

368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410

.REM -

IDENTIFICATION  
-----

PRODUCT CODE: AC-S210A-MC  
PRODUCT NAME: CXTMDA0 DEC/X11 TM78 MAGTAPE EXERCISER MODULE  
DATE: JUNE 1,1980  
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1980  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MA.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A  
SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION  
OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER  
COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE  
TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO  
AGREES TO ALL LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE  
SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT  
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
EQUIPMENT CORPORATION.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE  
USE OR RELIABILITY OF ITS SOFTWARE IN EQUIPMENT WHICH IS NOT  
SUPPLIED BY DEC.



.REM -

TABLE OF CONTENTS

-----

|      |                                            |
|------|--------------------------------------------|
| 1.0  | ABSTRACT                                   |
| 2.0  | REQUIREMENTS                               |
| 3.0  | PASS DEFINITION                            |
| 4.0  | EXECUTION TIME                             |
| 5.0  | CONFIGURATION REQUIREMENTS                 |
| 6.0  | DEVICE/OPTION SETUP                        |
| 7.0  | MODULE OPERATION TEST SEQUENCE             |
| 8.0  | OPERATION OPTIONS                          |
| 9.0  | SELECTING SLAVES                           |
| 10.0 | DUAL PORT OPTION                           |
| 11.0 | HIERARCHY OF THE TM78 DEC/X11 DRIVER       |
| 12.0 | DATA STRUCTURES OF THE TM78 DEC/X11 DRIVER |
| 13.0 | PDL FOR THE TM78 DEC/X11 DRIVER            |
| 14.0 | ERROR DEFINITIONS FOR TM78 DEC/X11 DRIVER  |

1.0 ABSTRACT

-----

THE TMD IS AN IOMODX MODULE THAT CAN EXERCISE UP TO 8 TM78 CONTROLLERS (DRIVES) WITH UP TO 4 TU78 TAPE TRANSPORTS (SLAVES) ON A SINGLE TM78. EACH TU78 WILL BE EXERCISED BY DOING A WRITE, READ REVERSE, READ FORWARD AND AN IN-CORE COMPARE. THIS SEQUENCE OF FUNCTIONS WILL BE DEFINED AS A CYCLE. AN "END OF PASS" WILL BE REACHED AFTER 500 CYCLES, OR WHEN END OF TAPE (EOT) IS DETECTED. WHEN "END OF PASS" IS REACHED, A "TAPE MARK" FUNCTION IS EXECUTED BEFORE CONTINUING.

WHEN EOT IS DETECTED AND AFTER THE TAPEMARK IS WRITTEN, THE ERROR SUMMARY IS PRINTED, ALL TU78(S) ARE REWOUND, AND THE DENSITY CHANGED.

THIS MODULE UTILIZES THE AUTO REPOSITIONING FEATURES OF THE TM78 TO ATTEMPT RECOVERY FROM WRITE AND READ STATUS ERRORS. THEREFORE, RECOVERY SEQUENCES ARE AS DEFINED FOR THE TM78 OPERATIONAL MICRO CODE, AND RECOVERABLE/NON-RECOVERABLE ERROR REPORTS AND STATISTIC UPDATE ARE BASED SOLEY ON THE OPERATIONAL MICROCODE RETRY ALGORITHM RESULTS.

STATISTICS MAINTAINED FOR EACH TU78 ARE:

468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523

RECOVERABLE READ ERRORS  
UNRECOVERABLE READ ERRORS  
RECOVERABLE WRITE ERRORS

READ STATUS ERRORS ARE NOT PRINTED ON A TRY BY TRY BASIS BUT  
RATHER THE FINAL OUTCOME OF EACH RETRY (RECOVERABLE OR NON-  
RECOVERABLE) IS PRINTED ON A RECORD BY RECORD BASIS.

WRITE STATUS ERRORS ARE NOT PRINTED ON A TRY BY TRY BASIS BUT  
ARE ALSO CLASSIFIED AS RECOVERABLE OR NON-RECOVERABLE. HOWEVER,  
A UNRECOVERABLE WRITE ERROR WILL CAUSE THE UNIT TO BE DROPPED  
AS THE OPERATIONAL MICRO CODE CANNOT CONTINUE, AND  
MAINTAIN A READABLE (ANSI STANDARD) TAPE.

## 2.0 REQUIREMENTS

HARDWARE: AT LEAST 1 TU78 SLAVE ON A TM78 CONTROLLER AND AN  
RH11/RH70.

SOFTWARE: TMB REQUIRES 6K BYTES OF STORAGE - (3K WORDS)

## 3.0 PASS DEFINITION

ONE PASS CONSISTS OF 500 CYCLES OF A WRITE, READ REVERSE,  
READ FORWARD, AND DATA CHECK OPERATIONS.

## 4.0 EXECUTION TIME

ONE PASS OF TMD TAKES APPROXIMATELY MINUTE.

## 5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS: DEVADR: 172400 VECTOR: 224 BR1: 5

REQUIRED PARAMETERS: NONE

## 6.0 DEVICE/OPTION SETUP

POWER UP ALL TU78'S AND MAKE READY AT LOAD POINT AND WRITE ENABLED.

## 7.0 MODULE OPERATION TEST SEQUENCE

REFER TO SECTION 13.0.

524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579

8.0 DRIVER OPERATION OPTIONS

-----

SR1 - THE DEFAULT FOR ALL OPTIONS IS ZERO (0).

---

BIT0 = 0: ALLOW DATA COMPARE

BIT0 = 1: INHIBIT DATA COMPARE

BIT1 = 0: MODULE WILL EXERCISE ALL TU78'S FOUND ON THE USER  
SPECIFIED MASSBUS CONTROLLER.

BIT1 = 1: OPERATOR MUST SELECT THE TU78(S) IN ADDITION TO THE  
TM78'S (DVID1) TO BE EXERCISED. SEE SECTION 10.0 FOR  
DETAILS.

BIT2 = 0: REPORT RECOVERABLE/NON-RECOVERABLE ERRORS AS THEY OCCUR.

BIT2 = 1: DO NOT REPORT RECOVERABLE/NON-RECOVERABLE ERRORS AS  
THEY OCCUR.

BIT3 = 0: PRINT ERROR SUMMARY AT EOT.

BIT3 = 1: DO NOT PRINT ERROR SUMMARY AT EOT.

BIT4 = 0: TEST GCR (6250 BPI)

BIT4 = 1: DO NOT TEST GCR (6250 BPI)

BIT5 = 0: TEST PE (1600 BPI)

BIT5 = 1: DO NOT TEST PE (1600 BPI)

9.0 SELECTING SLAVES

-----

BIT 1 OF SR1 = 1

WHEN BIT 1 IS SET IN SR1, THE OPERATOR MUST SELECT THE TU78(S)  
TO BE TESTED.

TO SELECT THE TU78(S) FOR A PARTICULAR TM78, A "1" MUST BE SET  
IN THE APPROPRIATE BIT POSITION FOR THE DESIRED TU78. REFER TO  
THE FOLLOWING TABLE.

UNIT TABLE FORMAT

-----

T U 7 8 #

<-----

|     | 15                                    | 4                                     | 3 | 2   | 1 | 0   |   |     |   |     |   |   |
|-----|---------------------------------------|---------------------------------------|---|-----|---|-----|---|-----|---|-----|---|---|
| 580 | +-----+-----+-----+-----+-----+-----+ |                                       |   |     |   |     |   |     |   |     |   |   |
| 581 | !                                     | NOT USED                              | ! | TM0 | ! | TM0 | ! | TM0 | ! | TM0 | ! |   |
| 582 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! | ! |
| 583 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 584 | !                                     | NOT USED                              | ! | TM1 | ! | TM1 | ! | TM1 | ! | TM1 | ! | ! |
| 585 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! | ! |
| 586 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 587 | !                                     | NOT USED                              | ! | TM2 | ! | TM2 | ! | TM2 | ! | TM2 | ! | ! |
| 588 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! | T |
| 589 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 590 | !                                     | NOT USED                              | ! | TM3 | ! | TM3 | ! | TM3 | ! | TM3 | ! | M |
| 591 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! |   |
| 592 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 593 | !                                     | NOT USED                              | ! | TM4 | ! | TM4 | ! | TM4 | ! | TM4 | ! | 7 |
| 594 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! |   |
| 595 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 596 | !                                     | NOT USED                              | ! | TM5 | ! | TM5 | ! | TM5 | ! | TM5 | ! | 8 |
| 597 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! |   |
| 598 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 599 | !                                     | NOT USED                              | ! | TM6 | ! | TM6 | ! | TM6 | ! | TM6 | ! |   |
| 600 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! |   |
| 601 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 602 | !                                     | NOT USED                              | ! | TM7 | ! | TM7 | ! | TM7 | ! | TM7 | ! |   |
| 603 | !                                     |                                       | ! | TU3 | ! | TU2 | ! | TU1 | ! | TU0 | ! |   |
| 604 | !                                     | +-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |   |
| 605 | !                                     |                                       |   |     |   |     |   |     |   |     |   | V |

REMEMBER THAT DVID1 MUST BE SET UP FOR THE PROPER TM78'S TO BE TESTED. IF THE TM78 IN NOT SELECTED IN DVID1, THE INFORMATION FOR THAT TM78 IN THE UNIT TABLE IS IGNORED.

BIT 1 OF SR1 = 0

THE DEC/X11 DRIVER WILL SEARCH FOR AND TEST ALL TU78 SLAVES THAT RESPOND READY, ON-LINE AND WRITE ENABLED. IN THIS MODE THE CONTENTS OF DVID1 IS NOT USED.

#### 10.0 DUAL PORT OPTION

THE TM78 PRODUCT WILL ALLOW TWO RH11/RH70 CONTROLLERS (PORTS A AND B) TO BE CONNECTED, AND EACH TU78 HAS A FRONT PANEL SWITCH THAT SPECIFIES THE PORTS THAT HAVE ACCESS TO THAT TU. THE CHOICES ARE A, B, A AND B, OR NONE.

THE A ONLY, B ONLY, OR NONE CONFIGURATIONS POSE NO PROBLEM TO THIS DRIVER, BUT THE A AND B CONFIGURATION IS A PROBLEM. THE PROBLEM EXISTS BECAUSE THE TM78 DOES NOT ARBITRATE PORT ACCESS, AND PROVIDES NO PATH TO ALLOW SOFTWARE TO DO IT.

THEREFORE, THE BOTH MODE OF A TU78 IS ONLY USEABLE FOR WRITE ONLY UTILIZATION, OR FOR INTERLEAVED OPERATIONS SYNCHRONIZED THROUGH SOME EXTERNAL MECHANISM.

AS A RESULT DUAL MB ACCESS TO A SINGLE TU78 IS NOT SUPPORTED

636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691

BY THIS DRIVER, AND CONFIGURATION IN THAT MANNER WILL RESULT IN  
UNPREDICTABLE DRIVER OPERATION.

TO SUPPORT DUAL PORT ACCESS TO THE A ONLY, B ONLY TU78'S REQUIRES  
TWO COPIES OF THIS DRIVER TO BE ACTIVE ONE FOR EACH RH11/70  
CONNECTED TO THE TM78.

11.0 HIERARCHY OF THE TM78 DEC/X11 DRIVER

```
-----  
1.1  
*****  
* BUILD RH11/70 *  
---* I/O *  
I * REGISTER MAP *  
I *****  
I  
I 1.2  
I *****  
I * *  
I--* RH11/70 CLEAR *  
I * *  
I *****  
I  
I 1.3  
I *****  
* * I * CHECK FOR *  
* INITIALIZE *--I--* TM78 *  
* * I * DRIVE TYPE *  
***** I *****  
I  
I 1.4  
I *****  
I * BUILD TU78 *  
I--* SLAVE *  
I * TABLE *  
I *****  
I  
I 1.5  
I *****  
I * INITIALIZE *  
I--* PROGRAM *  
I * VARIABLES *  
I *****  
I  
I 1.6  
I *****  
I * *
```

692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747

```

---*      ERRORS      *
 *      *      *
*****

```

```

2.1
*****
*
---*DENSITY CONTROL*---
I * * I *****
I ***** I * *
I 2.2 I---* ISSUE WRITE *---
I ***** I * I
I * * I ***** I
I---* BOT CONTROL *---I ***** 3.2 I
I * * I ***** I
I ***** I * ISSUE * I
I 2.3 I---* READ *---I
I ***** I * FORWARD * I
I * * I ***** I
I---* WRITE CONTROL *---I 3.3 I
I * * I ***** I
I ***** I * ISSUE * I
I 2.4 I---* READ *---I
I ***** I * REVERSE * I *****
* SEQUENCE * I * READ * I ***** I * INT *
* CONTROL *---I---* FORWARD *---I ***** 3.4 I---* CTL *
* * I * CONTROL * I ***** I * *
***** I ***** I * ISSUE * I *****
I 2.5 I---* TAPE *---I
I ***** I * MARK * I
I * READ * I ***** I
I---* REVERSE *---I 3.5 I
I * CONTROL * I ***** I
I ***** I * * I
I 2.6 I---* ISSUE REWIND *---I
I ***** I * * I
I * TAPE * I ***** I
I---* MARK *---I 3.6 I
I * CONTROL * I ***** I
I ***** I * * I
I 2.7 I---* ISSUE SENSE *---
I ***** I * *
I * * I *****
I---*REWIND CONTROL *---
I * *
I *****
I
I
I
I

```

12.0 DATA STRUCTURES OF THE TM78 DEC/X11 DRIVER

|        | 15 | 0 |                                    |
|--------|----|---|------------------------------------|
| TMBIT  | +  | + | TM78 # CONVERTED TO A BIT POSITION |
| TMBIT2 | +  | + | TM78 # *2                          |
| TUBIT  | +  | + | TU78 # CONVERTED TO A BIT POSITION |
| TUBIT2 | +  | + | TU78 # *2                          |
| UCNT   | +  | + | UNIT COUNT                         |
| TM78   | +  | + | TM78 #                             |
| TU78   | +  | + | TU78 #                             |
| EOTCNT | +  | + | EOT COUNT                          |
| TINTCD | +  | + | TERMINATION INTERRUPT CODE         |
| ITM78  | +  | + | INTERRUPTING TM78                  |
| ITU78  | +  | + | INTERRUPTING TU78                  |
| BOTCNT | +  | + | # OF TU78'S AT BOT                 |
| SENSE  | +  | + | SENSE COMMAND INFORMATION          |
| CNDTI  | +  | + | CLEAR ATTENTION INTERRUPT          |
| ERRIDX | +  | + | MODULE ERROR INDEX                 |
| A16AD  | +  | + | EXTENDED ADDRESS BITS 16 AND 17    |
| PA18   | +  | + | LOWER 16 OF BUFFER ADDRESS         |
| XMEM   | +  | + | EA BITS OF 18 BIT ADDRESS          |
| PA22   | +  | + | LOWER 16 OF 22 BIT ADDRESS         |
| EA22   | +  | + | EA BITS OF 22 BIT ADDRESS          |
| FILE   | +  | + | FILE NUMBER                        |
| RECORD | +  | + | RECORD NUMBER                      |

\* = THESE LOCATIONS MUST BE IN THIS ORDER

"FLAGS"

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

```

804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859

```

| <div>! UNW! EXT! DAT! RTY! BOT! DEN!</div> <div>! INT! ! ! ! PAS! FLG!</div>                                                                                                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|                                                                                                                                                                                                                    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <div>--&gt; DENSITY FLAG</div> <div>--&gt; BOT PASS FLAG</div> <div>--&gt; RETRY IN PROGRESS FLAG</div> <div>--&gt; DATA TRANSFER FLAG</div> <div>--&gt; EXIT FLAG</div> <div>--&gt; INTERRUPT EXPECTED FLAG</div> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

"TMUNIT"

| 15                                                                                                                                                                                                                                                                      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| NOT USED                                                                                                                                                                                                                                                                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                                                                                                                                                                                                                                                         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <div>--&gt; TM78 #0 SELECT</div> <div>--&gt; TM78 #1 SELECT</div> <div>--&gt; TM78 #2 SELECT</div> <div>--&gt; TM78 #3 SELECT</div> <div>--&gt; TM78 #4 SELECT</div> <div>--&gt; TM78 #5 SELECT</div> <div>--&gt; TM78 #6 SELECT</div> <div>--&gt; TM78 #7 SELECT</div> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

0 = NOT SELECTED  
1 = SELECTED

"DVID1"

| 15                                                                                                                                                                                                                                                                      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| NOT USED                                                                                                                                                                                                                                                                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|                                                                                                                                                                                                                                                                         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <div>--&gt; TM78 #0 SELECT</div> <div>--&gt; TM78 #1 SELECT</div> <div>--&gt; TM78 #2 SELECT</div> <div>--&gt; TM78 #3 SELECT</div> <div>--&gt; TM78 #4 SELECT</div> <div>--&gt; TM78 #5 SELECT</div> <div>--&gt; TM78 #6 SELECT</div> <div>--&gt; TM78 #7 SELECT</div> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

0 = NOT SELECTED  
1 = SELECTED

"SR1"

| 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| NOT USED |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

```

860      ! ! ! ! ! --> 0=ALLOW DATA COMPARE
861      ! ! ! ! !      1=INHIBIT DATA COMPARE
862      ! ! ! ! ! --> 0=TEST AVAIL. TU78'S
863      ! ! ! ! !      1=TEST USER SEL. TU78'S
864      ! ! ! --> 0=REPORT ERRORS
865      ! ! !      1=DO NOT REPORT ERRORS
866      ! ! --> 0=STATISTICS AT EOT
867      ! !      1=NO STATISTICS AT EOT
868      ! --> 0=TEST GCR
869      !      1=DO NOT TEST GCR
870      -->0=TEST PE
871      1=DO NOT TEST PE

```

UNIT TABLE

|      |  | T U 7 8 #    |         |         |         |         |     |
|------|--|--------------|---------|---------|---------|---------|-----|
|      |  | 15           | 4       | 3       | 2       | 1       | 0   |
| UTBL |  | ! NOT USED ! | ! TM0 ! | ! TM0 ! | ! TM0 ! | ! TM0 ! | !   |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | !   |
|      |  | ! NOT USED ! | ! TM1 ! | ! TM1 ! | ! TM1 ! | ! TM1 ! | !   |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | !   |
|      |  | ! NOT USED ! | ! TM2 ! | ! TM2 ! | ! TM2 ! | ! TM2 ! | !   |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | ! T |
|      |  | ! NOT USED ! | ! TM3 ! | ! TM3 ! | ! TM3 ! | ! TM3 ! | ! M |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | ! 7 |
|      |  | ! NOT USED ! | ! TM4 ! | ! TM4 ! | ! TM4 ! | ! TM4 ! | ! 8 |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | ! # |
|      |  | ! NOT USED ! | ! TM6 ! | ! TM6 ! | ! TM6 ! | ! TM6 ! | !   |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | !   |
|      |  | ! NOT USED ! | ! TM7 ! | ! TM7 ! | ! TM7 ! | ! TM7 ! | !   |
|      |  | ! !          | ! TU3 ! | ! TU2 ! | ! TU1 ! | ! TU0 ! | ! V |

0 = TU NOT SELECTED FOR TEST (USER SELECTED)  
OR  
TU NOT ONLINE-WRITE ENABLE-AVAILABLE (PROGRAM SELECTED)

1 = TU SELECTED FOR TEST  
EOT TABLE

| T U 7 8 # |          |          |       |       |       |       |       |       |   |   |
|-----------|----------|----------|-------|-------|-------|-------|-------|-------|---|---|
| ←-----    |          |          |       |       |       |       |       |       |   |   |
|           | 15       |          | 4     | 3     | 2     | 1     | 0     |       |   |   |
| EOT TBL   | +        | -----    | +     | ----- | +     | ----- | +     | ----- | + |   |
|           | !        | NOT USED | !     | TM0   | !     | TM0   | !     | TM0   | ! | ! |
|           | !        |          | !     | TU3   | !     | TU2   | !     | TU1   | ! | ! |
|           | !        | -----    | +     | ----- | +     | ----- | +     | ----- | + | ! |
|           | !        | NOT USED | !     | TM1   | !     | TM1   | !     | TM1   | ! | ! |
|           | !        |          | !     | TU3   | !     | TU2   | !     | TU1   | ! | ! |
|           | !        | -----    | +     | ----- | +     | ----- | +     | ----- | + | ! |
|           | !        | NOT USED | !     | TM2   | !     | TM2   | !     | TM2   | ! | ! |
|           | !        |          | !     | TU3   | !     | TU2   | !     | TU1   | ! | ! |
|           | !        | -----    | +     | ----- | +     | ----- | +     | ----- | + | ! |
|           | !        | NOT USED | !     | TM3   | !     | TM3   | !     | TM3   | ! | ! |
|           | !        |          | !     | TU3   | !     | TU2   | !     | TU1   | ! | ! |
|           | !        | -----    | +     | ----- | +     | ----- | +     | ----- | + | ! |
|           | !        | NOT USED | !     | TM4   | !     | TM4   | !     | TM4   | ! | ! |
|           | !        |          | !     | TU3   | !     | TU2   | !     | TU1   | ! | ! |
|           | !        | -----    | +     | ----- | +     | ----- | +     | ----- | + | ! |
| !         | NOT USED | !        | TM5   | !     | TM5   | !     | TM5   | !     | ! |   |
| !         |          | !        | TU3   | !     | TU2   | !     | TU1   | !     | ! |   |
| !         | -----    | +        | ----- | +     | ----- | +     | ----- | +     | ! |   |
| !         | NOT USED | !        | TM6   | !     | TM6   | !     | TM6   | !     | ! |   |
| !         |          | !        | TU3   | !     | TU2   | !     | TU1   | !     | ! |   |
| !         | -----    | +        | ----- | +     | ----- | +     | ----- | +     | ! |   |
| !         | NOT USED | !        | TM7   | !     | TM7   | !     | TM7   | !     | ! |   |
| !         |          | !        | TU3   | !     | TU2   | !     | TU1   | !     | ! |   |
| !         | -----    | +        | ----- | +     | ----- | +     | ----- | +     | ! |   |

|   |
|---|
| T |
| M |
| 7 |
| 8 |
| # |
| V |

0 = TU NOT AT EOT

1 = TU AT EOT

SCRATCH UNIT TABLE

|          | 15       | 4   | 3   | 2   | 1   | 0 |   |
|----------|----------|-----|-----|-----|-----|---|---|
| SUTRL    | NOT USED | TM0 | TM0 | TM0 | TM0 |   | ! |
|          |          | TU3 | TU2 | TU1 | TU0 |   | ! |
|          | NOT USED | TM1 | TM1 | TM1 | TM1 |   | ! |
|          |          | TU3 | TU2 | TU1 | TU0 |   | ! |
|          | NOT USED | TM2 | TM2 | TM2 | TM2 |   | ! |
|          |          | TU3 | TU2 | TU1 | TU0 |   | ! |
|          | NOT USED | TM3 | TM3 | TM3 | TM3 |   | ! |
|          |          | TU3 | TU2 | TU1 | TU0 |   | ! |
|          | NOT USED | TM4 | TM4 | TM4 | TM4 |   | ! |
|          |          | TU3 | TU2 | TU1 | TU0 |   | ! |
| NOT USED | TM5      | TM5 | TM5 | TM5 |     | ! |   |
|          | TU3      | TU2 | TU1 | TU0 |     | ! |   |
| NOT USED | TM6      | TM6 | TM6 | TM6 |     | ! |   |
|          | TU3      | TU2 | TU1 | TU0 |     | ! |   |
| NOT USED | TM7      | TM7 | TM7 | TM7 |     | ! |   |
|          | TU3      | TU2 | TU1 | TU0 |     | ! |   |

1 = TU BEING TESTED

1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027

| 15 | 00 | TM78# | TU78# |
|----|----|-------|-------|
|    |    | 0     | 0     |
|    |    | 0     | 1     |
|    |    | 0     | 2     |
|    |    | 0     | 3     |
|    |    | 1     | 0     |
|    |    | 1     | 1     |
|    |    | 1     | 2     |

