005067          CLR ERRRF
      177502
00320 005067          CLR ERRRFC
      177500
00324 005067          CLR ERRWCH
      177476
00330 005067          CLR ERRWCS
      177474
00334 012767 S12:     MOV #-24.,RWC   ;24 WORDS OF HEADERS
      177750
      177474
00342 012767          MOV #-6144.,WC  ;6K WORDS FOR WRITES
      164000
      177470
00350 012701 COMMON:  MOV #-6,R1      ;SCRAPE SURFACE 6 TIMES
```

```
        177772
96 00354 012777 COM:    MOV #14500,@RKDA        ;SEEK CYL. 202
        014500
        177430
97 00362 056777         BIS DSKTMP,@RKDA        ; ON PROPER UNIT
        177432
        177422
98 00370 105777 1$:     TSTB @RKCS
        177410
99 00374 100375         BPL 1$
100 0376 012777         MOV #11,@RKCS   ;SEEK
        000011
        177400
101 0404 032777         BIT #BIT7,@RKER ;DOES THIS DISK EXIST?
        000200
        177370
102 0412 001415         BEQ 2$
103 0414                .PRINT #11$
104 0422 000167         JMP FORMAT
        177424
105                     .NLIST BIN
106 0426        11$:    .ASCIZ <15><12>/NO SUCH DISK/
107                     .EVEN
108                     .LIST BIN
109 0446 032777 2$:     BIT #BIT6,@RKDS ;WAIT FOR SEEK DONE
        000100
        177324
110 0454 001774         BEQ 2$
111 0456 105777 3$:     TSTB @RKCS
        177322
112 0462 100375         BPL 3$
113 0464 012777         MOV #15,@RKCS   ;DRIVE RESET
        000015
        177312
114 0472 032777 4$:     BIT #BIT6,@RKDS
        000100
        177300
115 0500 001774         BEQ 4$          ;WAIT FOR DRIVE
116 0502 005201         INC R1
117 0504 001323         BNE COM         ;SEEK 6 TIMES
118 0506 005067         CLR DA  ;BEGIN AT THE BEGINNING
        177304
119 0512 012777 NEXT:   MOV #-1,@BA             ;WRITE ALL 1'S
        177777
        177274
120 0520 004167         JSR R1,IO
        000102
121 0524 005077         CLR @BA         ;WRITE ALL 0'S
        177264
122 0530 004167         JSR R1,IO
        000072
123 0534 012777         MOV #125252,@BA ;WRITE PATTERN
        125252
        177252
124 0542 004167         JSR R1,IO
        000060
125 0546 005177         COM @BA ;WRITE COMPLEMENT PATTERN
```

```
         177242
126 0552 004167         JSR R1,IO
         000050
127 0556 062767         ADD #40,DA      ;NEXT CYLINDER
         000040
         177232
128 0564 026727         CMP DA,#14540
         177226
         014540
129 0572 001347         BNE NEXT
130 0574        GOOD:   .PRINT #1$
131 0602 000407         BR 2$
132                     .NLIST BIN
133 0604        1$:     .ASCIZ <15><12>/PACK GOOD./
134                     .EVEN
135                     .LIST BIN
136 0622 000167 2$:     JMP FORMAT
         177224
137
138 0626 016777 IO:     MOV WC,@RKWC    ;SET UP FOR WRITE FORMAT
         177206
         177152
139 0634 016777         MOV DA,@RKDA
         177156
         177150
140 0642 056777         BIS DSKTMP,@RKDA        ;SET UNIT NUMBER
         177152
         177142
141 0650 016777         MOV BA,@RKBA
         177140
         177132
142 0656 005077         CLR @RKCS
         177122
143 0662 052777         BIS #BIT10+BIT11,@RKCS
         006000
         177114
144                                     ;FORMAT,INHIBIT BA INCR
145 0670 052777         BIS #BIT1,@RKCS ;WRITE
         000002
         177106
146 0676 005277         INC @RKCS               ;GO
         177102
147 0702 105777 1$:     TSTB @RKCS
         177076
148 0706 100375         BPL 1$
149 0710 005777         TST @RKCS               ;ERROR?
         177070
150 0714 100520         BMI WFERR               ;GO HANDLE ERROR
151 0716 005067         CLR ERRWF
         177110
152
153 0722 016777         MOV RWC.@RKWC   ;SET UP FOR READ FORMAT
         177110
         177056
154 0730 016777         MOV DA,@RKDA
         177062
         177054
```

