

Micro Fiche Scan

Name of device(s) tested:

DJ11

Test description:

DJ11 MODULE

MAINDEC Number or Package Identifier (after SEP 1977):

CXDJAL0

Fiche Document Part Number:

AH-E723L-MC

Fiche preparation date unknown, using copyright year:

1982

Image resolution:

1-bit black&white, compressed for minimal file size

COPYRIGHT (C) 73-82 by d|i|g|i|t|a|l

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-E721L-MC

PRODUCT NAME: CXDJALO DJ11 MODULE

DATE: JUNE 1981

MAINTAINER: DEC/X11 SUPPORT GROUP

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 ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1982 DIGITAL EQUIPMENT CORPORATION

2

REM %
1.

ABSTRACT

THIS MODULE EXERCISES ANY COMBINATION OF ONE TO SIXTEEN LINES ON ANY COMBINATION OF FROM ONE TO FOUR DJ11 ASYNCHRONOUS, SIXTEEN LINE COMMUNICATION DEVICES. 65 CHARACTERS ARE TRANSMITTED ON EACH DJ11 UNDER INTERRUPT CONTROL. TRANSMISSION OF THE 65TH CHARACTER CAUSES A RECEIVER BUFFER FULL INTERRUPT ON THE DJ11. THE ROUTINE DISABLES INTERRUPTS ON THE RESPECTIVE DJ11 AND EMPTIES THE HARDWARE SILO INTO A SOFTWARE BUFFER. THE DATA IS CHECKED, THEN INTERRUPTS ARE REENABLED AND TRANSMISSION RESUMED FOR THE GIVEN DJ11. AS EACH LINE FINISHES TRANSMISSION OF THE FULL 8-BIT COUNT PATTERN, IT IS DISABLED. THUS, THERE ARE TIME WHEN ONLY A SUBSET OF THE SELECTED LINES ARE ACTUALLY RUNNING. WHEN ALL LINES ON A DJ11 HAVE TRANSMITTED THE ENTIRE COUNT PATTERN, THEY ARE ALL REENABLED AND THE PATTERN TRANSMISSION IS RESTARTED. A PASS IS DEFINED AS 200 (OCTAL) TRANSMISSIONS OF 65 CHARACTERS EACH ON EACH ACTIVE DJ11. THE MODULE ASSUMES THAT ALL SIXTEEN LINES WILL BE EXERCISED AND WILL BE SENDING 8 BIT CHARACTERS. SEE 'OPERATION OPTIONS' BELOW FOR NECESSARY CHANGES REQUIRED FOR OTHER CONFIGURATIONS.
A BACKGROUND 'WATCHDOG TIMER' IS ALSO BY THE MODULE TO ALLOW DETECTION OF THE 'DEVICE HUNG' CONDITION (AN EXPECTED INTERRUPT THAT FAILED TO OCCUR). THE TIMER IS SET TO RUN LONGER THAN THE LONGEST POSSIBLE MODULE PASS TIME, AND ON A HEAVILY LOADED SYSTEM WILL IN FACT TAKE A LONG TIME TO TIMEOUT (I.E. 10-15 MINUTES). WHEN THE 'DEVICE HUNG' CONDITION IS DETECTED A 'DEVICE HUNG' MESSAGE IS OUTPUT AND THE MODULE IS THEN DROPPED. THIS PREVENTS A LONG RUN FROM BEING PERMANENTLY UNABLE TO RE-START DUE TO A 'LOST' INTERRUPT.

2. REQUIREMENTS

HARDWARE: DJ11 SIXTEEN LINE ASYNCHRONOUS COMMUNICATION DEVICE.
STORAGE:: DJA REQUIRES:
1. DECIMAL WORDS: 01169
2. OCTAL WORDS: 02217
3. OCTAL BYTES: 04436

3. PASS DEFINITION

ONE PASS OF THE DJA MODULE CONSISTS OF TRANSMITTING, RECEIVING, AND CHECKING 8320. CHARACTERS PER DJ11.

4. EXECUTION TIMES

WORST CASE (1 LINE ONLY ON EACH OF 4 DJ11'S AT 75 BAUD) TAKES APPROXIMATELY 3 MINUTES PER PASS. RUNNING ALL LINES (DEFAULT CASE) WILL CAUSE SEVERAL PASSES PER MINUTE TO BE COMPLETED.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
BR1: 5
DEVICE COUNT(DVID1): 1

REQUIRED PARAMETERS:

FIRST DEVICE ADDRESS: THE FIRST DEVICE REGISTER ADDRESS
MUST BE SPECIFIED.
VECTOR: THE VECTOR ADDRESS OF THE FIRST DJ11 MUST BE GIVEN
DVID1: IF MORE THAN 1 UNIT IS BEING RUN, VALUE MUST BE GIVEN
(4 UNITS MAX)

6. DEVICE/OPTION SETUP

ALL SIXTEEN LINES OPERATIONAL (DEFAULT) OR ALL SELECTED LINES OPERATIONAL
(IF NOT ALL LINES SELECTED).