1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083

|   |   |   |
|---|---|---|
|   | 1 | 3 |
| . | . | . |
| : | : | : |
| . | . | . |
| : | : | : |
| . | . | . |
| 6 | 0 |   |
| 6 | 1 |   |
| 6 | 2 |   |
| 6 | 3 |   |
| 7 | 0 |   |
| 7 | 1 |   |
| 7 | 2 |   |
| 7 | 3 |   |

INDEX = ((TM78#(8))+(TU78#(2)))

## RECOVERABLE READ ERROR STATISTIC TABLE

"RREAD"

| 15    | 00 | TM78# | TU78# |
|-------|----|-------|-------|
| ----- |    | 0     | 0     |
| ----- |    | 0     | 1     |
| ----- |    | 0     | 2     |
| ----- |    | 0     | 3     |
| ----- |    | 1     | 0     |
| ----- |    | 1     | 1     |
| ----- |    | 1     | 2     |
| ----- |    | 1     | 3     |
| ----- |    | .     | .     |
| ----- | .  | .     | .     |
| ----- | .  | .     | .     |
| ----- | .  | .     | .     |

|      |   |   |   |
|------|---|---|---|
| 1084 | ! | . | . |
| 1085 | ! |   |   |
| 1086 | ! | 6 | 0 |
| 1087 | ! |   |   |
| 1088 | ! | 6 | 1 |
| 1089 | ! |   |   |
| 1090 | ! | 6 | 2 |
| 1091 | ! |   |   |
| 1092 | ! | 6 | 3 |
| 1093 | ! |   |   |
| 1094 | ! | 7 | 0 |
| 1095 | ! |   |   |
| 1096 | ! | 7 | 1 |
| 1097 | ! |   |   |
| 1098 | ! | 7 | 2 |
| 1099 | ! |   |   |
| 1100 | ! | 7 | 3 |
| 1101 | + |   |   |

INDEX = ((TM78#(8))+(TU78#(2)))

RECOVERABLE WRITE ERROR STATISTIC TABLE

"RWRIT"

|      | 15 | 00 | TM78# | TU78# |
|------|----|----|-------|-------|
| 1111 | +  |    |       |       |
| 1112 | !  |    | 0     | 0     |
| 1113 | !  |    | 0     | 1     |
| 1114 | !  |    |       |       |
| 1115 | !  |    | 0     | 2     |
| 1116 | !  |    |       |       |
| 1117 | !  |    | 0     | 3     |
| 1118 | !  |    |       |       |
| 1119 | !  |    | 1     | 0     |
| 1120 | !  |    | 1     | 1     |
| 1121 | !  |    | 1     | 2     |
| 1122 | !  |    | 1     | 3     |
| 1123 | !  |    | .     | .     |
| 1124 | !  |    | .     | .     |
| 1125 | !  |    | .     | .     |
| 1126 | !  |    | .     | .     |
| 1127 | !  |    | .     | .     |
| 1128 | !  |    | .     | .     |
| 1129 | !  |    | .     | .     |
| 1130 | !  |    | .     | .     |
| 1131 | !  |    | .     | .     |
| 1132 | !  |    | .     | .     |
| 1133 | !  |    | 6     | 0     |
| 1134 | !  |    | 6     | 1     |
| 1135 | !  |    | 6     | 2     |
| 1136 | !  |    |       |       |
| 1137 | !  |    |       |       |
| 1138 | !  |    |       |       |
| 1139 | !  |    |       |       |

|      |         |   |   |   |
|------|---------|---|---|---|
| 1140 | !       | ! | 6 | 3 |
| 1141 | !-----! |   |   |   |
| 1142 | !       | ! | 7 | 0 |
| 1143 | !-----! |   |   |   |
| 1144 | !       | ! | 7 | 1 |
| 1145 | !-----! |   |   |   |
| 1146 | !       | ! | 7 | 2 |
| 1147 | !-----! |   |   |   |
| 1148 | !       | ! | 7 | 3 |
| 1149 | +-----+ |   |   |   |
| 1150 |         |   |   |   |
| 1151 |         |   |   |   |
| 1152 |         |   |   |   |
| 1153 |         |   |   |   |

INDEX = ((TM78\*(8))+(TU78\*(2)))

-

```

1155 000000' IOMODX <TM78>,172400,224,5,0,0,1,0,0,BUFIN,512,512.
(1) 000000' MODULE 150000, TM78, 172400, 224, 5, 0, 0, 1, 0, 0, BUFIN, 512, 512.
(2) .TITLE TM78 DEC/X11 SYSTEM EXERCISER MODULE
(2) ; DDSCOM VERSION 6 23-MAY-78
(2) .LIST BIN
(2) ;*****
(2) BEGIN:
(2) 000000' 046524 040504 040 MODNAM: .ASCII /TM78 / :MODULE NAME.
(2) 000005' 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
(2) 000006' 172400 ADDR: 172400+0 ;1ST DEVICE ADDR.
(2) 000010' 000224 VECTOR: 224+0 ;1ST DEVICE VECTOR.
(2) 000012' 240 BR1: .BYTE PRTY5+0 ;1ST BR LEVEL.
(2) 000013' 000 BR2: .BYTE PRTY0+0 ;2ND BR LEVEL.
(2) 000014' 000001 DVID1: 0+1 ;DEVICE INDICATOR 1.
(2) 000016' 000000 SR1: OPEN ;SWITCH REGISTER 1
(2) 000020' 000000 SR2: OPEN ;SWITCH REGISTER 2
(2) 000022' 000000 SR3: OPEN ;SWITCH REGISTER 3
(2) 000024' 000000 SR4: OPEN ;SWITCH REGISTER 4
(2) ;*****
(2) 000026' 150000 STAT: 150000 ;STATUS WORD.
(2) 000030' 000252' INIT: START ;MODULE START ADDR.
(2) 000032' 000252' SPOINT: MODSP ;MODULE STACK POINTER.
(2) 000034' 000000 PASCNT: 0 ;PASS COUNTER.
(2) 000036' 000001 ICONT: 1. ;# OF ITERATIONS PER PASS=1.
(2) 000040' 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
(2) 000042' 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
(2) 000044' 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
(2) 000046' 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
(2) 000050' 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
(2) 000052' 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
(2) 000054' 000000 RANUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
(2) 000056' CONFIG: ;RESERVED FOR MONITOR USE
(2) 000056' 000000 RES1: 0 ;RESERVED FOR MONITOR USE
(2) 000060' 000000 RES2: 0 ;RESERVED FOR MONITOR USE
(2) 000062' 000000 SVR0: OPEN ;LOC TO SAVE R0.
(2) 000064' 000000 SVR1: OPEN ;LOC TO SAVE R1.
(2) 000066' 000000 SVR2: OPEN ;LOC TO SAVE R2.
(2) 000070' 000000 SVR3: OPEN ;LOC TO SAVE R3.
(2) 000072' 000000 SVR4: OPEN ;LOC TO SAVE R4.
(2) 000074' 000000 SVR5: OPEN ;LOC TO SAVE R5.
(2) 000076' 000000 SVR6: OPEN ;LOC TO SAVE R6.
(2) 000100' 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
(2) 000102' SBADR: ;ADDR OF GOOD DATA, OR
(2) 000102' 000000 ACSR: OPEN ;CONTENTS OF CSR.
(2) 000104' WASADR: ;ADDR OF BAD DATA, OR
(2) 000104' 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
(2) 000106' ERRTP: ;TYPE OF ERROR
(2) 000106' 000000 ASB: OPEN ;EXPECTED DATA.
(2) 000110' 000000 AWAS: OPEN ;ACTUAL DATA.
(2) 000112' 001566' RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
(2) 000114' 000000 WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
(2) 000116' 000000 WDFF: OPEN ;WORDS FROM MEMORY PER ITERATION
(2) 000120' 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
(2) 000122' 000000 IDNUM: 0 ;MODULE IDENTIFICATION NUMBER=0
(2) 000124' 011174' RBUFA: BUFIN ;READ BUFFER VIRTUAL ADDRESS
(2) 000126' 000000 RBUFFA: OPEN ;READ BUFFER PHYSICAL ADDRESS

```

```

(2) 000130' 000000 RBUFEA: OPEN ;READ BUFFER EA BITS
(2) 000132' 001000 RBUFSZ: 512. ;SIZE OF THE READ BUFFER
(2) 000134' 000000 WBUFEA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
(2) 000136' 000000 WBUFEA: OPEN ;WRITE BUFFER EA BITS
(2) 000140' 001000 WBUFRQ: 512. ;WRITE BUFFER SIZE REQUESTED
(2) 000142' 000000 WBUFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
(2) 000144' 000000 CDRECT: OPEN ;CDATA/DATCK ERROR COUNT
(2) 000146' 000000 CDWDCT: OPEN ;CDATA/DATCK WORD COUNT
(2) 000150' 000000 FREE: OPEN ;RESERVED FOR FUTURE USE
(2) 000040 ;MODULE STACK STARTS HERE.
(2) .REPT SPSIZ
(2) .NLIST
(2) .WORD 0
(2) .LIST
(2) .ENDP
(2) 000252' MODSP:
(2) ;*****
1156 .REM
1157 13.0 PDL FOR THE TM78 DEC/X11 DRIVER
1158 -----
1159
1160
1161 BEGINROUTINE MODULE 1.0 (DRIVER INITIALIZATION)
1162 : CLEAR "FLAGS"
1163 : CALL MODULE 1.1 (BUILD I/O REGISTER MAP)
1164 : CALL MODULE 1.2 (RH11/70 INITIALIZE)
1165 : CALL MODULE 1.3 (CHECK TM78 DRIVE TYPE)
1166 : IF TMUNIT = 0
1167 : : THEN-CLEAR RH11/70 REGISTER CS1
1168 : : PRINT ERROR 1
1169 : : DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
1170 : : ELSE-CONTINUE
1171 : ENDF
1172 : CALL MODULE 1.4 (BUILD TU78 UNIT TABLE)
1173 : IF UNIT COUNT = 0
1174 : : THEN-CLEAR RH11/70 REGISTER CS1
1175 : : PRINT ERROR 11
1176 : : DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
1177 : : ELSE-CONTINUE
1178 : ENDF
1179 : CALL MODULE 1.5 (INITIALIZE PROGRAM DATA STRUCTURES)
1180 : IF "SR1" BIT4 AND BIT5=1 (NO DENSITY SELECTED)
1181 : : THEN-CLEAR RH11/70 REGISTER CS1
1182 : : PRINT ERROR 12
1183 : : DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
1184 : : ELSE-CONTINUE
1185 : ENDF
1186 ENDROUTINE
1187
1188
1189
1190 000252' 005067 010216 START: CLR FLAGS ;CLEAR THE FLAGS
1191 000256' 004767 000144 CALL M1.1 ;CALL MODULE 1.1 (BUILD I/O REGISTER MAP)
1192 000262' 004767 000174 CALL M1.2 ;CALL MODULE 1.2 (RH11/70 INITIALIZE)
1193 000266' 004767 000200 CALL M1.3 ;CALL MODULE 1.3 (CHECK TM78 DRIVE TYPE)
1194 000272' 005767 010200 TST TMUNIT ;IF TMUNIT=0
1195 000276' 001011 BNE 18 ;ELSE-CONTINUE

```

```

1196 000300' 005077 010570          CLR    #CS1          ;THEN-CLEAR TH/70 REGISTER CS1
1197 000304' 012767 000001 010146    MOV    #1,ERRINX      ;PRINT ERROR 1
1198 000312' 004767 006316          CALL    M4.1
1199 000316' 104410 000000'          ENDS,BEGIN
1200 000322' 004767 000366          1S:  CALL    M1.4          ;CALL MODULE 1.4 (BUILD TU78 UNIT TABLE)
1201 000326' 005767 010106          TST     UCNT          ;IF UNIT COUNT=0
1202 000332' 001011          BNE     2S          ;ELSE-CONTINUE
1203 000334' 005077 010534          CLR    #CS1          ;THEN-CLEAR RH11/70 REGISTER CS1
1204 000340' 012767 000011 010112    MOV    #11,ERRINX     ;PRINT ERROR 11
1205 000346' 004767 006262          CALL    M4.1
1206 000352' 104410 000000'          ENDS,BEGIN
1207 000356' 004767 000762          2S:  CALL    M1.5          ;CALL MODULE 1.5 (INITIALIZE DATA STRUCTURES)
1208 000362' 016701 177430          MOV     SR1,R1
1209 000366' 042701 177717          BIC     #1-BIT4-BIT5,R1 ;
1210 000372' 022701 000060          CMP     #BIT4+BIT5,R1 ;IF NO DENSITY SELECTED
1211 000376' 001011          BNE     3S          ;ELSE-CONTINUE
1212 000400' 005077 010470          CLR    #CS1          ;THEN-CLEAR RH11/70 REGISTER CS1
1213 000404' 012767 000012 010046    MOV    #12,ERRINX     ;PRINT ERROR 12
1214 000412' 004767 006216          CALL    M4.1
1215 000416' 104410 000000'          ENDS,BEGIN
1216 000422' 000167 001142          3S:  JMP     M2.0          ;DEC/X11 RESTART CODE

```

```

1218          .REM
1219          BEGINROUTINE  MODULE 1.1 (BUILD I/O REGISTER MAP)
1220          : GET DEC/X11 HEADER WORD "ADDR"
1221          : CLEAR LOOP COUNT
1222          : BGND0
1223          : : STORE ("ADDR"+LOOP COUNT) AT (RHTBL+LOOP COUNT)
1224          : : ACCESS THE UNIBUS ADDRESS ("ADDR"+LOOP COUNT)
1225          : : ADD 2 TO LOOP COUNT
1226          : : DO UNTIL LOOP COUNT = 100(8)
1227          : ENDD0
1228          ENDROUTINE
1229
1230          -
1231
1232 000426' 016701 177354          M1.1:  MOV     ADDR,R1          ;GET DEC/X11 HEADER WORD "ADDR"
1233 000432' 005711          TST     (R1)          ;CATCH WRONG DEVICE ADDRESS EARLY
1234 000434' 005002          CLR     R2          ;CLEAR THE LOOP COUNT
1235 000436' 010162 011074'          1S:  MOV     R1,RHTBL(R2)      ;STORE ("ADDR"+LOOP COUNT) AT (RHTBL+LOOP COUNT)
1236 000442' 062701 000002          ADD     #2,R1          ;NEXT REGISTER
1237 000446' 062702 000002          ADD     #2,R2          ;ADD 2 TO THE LOOP COUNT
1238 000452' 022702 000100          CMP     #100,R2         ;DO UNTIL THE LOOP COUNT=100
1239 000456' 001367          BNE     1S          ;
1240 000460' 000207          RTS     PC          ;ENDROUTINE

```

```

1242      .REM
1243      BEGINROUTINE      MODULE 1.2 (RH11/70 INITIALIZE)
1244      : SET THE "CLR" BIT IN RH11/70 CS2 (000040)
1245      ENDROUTINE
1246
1247
1248      -
1249      000462' 052777 000040 010414 M1.2: BIS      #BIT5,ACS2      ;SET THE CLEAR BIT IN RH11/70 CS2
1250      000470' 000207      RTS      PC      ;ENDROUTINE
1251
1252      .REM
1253      BEGINROUTINE      MODULE 1.3 (CHECK TM78 DRIVE TYPE)
1254      : CLEAR THE TM78#
1255      : IF BIT1 OF SR1 = 1 (USER SELECTED TU78'S)
1256      : : THEN - COPY "DVID1" TO "TMUNIT"
1257      : : ELSE - SET "TMUNIT" = 377(8)
1258      : ENDF
1259      : BGNDO
1260      : : IF BIT#(TM78#) OF "TMUNIT" = 0
1261      : : : THEN-CONTINUE
1262      : : : ELSE-LOAD RH11/70 CS2 REG WITH TM78#
1263      : : : : SET TRE IN RH11/70 IN REG CS1
1264      : : : : CLEAR REGISTER CS1
1265      : : : : IF NED BIT IN RH11/70 REGISTER CS2 = 0
1266      : : : : : THEN - IF BITS 8:0 OF RH11/70 REGISTER 26 = 101(8)
1267      : : : : : : THEN-CONTINUE
1268      : : : : : : ELSE-CLEAR BIT#(TM78#) OF "TMUNIT"
1269      : : : : : : ENDF
1270      : : : : : ELSE - CLEAR BIT#(TM78#) OF "TMUNIT"
1271      : : : : ENDF
1272      : : ENDF
1273      : : INCREMENT THE TM78#
1274      : : DO UNTIL THE TM78# = 10(8)
1275      : ENDDO
1276      ENDROUTINE
1277
1278      -
1279      000472' 005067 007744 M1.3: CLR      TM78      ;CLEAR THE TM78#
1280      000476' 012767 000377 007772 MOV      #377,TMUNIT ;SELECT ALL TM78'S
1281      000504' 032767 000002 177304 BIT      #BIT1,SR1 ;IF SR1 BIT1 = 1 (USER SELECTED TU78'S)
1282      000512' 001403      BEQ      1$ ;ELSE - CONTINUE
1283      000514' 016767 177274 007754 MOV      DVID1,TMUNIT ;THEN - COPY DEC/X11 VARIABLE "DVID1" TO "TMUNIT"
1284      000522' 004767 000734      CALL      TMCONV
1285      000526' 036767 007676 007742 BIT      TMBIT,TMUNIT ;IF BIT#(TM78#) OF "TMUNIT"=0
1286      000534' 001433      BEQ      3$ ;THEN-CONTINUE
1287      000536' 016777 007700 010340 MOV      TM78,ACS2 ;SELECT UNIT #
1288      000544' 000240      NOP
1289      000546' 012777 040000 010320 MOV      #BIT14,ACS1 ;SET BIT TRE IN RH11/70 REGISTER CS1
1290      000554' 005077 010314      CLR      ACS1 ;CLEAR BIT TRE IN RH11/70 REGISTER CS1
1291      000560' 032777 010000 010316 BIT      #BIT12,ACS2 ;IF BIT NED IN RH11/70 REGISTER CS2 = 0
1292      000566' 001404      BEQ      2$ ;THEN - GO TO THEN
1293      000570' 046767 007634 007700 BIC      TMBIT,TMUNIT ;ELSE - CLEAR BIT#(TM78#) OF "TMUNIT"
1294      000576' 000412      BR      3$ ;CONTINUE
1295      000600' 017701 010316      MOV      #DT,R1 ;IF BITS 8:0 OF RH11/70 REGISTER26=101
1296      000604' 042701 177000      BIC      #177000,R1
1297      000610' 022701 000101      CMP      #101,R1

```

```

1298      000614' 001412      BEQ      4$ ;THEN-CONTINUE
1299      000616' 046767 007606 007652 BIC      TMBIT,TMUNIT ;ELSE-CLEAR BIT#(TM78#) OF "TMUNIT"
1300      000624' 005267 007612      INC      TM78 ;INCREMENT THE TM78#
1301      000630' 022767 000010 007604 CMP      #10,TM78 ;DO UNTIL THE TM78#=10
1302      000636' 001331      BNE      1$
1303      000640' 000424      BR      6$ ;EXIT
1304      000642' 012704 000500      MOV      #500,R4 ;SET LOOP COUNTER
1305      000646' 005777 010274      TST      @ST78 ;TM READY ?
1306      000652' 100764      BMI      3$ ;YES , CHECK NEXT TM78
1307      000654' 104407 000000' BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
1308      000660' 104407 000000' BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1309      000664' 005304      DEC      R4 ;COUNT DOWN
1310      000666' 001367      BNE      5$ ;BR IF NOT TIMED OUT
1311      000670' 012767 000013 007562 MOV      #13,ERRINX ;SET ERROR #13
1312      000676' 004767 005732      CALL      M4.1 ;GO TO ERROR TYPE ROUTINE
1313      000702' 046767 007522 007566 BIC      TMBIT,TMUNIT ;DROP THIS TM78
1314      000710' 000745      BR      3$ ;CHECK NEXT TM78
1315      000712' 000207      RTS      PC ;ENDROUTINE

```

```

1316 .REM
1317 BEGINROUTINE MODULE 1.4 (BUILD TU78 UNIT TABLE)
1318 : SET UP INTERRUPT VECTOR (MODULE 4.0) AND BR LEVEL
1319 : IF SR1 BIT1=1 (USER SELECTED TU78'S)
1320 : : THEN-COPY USER TABLE "UTBL" TO SCRATCH UNIT TABLE "SUTBL"
1321 : : CLEAR TM78# "TM78"
1322 : : CLEAR UNIT COUNT "UCNT"
1323 : : BGNDO
1324 : : : IF BIT*(TM78#) OF "TMUNIT" = 0
1325 : : : : THEN-CONTINUE
1326 : : : : ELSE-LOAD RH11/70 CS2 REGISTER WITH TM78#
1327 : : : : CLEAR THE TU78#
1328 : : : : BGNDO
1329 : : : : : IF SCRATCH UNIT TABLE (TM78#,TU78#) = 1
1330 : : : : : : THEN-CALL MODULE 3.6 (SENSE)
1331 : : : : : : IF TERM. INT. CODE=DONE
1332 : : : : : : : THEN-CONTINUE
1333 : : : : : : : ELSE-CLEAR THE TU78 STATUS
1334 : : : : : : : ENDIF
1335 : : : : : IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
1336 : : : : : : THEN-INCREMENT UNIT COUNT
1337 : : : : : : ELSE-DROP TU78 - CLEAR BIT (SCRATCH UNIT TABLE(TM78#
1338 : : : : : : : ENDIF
1339 : : : : : : ELSE-CONTINUE
1340 : : : : : : ENDIF
1341 : : : : : INCREMENT THE TU78#
1342 : : : : : DO UNTIL THE TU78# = 4
1343 : : : : : ENDDO
1344 : : : : : ENDIF
1345 : : : : : INCREMENT THE TM78#
1346 : : : : : DO UNTIL THE TM78# = 10(8)
1347 : : : : : ENDDO
1348 : : : ELSE-CLEAR THE TM78#
1349 : : : : CLEAR THE UNIT COUNT
1350 : : : : BGNDO
1351 : : : : : IF BIT*(TM78#) OF TMUNIT = 0
1352 : : : : : : THEN - CLEAR (SCRATCH UNIT TABLE(TM78#(2)))
1353 : : : : : : : CLEAR (UNIT TABLE(TM78#(2)))
1354 : : : : : : : ELSE - LOAD THE RH11/70 CS2 REGISTER WITH TM78#
1355 : : : : : : : CLEAR THE TU78#
1356 : : : : : : : BGNDO
1357 : : : : : : : : CALL MODULE 3.6 (SENSE)
1358 : : : : : : : : IF TERM. INT. CODE=DONE
1359 : : : : : : : : THEN-CONTINUE
1360 : : : : : : : : ELSE-CLEAR THE TU78 STATUS
1361 : : : : : : : : ENDIF
1362 : : : : : : : IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
1363 : : : : : : : : THEN-SET BIT (SCRATCH UNIT TABLE(TM78#,TU78#))
1364 : : : : : : : : : SET BIT (UNIT TABLE(TM78#, TU78#))
1365 : : : : : : : : : INCREMENT UNIT COUNT
1366 : : : : : : : : : ELSE-CONTINUE
1367 : : : : : : : : : ENDIF
1368 : : : : : : : : INCREMENT THE TU78#
1369 : : : : : : : : DO UNTIL THE TU78# = 4
1370 : : : : : : : : ENDDO
1371 : : : : : : : ENDIF