```
0736 056777         BIS DSKTMP,@RKDA
     177056
     177046
0744 016777         MOV RBA,@RKBA
     177064
     177036
0752 005077         CLR @RKCS
     177026
0756 052777         BIS #BIT10,@RKCS         ;FORMAT
     002000
     177020
0764 052777         BIS #BIT2,@RKCS          ;READ
     000004
     177012
0772 005277         INC @RKCS                ;GO
     177006
0776 105777 2$:     TSTB @RKCS
     177002
1002 100375         BPL 2$
1004 032777         BIT #BIT14,@RKCS         ;HARD ERROR?
     040000
     176772
1012 001115         BNE RFERR
1014 005067         CLR ERRRF                ;OK
     177002

1020 016777         MOV WC,@RKWC     ;SET UP WRITE CHECK
     177014
     176760
1026 016777         MOV DA,@RKDA
     176764
     176756
1034 056777         BIS DSKTMP,@RKDA
     176760
     176750
1042 016777         MOV BA,@RKBA
     176746
     176740
1050 005077         CLR @RKCS
     176730
1054 052777         BIS #BIT11+BIT8,@RKCS
     004400
     176722
                    ;INHIBIT BA INCR, STOP ON SOFT ERR.
1062 052777         BIS #BIT1+BIT2,@RKCS     ;WRITE CHECK
     000006
     176714
1070 005277         INC @RKCS                ;GO
     176710

1074 016703         MOV RWC,R3       ;MEANWHILE, CHECK HEADER
     176736
1100 005403         NEG R3
1102 066703         ADD RBA,R3
     176726
1106 016702         MOV RBA,R2
     176722
```

```
181 1112 026722 MORE:   CMP DA,(R2)+
         176700
182 1116 001062         BNE RFCERR      ;HEADER WAS WRONG
183 1120 020302         CMP R3,R2
184 1122 001373         BNE MORE
185 1124 005067         CLR ERRRFC              ;OK
         176674
186
187 1130 105777 1$:     TSTB @RKCS      ;WAIT FOR WRITE CHECK
         176650
188 1134 100375         BPL 1$
189 1136 005777         TST @RKCS               ;ERROR?
         176642
190 1142 100462         BMI WCERR
191 1144 005067         CLR ERRWCH
         176656
192 1150 005067         CLR ERRWCS
         176654
193 1154 000201         RTS R1          ;RETURN TO MAIN LINE
194
195 1156 005267 WFERR:  INC ERRWF       ;WRITE FORMAT ERRORS
         176650
196 1162 026727         CMP ERRWF,#4
         176644
         000004
197 1170 001013         BNE RETRY
198 1172        SYSER:  .PRINT #1$
199 1200                .EXIT
200                     .NLIST BIN
201 1202        1$:     .ASCIZ /SYSTEM ERROR/
202                     .EVEN
203                     .LIST BIN
204
205 1220 005077 RETRY:  CLR @RKCS
         176560
206 1224 012777         MOV #15,@RKCS   ;DRIVE RESET
         000015
         176552
207 1232 032777 1$:     BIT #BIT6,@RKDS
         000100
         176540
208 1240 001774         BEQ 1$
209 1242 000167         JMP IO          ;TRY AGAIN
         177360
210
211 1246 005267 RFERR:  INC ERRRF       ;READ FORMAT ERRORS
         176550
212 1252 026727         CMP ERRRF,#4
         176544
         000004
213 1260 001357         BNE RETRY
214 1262 000743         BR SYSER
215
216                     ;READ FORMAT ERROR FOUND BY SOFTWARE
217 1264 005267 RFCERR: INC ERRRFC
         176534
218 1270 105777 1$:     TSTB @RKCS
```