7. MODULE OPERATION

- A. INITIALIZE FIRST VALUES TO BE TRANSMITTED FRO EACH LINE
(SET TO ZERO).
- B. INITIALIZE INTERNAL QUEUES AND PARAMETERS, SETUP LINKAGE OF DEVICE
VECTORS TO LINKAGE TABLE, AND INITIALIZE SELECTED DJ11'S
VIA SETTING 'MOS CLEAR'.
- C. SET TRANSMITTER INTERRUPT ENABLE AND SILO FULL INTERRUPT ENABLE,
THEN SET TRANSMITTER ENABLE FOR ALL SELECTED LINES ON EACH
SELECTED DJ11. COUNT DJ11'S SELECTED AS THEY GET TURNED ON.
- D. INITIALIZE WATCHDOG TIMER, WHICH RUNS VIA A SET OF 'BREAK'
LOOPS. THE LOOP MONITORS THE COUNT OF ACTIVE DJ11'S. IF THE DJ11
COUNT GOES TO ZERO BEFORE THE TIMER TIMES OUT, AN END OF PASS CALL
IS MADE. (WHICH WILL EVENTUALLY CAUSE THE MODULE TO BE RESTARTED
AT STEP B). IF THE TIMER TIMES OUT BEFORE THE DJ11 ACTIVE
COUNT GOES TO ZERO, A 'DEVICE HUNG' MESSAGE IS OUTPUT AND
THE MODULE IS DROPPED.
- E. WHEN A TRANSMIT INTERRUPT OCCURS, TRANSMIT INTERRUPT ENABLE IS
TURNED OFF FOR THAT DJ11, A UNIT IDENTIFICATION IS SAVED IN
THE MODULE'S TRANSMIT QUEUE, AND THE MODULE PIRQ'S. WHEN THE MONITOR
RETURNS TO THE TRANSMIT ISR, THE NEXT CHARACTER FOR THE
INTERRUPTING LINE IS TRANSMITTED. IF THE CHARACTER TRANSMITTED
WAS 377 (OCTAL), THEN THAT LINE IS DISABLED FROM BEING
RUN UNTIL ALL THE OTHER LINES HAVE ALSO REACHED THE MAX VALUE
(ACTUALLY, IT WON'T BE DISABLED UNTIL THE NEXT 65 CHARACTER
TRANSMISSION ON THIS DJ11). IF ALL LINES ARE DISABLED,
THE SELECTED LINES GET REENABLED. IF 65 (DECIMAL) CHARACTERS
HAVE BEEN TRANSMITTED, THE TRANSMITTER IS DISABLED BY CLEARING THE
TRANSMITTER CONTROL REGISTER. IF LESS THAN 65 CHARACTERS HAVE
BEEN TRANSMITTED, TRANSMIT INTERRUPT ENABLE IS REENABLED. THE
TRANSMITTER SERVICE THEN EXITS.
- F. WHEN 65 CHARACTERS HAVE BEEN TRANSMITTED ON A GIVEN DJ11, A
RECEIVER BUFFER ('SILO') FULL INTERRUPT SHOULD OCCUR. THE
MODULE THEN DISABLES THE 'SILO FULL' INTERRUPT, QUEUES A
UNIT IDENTIFICATION IN IT'S RECEIVER QUEUE, AND PIRQ'S. WHEN THE
MONITOR RETURNS TO THE RECEIVER SERVICE, THE MODULE DUMPS
THE CONTENTS OF THE HARDWARE SILO OF THE INTERRUPTING DJ11 INTO A
SOFTWARE BUFFER AND CHECKS THE DATA. IF THE REQUIRED NUMBER OF
TRANSMISSIONS (STORED IN ENDSEL) HAVE BEEN PERFORMED BY THIS

DJ11, IT IS TURNED OFF AND THE DJ11 ACTIVE COUNT IS
DECREMENTED. IF NOT ALL DONE, THE RECEIVER DATA TABLE IS RESYNCD
TO MATCH THE TRANSMITTER TABLE AND THE DJ11 IS REENABLED TO
TRANSMIT ANOTHER 65 CHARACTERS. WHEN THE DJ11 ACTIVE COUNT
GOES TO 0, THE WATCHDOG TIMER WILL DETECT IT AND ISSUE AN
END OF PASS (SEE STEP D).

THE SELECTED LINES GET REENABLED. IF 65 (DECIMAL) CHARACTERS

8. OPERATION OPTIONS

A. TO MAKE THE TIME INTERVAL BETWEEN ENDPAS STATEMENTS LONGER OR SHORTER,
EITHER INCREASE OR DECREASE THE 'ENDSEL' WORD.

B. TO EXERCISE A CONFIGURATION OF LINES OTHER THAN ALL SIXTEEN AT ONCE,
CHANGE THE APPROPRIATE BITS OF THE 'XMITEN' WORD FOR THE SELECTED DJ11
AND THE SAME BITS OF THE 'XMTSEL' WORD. A ONE INDICATES THAT
THE LINE IS TO BE EXERCISED, A ZERO INDICATES THAT THE LINE IS OFF.

C. MEANING OF SR1

1. FOR 5 BIT DATA SET BIT 2 (SR1=000004)
FOR 6 BIT DATA SET BIT 1 (SR1=000002)
FOR 7 BIT DATA SET BIT 0 (SR1=000001)
FOR 8 BIT DATA SET NO BITS (SR1=000000)
DEFAULT VALUE IS 8 BIT DATA LENGTH
CHECK HARDWARE STRAPPING TO DETERMINE SR1 SETTING
2. ITERATION COUNT - USED TO ADJUST PASS TIME TO BAUD RATE
FOR 75 BAUD SET BIT 5
FOR 110 BAUD SET BIT 6
FOR 134.5 BAUD SET BIT 7
FOR 150 BAUD SET BIT 8
FOR 300 BAUD SET BIT 9
FOR 600 BAUD SET BIT 10
FOR 1200 BAUD SET BIT 11
FOR 1800 BAUD SET BIT 12
FOR 2400 BAUD SET BIT 13
FOR 4800 BAUD SET BIT 14
FOR 9600 BAUD SET BIT 15 OR NO BITS (5-15=0)

9. NON STANDARD PRINTOUTS

THERE ARE 2 SPECIAL MESSAGES WHICH MAY BE PRINTED OUT. THEY ARE:

'UNIT DROPPED'- THIS INDICATES THAT A DJ11 WHICH HAS JUST PREVIOUSLY
REPORTED A FAILURE HAS BEEN DROPPED. DJ11'S ARE DROPPED IF
'BUSY CLEAR' FAILS TO CLEAR WHEN THEY ARE INITIALIZED.