```

```

1372 : : : INCREMENT THE TM78#
1373 : : : DO UNTIL THE TM78# = 10(8)
1374 : : : ENDDO
1375 : : ENDIF
1376 ENDROUTINE
1377 -
1378
1379 000714' 016700 177070 M1.4: MOV VECTOR,R0 ;SET UP INTERRUPT VECTOR
1380 000720' 012720 006314' MOV #M4.0,(R0)+
1381 000724' 116720 177062 MOV# BR1,(R0)+ ;SETUP BR LEVEL
1382 000730' 105010 CLR# (R0)
1383 000732' 032767 000002 177056 BIT #BIT1,SR1 ;IF SR1 BIT1=1
1384 000740' 001504 BEQ M1.4X ;ELSE-GO TO ELSE CODE
1385 000742' 005001 CLR R1 ;SET LOOP COUNT
1386 000744' 016161 010514' 010554' 1s: MOV UTBL(R1),SUTBL(R1) ;COPY UNIT TABLE TO SCRATCH UNIT TABLE
1387 000752' 062701 000002 ADD #2,R1
1388 000756' 022701 000020 CMP #20,R1
1389 000762' 001370 BNE 1s
1390 000764' 005067 007452 CLR TM78 ;CLEAR THE TM78#
1391 000770' 005067 007444 CLR UCNT ;CLEAR THE UNIT COUNT
1392 000774' 004767 000462 2s: CALL TMCONV
1393 001000' 036767 007424 007470 BIT TMBIT,TMUNIT ;IF BIT*(TM78#) OF "TMUNIT"=0
1394 001006' 001452 BEQ 7s ;THEN-CONTINUE
1395 001010' 016777 007426 010066 MOV TM78,#CS2 ;ELSE-LOAD RH11/70 CS2 REGISTER WITH TM78#
1396 001016' 005067 007422 CLR TU78 ;CLEAR THE TU78#
1397 001022' 004767 000476 3s: CALL TUCONV
1398 001026' 016701 007400 MOV TMBIT2,R1
1399 001032' 036761 007376 010554' BIT TUBIT,SUTBL(R1) ;IF SCRATCH UNIT TABLE (TM78#,TU78#)=1
1400 001040' 001427 BEQ 6s ;ELSE-CONTINUE
1401 001042' 004767 005164 CALL M3.6 ;CALL MODULE 3.6 (SENSE)
1402 001046' 022767 000001 007374 CMP #DONE,TINTCD ;IF SENSE TERMINATION INTERRUPT CODE=DONE
1403 001054' 001402 BEQ 4s ;THEN-CONTINUE
1404 001056' 005067 007374 CLR SENSE ;ELSE-CLEAR SENSE
1405 001062' 016701 007370 4s: MOV SENSE,R1
1406 001066' 042701 017177 BIC #017177,R1
1407 001072' 022701 160200 CMP #160200,R1 ;IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
1408 001076' 001406 BEQ 5s ;THEN-GO DO THEN
1409 001100' 016701 007326 MOV TMBIT2,R1 ;ELSE-DROP TU78
1410 001104' 046761 007324 010554' BIC TUBIT,SUTBL(R1) ;CLEAR BIT (SCRATCH UNIT TABLE(TM78#,TU78#))
1411 001112' 000402 BR 6s
1412 001114' 005267 007320 5s: INC UCNT ;INCREMENT THE UNIT COUNT
1413 001120' 005267 007320 6s: INC TU78 ;INCREMENT THE TU78#
1414 001124' 022767 000004 007312 CMP #4,TU78 ;DO UNTIL THE TU78#=4
1415 001132' 001333 BNE 3s
1416 001134' 005267 007302 7s: INC TM78 ;INCREMENT THE TM78#
1417 001140' 022767 000010 007274 CMP #10,TM78 ;DO UNTIL THE TM78#=10
1418 001146' 001312 BNE 2s
1419 001150' 000474 BR M1.4OT
1420 001152' 005067 007264 M1.4X: CLR TM78 ;CLEAR THE TM78#
1421 001156' 005067 007256 CLR UCNT ;CLEAR THE UNIT COUNT
1422 001162' 004767 000274 1s: CALL TMCONV
1423 001166' 016701 007240 MOV TMBIT2,R1
1424 001172' 005061 010554' CLR SUTBL(R1) ;CLEAR THE SCRATCH UNIT TABLE FOR THIS TM78
1425 001176' 005061 010514' CLR UTBL(R1) ;CLEAR THE UNIT TABLE FOR THIS TM78
1426 001202' 036767 007222 007266 BIT TMBIT,TMUNIT ;IF BIT*(TM78#) OF TMUNIT = 0
1427 001210' 001446 BEQ 4s ;THEN - CONTINUE

```

```

1428 001212' 016777 007224 007664      MOV    TM78,ACS2      ;ELSE - LOAD THE RH11/70 CS2 REGISTER WITH TM78#
1429 001220' 005067 007220                CLR    TU78          ;CLEAR THE TU78#
1430 001224' 004767 000274                2S:   CALL   TUCONV          ;
1431 001230' 004767 004776                CALL   M3.6          ;CALL MODULE 3.6 (SENSE)
1432 001234' 022767 000001 007206        CMP    #DONE,TINTCD  ;IF SENSE TERMINATION INTERRUPT CODE=DONE
1433 001242' 001402                BEQ    5S              ;THEN-CONTINUE
1434 001244' 005067 007206                CLR    SENSE          ;ELSE-CLEAR SENSE
1435 001250' 016701 007202                5S:   MOV    SENSE,R1
1436 001254' 042701 017177                BIC    #017177,R1
1437 001260' 022701 160200                CMP    #160200,R1    ;IF TU78 IS RDY.PRES.UNL.AVAIL AND NOT FPT
1438 001264' 001012                BNE    3S              ;ELSE-CONTINUE
1439 001266' 016701 007140                MOV    TMBIT2,R1
1440 001272' 056761 007136 010554'        BIS    TUBIT,SUTBL(R1) ;SET BIT SCRATCH UNIT TABLE(TM78#,TU78#)
1441 001300' 056761 007130 010514'        BIS    TUBIT,UTBL(R1) ;SET BIT UNIT TABLE(TM78#,TU78#)
1442 001306' 005267 007126                INC    UCNT          ;INCREMENT THE UNIT COUNT
1443 001312' 005267 007126                3S:   INC    TU78          ;INCREMENT THE TU78#
1444 001316' 022767 000004 007120        CMP    #4,TU78      ;DO UNTIL THE TU78#4
1445 001324' 001337                BNE    2S              ;
1446 001326' 005267 007110                4S:   INC    TM78          ;INCREMENT THE TM78#
1447 001332' 022767 000010 007102        CMP    #10,TM78     ;DO UNTIL THE TM78#10
1448 001340' 001310                BNE    1S              ;
1449 001342' 000207                M1.4UT: RTS         PC      ;RETURN TO USER.

```

```

1451      .REM
1452      BEGINROUTINE      MODULE 1.5 (INITIALIZE PROGRAM DATA STRUCTURES)
1453      : CLEAR THE RECOVERABLE WRITE STATISTIC TABLE
1454      : CLEAR THE RECOVERABLE READ STATISTIC TABLE
1455      : CLEAR THE UNRECOVERABLE READ STATISTIC TABLE
1456      : CLEAR THE EOT TABLE
1457      : IF SR1 BIT4 = 0 (TEST GCR)
1458      : : THEN - SET BIT0 OF FLAGS (GCR)
1459      : : ELSE - CLEAR BIT0 OF FLAGS (PE)
1460      : ENDDIF
1461      : SET THE BOT PASS FLAG
1462      : CLEAR RECORD NUMBER
1463      : SET FILE NUMBER=1
1464      ENDRROUTINE
1465
1466      -
1467
1468 001344' 005001                M1.5: CLR    R1              ;CLEAR THE LOOP COUNT
1469 001346' 005061 010774'        1S:   CLR    RWRITE(R1)         ;CLEAR RECOVERABLE WRITE STATISTIC TABLE
1470 001352' 005061 010674'        CLR    RREAD(R1)          ;CLEAR RECOVERABLE READ STATISTIC TABLE
1471 001356' 005061 010574'        CLR    URREAD(R1)         ;CLEAR UNRECOVERABLE READ STATISTIC TABLE
1472 001362' 062701 000002        ADD    #2,R1              ;INCREMENT THE LOOP COUNT
1473 001366' 022701 000100        CMP    #64..R1             ;DO UNTIL THE LOOP COUNT=64.
1474 001372' 001365                BNE    1S
1475 001374' 005001                CLR    R1              ;CLEAR THE LOOP COUNT
1476 001376' 005061 010534'        2S:   CLR    EOTTHL(R1)        ;CLEAR THE EOT TABLE
1477 001402' 062701 000002        ADD    #2,R1              ;INCREMENT THE LOOP COUNT
1478 001406' 022701 000020        CMP    #16..R1             ;DO UNTIL THE LOOP COUNT=16
1479 001412' 001371                BNE    2S
1480 001414' 052767 000001 007052        BIS    #BIT0,FLAGS      ;SET THE DENSITY TO GCR
1481 001422' 032767 000020 176366        BIT    #BIT4,SR1      ;IF SR1 BIT4 = 0 (TEST GCR)
1482 001430' 001410                BEQ    3S              ;THEN - SET THE DENSITY TO GCR
1483 001432' 042767 000001 007034        BIC    #BIT0,FLAGS      ;ELSE - SET THE DENSITY TO PE
1484 001440' 005067 007040                CLR    RECORD          ;CLEAR RECORD NUMBER
1485 001444' 012767 000001 007030        MOV    #1,FILE        ;SET FILE NUMBER=1
1486 001452' 052767 000002 007014        3S:   BIS    #BIT1,FLAGS      ;SET THE BOT PASS FLAG
1487 001460' 000207                RTS    PC              ;ENDROUTINE
1488
1489 001462' 016701 006754                TMCONV: MOV    TM78,R1
1490 001466' 012702 000001                MOV    #1,R2
1491 001472' 005701                1S:   TST    R1
1492 001474' 001403                BEQ    2S
1493 001476' 005301                DEC    R1
1494 001500' 006302                ASL    R2
1495 001502' 000773                BR     1S
1496 001504' 010267 006720                2S:   MOV    R2,TMBIT
1497 001510' 016701 006726                MOV    TM78,R1
1498 001514' 006301                ASL    R1
1499 001516' 010167 006710                MOV    R1,TMBIT2
1500 001522' 000207                RTS    PC

```

```

1502 001524' 016701 006714 TUCONV: MOV TU78,R1
1503 001530' 012702 000001      MOV #1,R2
1504 001534' 005701      1s: TST R1
1505 001536' 001403      BEQ 2s
1506 001540' 005301      DEC R1
1507 001542' 006302      ASL R2
1508 001544' 000773      BR 1s
1509 001546' 010267 006662      2s: MOV R2,TUBIT
1510 001552' 016701 006666      MOV TU78,R1
1511 001556' 006301      ASL R1
1512 001560' 010167 006652      MOV R1,TUBIT2
1513 001564' 000207      RTS PC
1514      .REM
1515      BEGINROUTINE MODULE 2.0 (DRIVER SEQUENCE CONTROL)
1516      : BGND0
1517      : : ENABLE RH11/70 INTERRUPTS
1518      : : CLEAR INTERRUPT EXPECTED FLAG
1519      : : CLEAR THE TM78#
1520      : : INCREMENT RECORD NUMBER
1521      : : BGND0
1522      : : : GET WRITE BUFFER (DEC/X11 GWBUFF)
1523      : : : CLEAR THE TU78#
1524      : : : BGND0
1525      : : : : IF THE TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#)) = 0
1526      : : : : THEN-CONTINUE
1527      : : : : ELSE-IF BOT PASS FLAG NOT=0
1528      : : : : : THEN-CALL MODULE 2.7 (REWIND CONTROL)
1529      : : : : : ELSE-IF EOT FLAG=1(EOTBL(TM78#,TU78#))
1530      : : : : : : THEN-CONTINUE
1531      : : : : : : ELSE-CALL MODULE 2.3 (WRITE CONTROL)
1532      : : : : : : IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#)
1533      : : : : : : : THEN-CALL MODULE 2.5 (READ REVERSE CONTROL)
1534      : : : : : : : ELSE-CONTINUE
1535      : : : : : : : ENDIF
1536      : : : : : : IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#)
1537      : : : : : : : THEN-CALL MODULE 2.8 (COMPARE DATA)
1538      : : : : : : : CALL MODULE 2.4 (READ FORWARD CONTROL)
1539      : : : : : : : ELSE-CONTINUE
1540      : : : : : : : ENDIF
1541      : : : : : : IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#)
1542      : : : : : : : THEN-CALL MODULE 2.8 (COMPARE DATA)
1543      : : : : : : : ELSE-CONTINUE
1544      : : : : : : : ENDIF
1545      : : : : : : : ENDIF
1546      : : : : : : : ENDIF
1547      : : : : : : : ENDIF
1548      : : : : : : : INCREMENT THE TU78#
1549      : : : : : : : DO UNTIL THE TU78# = 4
1550      : : : : : : : ENDDO
1551      : : : : : : : INCREMENT THE TM78# BY 1
1552      : : : : : : : DO UNTIL THE TM78# = 10(8)
1553      : : : : : : : ENDDO
1554      : : : : : : : CALL MODULE 2.10 (UNIT CONTROL)
1555      : : : : : : : IF UNIT COUNT=0
1556      : : : : : : : : THEN-CLEAR RH11/70 REGISTER CS1
1557      : : : : : : : : PRINT ERROR 11

```

```

1558      : : : DROP THIS DEC/X11 DRIVER (ENDMOD)
1559      : : : ELSE-CONTINUE
1560      : : : ENDIF
1561      : : : IF BOT PASS FLAG=1
1562      : : : : THEN-BGND0
1563      : : : : : CALL MODULE 2.10 (UNIT CONTROL)
1564      : : : : : CALL MODULE 2.2 (BOT CONTROL)
1565      : : : : : DO UNTIL UNIT COUNT=0 OR UNIT COUNT=BOT COUNT
1566      : : : : : ENDDO
1567      : : : : : CLEAR THE EOT COUNT
1568      : : : : : CLEAR THE EOT TABLE
1569      : : : : : SET FILE NUMBER=1
1570      : : : : : CLEAR RECORD NUMBER
1571      : : : : : ELSE-CONTINUE
1572      : : : : : ENDIF
1573      : : : : : CLEAR THE BOT PASS FLAG
1574      : : : : : IF UNIT COUNT=EOT COUNT
1575      : : : : : : THEN-SET BOT PASS FLAG
1576      : : : : : : CALL MODULE 2.9 (STATISTICS)
1577      : : : : : : CALL MODULE 2.1 (DENSITY CONTROL)
1578      : : : : : : ELSE-CONTINUE
1579      : : : : : : ENDIF
1580      : : : : : : IF RECORD NUMBER=500. OR BOT PASS FLAG=1
1581      : : : : : : : THEN-CALL MODULE 2.6 (TAPE MARK CONTROL)
1582      : : : : : : : INCREMENT FILE NUMBER
1583      : : : : : : : ELSE-CONTINUE
1584      : : : : : : : ENDIF
1585      : : : : : : : DO UNTIL RECORD NUMBER=500. OR BOT PASS FLAG=1
1586      : : : : : : : ENDDO
1587      : : : : : : : CLEAR THE RECORD NUMBER
1588      : : : : : : : END THE PASS (DEC/X11 ENDIT)
1589      : : : : : : : ENDRROUTINE
1590
1591      -
1592 001566' 000240      RESTRT: NOP
1593 001570' 042767 000040 006676 M2.0: BIC #BITS,FLAGS ;DEC/X11 RESTART TAG
1594 001576' 012777 000100 007270      MOV #BIT6,CS1 ;CLEAR INTERRUPT EXPECTED FLAG
1595 001604' 005067 006632      CLR TM78 ;ENABLE RH11/70 INTERRUPTS
1596 001610' 104414 000000'      GWBUFF, BEGIN ;CLEAR THE TM78#
1597 001614' 005267 006664      INC RECORD ;GET WRITE BUFFER INFORMATION
1598 001620' 005067 006620      1s: CLR TU78 ;INCREMENT THE RECORD NUMBER
1599 001624' 004767 177632      2s: CALL TUCONV ;CLEAR TU78#
1600 001630' 004767 177670      CALL TUCONV
1601 001634' 016701 006572      MOV TUBIT2,R1
1602 001640' 036761 010554'      BIT TUBIT,SUTBL(R1) ;IF TU78 SELECTED BIT=0
1603 001646' 001451      BEQ 6s ;THEN-CONTINUE
1604 001650' 032767 000002 006616      BIT #BIT1,FLAGS ;ELSE-IF BOT PASS FLAG NOT=0
1605 001656' 001403      BEQ 3s
1606 001660' 004767 002450      CALL M2.7 ;THEN-CALL MODULE 2.7 (REWIND CONTROL)
1607 001664' 000442      BR 6s ;
1608 001666' 016701 006540      3s: MOV TMBIT2,R1 ;ELSE-IF EOT FLAG BIT=1
1609 001672' 036761 006536 010534'      BIT TUBIT,EOTBL(R1) ;ELSE-IF EOT FLAG BIT=1
1610 001700' 001034      BNE 6s ;THEN-CONTINUE
1611 001702' 004767 000666      CALL M2.3 ;ELSE-CALL MODULE 2.3 (WRITE CONTROL)
1612 001706' 016701 006520      MOV TMBIT2,R1
1613 001712' 036761 006516 010554'      BIT TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1

```

```

1614 001720' 001402          BEQ      4S          ;ELSE-CONTINUE
1615 001722' 004767 001620    CALL     M2.5        ;THEN-CALL MODULE 2.5 (READ REVERSE CONTROL)
1616 001726' 016701 006500    MOV      TMBIT2,R1
1617 001732' 036761 006476 010554' BIT      TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1
1618 001740' 001404          BEQ      5S          ;ELSE-CONTINUE
1619 001742' 004767 002520    CALL     M2.8        ;THEN-CALL MODULE 2.8 (COMPARE DATA)
1620 001746' 004767 001224    CALL     M2.4        ;CALL MODULE 2.4 (READ FORWARD CONTROL)
1621 001752' 016701 006454    MOV      TMBIT2,R1
1622 001756' 036761 006452 010554' BIT      TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1
1623 001764' 001402          BEQ      6S          ;ELSE-CONTINUE
1624 001766' 004767 002474    CALL     M2.8        ;THEN-CALL MODULE 2.8 (COMPARE DATA)
1625 001772' 005267 006446    INC      TU78        ;INCREMENT THE TU78#
1626 001776' 022767 000004 006440 CMP      #4,TU78    ;DO UNTIL THE TU78#=4
1627 002004' 001307          BNE      2S
1628
1629 002006' 005267 006430    INC      TM78        ;INCREMENT THE TM78#
1630 002012' 022767 000010 006422 CMP      #10,TM78   ;DO UNTIL THE TM78#=10
1631 002020' 001277          BNE      1S
1632
1633 002022' 004767 003114    CALL     M2.10       ;CALL MODULE 2.10 (UNIT CONTROL)
1634 002026' 005767 006406    TST      UCNT        ;IF UNIT COUNT=0
1635 002032' 001011          BNE      7S          ;ELSE-CONTINUE
1636 002034' 012767 000011 006416 MOV      #11,ERRINX ;PRINT ERROR 11
1637 002042' 004767 004566    CALL     M4.1
1638 002046' 005077 007022    CLR      @CS1        ;THEN-CLEAR RH11/70 REGISTER CS1
1639 002052' 104410 000000'    ENDS,BEGIN
1640
1641 002056' 032767 000002 006410 7S: BIT      #BIT1,FLAGS ;IF BOT PASS FLAG=1
1642 002064' 001432          BEQ      1S          ;ELSE-CONTINUE
1643 002066' 004767 003050    CALL     M2.10       ;CALL MODULE 2.10 (UNIT CONTROL)
1644 002072' 004767 000244    CALL     M2.2        ;CALL MODULE 2.2 (BOT CONTROL)
1645 002076' 005767 006336    TST      UCNT        ;DO UNTIL THE UNIT
1646 002102' 001404          BEQ      9S          ;COUNT=0 OR THE UNIT
1647 002104' 026767 006344 006326 CMP      BOTCNT,UCNT ;COUNT=THE BOT COUNT
1648 002112' 001365          BNE      8S
1649 002114' 005001          CLR      R1
1650 002116' 005067 006324    CLR      EOTCNT        ;CLEAR THE EOT COUNT
1651 002122' 005061 010534'    CLR      EOTBL(R1) ;CLEAR THE EOT TABLE
1652 002126' 062701 000002    ADD      #2,R1
1653 002132' 022701 000020    CMP      #20,R1
1654 002136' 001371          BNE      10S
1655 002140' 012767 000001 006334 MOV      #1,FILE    ;SET FILE NUMBER=1
1656 002146' 005067 006332    CLR      RECORD        ;CLEAR RECORD NUMBER
1657
1658 002152' 042767 000002 006314 11S: BIC      #BIT1,FLAGS ;CLEAR THE BOT PASS FLAG
1659 002160' 026767 006254 006260 CMP      UCNT,EOTCNT ;IF UNIT COUNT=EOT COUNT
1660 002166' 001007          BNE      12S
1661 002170' 052767 000002 006276 BIC      #BIT1,FLAGS ;SET THE BOT PASS FLAG
1662 002176' 004767 002316    CALL     M2.9        ;THEN-CALL MODULE 2.9 (STATISTICS)
1663 002202' 004767 000064    CALL     M2.1        ;CALL MODULE 2.1 (DENSITY CONTROL)
1664
1665 002206' 022767 000764 006270 12S: CMP      #500,,RECORD ;IF RECORD NUMBER=500.
1666 002214' 001404          BEQ      13S
1667 002216' 032767 000002 006250 BIT      #BIT1,FLAGS ;FOR BOT PASS FLAG=1
1668 002224' 001404          BEQ      14S
1669 002226' 004767 001660 13S: CALL     M2.6        ;ELSE-CONTINUE
;THEN-CALL MODULE 2.6 (TAPE MARK CONTROL)

```

```

1670 002232' 005267 006244    INC      FILE          ;INCREMENT THE FILE NUMBER
1671 002236' 022767 000764 006240 14S: CMP      #500,,RECORD ;DO UNTIL RECORD NUMBER=500.
1672 002244' 001406          BEQ      15S
1673 002246' 032767 000002 006220 BIT      #BIT1,FLAGS ;FOR BOT PASS FLAG=1
1674 002254' 001002          BNE      15S
1675 002256' 000167 177306    JMP      M2.0
1676 002262' 005067 006216 15S: CLR      RECORD        ;SIGNAL END OF ITERATION.
1677 002266' 104413 000000'    ENDS,BEGIN        ;MONITOR SHALL TEST END OF PASS
(1)
1678      .REM
1679      BEGINROUTINE      MODULE 2.1 (DENSITY CONTROL)
1680      : IF DENSITY FLAG=0
1681      : : THEN-IF BIT4 OF SR1 (TEST GCR SWITCH)=0
1682      : : : THEN-SET THE DENSITY FLAG (GCR)
1683      : : : ELSE-CONTINUE
1684      : : ENDIF
1685      : : ELSE-IF BIT5 OF SR1 (TEST PE SWITCH)=0
1686      : : : THEN-CLEAR THE DENSITY FLAG (PE)
1687      : : : ELSE-CONTINUE
1688      : : ENDIF
1689      : ENDIF
1690      ENDRROUTINE
1691
1692
1693
1694 002272' 032767 000001 006174 M2.1: BIT      #BIT0,FLAGS ;IF DENSITY FLAG=0
1695 002300' 001010          BNE      1S          ;ELSE-GO TO ELSE
1696 002302' 032767 000020 175506 BIT      #BIT4,SR1 ;THEN-IF BIT4 OF SR1=0
1697 002310' 001013          BNE      2S          ;ELSE-CONTINUE
1698 002312' 052767 000001 006154 BIC      #BIT0,FLAGS ;THEN-SET THE DENSITY FLAG
1699 002320' 000407          BR      2S
1700 002322' 032767 000040 175466 1S: BIT      #BIT5,SR1 ;IF BIT5 OF SR1=0
1701 002330' 001003          BNE      2S          ;ELSE-CONTINUE
1702 002332' 042767 000001 006134 BIC      #BIT0,FLAGS ;THEN-CLEAR THE DENSITY FLAG
1703 002340' 000207          RTS      PC          ;RETURN