```
     176510
1274 100375         BPL 1$  ;WAIT FOR WRITE CHECK TO FINISH
1276 026727         CMP ERRRFC,#4
     176522
     000004
1304 001425         BEQ FAILED
1306 000744         BR RETRY

1310 032777 WCERR:  BIT #BIT14,@RKCS ;WRITE CHECK ERRORS
     040000
     176466
1316 001010         BNE WCHERR       IT WAS A HARD ERROR
1320 005267         INC ERRWCS
     176504
1324 026727         CMP ERRWCS,#4
     176500
     000004
1332 001421         BEQ FAIL
1334 000167         JMP IO          ;RETRY
     177266
1340 005267 WCHERR: INC ERRWCH
     176462
1344 026727         CMP ERRWCH,#4
     176456
     000004
1352 001707         BEQ SYSER
1354 000167         JMP RETRY
     177640

1360 042777 FAILED: BIC #37,@RKDA
     000037
     176424
1366 042702         BIC #^C37,R2
     177740
1372 060277         ADD R2,@RKDA
     176414
1376        FAIL:   .PRINT #1$
1404 000415         BR 2$
                    .NLIST BIN
1406        1$:     .ASCIZ <15><12>/PACK FAILED AT (OCTAL)/
                    .EVEN
                    .LIST BIN
1440 017701 2$:     MOV @RKDA,R1
     176346
1444 010102         MOV R1,R2
1446 042702         BIC #^C7,R2
     177770
1452 062702         ADD #'0,R2
     000060
1456 110267         MOVB R2,SEC+1
     000151
1462 004367         JSR R3;SHF3
     000164
1466 042702         BIC #^C1,R2
     177776
1472 062702         ADD #'0,R2
     000060
```