'UNIT HUNG'- THIS INDICATES THAT THE MODULE'S WATCHDOG TIMER
HAS TIMED OUT BEFORE THE DJ11 DEVICE ACTIVE COUNT WENT TO ZERO.
THIS OCCURS WHEN AN EXPECTED TRANSMITTER OR RECEIVER INTERRUPT
FAILS TO OCCUR ON ONE OR MORE DJ11'S.

*
* EDIT: BY: DATE: REASON: *

★
★ 003 R. GAUDIN 19-JUN-81 ADD ERROR PRINTOUT WHEN THE ★
★ VECTOR + ADDR LOCS. ARE ZERO. ★
★

%

```

.SBTTL MODULE HEADER
000000' IOMOD <DJAL>,0,0,5,,0,36
000000' MODULE 140000,DJAL,0,0,5,,0,36,
.TITLE DJAL DEC/X11 SYSTEM EXERCISER MODULE
; DDXCOM VERSION 6.2 23-APR-81
.LIST BIN
*****
000000' BEGIN:
000000' 045104 046101 040 MODNAM: .ASCII /DJAL / ;MODULE NAME.
000005' 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000000 ADDR: 0+0 ;1ST DEVICE ADDR.
000010' 000000 VECTOR: 0+0 ;1ST DEVICE VECTOR.
000012' 240 BR1: .BYTE PRTY5+0 ;1ST BR LEVEL.
000013' 000 BR2: .BYTE PRTY+0 ;2ND BR LEVEL.
000014' 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000016' 000000 SR1: OPEN ;SWITCH REGISTER 1
000020' 000000 SR2: OPEN ;SWITCH REGISTER 2
000022' 000000 SR3: OPEN ;SWITCH REGISTER 3
000024' 000000 SR4: OPEN ;SWITCH REGISTER 4
*****
000C26' 140000 STAT: 140000 ;STATUS WORD.
000030' 002056' INIT: START ;MODULE START ADDR.
000032' 000224' SPOINT: MODSP ;MODULE STACK POINTER.
000034' 000000 PASCNT: 0 ;PASS COUNTER.
000036' 000000 ICONT: 0 ;# OF ITERATIONS PER PASS=0
000040' 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
000042' 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054' 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056' 000000 CONFIG: ;RESERVED FOR MONITOR USE
000056' 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060' 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062' 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064' 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066' 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070' 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072' 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074' 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076' 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100' 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102' 000000 SBADR: ;ADDR OF GOOD DATA, OR
000102' 000000 ACSR: OPEN ;CONTENTS OF CSR.
000104' 000000 WASADR: ;ADDR OF BAD DATA, OR
000104' 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
000106' 000000 ERRTP: ;TYPE OF ERROR
000106' 000000 ASB: OPEN ;EXPECTED DATA.
000110' 000000 AWAS: OPEN ;ACTUAL DATA.
000112' 002440' RSTRT: RESTRT ;RESTART ADDRESS AFTER END OF PASS
000114' 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
000116' 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000120' 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122' 000036 IDNUM: 36 ;MODULE IDENTIFICATION NUMBER=36
000040' 000040 .REPT SPS1Z ;MODULE STACK STARTS HERE.

```

.NLIST
.WORD 0
.LIST
.ENDR

000224'

MODSP:

;
;+
; THE NEXT LOCATION CONTAINS A NUMBER THAT CAN BE EXAMINED IN CORE TO BE
; SURE THAT THE LISTING AND ACTUAL CODE ARE THE SAME. THIS LOCATION IS TO
; BE INCREMENTED EACH TIME THE SOURCE CODE IS UPDATED. LOCATION 'MODREV'
; IS NOT USED BY THE PROGRAM.
;-

MODREV: 1+1+1+1+1+0

.SBTTL SOME VARIABLES AND CONSTANTS UNIQUE TO THIS ROUTINE

DJLINK: JSR R5,RCVINT ;UNIT 0 RECEIVER
0
JSR R5,XMTINT ;UNIT 0 TRANSMITTER
0
JSR R5,RCVINT ;UNIT 1 RECEIVER
10
JSR R5,XMTINT ;UNIT 1 TRANSMITTER
10
JSR R5,RCVINT ;UNIT 2 RECEIVER
20
JSR R5,XMTINT ;UNIT 2 TRANSMITTER
20
JSR R5,RCVINT ;UNIT 3 RECEIVER
30
JSR R5,XMTINT ;UNIT 3 TRANSMITTER
30

OPEN=0

XMTTBL: .BLKW 32.

;A TABLE FOR RECORDING CHARACTERS WHICH HAVE
;BEEN TRANSMITTED

RCVTBL: .BLKW 64.

;A TABLE FOR RECORDING CHARACTERS WHICH YOU
;EXPECT TO RECEIVE

SILO: .BLKW 260.

;A BUFFER AREA TO STORE THE DATA RECORDED IN
;THE DJ'S RECEIVER SILO

XQ: .BLKW 4.

;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
;TRANSMIT INTERRUPT IS IN 'PIRQ'

RQ: .BLKW 4.

;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
;RECEIVER INTERRUPT IS IN 'PIRQ'

ERRQ: .BLKW 16.