```

```

1705 .REM -
1706 BEGINROUTINE MODULE 2.2 (BOT CONTROL)
1707 : CLEAR THE BOT COUNT
1708 : CLEAR THE TM78#
1709 : BGND0
1710 : : CLEAR THE TU78#
1711 : : RGND0
1712 : : IF BIT SET (SCRATCH UNIT TABLE(TM78#, TU78#))
1713 : : : THEN-CALL MODULE 3.6 (SENSE)
1714 : : : SELECT TERMINATION INTERRUPT CODE
1715 : : : : DONE-IF BIT10 OF SENSE=1 (TU AT BOT)
1716 : : : : THEN-INCREMENT BOT COUNT
1717 : : : : ELSE-CONTINUE
1718 : : : : ENDF
1719 : : : : IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
1720 : : : : THEN-PRINT ERROR 10
1721 : : : : ELSE-CONTINUE
1722 : : : : ENDF
1723 : : : : TM FAULT B-PRINT ERROR 7
1724 : : : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
1725 : : : : ALL OTHERS-PRINT ERROR 2
1726 : : : : DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
1727 : : : : ENDSELECT
1728 : : : : ELSE-CONTINUE
1729 : : : ENDF
1730 : : INCREMENT THE TU78#
1731 : : DO UNTIL THE TU78#=#4
1732 : : ENDDO
1733 : : INCREMENT THE TM78#
1734 : : DO UNTIL THE TM78#=#10(8)
1735 : : ENDDO
1736 ENDROUTINE
1737
1738 -
1739
1740 002342' 005067 006106 M2.2: CLR BUTCNT ;CLEAR BOT COUNT
1741 002346' 005067 006070 CLR TM78 ;CLEAR THE TM78#
1742 002352' 005067 006066 1s: CLR TU78 ;CLEAR THE TU78#
1743 002356' 004767 177100 2s: CALL TUCONV ;
1744 002362' 004767 177136 CALL TUCONV
1745 002366' 016701 006040 MOV TMBIT2,R1
1746 002372' 036761 006036 010554' BIT TUBIT,SUTBL(R1) ;IF BIT SET (SCRATCH UNIT TABLE(TM78#,TU78#))
1747 002400' 001460 BEQ 6S ;ELSE-CONTINUE
1748 002402' 004767 003624 CALL M3.6 ;THEN-CALL MODULE 3.6 (ISSUE SENSE)
1749 002406' 016701 006020 MOV TMBIT2,R1
1750 002412' 022767 000001 006030 CMP #DONE,TINTCD ;DONE?
1751 002420' 001020 BNE 4S ;NO-CONTINUE
1752 002422' 032767 002000 006026 BIT #BIT10,SENSE ;YES-IF TU AT BOT?
1753 002430' 001402 BEQ 3S ;ELSE-CONTINUE
1754 002432' 005267 006016 INC BUTCNT ;THEN-INCREMENT BOT COUNT
1755 002436' 032777 040000 006430 3s: BIT #BIT14,#CS1 ;IF 'TRE' BIT IN RH11/70 CS1=1
1756 002444' 001436 BEQ 6S ;ELSE-CONTINUE
1757 ;THEN-
1758 002446' 012767 000010 006004 MOV #10,ERRINX
1759 002454' 004767 004154 CALL M4.1
1760 002460' 000430 BR 6S

```

```

1761 002462' 022767 000032 005760 4s: CMP #TMFB,TINTCD ;TM FAULT B?
1762 002470' 001012 BNE 5S ;NO-CONTINUE
1763 002472' 012767 000007 005760 MOV #7,ERRINX
1764 002500' 004767 004130 CALL M4.1
1765 002504' 016701 005722 MOV TMBIT2,R1 ;DROP TM78
1766 002510' 005061 010554' CLR SUTBL(R1)
1767 002514' 000412 BR 6S
1768 002516' 012767 000002 005734 5s: MOV #2,ERRINX
1769 002524' 004767 004104 CALL M4.1
1770 002530' 016701 005676 MOV TMBIT2,R1 ;DROP TU78
1771 002534' 046761 005674 010554' BIC TUBIT,SUTBL(R1)
1772 002542' 005267 005676 6s: INC TU78 ;INCREMENT THE TU78#
1773 002546' 022767 000004 005670 CMP #4,TU78 ;DO UNTIL THE TU78#=#4
1774 002554' 001300 BNE 2S
1775
1776 002556' 005267 005660 INC TM78 ;INCREMENT THE TM78#
1777 002562' 022767 000010 005652 CMP #10,TM78 ;DO UNTIL THE TM78#=#10
1778 002570' 001270 HNE 1S
1779 002572' 000207 RTS PC ;RETURN

```

```

1781 .REM
1782 BEGINROUTINE MODULE 2.3 (WRITE CONTROL)
1783 : CLEAR RETRY IN PROGRESS FLAG
1784 : CLEAR THE EXIT FLAG
1785 : RGND0
1786 : : CALL MODULE 3.1 (WRITE)
1787 : : SFELECT TERMINATION INTERRUPT CODE
1788 : : : DONE-SET THE EXIT FLAG
1789 : : : IF RETRY IN PROGRESS FLAG=1
1790 : : : : THEN-UPDATE RECOVERABLE WRITE ERROR STATISTICS
1791 : : : : PRINT ERROR 3
1792 : : : : ELSE-CONTINUE
1793 : : : : ENDF
1794 : : : IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
1795 : : : : THEN-PRINT ERROR 10
1796 : : : : ELSE-CONTINUE
1797 : : : : ENDF
1798 : : : EOT -SET THE EOT FLAG (EOT TABLE (TM78#,TU78#))
1799 : : : SET THE EXIT FLAG
1800 : : : IF RETRY IN PROGRESS FLAG NOT=0
1801 : : : : THEN-UPDATE RECOVERABLE WRITE ERROR STATISTICS
1802 : : : : PRINT ERROR 3
1803 : : : : ELSE-CONTINUE
1804 : : : : ENDF
1805 : : : BAD TAPE-SET THE EXIT FLAG
1806 : : : : PRINT ERROR 4
1807 : : : DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
1808 : : : RETRY-SET RETRY IN PROGRESS FLAG
1809 : : : TM FAULT B-PRINT ERROR 7
1810 : : : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
1811 : : : : SET THE EXIT FLAG
1812 : : : : ALL OTHER-PRINT ERROR 2
1813 : : : : DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
1814 : : : : SET THE EXIT FLAG
1815 : : : : ENDF
1816 : : : DO UNTIL THE EXIT FLAG NOT=0
1817 : : : : ENDDO
1818 ENDRROUTINE
1819
1820
1821
1822 002574' 042767 000004 005672 M2.3: BIC #BIT2,FLAGS ;CLEAR RETRY IN PROGRESS FLAG
1823 002602' 042767 000020 005664 BIC #BIT4,FLAGS ;CLEAR EXIT FLAG
1824 002610' 004767 002444 18: CALL M4.1 ;CALL MODULE 3.1 (WRITE)
1825 002614' 016701 005612 MOV TMBIT2,R1
1826 002620' 022767 000001 005622 CMP #DONE,TINTCD ;DONE?
1827 002626' 001034 BNE 48 ;NO-CONTINUE
1828 002630' 052767 000020 005636 28: BIS #BIT4,FLAGS ;YES-SET THE EXIT FLAG
1829 002636' 032767 000004 005630 BIT #BIT2,FLAGS ;IF RETRY IN PROGRESS FLAG=1
1830 002644' 001413 BEQ 38 ;ELSE-CONTINUE
1831 002646' 016701 005560 MOV TMBIT2,R1
1832 002652' 066701 005560 ADD TUBIT2,R1 ;ADD IN THE TU DISPLACEMENT
1833 002656' 005261 010774' INC RWRIT(R1) ;THEN-UPDATE RECOVERABLE WRITE ERROR STATS.
1834 002662' 012767 000003 005570 MOV #3,ERRINX
1835 002670' 004767 003740 CALL M4.1
1836

```

```

1837 002674' 032777 040000 006172 38: HIT #BIT14,ACS1 ;IF 'TRE' BIT IN RH11/70 CS1=1
1838 002702' 001530 BEQ 98 ;ELSE-CONTINUE
1839 002704' 012767 000010 005546 MOV #10,ERRINX
1840 002712' 004767 003716 CALL M4.1
1841 002716' 000522 BR 98
1842
1843 002720' 022767 000004 005522 48: CMP #EOT,TINTCD ;EOT?
1844 002726' 001026 BNE 58 ;NO-CONTINUE
1845 002730' 016701 005476 MOV TMBIT2,R1 ;SET THE EOT FLAG
1846 002734' 056761 005474 010534' BIS TUBIT,EOTBL(R1)
1847 002742' 052767 000020 005524 HIS #BIT4,FLAGS ;YES-SET THE EXIT FLAG
1848 002750' 032767 000004 005516 BIT #BIT2,FLAGS ;IF RETRY IN PROGRESS FLAG=1
1849 002756' 001502 BEQ 98 ;ELSE-CONTINUE
1850 002760' 066701 005452 ADD TUBIT2,R1 ;ADD IN THE TU DISPLACEMENT
1851 002764' 005261 010774' INC RWRIT(R1) ;THEN-UPDATE RECOVERABLE WRITE ERROR STATS.
1852 002770' 012767 000003 005462 MOV #3,ERRINX
1853 002776' 004767 003632 CALL M4.1
1854 003002' 000470 BR 98
1855
1856 003004' 022767 000027 005436 58: CMP #BTAPE,TINTCD ;BAD TAPE?
1857 003012' 001016 BNE 68 ;NO-CONTINUE
1858 003014' 052767 000020 005452 BIS #BIT4,FLAGS ;YES-SET THE EXIT FLAG
1859 003022' 012767 000004 005430 MOV #4,ERRINX
1860 003030' 004767 003600 CALL M4.1
1861 003034' 016701 005372 MOV TMBIT2,R1 ;DROP TU78
1862 003040' 046761 005370 010554' TUBIT,SUTBL(R1)
1863 003046' 000446 BR 98
1864
1865 003050' 022767 000022 005372 68: CMP #RTY,TINTCD ;RETRY?
1866 003056' 001004 BNE 78 ;NO-CONTINUE
1867 003060' 052767 000004 005406 BIS #BIT2,FLAGS ;YES-SET RETRY IN PROGRESS FLAG
1868 003066' 000436 BR 98
1869 003070' 022767 000032 005352 78: CMP #TMFB,TINTCD ;TM FAULT B?
1870 003076' 001015 BNE 88 ;NO-CONTINUE
1871 003100' 012767 000007 005352 MOV #7,ERRINX
1872 003106' 004767 003522 CALL M4.1
1873 003112' 016701 005314 MOV TMBIT2,R1 ;DROP TM78
1874 003116' 005061 010554' CLR SUTBL(R1)
1875 003122' 052767 000020 005344 HIS #BIT4,FLAGS ;SET THE EXIT FLAG
1876 003130' 000415 BR 98
1877
1878 003132' 052767 000020 005334 88: BIS #BIT4,FLAGS ;SET THE EXIT FLAG
1879 003140' 012767 000002 005312 MOV #2,ERRINX
1880 003146' 004767 003462 CALL M4.1
1881 003152' 016701 005254 MOV TMBIT2,R1 ;DROP TU78
1882 003156' 046761 005252 010554' TUBIT,SUTBL(R1)
1883
1884 003164' 032767 000020 005302 98: BIT #BIT4,FLAGS ;DO UNTIL EXIT FLAG=1
1885 003172' 001606 BEQ 18 ;
1886 003174' 000207 RTS ;RETURN

```

```

1888 .REM
1889 BEGINROUTINE MODULE 2.4 (READ FORWARD CONTROL)
1890 : CLEAR THE RETRY IN PROGRESS FLAG
1891 : CLEAR THE EXIT FLAG
1892 : CALL MODULE 3.2 (READ FORWARD)
1893 : BGNDO
1894 : : SELECT TERMINATION INTERRUPT CODE
1895 : : : DONE-SET THE EXIT FLAG
1896 : : : IF RETRY IN PROGRESS FLAG=1
1897 : : : THEN-UPDATE RECOVERABLE READ ERROR STATISTICS
1898 : : : : PRINT ERROR 5
1899 : : : : ELSE-CONTINUE
1900 : : : : ENDOF
1901 : : : IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
1902 : : : : THEN-PRINT ERROR 10
1903 : : : : ELSE-CONTINUE
1904 : : : : ENDOF
1905 : : : RETRY-SET THE RETRY IN PROGRESS FLAG
1906 : : : CALL MODULE 3.2 (READ FORWARD)
1907 : : : READ OPP-SET THE RETRY IN PROGRESS FLAG
1908 : : : CALL MODULE 3.3 (READ REVERSE)
1909 : : : UNREADABLE-SET THE EXIT FLAG
1910 : : : UPDATE UNRECOVERABLE READ ERROR STATISTICS
1911 : : : : PRINT ERROR 6
1912 : : : : TM FAULT B-PRINT ERROR 7
1913 : : : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
1914 : : : : SET THE EXIT FLAG
1915 : : : : ALL OTHER-PRINT ERROR 2
1916 : : : : DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
1917 : : : : SET THE EXIT FLAG
1918 : : : : ENDSELECT
1919 : : : DO UNTIL THE EXIT FLAG NOT=0
1920 : : : ENDDO
1921 ENDRROUTINE
1922
1923
1924
1925 003176' 042767 000004 005270 M2.4: BIC #BIT2,FLAGS ;CLEAR RETRY IN PROGRESS FLAG
1926 003204' 042767 000020 005262 BIC #BIT4,FLAGS ;CLEAR EXIT FLAG
1927 003212' 044767 002310 CALL M3.2 ;CALL MODULE 3.2 (READ FORWARD)
1928 003216' 016701 005210 MOV TMBIT2,R1
1929
1930 003222' 022767 000001 005220 1S: CMP #DONE,TINTCD ;DONE?
1931 003230' 001034 BNE 3S ;NO-CONTINUE
1932 003232' 052767 000020 005234 BIS #BIT4,FLAGS ;YES-SET EXIT FLAG
1933 003240' 032767 000004 005226 BIT #BIT2,FLAGS ;IF RETRY IN PROGRESS=1
1934 003246' 001413 BEQ 2S
1935 003250' 016701 005156 MOV TMBIT2,R1 ;THEN-UPDATE RECOVERABLE READ ERRORS
1936 003254' 066701 005156 ADD TUBIT2,R1 ;ADD IN THE TU DISPLACEMENT
1937 003260' 005261 010674' INC RREAD(R1)
1938 003264' 012767 000005 005166 MOV #5,ERRINX
1939 003272' 044767 003336 CALL M4.1
1940
1941 003276' 032777 040000 005570 2S: BIT #BIT14,CS1 ;IF TRE BIT IN RH11/70 CS1=1
1942 003304' 001513 BEQ 8S ;ELSE-CONTINUE
1943 003306' 012767 000010 005144 MOV #10,ERRINX

```

```

1944 003314' 044767 003314 CALL M4.1
1945 003320' 000505 BR 8S
1946 003322' 022767 000022 005120 3S: CMP #RTY,TINTCD ;RETRY
1947 003330' 001006 BNE 4S ;NO-CONTINUE
1948 003332' 052767 000004 005134 BIS #BIT2,FLAGS ;YES-SET RETRY IN PROGRESS FLAG
1949 003340' 044767 002162 CALL M3.2 ;CALL MODULE 3.2 (READ FORWARD)
1950 003344' 000473 BR 8S
1951
1952 003346' 022767 000023 005074 4S: CMP #RDOPP,TINTCD ;READ OPPOSITE
1953 003354' 001006 BNE 5S ;NO-CONTINUE
1954 003356' 052767 000004 005110 BIS #BIT2,FLAGS ;YES-SET RETRY IN PROGRESS FLAG
1955 003364' 044767 002414 CALL M3.3 ;CALL MODULE 3.3 (READ REVERSE)
1956 003370' 000461 BR 8S
1957
1958 003372' 022767 000024 005050 5S: CMP #UNREAD,TINTCD ;UNREADABLE?
1959 003400' 001017 BNE 6S ;NO-CONTINUE
1960 003402' 052767 000020 005064 BIS #BIT4,FLAGS ;YES-SET EXIT FLAG
1961 003410' 016701 005016 MOV TMBIT2,R1 ;UPDATE UNRECOVERABLE READ ERRORS
1962 003414' 066701 005016 ADD TUBIT2,R1 ;ADD IN THE TU DISPLACEMENT
1963 003420' 005261 010574' INC UNREAD(R1)
1964 003424' 012767 000006 005026 MOV #6,ERRINX ;PRINT ERROR 6
1965 003432' 044767 003176 CALL M4.1
1966 003436' 000436 BR 8S
1967
1968 003440' 022767 000032 005002 6S: CMP #TMFB,TINTCD ;TM FAULT B?
1969 003446' 001015 BNE 7S ;NO-CONTINUE
1970 003450' 052767 000020 005016 BIS #BIT4,FLAGS ;YES-SET THE EXIT FLAG
1971 003456' 012767 000007 004774 MOV #7,ERRINX
1972 003464' 044767 003144 CALL M4.1
1973 003470' 016701 004736 MOV TMBIT2,R1 ;DROP TM78
1974 003474' 005061 010554' CLR SUTHL(R1)
1975 003500' 000415 BR 8S
1976
1977 003502' 052767 000020 004764 7S: BIS #BIT4,FLAGS ;SET THE EXIT FLAG
1978 003510' 012767 000002 004742 MOV #2,ERRINX
1979 003516' 044767 003112 CALL M4.1
1980 003522' 016701 004704 MOV TMBIT2,R1 ;DROP TU78
1981 003526' 046761 004702 010554' BIC TUBIT,SUTBL(R1)
1982
1983 003534' 032767 000020 004732 8S: BIT #BIT4,FLAGS ;DO UNTIL EXIT FLAG=1
1984 003542' 001627 BEQ 1S
1985 003544' 000207 RTS PC ;RETURN

```

```

1987 .REM
1988 BEGINROUTINE MODULE 2.5 (READ REVERSE CONTROL)
1989 : CLEAR THE RETRY IN PROGRESS FLAG
1990 : CLEAR THE EXIT FLAG
1991 : CALL MODULE 3.3 (READ REVERSE)
1992 : BGNDDO
1993 : : SELECT TERMINATION INTERRUPT CODE
1994 : : : DONE-SET THE EXIT FLAG
1995 : : : IF RETRY IN PROGRESS FLAG NOT=0
1996 : : : : THEN-UPDATE RECOVERABLE READ ERROR STATISTICS
1997 : : : : PRINT ERROR 5
1998 : : : : ELSE-CONTINUE
1999 : : : : ENDIF
2000 : : : IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
2001 : : : : THEN-PRINT ERROR 10
2002 : : : : ELSE-CONTINUE
2003 : : : : ENDIF
2004 : : : RETRY-SET THE RETRY IN PROGRESS FLAG
2005 : : : CALL MODULE 3.3 (READ REVERSE)
2006 : : : READ OPP-SET THE RETRY IN PROGRESS FLAG
2007 : : : CALL MODULE 3.2 (READ FORWARD)
2008 : : : UNREADABLE-SET THE EXIT FLAG
2009 : : : UPDATE UNRECOVERABLE READ ERROR STATISTICS
2010 : : : : PRINT ERROR 6
2011 : : : : TM FAULT B-PRINT ERROR 7
2012 : : : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
2013 : : : : SET THE EXIT FLAG
2014 : : : : ALL OTHER-PRINT ERROR 2
2015 : : : : DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
2016 : : : : SET THE EXIT FLAG
2017 : : : ENDSELECT
2018 : : : DO UNTIL THE EXIT FLAG NOT=0
2019 : : : ENDDO
2020 ENDROUTINE
2021
2022
2023
2024 003546' 042767 000004 004720 M2.5: BIC #BIT2,FLAGS :CLEAR RETRY IN PROGRESS FLAG
2025 003554' 042767 000020 004712 BIC #BIT4,FLAGS :CLEAR EXIT FLAG
2026 003562' 004767 002216 CALL M3.3 :CALL MODULE 3.3 (READ REVERSE)
2027
2028 003566' 022767 000001 004654 1s: CMP #DONE,TINTCD :DONE?
2029 003574' 001034 BNE 3s :NO-CONTINUE
2030 003576' 052767 000020 004670 BIS #BIT4,FLAGS :YES-SET EXIT FLAG
2031 003604' 032767 000004 004662 BIT #BIT2,FLAGS :IF RETRY IN PROGRESS=1
2032 003612' 001413 BEQ 2s
2033 003614' 016701 004612 MOV TMBIT2,R1 :THEN-UPDATE RECOVERABLE READ ERRORS
2034 003620' 066701 004612 ADD TUBIT2,R1 :ADD IN THE TU DISPLACEMENT
2035 003624' 005261 010674' INC RREAD(R1)
2036 003630' 012767 000005 004622 MOV #5,ERRINX
2037 003636' 004767 002772 CALL M4.1
2038
2039 003642' 032777 040000 005224 2s: BIT #BIT14,CS1 :IF TRE BIT IN RH11/70 CS1=1
2040 003650' 001513 BEQ 8s :ELSE-CONTINUE
2041 003652' 012767 000010 004600 MOV #10,ERRINX
2042 003660' 004767 002750 CALL M4.1

```

```

2043 003664' 000505 BR 8s
2044 003666' 022767 000022 004554 3s: CMP #RTRY,TINTCD :RETRY
2045 003674' 001006 BNE 4s :NO-CONTINUE
2046 003676' 052767 000004 004570 BIS #BIT2,FLAGS :YES-SET RETRY IN PROGRESS FLAG
2047 003704' 004767 002074 CALL M3.3 :CALL MODULE 3.3 (READ REVERSE)
2048 003710' 000473 BR 8s
2049
2050 003712' 022767 000023 004530 4s: CMP #RDOPP,TINTCD :READ OPPOSITE
2051 003720' 001006 BNE 5s :NO-CONTINUE
2052 003722' 052767 000004 004544 BIS #BIT2,FLAGS :YES-SET RETRY IN PROGRESS FLAG
2053 003730' 004767 001572 CALL M3.2 :CALL MODULE 3.2 (READ FORWARD)
2054 003734' 000461 BR 8s
2055
2056 003736' 022767 000024 004504 5s: CMP #UNREAD,TINTCD :UNREADABLE?
2057 003744' 001017 BNE 6s :NO-CONTINUE
2058 003746' 052767 000020 004520 BIS #BIT4,FLAGS :YES-SET EXIT FLAG
2059 003754' 016701 004452 MOV TMBIT2,R1 :UPDATE UNRECOVERABLE READ ERRORS
2060 003760' 066701 004452 ADD TUBIT2,R1 :ADD IN THE TU DISPLACEMENT
2061 003764' 005261 010574' INC UNREAD(R1)
2062 003770' 012767 000006 004462 MOV #6,ERRINX :PRINT ERROR 6
2063 003776' 004767 002632 CALL M4.1
2064 004002' 000436 BR 8s
2065
2066 004004' 022767 000032 004436 6s: CMP #TMFB,TINTCD :TM FAULT B?
2067 004012' 001015 BNE 7s :NO-CONTINUE
2068 004014' 052767 000020 004452 BIS #BIT4,FLAGS :YES-SET THE EXIT FLAG
2069 004022' 012767 000007 004430 MOV #7,ERRINX
2070 004030' 004767 002600 CALL M4.1
2071 004034' 016701 004372 MOV TMBIT2,R1 :DROP TM78
2072 004040' 005061 010554' CLR SUTBL(R1)
2073 004044' 000415 BR 8s
2074
2075 004046' 052767 000020 004420 7s: BIS #BIT4,FLAGS :SET THE EXIT FLAG
2076 004054' 012767 000002 004376 MOV #2,ERRINX
2077 004062' 004767 002546 CALL M4.1
2078 004066' 016701 004340 MOV TMBIT2,R1 :DROP TU78
2079 004072' 046761 004336 010554' BIC TUBIT,SUTBL(R1)
2080
2081 004100' 032767 000020 004366 8s: BIT #BIT4,FLAGS :DO UNTIL EXIT FLAG=1
2082 004106' 001627 BEQ 1s
2083 004110' 000207 RTS PC :RETURN