```
252 1476 110267         MOVB R2,SEC     ;CONVERT SECTOR NUMBER
         000130
253 1502 004367         JSR R3,SHF1
         000150
254 1506 042702         BIC #^C1,R2
         177776
255 1512 062702         ADD #'0,R2
         000060
256 1516 110267         MOVB R2,SUR             ;SURFACE NUMBER
         000122
257 1522 004367         JSR R3,SHF1
         000130
258 1526 042702         BIC #^C7,R2
         177770
259 1532 062702         ADD #'0,R2
         000060
260 1536 110267         MOVB R2,CYL+2
         000060
261 1542 004367         JSR R3,SHF3
         000104
262 1546 042702         BIC #^C7,R2
         177770
263 1552 062702         ADD #'0,R2
         000060
264 1556 110267         MOVB R2,CYL+1
         000037
265 1562 004367         JSR R3,SHF3
         000064
266 1566 042702         BIC #^C3,R2
         177774
267 1572 062702         ADD #'0,R2
         000060
268 1576 110267         MOVB R2,CYL             ;CYLINDER NUMBER
         000016
269 1602                .PRINT #3$
270 1610 000417         BR STOP
271                     .NLIST BIN
272 1612        3$:     .ASCII /CYL.  /
273 1620        CYL:    .ASCII /000 /
274 1624                .ASCII /SEC.  /
275 1632        SEC:    .ASCII /00 /
276 1635                .ASCII /SURF.  /
277 1644        SUR:    .ASCII /0 /
278 1646                .BYTE 15,12
279                     .EVEN
280                     .LIST BIN
281 1650        STOP:   .EXIT
282
283 1652 006201 SHF3:   ASR R1          ;SHIFT SUBROUTINES
284 1654 006201 SHF2:   ASR R1
285 1656 006201 SHF1:   ASR R1
286 1660 010102         MOV R1,R2
287 1662 000203         RTS R3
288
289 1664        RBUFF:  .BLKW 24.
290
291      000044'        .END START
```

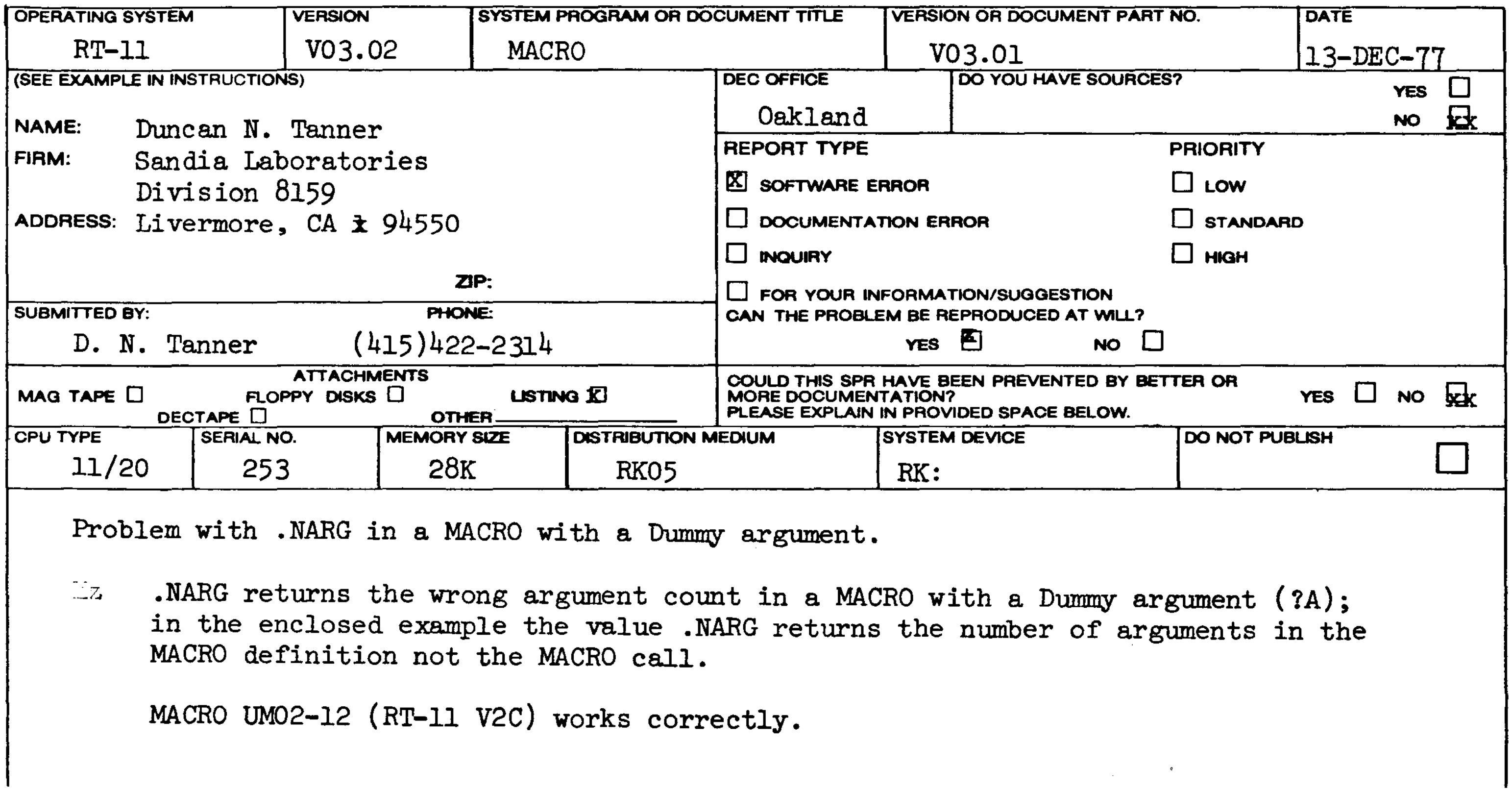
```
SYMBOL TABLE