ERRQ1: OPEN

ERRQ0: OPEN

XCNT: .WORD 101,101,101,101

;NUMBER OF CHARACTERS LEFT TO BE TRANSMITTED DURING

;THIS INTERRUPT. A WORD FOR EACH DEVICE

281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

000224' 000005

000226' 004567 003346
000232' 000000
000234' 004567 003004
000240' 000000
000242' 004567 003332
000246' 000010
000250' 004567 002770
000254' 000010
000256' 004567 003316
000262' 000020
000264' 004567 002754
000270' 000020
000272' 004567 003302
000276' 000030
000300' 004567 002740
000304' 000030

000000

000306' 000040

000406' 000100

000606' 000404

001616' 000004

001526' 000004

001636' 000020

001676' 000000

001700' 000000

001702' 000101 000101 000101

001710' 000101

```

331
332 001712' 000101 000101 000101 RCNT:.WORD 101,101,101,101 ;NUMBER OF CHARACTERS TO BE PROCESSED BY THE
333 001720' 000101 ;RECEIVE ROUTINE. A WORD FOR EACH DEVICE
334
335
336
337 001722' 000200 000200 000200 ENDCNT:.WORD 200,200,200,200 ;NUMBER OF '65 CHARACTER CHUNKS'
338 001730' 000200 ;TO BE PROCESSED BY THE MODULE PER PASS
339
340
341 001732' 177777 177777 177777 XMTSEL: .WORD 177777,177777,177777,177777 ;LINES TO BE RUN
342 001740' 177777
343
344 001742' 177777 177777 177777 XMITEN:.WORD 177777,177777,177777,177777 ;WORD USED TO FILL TRANSMITTER COMMAND
345 001750' 177777 ;REGISTER 16XXX2,CORRESPONDS TO
346 ;A ONE FOR EACH ACTIVE LINE
347
348
349 001752' 000000 000000 000000 XMITSV: .WORD 0,0,0,0 ;SAVE LINES ACTIVATED IN CASE OF HANG
350 001760' 000000
351
352
353
354
355
356
357 001762' 001616' XPNTW: XQ ;POINTERS USED FOR THE BOOKKEEPING OF
358 001764' 001616' XPNTR: XQ ;THE TRANSMIT QUEUE
359 001766' 001626' RPNTW: RQ ;POINTERS USED FOR THE BOOKKEEPING OF
360 001770' 001626' RPNTR: RQ ;THE RECEIVE QUEUE
361
362 001772' 000000 I: 000 ;TEMPORARY STORAGE LOCATION,USED DURING RECEIVE DATA CHE
363 001774' 000001 SELMSK: 1 ;MASK TO CHECK FOR UNIT SELECTED
364 001776' 000000 SELECT: 0
365 002000' 100000 EXPLIN: 100000 ;USED DURING INITIALIZATION,THE ONE REPRESENTS THE
366 ;DONE BIT IN THE EXPECTED RECEIVE DATA TABLE
367 002002' 000200 ENDSEL: 200 ;REFILL OF 'ENDSEL' WORD AFTER ENDPASS
368 002004' 000101 CNTSEL: 101 ;REFILL OF 'RCNT' AND 'XCNT' AFTER AN INTERRUPT SERVICE
369 ;NOT VARIABLE SINCE 65. CHARS REQD TO GET SILO FULL INT
370 002006' 000000 VCT: OPEN ;HOLDS VECTOR ADDRESS DURING SETUP ERROR PRINTOUT
371 002010' 000000 LINK: OPEN ;HOLDS LINK ADDRESS DURING SETUP ERROR PRINTOUT
372 002012' 000000 SCSR: OPEN
373 002014' 000000 ENDFLG: OPEN
374 002016' 000110 TIME: 110 ;NUMBER OF DJ11'S STILL RUNNING
375 002020' 000000 TIMER: 0 ;WATCHDOG TIMER OUTER LOOP VALUE
376 002022' 000000 TIME1: 0
377 002024' 000000 MASK: 0 ;-DATA BIT LENGTH (5-8) MASK
378 002026' 000000 CHAR: 0 ;-TEMP HOLDING FOR MASK & ITERATION COUNT &
379 ;-DATA COMPARISON DURING TEST.
380 002030' 017000 TMETBL: 17000 ;-9600 BPS VARIABLE
381 002032' 011700 11700 ;-4800 BPS ITERATION
382 002034' 006000 6000 ;-2400 BPS COUNTS
383 002036' 005100 5100 ;-1800 BPS PER
384 002040' 003000 3000 ;-1200 BPS LINE
385 002042' 001600 1600 ;-600 BPS SPEEDS
386 002044' 000750 750 ;-300 BPS

```

387	002046'	000350	350	: -150	BPS
388	002050'	000300	300	: -134.5	BPS
389	002052'	000260	260	: -110	BPS
390	002054'	000165	165	: -75	BPS

.SBTTL MODULE CODE

;MODULE INITIALIZATION--THIS ROUTINE INITIALIZES THE DJ11 ASYNCHRONOUS
;XMITTER AND RECEIVER, POINTING INTERRUPT VECTORS TO INTERRUPT HANDLERS (VIA A
;LINKAGE TABLE) AND INITIALIZING PARAMETERS AND CONTROL REGISTERS