```

```

2085 .REM =
2086 BEGINROUTINE MODULE 2.6 (TAPE MARK CONTROL)
2087 : CLEAR THE TM78#
2088 : HGND0
2089 : : CLEAR THE TU78#
2090 : : HGND0
2091 : : IF BIT (SCRATCH UNIT TABLE(TM78#, TU78#)) = 1, AND
2092 : : : BIT (EOT TABLE(TM78#, TU78#)) = 0
2093 : : : THEN-CALL MODULE 3.4 (TAPE MARK)
2094 : : : SELECT TERMINATION INTERRUPT CODE
2095 : : : : DONE-IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
2096 : : : : : THEN-PRINT ERROR 10
2097 : : : : : ELSE-CONTINUE
2098 : : : : : ENDIF
2099 : : : : TM FAULT B-PRINT ERROR 7
2100 : : : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
2101 : : : : ALL OTHERS-PRINT ERROR 2
2102 : : : : : DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
2103 : : : : ENDSLECT
2104 : : : : ELSE-CONTINUE
2105 : : : : ENDF
2106 : : : INCREMENT THE TU78#
2107 : : : DO UNTIL THE TU78#=4
2108 : : ENDDO
2109 : : INCREMENT THE TM78#
2110 : : DO UNTIL THE TM78#=10(8)
2111 : ENDDO
2112 ENDRROUTINE
2113
2114
2115
2116 004112' 005067 004324 M2.6: CLR TM78 ;CLEAR THE TM78#
2117 004116' 005067 004322 1s: CLR TU78 ;CLEAR THE TU78#
2118 004122' 004767 175334 2s: CALL TMCONV ;
2119 004126' 004767 175372 CALL TUCONV ;
2120 004132' 016701 004274 MOV TMBIT2,R1
2121 004136' 036761 004272 010554' BIT TUBIT,SUTBL(R1) ;IF HIT SET (SCRATCH UNIT TABLE(TM78#,TU78#))
2122 004144' 001456 BEQ 5s ;ELSE-CONTINUE
2123 004146' 036761 004262 010534' BIT TUBIT,EOTBL(R1) ;IF HIT (EOT TABLE(TM78#, TU78#)) = 0
2124 004154' 001052 BNE 5s ;ELSE - CONTINUE
2125 004156' 004767 001664 CALL M3.4 ;CALL MODULE 3.4 (TAPE MARK CONTROL)
2126 004162' 016701 004244 MOV TMBIT2,R1
2127 004166' 022767 000001 004254 CMP #DONE,TINTCD ;DONE?
2128 004174' 001012 BNE 3s ;NO-CONTINUE
2129 004176' 032777 040000 004670 BIT #BIT14,ACS1 ;YES-IF TRE BIT IN RH11/70 CS1=1
2130 004204' 001436 BEQ 5s ;ELSE-CONTINUE
2131 004206' 012767 000010 004244 MOV #10,ERRINX
2132 004214' 004767 002414 CALL M4.1
2133 004220' 000430 BR 5s
2134
2135 004222' 022767 000032 004220 3s: CMP #TMFB,TINTCD ;TM FAULT B?
2136 004230' 001012 BNE 4s ;NO-CONTINUE
2137 004232' 012767 000007 004220 MOV #7,ERRINX ;YES - PRINT ERROR 7
2138 004240' 004767 002370 CALL M4.1
2139 004244' 016701 004162 MOV TMBIT2,R1 ;DROP TM78
2140 004250' 005061 010554' CLR SUTBL(R1)

```

```

2141 004254' 000412 BR 5s
2142
2143 004256' 012767 000002 004174 4s: MOV #2,ERRINX ;PRINT ERROR 2
2144 004264' 004767 002344 CALL M4.1
2145 004270' 016701 004136 MOV TMBIT2,R1 ;DROP TU78
2146 004274' 046761 004134 010554' BIC TUBIT,SUTBL(R1)
2147
2148 004302' 005267 004136 5s: INC TU78 ;INCREMENT THE TU78#
2149 004306' 022767 000004 004130 CMP #4,TU78 ;DO UNTIL THE TU78#=4
2150 004314' 001302 BNE 2s
2151
2152 004316' 005267 004120 INC TM78 ;INCREMENT THE TM78#
2153 004322' 022767 000010 004112 CMP #10,TM78 ;DO UNTIL THE TM78#=10
2154 004330' 001272 BNE 1s
2155 004332' 000207 RTS PC ;RETURN

```

```

2157 .REM
2158 BEGINROUTINE MODULE 2.7 (REWIND CONTROL)
2159 : CALL MODULE 3.5 (REWIND)
2160 : SELECT TERMINATION INTERRUPT
2161 : : DONE-IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
2162 : : : THEN-PRINT ERROR 10
2163 : : : ELSE-CONTINUE
2164 : : ENDIF
2165 : : REWINDING-CONTINUE
2166 : : IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
2167 : : : THEN-PRINT ERROR 8
2168 : : : ELSE-CONTINUE
2169 : : ENDIF
2170 : : TM FAULT 8-PRINT ERROR 7
2171 : : DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
2172 : : ALL OTHERS-PRINT ERROR 2
2173 : : DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
2174 : ENDSELECT
2175 ENDROUTINE
2176
2177
2178
2179 004334' 004767 001610 M2.7: CALL M3.5 ;CALL MODULE 3.5 (REWIND CONTROL)
2180 004340' 022767 000001 004102 CMP #DONE,TINTCD ;DONE?
2181 004346' 001012 BNE 28 ;NO-CONTINUE
2182 004350' 032777 040000 004516 18: BIT #BIT14,CS1 ;YES-IF IN RH11/70 CS1=1
2183 004356' 001442 BEQ 48 ;ELSE-CONTINUE
2184 : THEN-PRINT ERROR 8
2185 004360' 012767 000010 004072 MOV #10,ERRINX
2186 004366' 004767 002242 CALL M4.1
2187 004372' 000434 BR 48
2188
2189 004374' 022767 000007 004046 28: CMP #REW,TINTCD ;REWINDING
2190 004402' 001762 BEQ 18 ;YES-CHEAT-USE DONE CODE
2191 004404' 022767 000032 004036 CMP #TMFB,TINTCD ;NO-FAULT 8?
2192 004412' 001012 BNE 35 ;NO-CONTINUE
2193 004414' 012767 000007 004036 MOV #7,ERRINX ;YES - PRINT ERROR 7
2194 004422' 004767 002206 CALL M4.1
2195 004426' 016701 004000 MOV TMBIT2,R1 ;DROP TM78
2196 004432' 005061 010554' CLR SUBBL(R1)
2197 004436' 000412 BR 48
2198 004440' 012767 000002 004012 35: MOV #2,ERRINX ;PRINT ERROR 2
2199 004446' 004767 002162 CALL M4.1
2200 004452' 016701 003754 MOV TMBIT2,R1 ;DROP TU78
2201 004456' 046761 003752 010554' BIC TMBIT,SUBBL(R1)
2202 004464' 000207 48: RTS PC ;RETURN

```

```

2204 .REM
2205 BEGINROUTINE MODULE 2.8 (DATA COMPARE)
2206 : IF INHIBIT DATA COMPARE FLAG (BIT0 OF SR1)=1
2207 : : THEN-CONTINUE
2208 : : ELSE-IF READ WAS A READ REVERSE
2209 : : : THEN-COMPARE WRITE BUFFER WITH READ BUFFER (DEC/X11 CKDATA)
2210 : : : ELSE-CONTINUE
2211 : : ENDIF
2212 : ENDIF
2213 ENDROUTINE
2214
2215
2216
2217 004466' 032767 000001 173322 M2.8: BIT #BIT0,SR1 ;IF INHIBIT DATA COMPARE FLAG=1
2218 004474' 001010 BNE 18 ;THEN-CONTINUE
2219 004476' 022767 000077 004002 CMP #RREV,RUFCH ;ELSE-WAS IT A READ REVERSE?
2220 004504' 001404 BEQ 18 ;YES - THEN NO DATA COMPARE
2221 004506' 104412 000000' 000126' CDATAS,BEGIN,RBUFA ; REQUEST FOR MONITOR TO CHECK DATA
2222 (1) 004514' 004516' .+2 ; IF ERROR, CONTINUE
2223 004516' 000207 18: RTS PC ;RETURN
2224
2225 :NOTE: IF ANYONE TRIES TO ADD A DATA COMPARE ON A READ REVERSE
2226 : OPERATION THERE WILL HAVE TO BE SOME WORK DONE WITH THE GETPA
2227 : AND MAP22 MACROS TO MAKE THE ADDRESSES APPEAR CORRECT FOR THE
2228 : READ REVERSE DATA COMPARE. THE REASON BEING THAT THE DATA
2229 : COMPARE MACRO UTILIZES THE "RBUFA" AND "RBUFEA" WORDS IN THE
2230 : HEADER AS A POINTER TO THE READ BUFFER. HOWEVER, ON A READ
2231 : REVERSE OPERATION THESE WORDS POINT TO THE END OF THE BUFFER,
2232 : AND THE DATA COMPARE MACRO ASSUMES THAT "RBUFA" AND "RBUFEA"
2233 : A STARTING BUFFER ADDRESSES.

```

```

2234 .REM -
2235 BEGINROUTINE MODULE 2.9 (STATISTIC PRINTOUT)
2236 : IF BIT3 OF SR1=0
2237 : : THEN-PRINT STATISTIC HEADER LINE
2238 : : CLEAR TM78#
2239 : : BGNDDO
2240 : : : CLEAR THE TU78#
2241 : : : BGNDDO
2242 : : : : IF BIT SET (UNIT TABLE(TM78#, TU78#))
2243 : : : : THEN-LET INDEX=((TM78#(8))+(TU78#(2)))
2244 : : : : GET (RECOVERABLE READ ERROR TABLE (INDEX))
2245 : : : : GET (RECOVERABLE WRITE ERROR TABLE (INDEX))
2246 : : : : GET UNRECOVERABLE READ ERROR TABLE (INDEX))
2247 : : : : PRINT TM78#, TU78#, REC. READ, UN-REC. READ, REC. WRITE
2248 : : : : ELSE-CONTINUE
2249 : : : : ENDDO
2250 : : : INCREMENT THE TU78#
2251 : : : DO UNTIL THE TU78#=4
2252 : : : ENDDO
2253 : : : INCREMENT THE TM78#
2254 : : : DO UNTIL THE TM78#=10(8)
2255 : : : ENDDO
2256 : : ELSE-CONTINUE
2257 : ENDDO
2258 ENDROUTINE
2259 -
2260
2261 004520' 032767 000010 173270 M2.9: BIT #BIT3,SR1 ;IF BIT3 OF SR1=0
2262 004526' 001103 BNE 48 ;ELSE-CONTINUE
2263 004530' 104401 000000' 004740' MSGS,BEGIN,STATHD ;ASCII MESSAGE CALL
2264 004536' 005067 003700 CLR TM78 ;CLEAR THE TM78#
2265 004542' 005067 003676 1S: CLR TU78 ;CLEAR THE TU78#
2266 004546' 004767 174710 CALL TMCONV
2267 004552' 004767 174746 2S: CALL TUCONV
2268 004556' 016701 003650 MOV TMBIT2,R1
2269 004562' 036761 003646 010514' BIT TUBIT,UTBL(R1) ;IF BIT SET (UNIT TABLE(TM78#,TU78#))
2270 004570' 001446 BEQ 3S ;ELSE-CONTINUE
2271 004572' 016702 003634 MOV TMBIT2,R2 ;THEN-LET INDEX=((TM78#(8))+(TU78#(2)))
2272 004576' 006302 ASL R2 ;TIMES 4
2273 004600' 006302 ASL R2 ;TIMES 8
2274 004602' 066702 003630 ADD TUBIT2,R2 ;ADD TU78# X 2
2275 004606' 016267 010674' 003664 MOV RREAD(R2),TEMP ;GET THE RECOVERABLE READS
2276 ;*****
2277 (1) ;CONVERT TEMP TO ASCII AND
2278 (1) ;STORE AT STRR
2279 004614' 104421 000000' 010500' BTODS,BEGIN,TEMP,STRR
2280 (1)
2281 004622' 005072'
2282 (1)
2283 004624' 016267 010774' 003646 ;*****
2284 (1) MOV RWRIT(R2),TEMP ;GET THE RECOVERABLE WRITES
2285 (1) ;*****
2286 (1) ;CONVERT TEMP TO ASCII AND
2287 (1) ;STORE AT STRW
2288 004632' 104421 000000' 010500' BTODS,BEGIN,TEMP,STRW
2289 004640' 005131'
2290 (1)
2291 004642' 016267 010574' 003630 ;*****
2292 (1) MOV URREAD(R2),TEMP ;GET THE UNRECOVERABLE READS
2293 (1)
2294 (1)
2295 (1)
2296 (1)
2297 (1)
2298 (1)
2299 (1)
2300 (1)
2301 (1)
2302 (1)
2303 (1)
2304 (1)
2305 (1)

```

```

2280 ;*****
2281 (1) ;CONVERT TEMP TO ASCII AND
2282 (1) ;STORE AT STURR
2283 004650' 104421 000000' 010500' BTODS,BEGIN,TEMP,STURR
2284 004656' 005111'
2285 (1)
2286 (1)
2287 (1)
2288 004660' 104420 000000' 010442' OTODS,BEGIN,TM78,STTM
2289 004666' 005046'
2290 (1)
2291 (1)
2292 (1)
2293 004670' 104420 000000' 010444' OTODS,BEGIN,TU78,STTU
2294 004676' 005057'
2295 (1)
2296 (1)
2297 (1)
2298 004700' 104401 000000' 005044' MSGS,BEGIN,STATLN ;ASCII MESSAGE CALL
2299 004706' 005267 003532 3S: INC TU78 ;INCREMENT THE TU78#
2300 004712' 022767 000004 003524 CMP #4,TU78 ;DO UNTIL THE TU78#=4
2301 004720' 001314 BNE 2S
2302 004722' 005267 003514 INC TM78 ;INCREMENT THE TM78#
2303 004726' 022767 000010 003506 CMP #10,TM78 ;DO UNTIL THE TM78#=10
2304 004734' 001302 BNE 1S
2305 004736' 000207 4S: RTS PC ;RETURN
2306 004740' 020045 020040 046524 STATHD: .ASCII '% TM78# TU78# REC. READS UN-REC. READS '
2307 004746' 034067 020043 020040
2308 004754' 052040 033525 021470
2309 004762' 020040 020040 042522
2310 004770' 027103 051040 040505
2311 004776' 051504 020040 020040
2312 005004' 047125 051055 041505
2313 005012' 020056 042522 042101
2314 005020' 020123 020040 040
2315 005025' 122 041505 020056 .ASCIIZ 'REC. WRITES %'
2316 005032' 051127 052111 051505
2317 005040' 022440 000
2318 005044'
2319 005044' 020040
2320 005046' 030060 030060 STATLN: .ASCII ' '
2321 005054' 020040 040 SITM: .ASCII '000000'
2322 005057' 060 030060 030060 SITU: .ASCII '000000'
2323 005064' 060
2324 005065' 040 020040 020040
2325 005072' 030060 030060 027060 STRR: .ASCII '00000.'
2326 005100' 020040 020040 020040 .ASCII ' '
2327 005106' 020040 040
2328 005111' 060 030060 030060 STURR: .ASCII '00000.'
2329 005116' 056
2330 005117' 040 020040 020040 .ASCII ' '
2331 005124' 020040 020040 040
2332 005131' 060 030060 030060 STRW: .ASCII '00000.'
2333 005136' 056
2334 005137' 045 000 .ASCIIZ '% '
2335 005142' .EVEN

```

```
2307 .REM -
2308 BEGINROUTINE MODULE 2.10 (UNIT CONTROL)
2309 : CLEAR THE UNIT COUNT
2310 : CLEAR THE EOT COUNT
2311 : CLEAR THE TM78#
2312 : BGNDO
2313 : : CLEAR THE TU78#
2314 : : BGNDO
2315 : : : IF BIT SET (SCRATCH UNIT TABLE(TM78#, TU78#))
2316 : : : THEN-INCREMENT THE UNIT COUNT
2317 : : : ELSE-CONTINUE
2318 : : : ENDF
2319 : : : IF BIT SET (EOT TABLE(TM78#, TU78#))
2320 : : : THEN-INCREMENT THE EOT COUNT
2321 : : : ELSE-CONTINUE
2322 : : : ENDF
2323 : : : INCREMENT THE TU78#
2324 : : : DO UNTIL THE TU78#=4
2325 : : : ENDDO
2326 : : : INCREMENT THE TM78#
2327 : : : DO UNTIL THE TM78#=10(8)
2328 : : : ENDDO
2329 ENDROUTINE
2330 -
2331
2332 005142' 005067 003272 M2.10: CLR UCNT ;CLEAR THE UNIT COUNT
2333 005146' 005067 003274 CLR EUTCNT ;CLEAR THE EOT COUNT
2334 005152' 005067 003264 CLR TM78 ;CLEAR THE TM78 #
2335 005156' 005067 003262 1S: CLR TU78 ;CLEAR THE TU78#
2336 005162' 004767 174274 CALL TMCONV
2337 005166' 004767 174332 2S: CALL TUCONV
2338 005172' 016701 003234 MOV TMBIT2,R1 ;
2339 005176' 036761 003232 010554' BIT TUBIT.SUTBL(R1) ;IF SCRATCH UNIT TABLE (TM78#,TU78#)=1
2340 005204' 001402 BEQ 3S ;ELSE-CONTINUE
2341 005206' 005267 003226 INC UCNT ;THEN-INCREMENT THE UNIT COUNT
2342 005212' 036761 003216 010534' 3S: BIT TUBIT.EOTTBL(R1) ;IF EOT TABLE (TM78#,TU78#)=1
2343 005220' 001402 BEQ 4S ;ELSE-CONTINUE
2344 005222' 005267 003220 INC EUTCNT ;THEN-INCREMENT THE EOT COUNT
2345 005226' 005267 003212 4S: INC TU78 ;INCREMENT THE TU78#
2346 005232' 022767 000004 003204 CMP #4,TU78 ;DO UNTIL THE TU78#=4
2347 005240' 001352 BNE 2S
2348 005242' 005267 003174 INC TM78 ;INCREMENT THE TM78#
2349 005246' 022767 000010 003166 CMP #10,TM78 ;DO UNTIL THE TM78#=10
2350 005254' 001340 BNE 1S
2351 005256' 000207 RTS PC ;RETURN
```

```
2353 .REM -
2354 BEGINROUTINE MODULE 3.1 (WRITE)
2355 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2356 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2357 : CLEAR THE RH11/70 CS1 REGISTER
2358 : LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFSZ)
2359 : LOAD THE RH11/70 BC REGISTER WITH (WBUFSZ*2)
2360 : CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2361 : LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2362 : IF 22 BIT ADDRESSING AVAILABLE (DEC/X11 BIT9 OF RES1)
2363 : : THEN-CONVERT THE 18 BIT WHITE BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2364 : : : LOAD THE RH70 BA REGISTER
2365 : : : LOAD THE RH70 EXTENDED BA REGISTER
2366 : : : ELSE-LOAD THE RH11 BA REGISTER
2367 : : ENDF
2368 : SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2369 : IF DENSITY FLAG=1 (GCR)
2370 : : THEN-LOAD RH11/70 CS1 WITH (IE * WRITE GCR FUNCTION CODE * A16 * A17)
2371 : : : ELSE-LOAD RH11/70 CS1 WITH (IE * WRITE PE FUNCTION CODE * A16 * A17)
2372 : : ENDF
2373 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2374 ENDROUTINE
2375 -
2376
2377 005260' 016777 003156 003616 M3.1: MOV TM78.#CS2 ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2378 005266' 016777 040000 003600 MOV BIT14,#CS1 ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2379 005274' 005077 003574 CLR #CS1 ;CLEAR RH11/70 CS1
2380 005300' 016777 172636 003570 MOV WBUFSZ,#WC ;LOAD THE RH11/70 WC REGISTER WITH
2381 005306' 005477 003564 NEG #WC ;THE 2'S COMPLEMENT OF BUFFER SIZE
2382 005312' 016777 172624 003562 MOV WBUFSZ,#BC ;LOAD THE RH11/70 BC REGISTER WITH
2383 005320' 006377 003556 ASL #BC ;THE WORD COUNT * 2
2384 005324' 005077 003556 CLR #XFRINT ;CLEAR THE DATA TRANSFER INTERRUPT CODE
2385 005330' 016777 003110 003552 MOV TU78,#TC ;LOAD RH11/70 REGISTER 14 WITH TU78#, FORMAT REC. CNT. S
2386 005336' 032767 001000 172512 BIT #ADDR22,RES1 ;IF 22 BIT ADDRESSING AVAILABLE
2387 005344' 001430 BEQ 1S ;GO DO ELSE
2388 005346' 016767 172562 003110 MOV WBUFPA,PA18 ;THEN-CONVERT THE 18 BIT
2389 005354' 016767 172556 003104 MOV WBUFEA,XMEM ;PHYSICAL ADDRESS TO 22 BITS
2390 005362' 104416 000000' 010464' MAP22S, BEGIN,PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR
2391 005370' 016777 003074 003502 MOV PA22,#BA ;LOAD THE RH70 BA REGISTER
2392 005376' 016777 003070 003564 MOV EA22,#EBA ;LOAD THE RH70 EBA REGISTER
2393 005404' 016767 003062 003050 MOV EA22,A16A17 ;SAVE ADDRESS BITS A16 AND A17
2394 005412' 042767 177774 003042 BIC #177774,A16A17
2395 005420' 000367 003036 SWAB A16A17
2396 005424' 000416 BR 2S
2397 005426' 016777 172502 003444 1S: MOV WBUFPA,#BA ;ELSE-LOAD THE RH11 BA REGISTER
2398 005434' 016767 172476 003020 MOV WBUFEA,A16A17 ;CALCULATE ADDRESS BITS A16 AND A17
2399 005442' 006367 003014 ASL A16A17 ;SHIFT EXTENDED ADDRESS
2400 005446' 006367 003010 ASL A16A17 ;BITS TO BIT POSITIONS
2401 005452' 006367 003004 ASL A16A17 ;8 AND 9
2402 005456' 006367 003000 ASL A16A17 ;
2403 005462' 052767 000050 003004 2S: BIS #BIT3:BIT5,FLAGS ;SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2404 005470' 032767 000001 002776 BIT #BIT0,FLAGS ;IF DENSITY FLAG=1 (GCR)
2405 005476' 001403 BEQ 3S ;GO DO ELSE
2406 005500' 012701 000163 MOV #WGCR:BIT6,R1 ;THEN-LOAD WRITE GCR FUNCTION CODE AND IE BIT
2407 005504' 000402 BR 4S ;
2408 005506' 012701 000161 3S: MOV #WPE:BIT6,R1 ;ELSE-LOAD WRITE PE FUNCTION CODE AND IE BIT
```

```

2409 005512' 056701 002744 4S: BIS A16A17,R1 ;FOR IN EXTENDED ADDRESS BITS
2410 005516' 010177 003352 MOV R1,PCSI ;ISSUE THE COMMAND
2411 005522' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2412
2413
2414 .REM
2415 BEGINROUTINE MODULE 3.2 (READ FORWARD)
2416 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2417 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2418 : CLEAR THE RH11/70 CS1 REGISTER
2419 : LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFVSZ)
2420 : LOAD THE RH11/70 BC REGISTER WITH (WBUFVSZ*2)
2421 : CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2422 : LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2423 : CONVERT THE READ BUFFER ADDRESS TO 18 BITS (DEC/X11 GETPA)
2424 : IF 22 BIT ADDRESSING AVAILABLE (DEC/X11 BIT9 OF RES1)
2425 : : THEN-CONVERT 18 BIT READ BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2426 : : : LOAD THE RH70 BA REGISTER
2427 : : : LOAD THE RH70 EXTENDED BA REGISTER
2428 : : ELSE-LOAD THE RH11 BA REGISTER
2429 : : :
2430 : SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2431 : LOAD RH11/70 CS1 WITH (1E * READ FORWARD FUNCTION CODE * A17 * A18)
2432 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2433 ENDROUTINE
2434
2435
2436 005526' 012767 000071 002752 M3.2: MOV #RWD,RDFCN
2437 005534' 012767 011174' 172362 MOV #BUFV,RBUFVA
2438
2439 005542' 016777 002674 003334 MRDCOM: MOV TM78,PCSI ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2440 005550' 012777 040000 003316 MOV #BIT14,PCSI ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2441 005556' 005077 003312 CLR #CSI ;CLEAR THE RH11/70 CS1 REGISTER
2442 005562' 016777 172354 003306 MOV WBUFVSZ,WVC ;LOAD THE RH11/70 WC REGISTER WITH THE
2443 005570' 005477 003302 NEG #VC ;2'S COMPLEMENT OF BUFFER SIZE
2444 005574' 016777 172342 003300 MOV WBUFVSZ,#BC ;LOAD THE RH11/70 BC REGISTER WITH BUFFER SIZE
2445 005602' 006377 003274 ASL #BC ;WORD COUNT * 2
2446 005606' 005077 003274 CLR #XFINT ;CLEAR THE RM78 DATA TRANSFER INTERRUPT CODE
2447 005612' 016777 002626 003270 MOV TU78,OTC ;LOAD RH11/70 REG. 14 WITH TU78#,FMT,REC-CNT,SKIP-CNT
2448 005620' 104415 000000' 000124' GETPA,BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
2449 005626' 032767 001000 172222 BIT #ADDR22,RES1 ;IF 22 BIT ADDRESSING AVAILABLE
2450 005634' 001430 BEQ IS ;GO DO ELSE
2451 005636' 016767 172264 002620 MOV RBUFPA,PA18 ;THEN-LOAD UP THE 18 BIT
2452 005644' 016767 172260 002614 MOV RBUFEA,XMEM ;ADDRESS FOR CONVERSION
2453 005652' 104416 000000' 010464' MAP22S, BEGIN,PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR
2454 005660' 016777 002604 003212 MOV PA22,#BA ;LOAD THE RH70 BA REGISTER
2455 005666' 016777 002600 003274 MOV EA22,#EBA ;LOAD THE RH70 EBA REGISTER
2456 005674' 016767 002572 002560 MOV EA22,A16A17 ;CALCULATE A16 + A17 FOR LATER
2457 005702' 042767 177774 002552 BIC #177774,A16A17 ;USE IN CS1
2458 005710' 000367 002546 SWAB A16A17 ;
2459 005714' 000416 BR ZS ;
2460 005716' 016777 172204 003154 1S: MOV RBUFPA,#BA ;ELSE-LOAD THE RH11 BA REGISTER
2461 005724' 016767 172200 002530 MOV RBUFEA,A16A17 ;CALCULATE A16 + A17 FOR LATER
2462 005732' 006367 002524 ASL A16A17 ;USE IN CS1
2463 005736' 006367 002520 ASL A16A17 ;
2464 005742' 006367 002514 ASL A16A17 ;