BA        000014R            BIT0    = 000001            BIT1    = 000002
BIT10   = 002000             BIT11   = 004000            BIT12   = 010000
BIT13   = 020000             BIT14   = 040000            BIT15   = 100000
BIT2    = 000004             BIT3    = 000010            BIT4    = 000020
BIT5    = 000040             BIT6    = 000100            BIT7    = 000200
BIT8    = 000400             BIT9    = 001000            COM       000354R
COMMON    000350R            CYL       001620R           DA        000016R
DSKTMP    000020R            ERRRF     000022R           ERRRFC    000024R
ERRWCH    000026R            ERRWCS    000030R           ERRWF     000032R
FAIL      001376R            FAILED    001360R           FORMAT    000052R
GOOD      000574R            IO        000626R           MORE      001112R
NEXT      000512R            PC      =%000007            RBA       000034R
RBUFF     001664R            RETRY     001220R           RFCERR    001264R
RFERR     001246R            RKBA      000010R           RKCS      000004R
RKDA      000012R            RKDS      000000R           RKER      000002R
RKWC      000006R            RWC       000036R           R0      =%000000
R1      =%000001             R2      =%000002            R3      =%000003
R4      =%000004             R5      =%000005            SEC       001632R
SHF1      001656R            SHF2      001654R           SHF3      001652R
SP      =%000006             START     000044R           STOP      001650R
SUR       001644R            SYSER     001172R           S12       000334R
WC        000040R            WCERR     001310R           WCHERR    001340R
WFERR     001156R
. ABS.    000000      000
          001744      001
ERRORS DETECTED: 0
FREE CORE: 15971. WORDS
```

----------------------------------------------------------------

SPRS

----------------------------------------------------------------



| | | | | |
|---|---|---|---|---|
| OPERATING SYSTEM<br>RT-11 | VERSION<br>V03.02 | SYSTEM PROGRAM OR DOCUMENT TITLE<br>MACRO | VERSION OR DOCUMENT PART NO.<br>V03.01 | DATE<br>13-DEC-77 |

(SEE EXAMPLE IN INSTRUCTIONS)

NAME: Duncan N. Tanner
FIRM: Sandia Laboratories
Division 8159
ADDRESS: Livermore, CA x 94550
ZIP:

DEC OFFICE: Oakland

DO YOU HAVE SOURCES? YES □ NO ☒

REPORT TYPE
☒ SOFTWARE ERROR
□ DOCUMENTATION ERROR
□ INQUIRY
□ FOR YOUR INFORMATION/SUGGESTION

PRIORITY
□ LOW
□ STANDARD
□ HIGH

SUBMITTED BY: D. N. Tanner
PHONE: (415)422-2314

CAN THE PROBLEM BE REPRODUCED AT WILL? YES ☒ NO □

ATTACHMENTS
MAG TAPE □ FLOPPY DISKS □ LISTING ☒
DECTAPE □ OTHER ______

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES □ NO ☒

| | | | | | |
|---|---|---|---|---|---|
| CPU TYPE<br>11/20 | SERIAL NO.<br>253 | MEMORY SIZE<br>28K | DISTRIBUTION MEDIUM<br>RK05 | SYSTEM DEVICE<br>RK: | DO NOT PUBLISH<br>□ |

Problem with .NARG in a MACRO with a Dummy argument.

.NARG returns the wrong argument count in a MACRO with a Dummy argument (?A); in the enclosed example the value .NARG returns the number of arguments in the MACRO definition not the MACRO call.

MACRO UM02-12 (RT-11 V2C) works correctly.

120494