399	002056'	012767	000101	176034	START:	MOV	#65.,INTR	:65 INTERRUPTS/ITERATION
400	002064'	012767	000100	176022		MOV	#64.,WDTO	:64 WORDS TO MEM/ITERATION
401	002072'	012767	000100	176016		MOV	#64.,WDFR	:64 WORDS FROM MEM/ITERATION
402	002100'	016767	175710	177670	MOV	DVID1,SELECT		
403	002106'	005067	175724			CLR	ICONT	:MK001
404	002112'	012704	000306'			MOV	#XMTTBL,R4	
405	002116'	005024			1\$:	CLR	(R4)+	:CLEAR XMT TABLE
406	002120'	020427	000406'			CMP	R4,#RCVTBL	
407	002124'	103774				BLO	1\$	
408	002126'	016702	175664			MOV	SR1,R2	: -SET UP TO FIND ITERATION CNT
409	002132'	042702	000037			BIC	#37,R2	: -STRIP UNWANTED BITS
410	002136'	012703	002030'			MOV	#TMETBL,R3	: -POINT TO ITERATION TABLE
411	002142'	006302			2\$:	ASL	R2	: -SHIFT TO FIND COUNT
412	002144'	005702				TST	R2	: -IS THIS IT?
413	002146'	001402				BEQ	3\$	: -YES LEAVE NOW
414	002150'	005723				TST	(R3)+	: -POINT TO NXT IN TBL
415	002152'	000773				BR	2\$	: -TRY AGAIN
416	002154'	011367	177622		3\$:	MOV	(R3),ENDSEL	:SET UP COUNT
417	002160'	016767	175630	177604		MOV	DVID1,I	:MK001
418	002166'	006267	177600		12\$:	ASR	I	:MK001
419	002172'	001403				BEQ	13\$	:MK001
420	002174'	006267	177602			ASR	ENDSEL	:MK001
421	002200'	000772				BR	12\$	:MK001
422	002202'	012700	000004		13\$:	MOV	#4,R0	:MK001
423	002206'	005001				CLR	R1	
424	002210'	016761	177570	001712'	4\$:	MOV	CNTSEL,RCNT(R1)	
425	002216'	016761	177562	001702'		MOV	CNTSEL,XCNT(R1)	
426	002224'	012761	000001	001722'		MOV	#1,ENDCNT(R1)	:MK001
427	002232'	062701	000002			ADD	#2,R1	
428	002236'	005300				DEC	R0	
429	002240'	001363				BNE	4\$	
430	002242'	005767	175570			TST	ICONT	:IS THIS THE FIRST TIME THRU ?
431	002246'	001031				BNE	9\$	:NO BRANCH
432	002250'	016767	177526	175560		MOV	ENDSEL,ICONT	:YES SET THE COUNT TO ICONT
433	002256'	005267	175554			INC	ICONT	:MUST BE GREATER BY ONE BECAUSE
434								:LAST ENDIT CALL IS DONE AFTER
435								:A TEST TO SEE IF ALL DEVS
436								:FUNCTIONED BEFORE WATCHDOG TIMER
437								:TIMED OUT
438	002262'	016701	175526			MOV	DVID1,R1	:SAVE DVC IN R1
439	002266'	001313				BNE	1\$	:IF SET UP OK - BRANCH
440	002270'	104410	000000'			END\$,BEGIN		
441	002274'	006201			11\$:	ASR	R1	:ANY DEVS LEFT ??
442	002276'	001707				BEQ	1\$	:NO - GOODBYE