```

```

2465 005746' 006367 002510 2S: ASL A16A17 ;
2466 005752' 052767 000050 002514 BIS #BIT3,BITS,FLAGS ;SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2467 005760' 016701 002522 MOV RDFCN,R1 ;GET THE READ COMMAND
2468 005764' 052701 000100 BIS #BIT6,R1 ;FOR IN THE 1E BIT
2469 005770' 056701 002466 BIS A16A17,R1 ;FOR IN THE EXTENDED ADDRESS BITS
2470 005774' 010177 003074 MOV R1,PCSI ;ISSUE THE COMMAND
2471 006000' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2472
2473 .REM
2474 BEGINROUTINE MODULE 3.3 (READ REVERSE)
2475 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2476 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2477 : CLEAR THE RH11/70 CS1 REGISTER
2478 : LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFVSZ)
2479 : LOAD THE RH11/70 BC REGISTER WITH (WBUFVSZ*2)
2480 : CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2481 : LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2482 : CONVERT THE READ BUFFER ADDRESS + "RBUFVSZ" TO 18 BITS (DEC/X11 GETPA)
2483 : IF RH70 SWITCH=0 (RH70)
2484 : : THEN-CONVERT 18 BIT READ BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2485 : : : LOAD THE RH70 BA REGISTER
2486 : : : LOAD THE RH70 EXTENDED BA REGISTER
2487 : : ELSE-LOAD THE RH11 BA REGISTER
2488 : : :
2489 : SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2490 : LOAD RH11/70 CS1 WITH (1E * READ REVERSE FUNCTION CODE * A17 * A18)
2491 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2492 ENDROUTINE
2493
2494
2495 006004' 012767 000077 002474 M3.3: MOV #RREV,RDFCN ;
2496 006012' 012767 011174' 172104 MOV #BUFV,RBUFVA ;
2497 006020' 066767 172116 172076 ADD WBUFVSZ,RBUFVA ;
2498 006026' 066767 172110 172070 ADD WBUFVSZ,RBUFVA ;
2499 006034' 162767 000002 172062 SUB #2,RBUFVA ;START ONE WORD BEFORE END
2500 006042' 000167 177474 JMP MRDCOM ;

```

```

2501 .REM -
2502 BEGINROUTINE MODULE 3.4 (TAPE MARK)
2503 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2504 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2505 : CLEAR THE RH11/70 CS1 REGISTER
2506 : CLEAR THE DATA TRANSFER FLAG
2507 : SET THE INTERRUPT EXPECTED FLAG
2508 : IF DENSITY FLAG=1 (GCR)
2509 : : THEN-LOAD RH11/70 REGISTER (40+(TU78**2)) WITH WTM GCR FUNCTION CODE
2510 : : ELSE-LOAD RH11/70 REGISTER (40+(TU78**2)) WITH WTM PE FUNCTION CODE
2511 : ENDIF
2512 : LOAD RH11/70 CS1 REGISTER WITH (IE)
2513 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2514 ENDRROUTINE
2515 -
2516
2517 006046' 016777 002370 003030 M3.4: MOV TM78, @CS2 ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2518 006054' 012777 040000 003012 MOV #BIT14, @CS1 ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2519 006062' 005077 003006 CLR @CS1 ;CLEAR THE RH11/70 CS1 REGISTER
2520 006066' 042767 000010 002400 BIC #BIT3, FLAGS ;CLEAR THE DATA TRANSFER FLAG
2521 006074' 052767 000040 002372 BIS #BIT5, FLAGS ;SET THE INTERRUPT EXPECTED FLAG
2522 006102' 016701 002336 MOV TU78, R1 ;GET THE TU78#
2523 006106' 006301 ASL R1 ;MULTIPLY BY 2
2524 006110' 032767 000001 002356 BIT #BIT0, FLAGS ;IF DENSITY FLAG=1
2525 006116' 001404 REQ 15 ;GO DO ELSE
2526 006120' 012771 000017 011134' MOV #GCR TM, @MOO(R1) ;THEN-LOAD RH11/70 REGISTER (40+(TU78**2)) WITH WTM GCR
2527 006126' 000403 BR 25 ;
2528 006130' 012771 000015 011134' 1s: MOV #PE TM, @MOO(R1) ;ELSE-LOAD RH11/70 REGISTER (40+(TU78**2)) WITH WTM PE F
2529 006136' 012777 000100 002730 2s: MOV #BIT6, @CS1 ;LOAD RH11/70 CS1 WITH THE IE BIT
2530 006144' 104400 000000' EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

2532 .REM -
2533 BEGINROUTINE MODULE 3.5 (REWIND)
2534 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2535 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2536 : CLEAR THE RH11/70 CS1 REGISTER
2537 : CLEAR THE DATA TRANSFER FLAG
2538 : SET THE INTERRUPT EXPECTED FLAG
2539 : LOAD THE RH11/70 REGISTER (40+(TU78**2)) WITH REWIND FUNCTION CODE
2540 : LOAD THE RH11/70 CS1 REGISTER WITH (IE)
2541 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2542 ENDRROUTINE
2543 -
2544
2545 006150' 016777 002266 002726 M3.5: MOV TM78, @CS2 ;LOAD THE RH11/70 CS2 REGISTER WITH TM78#
2546 006156' 012777 040000 002710 MOV #BIT14, @CS1 ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2547 006164' 005077 002704 CLR @CS1 ;CLEAR THE RH11/70 CS1 REGISTER
2548 006170' 042767 000010 002276 BIC #BIT3, FLAGS ;CLEAR THE DATA TRANSFER FLAG
2549 006176' 052767 000040 002270 BIS #BIT5, FLAGS ;SET THE INTERRUPT EXPECTED FLAG
2550 006204' 016701 002234 MOV TU78, R1 ;GET THE TU78#
2551 006210' 006301 ASL R1 ;MULTIPLY BY 2
2552 006212' 012771 000007 011134' MOV #REW, @MOO(R1) ;LOAD RH11/70 REGISTER (40+(TU78**2)) WITH THE REWIND FU
2553 006220' 012777 000100 002646 MOV #BIT6, @CS1 ;LOAD RH11/70 REGISTER CS1 WITH THE IE BIT
2554 006226' 104400 000000' EXITS, BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

2556 .REM -
2557 BEGINROUTINE MODULE 3.6 (SENSE)
2558 : LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2559 : LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2560 : CLEAR THE RH11/70 CS1 REGISTER
2561 : CLEAR THE DATA TRANSFER FLAG
2562 : SET THE INTERRUPT EXPECTED FLAG
2563 : LOAD THE RH11/70 REGISTER (40+(TU78**2)) WITH SENSE FUNCTION CODE
2564 : LOAD RH11/70 CS1 REGISTER WITH (IE)
2565 : WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2566 ENDROUTINE
2567 -
2568
2569 006232' 016777 002204 002644 M3.6: MOV TM78,#CS2 ;LOAD THE RH11/70 CS2 REGISTER WITH TM78#
2570 006240' 012777 040000 002626 MOV #BIT14,#CS1 ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2571 006246' 005077 002622 CLR #CS1 ;CLEAR THE RH11/70 CS1 REGISTER
2572 006252' 042767 000010 002214 BIC #BIT3,FLAGS ;CLEAR THE DATA TRANSFER FLAG
2573 006260' 052767 000040 002206 BIS #BIT5,FLAGS ;SET THE INTERRUPT EXPECTED FLAG
2574 006266' 016701 002152 MOV TU78,R1 ;GET THE TU78#
2575 006272' 006301 ASL R1 ;MULTIPLY BY 2
2576 006274' 012771 000011 011134' MOV #SEN,#M00(R1) ;LOAD THE RH11/70 REGISTER (R0+(TU78**2)) WITH THE SENSE
2577 006302' 012777 000100 002564 MOV #BIT6,#CS1 ;LOAD RH11/70 REGISTER CS1 WITH THE IE BIT
2578 006310' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

2580 .REM -
2581 BEGINROUTINE MODULE 4.0 (INTERRUPT)
2582 -
2583
2584
2585
2586 006314' 042777 000100 002552 M4.0: BIC #BIT6,#CS1 ;CLEAR THE RH11/70 IE BIT
2587 006322' 016767 002546 171550 MOV CS1,CSRA ;COPY CSR ADDRESS FOR ERROR PRINTOUT
2588 006330' 017767 002540 171544 MOV #CS1,ACSR ;COPY CONTENTS OF CS1 TO HEADER
2589 006336' 017767 002542 171540 MOV #CS2,ASTAT ;COPY CS2 CONTENTS
2590 006344' 032767 000040 002122 BIT #BIT5,FLAGS ;INTERRUPT EXPECTED?
2591 006352' 001007 BNE Z6 ;YES-GO PROCESS
2592 006354' 017777 002532 002530 MOV #AS,#AS ;NO-CLEAR PENDING INTERRUPTS
2593 006362' 012777 000100 002504 1S: MOV #BIT6,#CS1 ;ENABLE-RH11/70 INTERRUPTS
2594 006370' 000002 RTI ;RETURN
2595
2596 006372' 032767 000010 002074 2S: BIT #BIT3,FLAGS ;DATA TRANSFER INTERRUPT EXPECTED
2597 006400' 001406 BEQ Z6 ;NO-
2598 006402' 042767 000040 002064 BIC #BIT5,FLAGS ;YES-CLEAR INTERRUPT EXPECTED FLAG
2599 ;-----
2600 (1) 006410' 000004 000000' 006532' PIRQS,BEGIN,M4.0C ; QUEUE UP TO CONTINUE AT M4.0C AND RTI
2601 (1) ;-----
2602 006416' 017767 002470 002066 3S: MOV #AS,ASREG ;COPY THE RH11/70 AS REGISTER
2603 006424' 036767 002000 002060 BIT TMBIT,ASREG ;EXPECTED TM?
2604 006432' 001004 BNE Z6 ;YES-GO CHECK TU
2605 006434' 016777 002052 002450 4S: MOV ASREG,#AS ;NO-CLEAR THE INTERRUPT(S)
2606 006442' 000747 BR 1S
2607
2608 006444' 016777 001772 002432 5S: MOV TM78,#CS2 ;SELECT INTERRUPTING TM78
2609 006452' 017767 002454 002030 MOV #M0INT,M0INR ;SAVE INTERRUPT CODE
2610 006460' 017767 002430 001770 MOV #TUSTAT,SENSE ;SAVE SENSE INFO.
2611 006466' 017767 002440 001756 MOV #M0INT,ITU78 ;GET THE TU78 #
2612 006474' 000367 001752 SWAB ITU78
2613 006500' 042767 177774 001744 BIC #177774,ITU78
2614 006506' 026767 001732 001736 CMP TU78,ITU78 ;EXPECTED TU?
2615 006514' 001347 HNE Z6 ;NO-GO CLEAR INTERRUPT
2616 006516' 042767 000040 001750 BIC #BIT5,FLAGS ;YES-CLEAR INTERRUPT EXPECTED FLAG
2617 ;-----
2618 (1) 006524' 000004 000000' 006532' PIRQS,BEGIN,M4.0C ; QUEUE UP TO CONTINUE AT M4.0C AND RTI
2619 (1) ;-----
2620
2621 006532' 032767 000010 001734 M4.0C: BIT #BIT3,FLAGS ;IF DATA TRANSFER FLAG=1
2622 006540' 001412 BEQ 1S ;GO DO ELSE
2623 006542' 016777 001674 002334 MOV TM78,#CS2 ;THEN-LOAD TM78 # TO RH11/70
2624 006550' 017767 002332 001672 MOV #XFRINT,TINTCD ;COPY DATA TRANSFER INTERRUPT
2625 006556' 042767 177700 001664 BIC #177700,TINTCD ;CODE TO TERMINATION INTERRUPT CODE
2626 006564' 000411 BR Z6
2627
2628 006566' 016767 001716 001654 1S: MOV #M0INR,TINTCD ;GET THE NON DATA
2629 006574' 042767 177700 001646 BIC #177700,TINTCD ;TRANSFER INTERRUPT CODE
2630 006602' 016777 001622 002302 MOV TMBIT,#AS
2631 006610' 042767 000010 001656 2S: BIC #BIT3,FLAGS ;CLEAR DATA TRANSFER FLAG
2632 006616' 042767 000040 001650 BIC #BIT5,FLAGS ;CLEAR INTERRUPT EXPECTED FLAG
2633 006624' 012777 000100 002242 MOV #BIT6,#CS1 ;ENABLE RH11/70 INTERRUPTS
2634 006632' 000207 RTS PC ;RETURN

```

```

2633      .REM
2634      ;*BEGINROUTINE MODULE 4.1 (ERRORS)
2635      -
2636
2637      006634' 032767 000004 171154 M4.1: BIT      #BIT2,SRI      ;IF BIT2 OF SRI=0 (PRINT ERROR MESSAGES)
2638      006642' 001063                HNE      35              ;ELSE - GET OUT
2639      006644' 032767 000001 001622      BIT      #BIT0,FLAGS    ;IF GCR DENSITY?
2640      006652' 001407                BEQ      15              ;ELSE-GO DO PE
2641      006654' 012702 000107                MOV      #107,R2      ;THEN-LOAD GCR CHARACTERS
2642      006660' 012703 000103                MOV      #103,R3
2643      006664' 012704 000122                MOV      #122,R4
2644      006670' 000406                BH      25
2645      006672' 012702 000120      1S:      MOV      #120,R2      ;ELSE-LOAD PE CHARACTERS
2646      006676' 012703 000105                MOV      #105,R3
2647      006702' 012704 000040                MOV      #040,R4
2648      006706' 110267 000354      2S:      MOVH     R2,TYPE1      ;STORE THE FORMAT
2649      006712' 110267 000442                MOVH     R2,TYPE2      ;TYPE IN THE
2650      006716' 110267 000521                MOVH     R2,TYPE3      ;ERROR PRINTOUT
2651      006722' 110267 000601                MOVH     R2,TYPE4      ;CHARACTER STRINGS
2652      006726' 110367 000335                MOVH     R3,TYPE1+1
2653      006732' 110367 000423                MOVH     R3,TYPE2+1
2654      006736' 110367 000502                MOVH     R3,TYPE3+1
2655      006742' 110367 000562                MOVH     R3,TYPE4+1
2656      006746' 110467 000316                MOVH     R4,TYPE1+2
2657      006752' 110467 000404                MOVH     R4,TYPE2+2
2658      006756' 110467 000463                MOVH     R4,TYPE3+2
2659      006762' 110467 000543                MOVH     R4,TYPE4+2
2660      006766' 005067 171114                CLR      ERRRTYP      ;CLEAR ERRTYPE
2661      006772' 016702 001462                MOV      ERRINX,R2      ;GET ERROR INDEX
2662      006776' 006302                ASL      R2            ;MULTIPLY BY 2
2663      007000' 004772 007014'                CALL     @M4.1TB(R2)    ;GO PRINT PROPER ERROR
2664      007004' 104401 000000' 010334'      MSGS,BEGIN,CRLF ;ASCII MESSAGE CALL
2665      007012' 000207                RTS      PC            ;RETURN
2666
2667      007014' 000000      M4.1TB: .WORD     0              ;UNDEFINED
2668      007016' 007044'      .WORD     ERR1            ;NO TM78 UNITS AVAILABLE
2669      007020' 007114'      .WORD     ERR2            ;UNEXPECTED INTERRUPT CODE
2670      007022' 007216'      .WORD     ERR3            ;RECOVERABLE WRITE ERROR
2671      007024' 007302'      .WORD     ERR4            ;UNRECOVERABLE WRITE ERROR
2672      007026' 007374'      .WORD     ERR5            ;RECOVERABLE READ ERROR
2673      007030' 007456'      .WORD     ERR6            ;UNRECOVERABLE READ ERROR
2674      007032' 007542'      .WORD     ERR7            ;TM FAULT INTERRUPT
2675      007034' 007614'      .WORD     ERR10           ;MASSBUS STATUS ERROR
2676      007036' 007760'      .WORD     ERR11           ;NO TU78 UNITS AVAILABLE
2677      007040' 010030'      .WORD     ERR12           ;BOTH DENSITIES DISABLED
2678      007042' 010100'      .WORD     ERR13           ;TM78 NOT READY
2679
2680      007044'      ERR1:
2681      (1) 007044' 104405 000000' 000000      ;*****
2682      (1) 007052' 104401 000000' 007062'      HDRS,BEGIN,NULL ;
2683      007060' 000207      MSGS,BEGIN,ERMS1      ;ASCII MESSAGE CALL
2684      007062' 047516 052040 033515      RTS      PC            ;RETURN
2685      007070' 020070 047125 052111      ERMS1: .ASCIIZ 'NO TM78 UNITS AVAILABLE%'
2686      007076' 020123 053101 044501

```

```

007104' 040514 046102 022505
007112' 000
2684      007114'      .EVEN
2685
2686      007114'      ERR2:
2687      (1)
2688      (1) 007114' 104405 000000' 000000      ;*****
2689      (1) 007122' 104420 000000' 010450'      HDRS,BEGIN,NULL ;
2690      (1) 007130' 007205'      ;*****
2691      (1) 007122' 104420 000000' 010450'      ;CONVERT TINTCD TO ASCII AND
2692      (1) 007130' 007205'      ;STORE AT ERMS2A
2693      (1) 007132' 104401 000000' 007152'      UTOAS,BEGIN,TINTCD,ERMS2A
2694      007140' 004767 001014      ;*****
2695      007144' 004767 001110      MSGS,BEGIN,ERMS2      ;ASCII MESSAGE CALL
2696      007150' 000207      CALL     UNITS      ;GO PRINT TM78#-TU78#
2697      007152' 047125 054105 042520      CALL     DROPTU     ;GO DROP UNIT MESSAGE
2698      007160' 052103 042105 044440      RTS      PC            ;RETURN
2699      007166' 052116 051105 052522      ERMS2: .ASCIIZ 'UNEXPECTED INTERRUPT CODE '
2700      007174' 052120 041440 042117
2701      007202' 020105 040
2702      007205' 060 030060 030060      ERMS2A: .ASCIIZ '000000%'
2703      007212' 022460 000
2704      007216'      .EVEN
2705
2706      007216'      ERR3:
2707      (1) 007216' 104406 000000' 000000      ;*****
2708      (1) 007224' 104401 000000' 007244'      SOFERS,BEGIN,NULL ;
2709      007232' 004767 000722      ;*****
2710      007236' 004767 001074      MSGS,BEGIN,ERMS3      ;ASCII MESSAGE CALL
2711      007242' 000207      CALL     UNITS      ;PRINT TM78#-TU78#
2712      007244' 042522 047503 042526      CALL     RECFIL     ;PRINT RECORD NUMBER FILE NUMBER
2713      007252' 040522 046102 020105      RTS      PC            ;RETURN
2714      007260' 051127 052111 020105      ERMS3: .ASCIIZ 'RECOVERABLE WRITE '
2715      007266' 041507 122
2716      007271' 040 051105 047522      TYPE1: .ASCIIZ 'GCH'
2717      007276' 022522 000      .ASCIIZ 'ERROR%'
2718
2719      007302'      .EVEN
2720
2721      007302'      ERR4:
2722      (1) 007302' 104405 000000' 000000      ;*****
2723      (1) 007310' 104401 000000' 007334'      HDRS,BEGIN,NULL ;
2724      007316' 004767 000636      ;*****
2725      007322' 004767 001010      MSGS,BEGIN,ERMS4      ;ASCII MESSAGE CALL
2726      007326' 004767 000726      CALL     UNITS      ;PRINT THE TM78#-TU78#
2727      007332' 000207      CALL     RECFIL     ;PRINT RECORD NUMBER FILE NUMBER
2728      007334' 047125 042522 047503      CALL     DROPTU     ;PRINT DROP MESSAGE
2729      007342' 042526 040522 046102      RTS      PC            ;RETURN
2730      007350' 020105 051127 052111      ERMS4: .ASCIIZ 'UNRECOVERABLE WRITE '
2731      007356' 020105

```

```

2713 007360' 041507 122 TYPE2: .ASCII 'GCR'
2714 007363' 040 051105 047522 .ASCIZ 'ERROR%'
2715 007370' 022522 000 .EVEN
2716 007374'
2717 (1) ERR5: ;*****
(1) 007374' 104406 000000' 000000 ;SOFERS,BEGIN,NULL ;
(1) ;*****
2718 007402' 104401 000000' 007422' MSGS,BEGIN,ERMS5 ;ASCII MESSAGE CALL
2719 007410' 004767 000544 CALL UNITS ;PRINT THE TM78*-TU78#
2720 007414' 004767 000716 CALL RECFIL ;PRINT RECORD NUMBER FILE NUMBER
2721 007420' 000207 RTS PC ;RETURN
2722 007422' 042522 047503 042526 ERMS5: .ASCII 'RECOVERABLE READ '
2723 007430' 040522 046102 020105
2724 007436' 042522 042101 040 TYPE3: .ASCII 'GCR'
2725 007443' 107 051103 .ASCIZ 'ERROR%'
2726 007446' 042440 051122 051117 .EVEN
2727 007456'
2728 (1) ERR6: ;*****
2729 (1) 007456' 104405 000000' 000000 ;HDRS,BEGIN,NULL ;
2730 (1) ;*****
2731 007464' 104401 000000' 007504' MSGS,BEGIN,ERMS6 ;ASCII MESSAGE CALL
2732 007472' 004767 000462 CALL UNITS ;PRINT THE TM78*-TU78#
2733 007476' 004767 000634 CALL RECFIL ;PRINT RECORD NUMBER FILE NUMBER
2734 007502' 000207 RTS PC ;RETURN
2735 007504' 047125 042522 047503 ERMS6: .ASCII 'UNRECOVERABLE READ '
2736 007512' 042526 040522 046102
2737 007520' 020105 042522 042101
2738 007526' 040
2739 007527' 107 051103 TYPE4: .ASCII 'GCR'
2740 007532' 042440 051122 051117 .ASCIZ 'ERROR%'
2741 007540' 000045 .EVEN
2742 007542'
2743 (1) ERR7: ;*****
2744 (1) 007542' 104405 000000' 000000 ;HDRS,BEGIN,NULL ;
2745 (1) ;*****
2746 007550' 104401 000000' 007570' MSGS,BEGIN,ERMS7 ;ASCII MESSAGE CALL
2747 007556' 004767 000376 CALL UNITS ;PRINT TM78*-TU78#
2748 007562' 004767 000520 CALL DROPTM ;DROP THE TM78
2749 007566' 000207 RTS PC ;RETURN
2750 007570' 046524 043040 052501 ERMS7: .ASCIZ 'TM FAULT INTERRUPT%'
2751 007576' 052114 044440 052116
2752 007604' 051105 052522 052120
2753 007612' 000045 .EVEN
2754 007614'
2755 (1) ERR10: ;*****
2756 (1) 007614' 104406 000000' 000000 ;SOFERS,BEGIN,NULL ;
2757 (1) ;*****