```
SPR MACRO       RT-11 MACRO VM02-12    13-DEC-77 00:13:39 PAGE 1

     1                                  .NLIST TTM
     2                                  .TITLE SPR MACRO
     3
     4                                  ;****************************************
     5
     6                                  ;       RUN ON MACRO VM02-12 (RT-11 V02C)
     7
     8                                  ;****************************************
     9
    10
    11                                  ;       PROBLEM WITH .NARG IN A MACRO WITH DUMMY ARGUMENT
    12                                  ;       .NARG DOES NOT RETURN THE CORRECT NUMBER OF ARGUMENTS.
    13                                  ;       IT RETURNS THE NUMBER OF ARGUMENTS IN THE MACRO DEFINITION.
    14
    15
    16                                  .MACRO DUM A1,A2,A3,A4,?A
    17                                  .NARG NUM
    18                                  A:      .WORD NUM
    19                                  .ENDM
    20
    21                                  .LIST MEB
    22
    23 000000                           START:  DUM A,B,C,D     ;NUM SHOULD BE 4
       000000  000004                   64$:    .WORD NUM
    24 000002                                   DUM A,B,C       ;NUM SHOULD BE 3
       000002  000003                   65$:    .WORD NUM
    25 000004                                   DUM A,B         ;NUM SHOULD BE 2
       000004  000002                   66$:    .WORD NUM
    26 000006                                   DUM A           ;NUM SHOULD BE 1
       000006  000001                   67$:    .WORD NUM
    27 000010                                   DUM             ;NUM SHOULD BE 0
       000010  000000                   68$:    .WORD NUM
    28 000012                                   DUM A,B,C,D,E   ;NUM SHOULD BE 5
       000012  000005                   E:      .WORD NUM
    29
    30
    31 000014  000000                   A:      .WORD 0
    32 000016  000000                   B:      .WORD 0
    33 000020  000000                   C:      .WORD 0
    34 000022  000000                   D:      .WORD 0
    35         000001'                          .END
SPR MACRO       RT-11 MACRO VM02-12    13-DEC-77 00:13:39 PAGE 1+
SYMBOL TABLE

A       000014R         B       000016R         C       000020R         D       000022R         E       000012R
NUM   = 000005          START   000000R
. ABS.  000000     000
        000024     001
ERRORS DETECTED: 0
FREE CORE: 18210. WORDS

.LP:=SPR1
```

CORRECT!

120494

```
SPR MACRO V03.01, RT11 V3        MACRO V03.01 13-DEC-77 00:06:46 PAGE 1

      1                                  .TITLE SPR MACRO V03.01, RT11 V3
      2
      3
      4                                  ;       PROBLEM WITH .NARG IN A MACRO WITH DUMMY ARGUMENT
      5                                  ;       .NARG DOES NOT RETURN THE CORRECT NUMBER OF ARGUMENTS.
      6                                  ;       IT RETURNS THE NUMBER OF ARGUMENTS IN THE MACRO DEFINITION.
      7
      8
      9                                  .MACRO DUM A1,A2,A3,A4,?A   <--  5 ARGUMENTS
     10                                  .NARG NUM
     11                                  A:      .WORD NUM
     12                                  .ENDM
     13
     14                                  .NLIST TTM
     15                                  .LIST MEB
     16
     17 000000                           START:  DUM A,B,C,D     ;NUM SHOULD BE 4
        000000  000005                   64$:    .WORD NUM
     18 000002                                   DUM A,B,C       ;NUM SHOULD BE 3
        000002  000005                   65$:    .WORD NUM
     19 000004                                   DUM A,B         ;NUM SHOULD BE 2        NUM  IS  ALWAYS  5
        000004  000005                   66$:    .WORD NUM
     20 000006                                   DUM A           ;NUM SHOULD BE 1
        000006  000005                   67$:    .WORD NUM
     21 000010                                   DUM             ;NUM SHOULD BE 0
        000010  000005                   68$:    .WORD NUM
     22 000012                                   DUM A,B,C,D,E   ;NUM SHOULD BE 5
        000012  000005                   E:      .WORD NUM
     23
     24
     25 000014  000000                   A:      .WORD 0
     26 000016  000000                   B:      .WORD 0
     27 000020  000000                   C:      .WORD 0
     28 000022  000000                   D:      .WORD 0
     29         000001                           .END

SPR MACRO V03.01, RT11 V3        MACRO V03.01 13-DEC-77 00:06:46 PAGE 1-1
SYMBOL TABLE

A       000014R         C       000020R         E       000012R         NUM   = 000005         START   000000R
B       000016R         D       000022R

. ABS.  000000     000
        000024     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  363 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  67 PAGES
.LP:=SPR
```