443	002300	066767	177476	175530		ADD	ENDSEL,ICONT	:YUP - SET ADDITIONAL COUNT
444	002306	062767	000101	175604		ADD	#65.,INTR	:65 MORE TRUPS
445	002314	062767	000100	175572		ADD	#64.,WDTO	:64. MORE WORDS TO
446	002322	062767	000100	175566		ADD	#64.,WDFR	:64. MORE WORDS FROM
447	002330	000761				BR	11\$	:GO CHECK FOR MORE
448	002332	012767	100000	177440	9\$:	MOV	#100000,EXPLIN	:SETUP TO PRESET THE RECEIVE TABLE
449	002340	012704	000406			MOV	#RCVTBL,R4	:GET START ADDRESS OF RECEIVE TABLE
450	002344	016724	177430		5\$:	MOV	EXPLIN,(R4)+	:LOAD RECEIVE TABLE WITH A DONE BIT AND LINE #
451	002350	105267	177425			INCB	EXPLIN+1	:INCREMENT TO THE NEXT LINE NUMBER
452	002354	042767	070000	177416		BIC	#070000,EXPLIN	:MAKE SURE LINE # NEVER GREATER THAN FIFTEEN
453	002362	020427	000606			CMP	R4,#SILO	:SEE IF DONE PRESETTING
454	002366	103766				BLO	5\$	:BRANCH BACK IF NOT
455	002370	012767	177400	177430		MOV	#177400,CHAR	:--PRESET MASK FOR 8 BIT DATA
456	002376	016767	175414	177420		MOV	SR1,MASK	:--GET OPERATOR INPUT
457	002404	042767	177770	177412		BIC	#177770,MASK	:--ELIMINATE ITERATION BAUD COUNT
458	002412	005767	177406			TST	MASK	:8-BIT
459	002416	001405				BEQ	7\$	:MK001
460	002420	006267	177402		6\$:	ASR	CHAR	:--SHIFT TO RIGHT-
461	002424	006267	177374			ASR	MASK	:--SIZE MASK.
462	002430	001373				BNE	6\$	:--DONE?
463	002432	116767	177370	177364	7\$:	MOVB	CHAR,MASK	:--SAVE IT.
464	002440	005767	177332		RESTRT:	TST	SELECT	:ANY DJ'S SELECTED?
465	002444	001404				BEQ	1\$	:BR IF NO
466	002446	032767	177760	177322		BIT	#177760, SELECT	:UNITS OTHER THAN 0-3 SELECTED?
467	002454	001402				BEQ	2\$	:NO- OK
468	002456				1\$:			
469	002456	104410	000000			END\$,BEGIN		:UNIT SELECTION (DVID1) INCORRECT
470	002462	012767	001616	177272	2\$:	MOV	#XQ,XPNTW	:SETUP QUEUE POINTERS
471	002470	012767	001616	177266		MOV	#XQ,XPNTR	
472	002476	012767	001626	177262		MOV	#RQ,RPNTW	
473	002504	012767	001626	177256		MOV	#RQ,RPNTR	
474	002512	012767	001636	177156		MOV	#ERRQ,ERRQI	
475	002520	012767	001636	177152		MOV	#ERRQ,ERRQO	
476	002526	005067	177262			CLR	ENDFLG	
477	002532	012701	000226			MOV	#DJLINK,R1	:GET ADDRESS OF DJ'S LINKING TABLE
478	002536	016700	175246			MOV	VECTOR,R0	:GET VECTOR ADDRESS
479	002542	001403				BEQ	3\$	:BR IF VECTOR = ZERO ;RG03
480	002544	016705	175236			MOV	ADDR,R5	:GET CSR ADDRESS
481	002550	001005				BNE	4\$	:BR IF ADDR NOT = ZERO ;RG03
482	002552				3\$:			
483	002552	104401	000000	004461		MSG\$,BEGIN,NOVEC		:ASCII MESSAGE CALL
484	002560	104410	000000			END\$,BEGIN		:DROP MODULE ;RG03
485	002564	012767	000001	177202	4\$:	MOV	#1,SELMSK	
486	002572	036767	177176	177176	NXTDEV:	BIT	SELMSK,SELECT	:HAS THIS DEVICE BEEN SELECTED?
487	002600	001530				BEQ	1\$	:NO- BRANCH
488	002602	010120				MOV	R1,(R0)+	:YES- LOAD RCV INT VECTOR
489	002604	116720	175202			MOVB	BR1,(R0)+	:LOAD RCV BR
490	002610	105020				CLRB	(R0)+	
491	002612	062701	000006			ADD	#6,R1	
492	002616	010120				MOV	R1,(R0)+	:LOAD XMT INT VECTOR
493	002620	116720	175166			MOVB	BR1,(R0)+	:LOAD XMT BR
494	002624	105020				CLRB	(R0)+	
495	002626	062701	000006			ADD	#6,R1	
496	002632	004767	001340			JSR	PC,TBLSNC	:SYNCH RCV AND XMT TABLES
497	002636	012715	000014			MOV	#14,(R5)	:CLEAR DJ BUFFER AND UARTS
498	002642	012704	000010			MOV	#10,R4	:SETUP WAIT LOOP COUNTER

```

499 002646' 032715 000020      10$: BIT      #20,(R5)          ;'BUSY CLEAR' GONE LOW YET?
500 002652' 001456              BEQ      11$          ;YES- BRANCH
501 002654' 104407 000000'      BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
502 002660' 104407 000000'      BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
503 002664' 005304              DEC      R4          ;WAIT LOOP COUNTER
504 002666' 001367              BNE      10$          ;LOOP IF NOT TIMED OUT
505 002670' 010167 177114      MOV      R1,LINK      ;SAVE INFO DURING ERROR (CAN BE DONE
506 002674' 010067 177106      MOV      R0,VCT      ;SINCE THIS IS A SEQUENTIAL LOOP
507 002700' 010567 177106      MOV      R5,SCSR      ;AND INFO NOT USED ELSEWHERE)
508 002704' 046767 177064 177064 BIC      SELMSK,SELECT ;CLEAR UNIT SELECTED BIT - DROP UNIT
509 002712' 010567 175162      MOV      R5,CSRA      ;SETUP FOR ERROR PRINTOUT
510 002716' 011567 175160      MOV      @R5,ACSR
511 002722' 005015              CLR      @R5          ;DISABLE UNIT
512 002724' 012767 000016 175154 MOV      #16,ERRTYP ;BUSY STUCK
513                                     ;*****
514 002732' 104405 000000' 000000 HRDERS$,BEGIN,NULL ;BUSY CLEAR DIDN'T CLEAR IN TIME
515                                     ;*****
516 002740' 104403 000000' 004422' MSGNS$,BEGIN,DROPM ;ASCII MESSAGE CALL WITH COMMON HEADER
517                                     ;'UNIT DROPPED'
518 002746' 016767 177044 177044 MOV      TIME,TIMER ;RESET TIMER DUE TO PRINTOUT DELAY
519 002754' 005767 177016      TST      SELECT      ;ANY DJ'S STILL SELECTED?
520 002760' 001002              BNE      12$          ;YES- BRANCH
521 002762' 104410 000000'      ENDS$,BEGIN      ;ALL UNITS DROPPED
522 002766' 016701 177016      12$: MOV      LINK,R1
523 002772' 016700 177010      MOV      VCT,R0
524 002776' 016705 177010      MOV      SCSR,R5
525 003002' 062705 000010      ADD      #10,R5
526 003006' 000433              BR      2$
527 003010' 012715 050405      11$: MOV      #50405,(R5) ;TURN ON RCVR AND XMIT ENABLE
528 003014' 010503              MOV      R5,R3
529 003016' 166703 174764      SUB      ADDR,R3
530 003022' 006203              ASR      R3
531 003024' 006203              ASR      R3          ;GET OFFSET (UNIT # TIMES 2)
532 003026' 016363 001732' 001742' MOV      XMTSEL(R3),XMITEN(R3)
533 003034' 016363 001742' 001752' MOV      XMITEN(R3),XMITSV(R3) ;SAVE IN CASE OF HANG TO SHOW ACTIVE LINES
534 003042' 016365 001742' 000004      MOV      XMITEN(R3),4(R5) ;TURN ON XMTR
535 003050' 005267 176740      INC      ENDFLG      ;COUNT UNITS RUNNING
536 003054' 062705 000010      ADD      #10,R5
537 003060' 000406              BR      2$
538 003062' 062701 000014      1$: ADD      #14,R1
539 003066' 062700 000010      ADD      #10,R0
540 003072' 062705 000010      ADD      #10,R5
541 003076' 006367 176672      2$: ASL      SELMSK      ;NEXT UNIT
542 003102' 026727 176666 000020      CMP      SELMSK,#20 ;DONE?
543 003110' 002630              BLT      NXTDEV      ;NO- LOOP
544 003112' 016767 176700 176700      MOV      TIME,TIMER
545 003120' 005067 176676      CLR      TIME1      ;YES- DELAY
546 003124'              3$:
547 003124' 104407 000000'      BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
548 003130' 104407 000000'      BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
549 003134' 005767 176654      TST      ENDFLG      ;DJ11'S DONE?
550 003140' 001004              BNE      4$          ;NO- BRANCH
551 003142' '04413 000000'      ENDITS$,BEGIN      ;SIGNAL END OF ITERATION.
552                                     ;MONITOR SHALL TEST END OF PASS
553 003146' 000167 177266      JMP      RESTRT      ;NEXT ITERATION
554 003152' 005367 176644      4$: DEC      TIME1      ;COUNT TIMER