```

```

2746 007622' 017767 001246 000650 MOV @CS1,TEMP ;GET THE CONTENTS OF CS1
2747 (1) ;*****
2748 (1) 007630' 104420 000000' 010500' ;CONVERT TEMP TO ASCII AND
2749 (1) 007636' 007731' ;STORE AT ERM10A
2750 (1) ;*****
2751 007640' 017767 001240 000632 MOV @CS2,TEMP ;GET THE CONTENTS OF CS2
2752 (1) ;*****
2753 (1) 007646' 104420 000000' 010500' ;CONVERT TEMP TO ASCII AND
2754 (1) 007654' 007747' ;STORE AT ERM10B
2755 (1) ;*****
2756 007656' 104401 000000' 007676' MSGS,BEGIN,ERMS10 ;ASCII MESSAGE CALL
2757 007664' 004767 000270 CALL UNITS ;PRINT TM78*-TU78#
2758 007670' 004767 000442 CALL RECFIL ;PRINT RECORD NUMBER FILE NUMBER
2759 007674' 000207 RTS PC ;RETURN
2760 007676' 040515 051523 052502 ERMS10: .ASCII 'MASSBUS STATUS ERROR CS1 = '
2761 007704' 020123 052123 052101
2762 007712' 051525 042440 051122
2763 007720' 051117 041440 030523
2764 007726' 036440 040
2765 007731' 060 030060 030060 ERM10A: .ASCII '000000 CS2 = '
2766 007736' 020060 041440 031123
2767 007744' 036440 040
2768 007747' 060 030060 030060 ERM10B: .ASCIZ '000000%'
2769 007754' 022460 000 .EVEN
2770 007760'
2771 (1) ERR11: ;*****
2772 (1) 007760' 104405 000000' 000000 ;HDRS,BEGIN,NULL ;
2773 (1) ;*****
2774 007766' 104401 000000' 007776' MSGS,BEGIN,ERMS11 ;ASCII MESSAGE CALL
2775 007774' 000207 RTS PC ;RETURN
2776 007776' 047516 052040 033525 ERMS11: .ASCIZ 'NO TU78 UNITS AVAILABLE%'
2777 010004' 020070 047125 052111
2778 010012' 020123 053101 044501
2779 010020' 040514 046102 022505
2780 010026' 000
2781 010030' .EVEN
2782 010030'
2783 (1) ERR12: ;*****
2784 (1) 010030' 104405 000000' 000000 ;HDRS,BEGIN,NULL ;
2785 (1) ;*****
2786 010036' 104401 000000' 010046' MSGS,BEGIN,ERMS12 ;ASCII MESSAGE CALL
2787 010044' 000207 RTS PC ;RETURN
2788 010046' 047502 044124 042040 ERMS12: .ASCIZ 'BOTH DENSITIES DISABLED%'
2789 010054' 047105 044523 044524
2790 010062' 051505 042040 051511
2791 010070' 041101 042514 022504
2792 010076' 000
2793 010100' .EVEN
2794 010100'
2795 (1) ERR13:
2796 (1)
2797 (1)

```

```

(1)
(1) 010100' 104405 000000' 000000 ;*****
(1) HDRS,BEGIN,NULL ;
2771 010106' 104401 000000' 010126' ;*****
2772 010114' 004767 000040 ;MSG,BEGIN,ERMS13 ;ASCII MESSAGE CALL
2773 010120' 004767 000162 CALL UNITS ;PRINT UNITS
2774 010124' 000207 CALL DROPTM ;PRINT DROPPING TM78
2775 010126' 046524 034067 051040 ERMS13: RTS PC ;RETURN
010134' 040505 054504 041040 ;ASCIZ/TM78 READY BIT TIME OUT%/
010142' 052111 052040 046511
010150' 020105 052517 022524
010156' 000
2776 010160' 016767 000256 000312 UNITS: .EVEN
2777 MOV TM78,TEMP ;GET THE TM78#
2778 ;*****
(1) ;CONVERT TEMP TO ASCII AND
(1) ;STORE AT UNITM
(1) 010166' 104420 000000' 010500' UTOAS,BEGIN,TEMP,UNITM
(1) 010174' 010231' ;*****
(1) ;*****
2779 010176' 016767 000242 000274 MOV TU78,TEMP ;GET THE TU78#
2780 ;*****
(1) ;CONVERT TEMP TO ASCII AND
(1) ;STORE AT UNITTU
(1) 010204' 104420 000000' 010500' UTOAS,BEGIN,TEMP,UNITTU
(1) 010212' 010247' ;*****
(1) ;*****
2781 010214' 104401 000000' 010224' MSGS,BEGIN,UNITMS ;ASCII MESSAGE CALL
2782 010222' 000207 RTS PC ;RETURN
2783 ;*****
2784 010224' 046524 034067 040 UNITMS: .ASCII 'TM78 '
2785 010231' 060 030060 030060 UNITTM: .ASCII '000000 TU78 '
010236' 020060 020040 052524
010244' 034067 040
2786 010247' 060 030060 030060 UNITTU: .ASCIZ '000000 %'
010254' 020060 000045
2787 .EVEN
2788 010260' DROPTU:
(1) 010260' 104401 000000' 010270' MSGS,BEGIN,MSDTU ;ASCII MESSAGE CALL
2789 010266' 000207 RTS PC
2790 ;*****
2791 010270' 052524 034067 042040 MSDTU: .ASCIZ 'TU78 DROPPED%'
010276' 047522 050120 042105
010304' 000045
2792 .EVEN
2793
2794 010306' DROPTM:
(1) 010306' 104401 000000' 010316' MSGS,BEGIN,MSDTM ;ASCII MESSAGE CALL
2795 010314' 000207 RTS PC
2796 ;*****
2797 010316' 046524 034067 042040 MSDTM: .ASCIZ 'TM78 DROPPED%'
010324' 047522 050120 042105
010332' 000045
2798 .EVEN
2799
2800 010334' 000045 CHLF: .ASCIZ '%'

```

```

2801 .EVEN
2802
2803 010336' RECFIL:
(1) ;*****
(1) ;CONVERT RECORD TO ASCII AND
(1) ;STORE AT RECMSG
(1) 010336' 104421 000000' 010504' BTODS,BEGIN,RECORD,RECMSG
(1) 010344' 010377' ;*****
(1) ;*****
2804 ;CONVERT FILE TO ASCII AND
(1) ;STORE AT FILMSG
(1) 010346' 104421 000000' 010502' BTODS,BEGIN,FILE,FILMSG
(1) 010354' 010417' ;*****
(1) ;*****
2805 010356' 104401 000000' 010366' MSGS,BEGIN,RFMSG ;ASCII MESSAGE CALL
2806 010364' 000207 RTS PC
2807 ;*****
2808 010366' 042522 047503 042122 RFMSG: .ASCII 'RECORD# '
010374' 020043 040
2809 010377' 060 030060 030060 RECMSG: .ASCII '00000. FILE# '
010404' 020056 020040 044506
010412' 042514 020043 040
2810 010417' 060 030060 030060 FILMSG: .ASCIZ '00000.%'
010424' 022456 000
2811 010430' .EVEN

```

2813  
2814 010430' 000000  
2815 010432' 000000  
2816 010434' 000000  
2817 010436' 000000  
2818 010440' 000000  
2819 010442' 000000  
2820 010444' 000000  
2821 010446' 000000  
2822 010450' 000000  
2823 010452' 000000  
2824 010454' 000000  
2825 010456' 000000  
2826 010460' 000000  
2827 010462' 000000  
2828 010464' 000000  
2829 010466' 000000  
2830 010470' 000000  
2831 010472' 000000  
2832 010474' 000000  
2833 010476' 000000  
2834 010500' 000000  
2835 010502' 000000  
2836 010504' 000000  
2837 010506' 000000  
2838 010510' 000000  
2839 010512' 000000

.SBTTL PROGRAM VARIABLE  
TMBIT: .WORD 0 ;TM78 # CONVERTED TO A BIT POSITION  
TMBIT2: .WORD 0 ;TM78 #2  
TUBIT: .WORD 0 ;TU78 # CONVERTED TO A BIT POSITION  
TUBIT2: .WORD 0 ;TU78 #2  
UCNT: .WORD 0 ;UNIT COUNT (NUMBER OF TU78'S ACTIVE)  
TM78: .WORD 0 ;TM78 #  
TU78: .WORD 0 ;TU78 #  
EOTCNT: .WORD 0 ;EOT COUNT (NUMBER OF TU78'S AT EOT)  
TINTCU: .WORD 0 ;TERMINATION INTERRUPT CODE  
ITU78: .WORD 0 ;INTERRUPTING TU78 #  
RUCNT: .WORD 0 ;RUT COUNT (NUMBER OF TU78'S AT RUT)  
SENSE: .WORD 0 ;SENSE COMMAND INFO  
ERRINX: .WORD 0 ;ERROR INDEX  
A16A17: .WORD 0 ;EXTENDED ADDRESS BITS 16 AND 17  
PA18: .WORD 0 ;LOWER 16 BITS OF 18 BIT MEMORY ADDRESS  
XMEM: .WORD 0 ;EXTENDED ADDRESS BITS OF AN 18 BIT ADDRESS  
PA22: .WORD 0 ;LOWER 16 BITS OF 22 BIT MEMORY ADDRESS  
EA22: .WORD 0 ;EXTENDED ADDRESS BITS OF A 22 BIT ADDRESS  
FLAGS: .WORD 0 ;FLAGS  
TMUNIT: .WORD 0 ;TM78 UNITS  
TEMP: .WORD 0 ;TEMP STORAGE  
FILE: .WORD 0 ;FILE NUMBER  
RECORD: .WORD 0 ;RECORD NUMBER  
RDFCN: .WORD 0 ;READ FUNCTION CODE  
MONINR: .WORD 0 ;COPY OF MOTION INTERRUPT REGISTER  
ASREG: .WORD 0 ;COPY OF ATTENTION SUMMARY REGISTER

2841  
2842  
2843 010514' 000000  
2844 010516' 000000  
2845 010520' 000000  
2846 010522' 000000  
2847 010524' 000000  
2848 010526' 000000  
2849 010530' 000000  
2850 010532' 000000  
2851  
2852  
2853  
2854 010534' 000000  
2855 010536' 000000  
2856 010540' 000000  
2857 010542' 000000  
2858 010544' 000000  
2859 010546' 000000  
2860 010550' 000000  
2861 010552' 000000  
2862  
2863  
2864  
2865 010554' 000000  
2866 010556' 000000  
2867 010560' 000000  
2868 010562' 000000  
2869 010564' 000000  
2870 010566' 000000  
2871 010570' 000000  
2872 010572' 000000

.SBTTL UNIT TABLE  
UTBL: .WORD 0 ;TM78 = 0 TU78 0 - 3  
          .WORD 0 ;TM78 = 1 TU78 0 - 3  
          .WORD 0 ;TM78 = 2 TU78 0 - 3  
          .WORD 0 ;TM78 = 3 TU78 0 - 3  
          .WORD 0 ;TM78 = 4 TU78 0 - 3  
          .WORD 0 ;TM78 = 5 TU78 0 - 3  
          .WORD 0 ;TM78 = 6 TU78 0 - 3  
          .WORD 0 ;TM78 = 7 TU78 0 - 3  
  
.SBTTL EOT TABLE  
EOTBL: .WORD 0 ;TM78 = 0 TU78 0 - 3  
          .WORD 0 ;TM78 = 1 TU78 0 - 3  
          .WORD 0 ;TM78 = 2 TU78 0 - 3  
          .WORD 0 ;TM78 = 3 TU78 0 - 3  
          .WORD 0 ;TM78 = 4 TU78 0 - 3  
          .WORD 0 ;TM78 = 5 TU78 0 - 3  
          .WORD 0 ;TM78 = 6 TU78 0 - 3  
          .WORD 0 ;TM78 = 7 TU78 0 - 3  
  
.SBTTL SCRATCH UNIT TABLE  
SUTBL: .WORD 0 ;TM78 = 0 TU78 0 - 3  
          .WORD 0 ;TM78 = 1 TU78 0 - 3  
          .WORD 0 ;TM78 = 2 TU78 0 - 3  
          .WORD 0 ;TM78 = 3 TU78 0 - 3  
          .WORD 0 ;TM78 = 4 TU78 0 - 3  
          .WORD 0 ;TM78 = 5 TU78 0 - 3  
          .WORD 0 ;TM78 = 6 TU78 0 - 3  
          .WORD 0 ;TM78 = 7 TU78 0 - 3

```

2874
2875
2876 010574' 000000
2877 010576' 000000
2878 010600' 000000
2879 010602' 000000
2880 010604' 000000
2881 010606' 000000
2882 010610' 000000
2883 010612' 000000
2884 010614' 000000
2885 010616' 000000
2886 010620' 000000
2887 010622' 000000
2888 010624' 000000
2889 010626' 000000
2890 010630' 000000
2891 010632' 000000
2892 010634' 000000
2893 010636' 000000
2894 010640' 000000
2895 010642' 000000
2896 010644' 000000
2897 010646' 000000
2898 010650' 000000
2899 010652' 000000
2900 010654' 000000
2901 010656' 000000
2902 010660' 000000
2903 010662' 000000
2904 010664' 000000
2905 010666' 000000
2906 010670' 000000
2907 010672' 000000
2908
2909
2910

```

```

.SBTTL UNRECOVERABLE READ STATISTIC TABLE
UPREAD: .WORD 0          ;TM78 0  TU78 0
         .WORD 0          ;TM78 0  TU78 1
         .WORD 0          ;TM78 0  TU78 2
         .WORD 0          ;TM78 0  TU78 3
         .WORD 0          ;TM78 1  TU78 0
         .WORD 0          ;TM78 1  TU78 1
         .WORD 0          ;TM78 1  TU78 2
         .WORD 0          ;TM78 1  TU78 3
         .WORD 0          ;TM78 2  TU78 0
         .WORD 0          ;TM78 2  TU78 1
         .WORD 0          ;TM78 2  TU78 2
         .WORD 0          ;TM78 2  TU78 3
         .WORD 0          ;TM78 3  TU78 0
         .WORD 0          ;TM78 3  TU78 1
         .WORD 0          ;TM78 3  TU78 2
         .WORD 0          ;TM78 3  TU78 3
         .WORD 0          ;TM78 4  TU78 0
         .WORD 0          ;TM78 4  TU78 1
         .WORD 0          ;TM78 4  TU78 2
         .WORD 0          ;TM78 4  TU78 3
         .WORD 0          ;TM78 5  TU78 0
         .WORD 0          ;TM78 5  TU78 1
         .WORD 0          ;TM78 5  TU78 2
         .WORD 0          ;TM78 5  TU78 3
         .WORD 0          ;TM78 6  TU78 0
         .WORD 0          ;TM78 6  TU78 1
         .WORD 0          ;TM78 6  TU78 2
         .WORD 0          ;TM78 6  TU78 3
         .WORD 0          ;TM78 7  TU78 0
         .WORD 0          ;TM78 7  TU78 1
         .WORD 0          ;TM78 7  TU78 2
         .WORD 0          ;TM78 7  TU78 3

```

```

2911 010674' 000000
2912 010676' 000000
2913 010700' 000000
2914 010702' 000000
2915 010704' 000000
2916 010706' 000000
2917 010710' 000000
2918 010712' 000000
2919 010714' 000000
2920 010716' 000000
2921 010720' 000000
2922 010722' 000000
2923 010724' 000000
2924 010726' 000000
2925 010730' 000000
2926 010732' 000000
2927 010734' 000000
2928 010736' 000000
2929 010740' 000000

```

```

.SBTTL RECOVERABLE READ STATISTIC TABLE
RREAD:  .WORD 0          ;TM78 0  TU78 0
         .WORD 0          ;TM78 0  TU78 1
         .WORD 0          ;TM78 0  TU78 2
         .WORD 0          ;TM78 0  TU78 3
         .WORD 0          ;TM78 1  TU78 0
         .WORD 0          ;TM78 1  TU78 1
         .WORD 0          ;TM78 1  TU78 2
         .WORD 0          ;TM78 1  TU78 3
         .WORD 0          ;TM78 2  TU78 0
         .WORD 0          ;TM78 2  TU78 1
         .WORD 0          ;TM78 2  TU78 2
         .WORD 0          ;TM78 2  TU78 3
         .WORD 0          ;TM78 3  TU78 0
         .WORD 0          ;TM78 3  TU78 1
         .WORD 0          ;TM78 3  TU78 2
         .WORD 0          ;TM78 3  TU78 3
         .WORD 0          ;TM78 4  TU78 0
         .WORD 0          ;TM78 4  TU78 1
         .WORD 0          ;TM78 4  TU78 2
         .WORD 0          ;TM78 4  TU78 3

```

```

2930 010742' 000000
2931 010744' 000000
2932 010746' 000000
2933 010750' 000000
2934 010752' 000000
2935 010754' 000000
2936 010756' 000000
2937 010760' 000000
2938 010762' 000000
2939 010764' 000000
2940 010766' 000000
2941 010770' 000000
2942 010772' 000000
2943
2944
2945

```

```

         .WORD 0          ;TM78 4  TU78 3
         .WORD 0          ;TM78 5  TU78 0
         .WORD 0          ;TM78 5  TU78 1
         .WORD 0          ;TM78 5  TU78 2
         .WORD 0          ;TM78 5  TU78 3
         .WORD 0          ;TM78 6  TU78 0
         .WORD 0          ;TM78 6  TU78 1
         .WORD 0          ;TM78 6  TU78 2
         .WORD 0          ;TM78 6  TU78 3
         .WORD 0          ;TM78 7  TU78 0
         .WORD 0          ;TM78 7  TU78 1
         .WORD 0          ;TM78 7  TU78 2
         .WORD 0          ;TM78 7  TU78 3

```

```

2946 010774' 000000
2947 010776' 000000
2948 011000' 000000
2949 011002' 000000
2950 011004' 000000
2951 011006' 000000
2952 011010' 000000
2953 011012' 000000
2954 011014' 000000
2955 011016' 000000
2956 011020' 000000
2957 011022' 000000
2958 011024' 000000
2959 011026' 000000
2960 011030' 000000
2961 011032' 000000
2962 011034' 000000
2963 011036' 000000
2964 011040' 000000
2965 011042' 000000
2966 011044' 000000
2967 011046' 000000
2968 011050' 000000
2969 011052' 000000
2970 011054' 000000
2971 011056' 000000
2972 011060' 000000
2973 011062' 000000
2974 011064' 000000
2975 011066' 000000
2976 011070' 000000
2977 011072' 000000
2978
2979 011074'
2980 011074' 000000
2981 011076' 000000
2982 011100' 000000
2983 011102' 000000
2984 011104' 000000
2985 011106' 000000

```

```

.SBTTL RECOVERABLE WRITE STATISTIC TABLE
RWRT:  .WORD 0          ;TM78 0  TU78 0
         .WORD 0          ;TM78 0  TU78 1
         .WORD 0          ;TM78 0  TU78 2
         .WORD 0          ;TM78 0  TU78 3
         .WORD 0          ;TM78 1  TU78 0
         .WORD 0          ;TM78 1  TU78 1
         .WORD 0          ;TM78 1  TU78 2
         .WORD 0          ;TM78 1  TU78 3
         .WORD 0          ;TM78 2  TU78 0
         .WORD 0          ;TM78 2  TU78 1
         .WORD 0          ;TM78 2  TU78 2
         .WORD 0          ;TM78 2  TU78 3
         .WORD 0          ;TM78 3  TU78 0
         .WORD 0          ;TM78 3  TU78 1
         .WORD 0          ;TM78 3  TU78 2
         .WORD 0          ;TM78 3  TU78 3
         .WORD 0          ;TM78 4  TU78 0
         .WORD 0          ;TM78 4  TU78 1
         .WORD 0          ;TM78 4  TU78 2
         .WORD 0          ;TM78 4  TU78 3
         .WORD 0          ;TM78 5  TU78 0
         .WORD 0          ;TM78 5  TU78 1
         .WORD 0          ;TM78 5  TU78 2
         .WORD 0          ;TM78 5  TU78 3
         .WORD 0          ;TM78 6  TU78 0
         .WORD 0          ;TM78 6  TU78 1
         .WORD 0          ;TM78 6  TU78 2
         .WORD 0          ;TM78 6  TU78 3
         .WORD 0          ;TM78 7  TU78 0
         .WORD 0          ;TM78 7  TU78 1
         .WORD 0          ;TM78 7  TU78 2
         .WORD 0          ;TM78 7  TU78 3

```

```

RHTBL:
CS1:  .WORD 0          ;MASSBUS REGISTER TABLE
WC:  .WORD 0          ;REGISTER +00
BA:  .WORD 0          ;REGISTER +02
BC:  .WORD 0          ;REGISTER +04
CS2: .WORD 0          ;REGISTER +06
XFRINT: .WORD 0       ;REGISTER +10
         .WORD 0       ;REGISTER +12

```

|      |         |        |         |       |       |                        |
|------|---------|--------|---------|-------|-------|------------------------|
| 2986 | 011110' | 000000 | TC:     | .WORD | 0     | ;REGISTER +14          |
| 2987 | 011112' | 000000 | AS:     | .WORD | 0     | ;REGISTER +16          |
| 2988 | 011114' | 000000 | TUSTAT: | .WORD | 0     | ;REGISTER +20          |
| 2989 | 011116' | 000000 | DB:     | .WORD | 0     | ;REGISTER +22          |
| 2990 | 011120' | 000000 | D11:    | .WORD | 0     | ;REGISTER +24          |
| 2991 | 011122' | 000000 | DT:     | .WORD | 0     | ;REGISTER +26          |
| 2992 | 011124' | 000000 | SN:     | .WORD | 0     | ;REGISTER +30          |
| 2993 | 011126' | 000000 | D12:    | .WORD | 0     | ;REGISTER +32          |
| 2994 | 011130' | 000000 | D13:    | .WORD | 0     | ;REGISTER +34          |
| 2995 | 011132' | 000000 | MDINT:  | .WORD | 0     | ;REGISTER +36          |
| 2996 | 011134' | 000000 | MD0:    | .WORD | 0     | ;REGISTER +40          |
| 2997 | 011136' | 000000 | MD1:    | .WORD | 0     | ;REGISTER +42          |
| 2998 | 011140' | 000000 | MD2:    | .WORD | 0     | ;REGISTER +44          |
| 2999 | 011142' | 000000 | MD3:    | .WORD | 0     | ;REGISTER +46          |
| 3000 | 011144' | 000000 | AD78:   | .WORD | 0     | ;REGISTER +50          |
| 3001 | 011146' | 000000 | ST78:   | .WORD | 0     | ;REGISTER +52          |
| 3002 | 011150' | 000000 |         | .WORD | 0     | ;REGISTER +54          |
| 3003 | 011152' | 000000 |         | .WORD | 0     | ;REGISTER +56          |
| 3004 | 011154' | 000000 |         | .WORD | 0     | ;REGISTER +60          |
| 3005 | 011156' | 000000 |         | .WORD | 0     | ;REGISTER +62          |
| 3006 | 011160' | 000000 |         | .WORD | 0     | ;REGISTER +64          |
| 3007 | 011162' | 000000 |         | .WORD | 0     | ;REGISTER +66          |
| 3008 | 011164' | 000000 |         | .WORD | 0     | ;REGISTER +70          |
| 3009 | 011166' | 000000 |         | .WORD | 0     | ;REGISTER +72          |
| 3010 | 011170' | 000000 | EBA:    | .WORD | 0     | ;REGISTER +74          |
| 3011 | 011172' | 000000 | CS3:    | .WORD | 0     | ;REGISTER +76          |
| 3012 |         |        |         |       |       |                        |
| 3013 | 011174' | 002000 | RUFIN:  | .BLKH | 1024. | ;1024 BYTE READ BUFFER |

|      |        |  |                             |   |        |                      |
|------|--------|--|-----------------------------|---|--------|----------------------|
| 3015 |        |  | .SHTTL PROGRAM DEFINITIONS  |   |        |                      |
| 3016 |        |  | ;                           |   |        |                      |
| 3017 |        |  | ;INTERRUPT CODE DEFINITIONS |   |        |                      |
| 3018 |        |  | ;                           |   |        |                      |
| 3019 |        |  |                             |   |        |                      |
| 3020 | 000027 |  | BTAPE                       | = | 000027 | ;BAD TAPE            |
| 3021 | 000001 |  | DONE                        | = | 000001 | ;DONE                |
| 3022 | 000004 |  | EDT                         | = | 000004 | ;END OF TAPE         |
| 3023 | 000023 |  | RDOPP                       | = | 000023 | ;READ OPPOSITE       |
| 3024 | 000022 |  | RTKY                        | = | 000022 | ;RETRY               |
| 3025 | 000032 |  | TMFH                        | = | 000032 | ;TM FAULT B          |
| 3026 | 000024 |  | UNREAD                      | = | 000024 | ;UNREADABLE          |
| 3027 |        |  |                             |   |        |                      |
| 3028 |        |  | ;                           |   |        |                      |
| 3029 |        |  | ;COMMAND CODE DEFINITIONS   |   |        |                      |
| 3030 |        |  | ;                           |   |        |                      |
| 3031 |        |  |                             |   |        |                      |
| 3032 | 000063 |  | WGCR                        | = | 000063 | ;WRITE GCR           |
| 3033 | 000061 |  | WPE                         | = | 000061 | ;WRITE PE            |
| 3034 | 000017 |  | GCRM                        | = | 000017 | ;WRITE GCR TAPE MARK |
| 3035 | 000015 |  | PETM                        | = | 000015 | ;WRITE PE TAPE MARK  |
| 3036 | 000007 |  | REW                         | = | 000007 | ;REWIND              |
| 3037 | 000011 |  | SEN                         | = | 000011 | ;SENSE               |
| 3038 | 000071 |  | RFWD                        | = | 000071 | ;READ FORWARD        |
| 3039 | 000077 |  | RREV                        | = | 000077 | ;READ REVERSE        |

SEQ 0065

|         |         |       |       |
|---------|---------|-------|-------|
| ACSR    | 000102P | 1155# | 2588* |
| ADDR    | 000006R | 1155# | 1232  |
| ADDR27= | 001000  | 1155# | 2386  |
| AD78    | 011144R | 3000# |       |
| AS      | 011112R | 2592* | 2601  |
| ASB     | 000106R | 1155# | 2604* |
| ASREG   | 010512R | 2601* | 2602  |
| ASTAT   | 000104R | 1155# | 2589* |
| AWAS    | 000110W | 1155# |       |
| A16A17  | 010462R | 2393* | 2394* |
|         |         | 2462* | 2463* |
|         |         | 2395* | 2396* |
|         |         | 2464* | 2465* |
|         |         | 2397* | 2398* |
|         |         | 2466* | 2467* |
|         |         | 2399* | 2400* |
|         |         | 2468* | 2469* |
|         |         | 2401* | 2402* |
|         |         | 2409  | 2456* |
|         |         | 2457* | 2458* |
|         |         | 2459* | 2461* |
| RA      | 011100R | 2391* | 2397* |
| HC      | 011102R | 2382* | 2383* |
| HEGIN   | 000000R | 1155# | 1199  |
|         |         | 2281  | 2282  |
|         |         | 2664  | 2680  |
|         |         | 2728  | 2737  |
|         |         | 2778  | 2780  |
| RIT0 =  | 000001  | 1155# | 1480  |
| RIT1 =  | 000002  | 1155# | 1281  |
| RIT10 = | 002000  | 1155# | 1752  |
| RIT11 = | 004000  | 1155# |       |
| RIT12 = | 010000  | 1155# | 1291  |
| RIT13 = | 020000  | 1155# |       |
| RIT14 = | 040000  | 1155# | 1289  |
| RIT15 = | 100000  | 1155# |       |
| RIT2 =  | 000004  | 1155# | 1822  |
|         |         | 2637  | 1829  |
| RIT3 =  | 000010  | 1155# | 2261  |
| HIT4 =  | 000020  | 1155# | 1209  |
|         |         | 1932  | 1960  |
| HIT5 =  | 000040  | 1155# | 1209  |
|         |         | 2615  | 2629  |
| RIT6 =  | 000100  | 1155# | 1594  |
| RIT7 =  | 000200  | 1155# |       |
| RIT8 =  | 000400  | 1155# |       |
| RIT9 =  | 001000  | 1155# |       |
| BOTCNT  | 010454R | 1647  | 1740* |
| BREAKS= | 104407  | 1155# | 1307  |
| BRI     | 000012R | 1155# | 1381  |
| BR2     | 000013R | 1155# |       |
| BTAPF = | 000027  | 1856  | 3020# |
| BDOS =  | 104421  | 1155# | 2276  |
| BUFIN   | 011174W | 1155  | 2437  |
| CDATAS= | 104412  | 1155# | 2221  |
| CDCRCT  | 000144R | 1155# |       |
| CWDCT   | 000146R | 1155# |       |
| CONFIC  | 000056R | 1155# |       |
| CRLF    | 010334R | 2664  | 2800# |
| CSRA    | 000100R | 1155# | 2587* |
| CS1     | 011074R | 1196* | 1203* |
|         |         | 2378* | 2410* |
|         |         | 2571* | 2577* |
|         |         | 1249* | 1287* |
|         |         | 2748  | 2984# |
|         |         | 3011# |       |
| CS2     | 011104R | 1291* | 1395* |
|         |         | 1428* | 1594* |
|         |         | 2377* | 2439* |
|         |         | 2518* | 2519* |
|         |         | 2630* | 2746  |
|         |         | 2439* | 2517* |
|         |         | 2545* | 2569* |
|         |         | 2589  | 2607* |
|         |         | 2620* |       |
| CS3     | 011172R | 3011# |       |

[illegible]

|        |          |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|        |          | 1980* | 1939* | 1944* | 1965* | 1972* | 1979* | 2037* | 2042* | 2063* | 2070* | 2077* | 2132* | 2138* |
|        |          | 2144* | 2186* | 2194* | 2199* | 2637* |       |       |       |       |       |       |       |       |
| M4.1TR | 007014R  | 2663* | 2667* |       |       |       |       |       |       |       |       |       |       |       |
| NULL   | = 000000 | 1155* | 2680  | 2686  | 2696  | 2706  | 2717  | 2727  | 2737  | 2745  | 2759  | 2765  | 2770  |       |
| OPEN   | = 000000 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| OTDAS  | = 104420 | 1155* | 2281  | 2282  | 2687  | 2747  | 2749  | 2778  | 2780  |       |       |       |       |       |
| PASCNT | 000034R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PA1R   | 010464R  | 2388* |       |       |       |       |       |       |       |       |       |       |       |       |
| PA22   | 010470R  | 2391  | 2454  | 2451* | 2453  | 2828* |       |       |       |       |       |       |       |       |
| PRTM   | = 000015 | 2528  | 3035* |       |       |       |       |       |       |       |       |       |       |       |
| PIHDS  | = 000004 | 1155* | 2599  | 2616  |       |       |       |       |       |       |       |       |       |       |
| POPS   | = 005726 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PUPSP2 | = 022626 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY   | = 000000 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY0  | = 000000 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY1  | = 000040 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY2  | = 000100 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY3  | = 000140 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY4  | = 000200 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY5  | = 000240 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY6  | = 000300 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PRTY7  | = 000340 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PS     | = 177776 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PSW    | = 177776 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PUSH   | = 005746 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| PUSH2  | = 024646 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RANDS  | = 104417 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RANRUM | 000054R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RHUFPA | 000130R  | 1155* | 2452  | 2461  |       |       |       |       |       |       |       |       |       |       |
| RRUFPA | 000126R  | 1155* | 2221  | 2451  | 2460  |       |       |       |       |       |       |       |       |       |
| RRUFSA | 000132R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RRUFVA | 000124R  | 1155* | 2437* | 2448  | 2495* | 2496* | 2497* | 2498* |       |       |       |       |       |       |
| RDFCN  | 010506R  | 2219  | 2436* | 2467  | 2494* | 2837* |       |       |       |       |       |       |       |       |
| RDOPP  | = 000023 | 1952  | 2050  | 3023* |       |       |       |       |       |       |       |       |       |       |
| RECF1L | 010336R  | 2699* | 2709* | 2720* | 2730* | 2752* | 2803* |       |       |       |       |       |       |       |
| RECMG  | 010377R  | 2803  | 2809* |       |       |       |       |       |       |       |       |       |       |       |
| RECORD | 010504R  | 1484* | 1597* | 1656* | 1665  | 1671  | 1676* | 2803  | 2836* |       |       |       |       |       |
| RESTR  | 001566R  | 1155  | 1592* |       |       |       |       |       |       |       |       |       |       |       |
| RES1   | 000056R  | 1155* | 2386  | 2449  |       |       |       |       |       |       |       |       |       |       |
| RES2   | 000060R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RFW    | = 000007 | 2189  | 2552  | 3036* |       |       |       |       |       |       |       |       |       |       |
| RFMSG  | 010366R  | 2805  | 2808* |       |       |       |       |       |       |       |       |       |       |       |
| RFWD   | = 000071 | 2436  | 3038* |       |       |       |       |       |       |       |       |       |       |       |
| RHTBL  | 011074R  | 1235* | 2979* |       |       |       |       |       |       |       |       |       |       |       |
| RREAD  | 010674R  | 1470* | 1937* | 2035* | 2275  | 2911* |       |       |       |       |       |       |       |       |
| RREV   | = 000077 | 2219  | 2494  | 3039* |       |       |       |       |       |       |       |       |       |       |
| RSTR   | 000112R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| RTRY   | = 000022 | 1865  | 1946  | 2044  | 3024* |       |       |       |       |       |       |       |       |       |
| RWRIT  | 010774R  | 1469* | 1833* | 1851* | 2277  | 2946* |       |       |       |       |       |       |       |       |
| SBADR  | 000102R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SEN    | = 000011 | 2576  | 3037* |       |       |       |       |       |       |       |       |       |       |       |
| SENSF  | 010456R  | 1404* | 1405  | 1434* | 1435  | 1752  | 2609* | 2825* |       |       |       |       |       |       |
| SN     | 011124R  | 2992* |       |       |       |       |       |       |       |       |       |       |       |       |
| SOPCNT | 000042R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SOFERS | = 104406 | 1155* | 2696  | 2717  | 2745  |       |       |       |       |       |       |       |       |       |
| SOPPAS | 000046R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |

|         |          |       |       |       |       |       |       |       |       |       |       |       |       |       |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| SPPOINT | 000032R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SPSIZ   | = 000040 | 7*    | 1155  |       |       |       |       |       |       |       |       |       |       |       |
| SR1     | 000016R  | 1155* | 1208  | 1281  | 1383  | 1481  | 1696  | 1700  | 2217  | 2261  | 2637  |       |       |       |
| SR2     | 000020R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SR3     | 000022R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SR4     | 000024R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| START   | 000252R  | 1155  | 1190* |       |       |       |       |       |       |       |       |       |       |       |
| STAT    | 000026R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| STATHD  | 004740R  | 2263  | 2291* |       |       |       |       |       |       |       |       |       |       |       |
| STATLN  | 005044R  | 2281  | 2294* |       |       |       |       |       |       |       |       |       |       |       |
| STHR    | 005072R  | 2276  | 2299* |       |       |       |       |       |       |       |       |       |       |       |
| STRW    | 005131R  | 2278  | 2303* |       |       |       |       |       |       |       |       |       |       |       |
| STTM    | 005046R  | 2281  | 2295* |       |       |       |       |       |       |       |       |       |       |       |
| STTU    | 005057R  | 2282  | 2297* |       |       |       |       |       |       |       |       |       |       |       |
| STUHR   | 005111R  | 2280  | 2301* |       |       |       |       |       |       |       |       |       |       |       |
| ST78    | 011146R  | 1305  | 3001* |       |       |       |       |       |       |       |       |       |       |       |
| SUTBL   | 010554R  | 1386* | 1399  | 1410* | 1424* | 1440* | 1602  | 1613  | 1617  | 1622  | 1746  | 1766* | 1771* | 1862* |
|         |          | 1874* | 1882* | 1974* | 1981* | 2072* | 2079* | 2121  | 2140* | 2146* | 2196* | 2201* | 2339  | 2865* |
| SVR0    | 000062R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR1    | 000064R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR2    | 000066R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR3    | 000070R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR4    | 000072R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR5    | 000074R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SVR6    | 000076R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| SYSCNT  | 000052R  | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| TC      | 011110R  | 2385* | 2447* | 2986* |       |       |       |       |       |       |       |       |       |       |
| TEMP    | 010500R  | 2275* | 2276  | 2277* | 2278  | 2279* | 2280  | 2746* | 2747  | 2748* | 2749  | 2777* | 2778  | 2779* |
|         |          | 2780  | 2834* |       |       |       |       |       |       |       |       |       |       |       |
| TINTCD  | 010450R  | 1402  | 1432  | 1750  | 1761  | 1826  | 1843  | 1856  | 1865  | 1869  | 1930  | 1946  | 1952  | 1958  |
|         |          | 1968  | 2028  | 2044  | 2050  | 2056  | 2066  | 2127  | 2135  | 2180  | 2189  | 2191  | 2621* | 2622* |
|         |          | 2625* | 2626* | 2687  | 2822* |       |       |       |       |       |       |       |       |       |
| TMHIT   | 010430R  | 1285  | 1293  | 1299  | 1312  | 1393  | 1426  | 1496* | 2602  | 2627  | 2814* |       |       |       |
| TMBIT2  | 010432R  | 1398  | 1409  | 1423  | 1439  | 1499* | 1601  | 1608  | 1612  | 1616  | 1621  | 1745  | 1749  | 1765  |
|         |          | 1770  | 1825  | 1831  | 1845  | 1861  | 1873  | 1881  | 1928  | 1935  | 1961  | 1973  | 1980  | 2033  |
|         |          | 2059  | 2071  | 2078  | 2120  | 2126  | 2139  | 2145  | 2195  | 2200  | 2268  | 2271  | 2338  | 2815* |
| TMCONV  | 001462R  | 1284* | 1392* | 1422* | 1489* | 1599* | 1743* | 2118* | 2266* | 2336* |       |       |       |       |
| TMFH    | = 000032 | 1761  | 1869  | 1968  | 2066  | 2135  | 2191  | 3025* |       |       |       |       |       |       |
| TMUNIT  | 010476R  | 1194  | 1280* | 1283* | 1285  | 1293* | 1299* | 1312* | 1393  | 1426  | 2833* |       |       |       |
| TM78    | 010442R  | 1279* | 1287  | 1300* | 1301  | 1390* | 1395  | 1416* | 1417  | 1420* | 1428  | 1446* | 1447  | 1489  |
|         |          | 1497  | 1595* | 1629* | 1630  | 1741* | 1776* | 1777  | 2116* | 2152* | 2153  | 2264* | 2281  | 2287* |
|         |          | 2288  | 2334* | 2348* | 2349  | 2377  | 2439  | 2517  | 2545  | 2569  | 2607  | 2620  | 2777  | 2819* |
| TRPDFD  | = 000022 | 1155* |       |       |       |       |       |       |       |       |       |       |       |       |
| TUBIT   | 010434R  | 1399  | 1410  | 1440  | 1441  | 1509* | 1602  | 1609  | 1613  | 1617  | 1622  | 1746  | 1771  | 1846  |
|         |          | 1862  | 1882  | 1981  | 2079  | 2121  | 2123  | 2146  | 2201  | 2269  | 2339  | 2342  | 2816* |       |
| TUBIT2  | 010436R  | 1512* | 1832  | 1850  | 1936  | 1962  | 2034  | 2060  | 2274  | 2817* |       |       |       |       |
| TUCONV  | 001524R  | 1397* | 1430* | 1502* | 1600* | 1744* | 2119* | 2267* | 2337* |       |       |       |       |       |
| TUSTAT  | 011114R  | 2609  | 2988* |       |       |       |       |       |       |       |       |       |       |       |
| TU78    | 010444R  | 1396* | 1413* | 1414  | 1429* | 1443* | 1444  | 1502  | 1598* | 1625* | 1626  | 1742* | 1772* |       |
|         |          | 1773  | 2117* | 2148* | 2149  | 2265* | 2282  | 2284* | 2285  | 2335* | 2345* | 2346  | 2385  | 2447  |
|         |          | 2522  | 2550  | 2574  | 2613  | 2779  | 2820* |       |       |       |       |       |       |       |
| TYPE1   | 007266R  | 2648* | 2652* | 2656* | 2702* |       |       |       |       |       |       |       |       |       |
| TYPE2   | 007360R  | 2649* | 2653* | 2657* | 2713* |       |       |       |       |       |       |       |       |       |
| TYPE3   | 007443R  | 2650* | 2654* | 2658* | 2723* |       |       |       |       |       |       |       |       |       |
| TYPE4   | 007527R  | 2651* | 2655* | 2659* | 2733* |       |       |       |       |       |       |       |       |       |
| UCNT    | 010440R  | 1201  | 1391* | 1412* | 1421* | 1442* | 1634  | 1645  | 1647  | 1659  | 2332* | 2341* | 2818* |       |

|                 |       |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|--|--|--|--|--|--|
| UNITMS 010224R  | 2781  | 2784# |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| UNITS 010160R   | 2689# | 2698# | 2708# | 2719# | 2729# | 2739# | 2751# | 2772# | 2777# |       |       |       |       |  |  |  |  |  |  |  |
| UNITTM 010231R  | 277#  | 2785# |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| UNITTU 010247R  | 2780  | 2786# |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| UNHEAD= 000024  | 1958  | 2056  | 3026# |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| URHEAD 010574R  | 1471# | 1963# | 2061# | 2279  | 2876# |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| UTAL 010514R    | 1386  | 1425# | 1441# | 2269  | 2843# |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| VECTOR 000010R  | 1155# | 1379  |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WASADR 000104R  | 1155# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WHUFFA 000136R  | 1155# | 2389  | 2398  |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WHUFFA 000134R  | 1155# | 2388  | 2397  |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WHUFFQ 000140R  | 1155# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WHUFFSZ 000142R | 1155# | 2380  | 2382  | 2442  | 2444  | 2496  | 2497  |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WC 011076R      | 2380# | 2381# | 2442# | 2443# | 2981# |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WDFR 000116R    | 1155# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WDT0 000114R    | 1155# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WGCR = 000063   | 2406  | 3032# |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| WPE = 000061    | 2408  | 3033# |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| XFLAG 000005R   | 1155# |       |       |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| XFRINT 011106R  | 2384# | 2446# | 2621  | 2985# |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| XMEM 010406R    | 2389# | 2452# | 2829# |       |       |       |       |       |       |       |       |       |       |  |  |  |  |  |  |  |
| . = 013174R     | 2221  | 2293# | 2305# | 2684# | 2694# | 2704# | 2715# | 2757# | 2763# | 2769# | 2776# | 2811# | 3013# |  |  |  |  |  |  |  |

|             |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--|--|--|--|--|--|
| BKMOD 95#   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| BREAK 185#  | 1307 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| BTOD 204#   | 2276 | 2278 | 2280 | 2803 | 2804 |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| CKDATA 240# | 2221 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| DATACK 249# |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| DATERR 138# |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| DFSEVN 272# | 1155 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| DSEVNT 282# | 1155 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| END 175#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| ENDIT 166#  | 1677 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| ENDMOD 171# | 1199 | 1206 | 1215 | 1639 |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| EQUATS 288# | 1155 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| EXIT 120#   | 2411 | 2471 | 2530 | 2554 | 2578 |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| GETPA 231#  | 2448 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| GWRUFF 219# | 1596 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| HRDER 128#  | 2680 | 2686 | 2706 | 2727 | 2737 | 2759 | 2765 | 2770 |      |      |      |      |      |      |  |  |  |  |  |  |
| IOMOD 91#   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| IOMODP 115# |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| IOMODR 111# |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| IOMODX 107# | 1155 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| MAP22 235#  | 2390 | 2453 |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| MODULF 8#   | 1155 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| MSG 154#    | 2263 | 2283 | 2664 | 2681 | 2688 | 2697 | 2707 | 2718 | 2728 | 2738 | 2750 | 2760 | 2766 | 2771 |  |  |  |  |  |  |
| 2781        | 2788 | 2794 | 2805 |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| MSGN 158#   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| MSGS 162#   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| NRKMOD 103# |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| OTOA 190#   | 2281 | 2282 | 2687 | 2747 | 2749 | 2778 | 2780 |      |      |      |      |      |      |      |  |  |  |  |  |  |
| PIRG 179#   | 2599 | 2616 |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| RAND 124#   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| SBKMOD 99#  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |
| SOFFP 144#  | 2696 | 2717 | 2745 |      |      |      |      |      |      |      |      |      |      |      |  |  |  |  |  |  |

. AHS. 000000 000  
 013174 001

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

XTMDA0.XTMDA0/CRF=DDXCOM.P11.XTMDA0.P11  
 RUN-TIME: 4 8 .7 SECONDS  
 RUN-TIME RATIO: 47/14=3.4  
 CORF USED: 10K (20 PAGES)

C

Component Part Number : AF-S210A-M1

DIGITAL EQUIPMENT CORPORATION  
DIAGNOSTIC ENGINEERING PATCH ORDERPAGE 1 OF 1  
Form : B

Component Part Description : ECO TALLY 01

ECO Purpose : Patch

Approved : Yes

Diagnostic Package Code : CXTMD-A1

PRODUCT IDENTIFICATION

Patch Level : 1

Product Date : 23-Sep-80

Description : CXTMDA1 DEC/X11 TM78 MODULE

1ST Copyright year : 1980

Last Copyright Year : 1980

Expanded Description :

PRODUCT COMPONENTS AFFECTED

PRODUCTS OBSOLETE

PROBLEM/DESCRIPTION

UNEXPECTED INTERRUPT CODE 00000 WHEN RUNNING MONITOR  
C ON AN 11/40.

SOLUTION/CORRECTION

CORRECT TYPE ERROR IN WRITE SUBROUTINE MODULE 3.1.  
MOVE BIT 14, \* CS1 SHOULD BE MOV #BIT 14, \* CS1.  
THIS IS A CONDITIONAL PATCH.

DEPO PATCH AREA

| Loc  | From  | To    | Mnemonic | Loc | From | To | Mnemonic |
|------|-------|-------|----------|-----|------|----|----------|
| 5266 | 16777 | 12777 |          |     |      |    |          |

PRODUCT CHARACTERISTICS

Run Time (seconds) 1st Pass : 75      2nd Pass : 75      ACT Sequence Number : 1351      Multimedia Affected : \*No

Processors : 05 10 20 34 35 40 44 45 50 55 60 70

Operational Environment : 02 03 04 06 50

Device Affected : TM78      TU78

Problem Reports Affected :

Kit Number : 7J129

AUTHORIZATION

|                        |               |                        |             |
|------------------------|---------------|------------------------|-------------|
| Author                 | : R. COOKE    | Maintaining Group      | : JO DIA    |
| Submitting Engineer    | : B. VANDETTE | Maintainer             | : DEKNIS    |
| Product Engineer       | :             | Field Service Engineer | :           |
| Manufacturing Engineer | : R. SACINO   | Release Engineer       | : M. HANSEN |
| Charge Number          | : V980-05346  | Coordination Number    | : MC3796    |