```

;DRB001

```

555 003156* 001362      BNE      3$              ;TO CATCH
556 003160* 005367 176634 DEC      TIMER              ;HUNG DJ11'S
557 003164* 001357      BNE      3$
558 003166* 104403 000000* 004426* MSGNS$,BEGIN,HUNG$M      ;ASCII MESSAGE CALL WITH COMMON HEADER
559 003174* 016705 174606      MOV      ADDR,R5
560 003200* 012767 000001 176566      MOV      #1,SELMSK
561 003206* 036767 176562 176562 5$: BIT      SELMSK,SELECT
562 003214* 001401      BEQ      6$              ;BRANCH IF NOT SELECTED
563 003216* 005015      CLR      @R5              ;IF SELECTED CLEAR THE CSR
564 003220* 062705 000010      ADD      #10,R5      ;NEXT CSR ADDRESS
565 003224* 006367 176544      ASL      SELMSK      ;NEXT UNIT SELECT BIT
566 003230* 026727 176540 000020      CMP      SELMSK,#20      ;ALL UNITS CHECKED?
567 003236* 002763      BLT      5$              ;BRANCH IF NOT
568 003240* 104410 000000*      ENDS$.BEGIN      ;DJ11 HUNG
569
570      ;ROUTINE TO HANDLE TRANSMITTER INTERRUPTS
571 003244* 010246      XMTINT: MOV      R2,-(SP)      ;SAVE REG 2 ON STACK
572 003246* 011502      MOV      (R5),R2      ;GET OFFSET INTO R2
573 003250* 066702 174532      ADD      ADDR,R2      ;FORM DEVICE ADDRESS
574 003254* 042712 040000      BIC      #040000,(R2)      ;TURN OFF XMT INT ENABLE WHILE YOU SERVICE
575 003260* 011577 176476      MOV      (R5),@XPNTW      ;SAVE OFFSET IN QUEUE
576 003264* 062767 000002 176470      ADD      #2,XPNTW      ;INCREMENT IN POINTER
577 003272* 026727 176464 001626*      CMP      XPNTW,#XQ+10      ;TIME TO WRAP?
578 003300* 001003      BNE      1$              ;NO
579 003302* 012767 001616* 176452      MOV      #XQ,XPNTW      ;YES
580 003310* 012602      MOV      (SP)+,R2      ;RESTORE R2
581 003312* 012605      MOV      (SP)+,R5      ;RESTORE R5
582
583 003314* 000004 000000* 003322*      PIRQS$,BEGIN,XMTRTN      ; QUEUE UP TO CONTINUE AT XMTRTN AND RTI
584
585
586 003322* 017703 176436      XMTRTN: MOV      @XPNTW,R3      ;GET OFFSET
587 003326* 062767 000002 176430      ADD      #2,XPNTW      ;INCREMENT OUT POINTER
588 003334* 026727 176424 001626*      CMP      XPNTW,#XQ+10      ;TIME TO WRAP?
589 003342* 001003      BNE      1$              ;NO
590 003344* 012767 001616* 176412      MOV      #XQ,XPNTW      ;YES
591 003352* 010302      1$: MOV      R3,R2      ;GET READY- COPY OFFSET
592 003354* 066702 174426      ADD      ADDR,R2      ;TO FORM DEVICE ADDRESS
593 003360* 005712      TST      (R2)      ;XMT READY BIT SET?
594 003362* 100422      BMI      XMTOK      ;BR IF OK
595 003364* 010267 174510      MOV      R2,CSRA      ;STORE COMMAND REGISTER ADDRESS
596 003370* 011267 174506      MOV      (R2),ACSR      ;STORE CONTENTS OF COMMAND REGISTER
597 003374* 012767 000011 174504      MOV      #11,ERRTP      ;ILLEGAL INTERRUPT
598
599 003402* 104405 000000* 000000      ;*****
600      HRDRS$,BEGIN,NULL      ;XMIT READY NOT SET ON XMIT INTERRUPT
601      ;*****
602 003410* 016767 176402 176402      MOV      TIME,TIMER      ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
603 003416* 052777 040000 174454      BIS      #040000,@CSRA      ;BETTER REENABLE TRANSMITTERS
604 003424* 104400 000000*      EXITS$,BEGIN      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
605
606 003430* 010300      XMTOK: MOV      R3,R0      ;PREPARE A ONE WORD OFFSET
607 003432* 006200      ASR      R0      ;BEGIN THE OFFSET WITH ONE SHIFT
608 003434* 006200      ASR      R0      ;COMPLETE THE OFFSET WITH 2ND SHIFT
609 003436* 006303      ASL      R3      ;OFFSET OF 20 TIMES UNIT #
610 003440* 116204 000007      MORXMT: MOV      7(R2),R4      ;PUT LINE # IN R4
611 003444* 042704 177760      BIC      #177760,R4      ;PURGE LEFTOVER DATA FROM R4

```

```

611 003450' 060304          ADD      R3,R4          ;FORM UNIQUE LINE/CHAR OFFSET
612 003452' 116467 000306' 176346  MOVB     XMTTBL(R4),CHAR ;TEMP HLD CHAR
613 003460' 146767 176340 176340  BICB     MASK,CHAR      ;FOR MASKING
614 003466' 116762 176334 000006  MOVB     CHAR,6(R2)     ;TRANSMIT CHARACTER
615 003474' 105264 000306'        INCB     XMTTBL(R4)      ;ADVANCE TO NEXT CHARACTER
616 003500' 001016        BNE      DECCNT      ;AVOID NEXT SECTION OF CODE
617                                ;IF SENT CHARS NOT WRAPPED TO ZERO
618 003502' 012701 000001        MOV      #1,R1          ;BIT POSITION OF THE ONE IS SAME AS XMT LINE #
619 003506' 042704 177760        BIC      #177760,R4      ;LEAVE ONLY LINE # IN R4
620 003512' 001403        BEQ      LINOFF          ;LINE 0 IS DONE,TURN IT OFF
621 003514' 006301 1$:          ASL      R1          ;SHIFT XMT BIT TO NEXT XMTTER LINE POSITION
622 003516' 005304        DEC      R4          ;DECREMENT ACTUAL LINE NUMBER
623 003520' 001375        BNE      1$          ;BRANCH BACK IF NOT DONE
624 003522' 040160 001742'  LINOFF: BIC      R1,XMITEN(R0) ;TURN OFF THE INDIVIDUAL XMTTER LINE
625 003526' 001003        BNE      DECCNT      ;ARE ALL LINES DONE NOW?
626 003530' 016060 001732' 001742' MOV      XMTSEL(R0),XMITEN(R0) ;IF YES, RESET ENABLE WORD
627 003536' 005360 001702'  DECCNT: DEC      XCNT(R0)      ;DECREMENT CHARACTER COUNT
628 003542' 003403        BLE      OFF          ;HAVE YOU SENT 65? BR IF YES
629 003544' 005712        TST      (R2)         ;ADDIT ONAL XMTTRS READY
630 003546' 100734        BMI      MORXMT        ;BR IF YES
631 003550' 000407        BR      XMTOUT        ;GO OUT AND WAIT FOR ANOTHER LINE TO INT
632 003552' 005062 000004        OFF:  CLR      4(R2)      ;TURN OFF XMTTER
633 003556' 016760 176222 001702' MOV      CNTSEL,XCNT(R0) ;RESET CHAR COUNT
634 003564' 104400 000000'        EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
635 003570' 052712 040000        XMTOUT: BIS      #040000,(R2) ;ENABLE MORE XMITR INTERRUPTS
636 003574' 104400 000000'        EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
637
638                                ;ROUTINE TO HANDLE RECEIVER INTERRUPTS
639 003600' 010246  RCVINT: MOV      R2,-(SP)          ;SAVE R2 ON STACK
640 003602' 011502        MOV      (R5),R2          ;GET OFFSET
641 003604' 066702 174176        ADD      ADDR,R2      ;FORM DEVICE ADDRESS
642 003610' 042712 010000        BIC      #010000,(R2) ;TURN OFF INTERRUPTS
643 003614' 011577 176146        MOV      (R5),@RPNTW   ;SAVE OFFSET IN QUEUE
644 003620' 062767 000002 176140  ADD      #2,RPNTW    ;INCREMENT WRITE POINTER
645 003626' 026727 176134 001636' CMP      RPNTW,#RQ+10 ;TIME TO WRAP?
646 003634' 001003        BNE      1$          ;NO
647 003636' 012767 001626' 176122  MOV      #RQ,RPNTW   ;YES
648 003644' 012602 1$:          MOV      (SP)+,R2      ;RESTORE R2
649 003646' 012605        MOV      (SP)+,R5          ;RESTORE R5
650
651 003650' 000004 000000' 003656' PIRQS,BEGIN,RCVRTN ; QUEUE UP TO CONTINUE AT RCVRTN AND RTI
652
653                                ;-----
654 003656' 017703 176106  RCVRTN: MOV      @RPNTR,R3    ;GET OFFSET
655 003662' 062767 000002 176100  ADD      #2,RPNTR    ;INCREMENT READ POINTER
656 003670' 026727 176074 001636' CMP      RPNTR,#RQ+10 ;TIME TO WRAP?
657 003676' 001003        BNE      1$          ;NO
658 003700' 012767 001626' 176062  MOV      #RQ,RPNTR   ;YES
659 003706' 010300 1$:          MOV      R3,RC          ;GET OFFSET AGAIN
660 003710' 006200        ASR      R0          ;FORM A ONE WORD
661 003712' 006200        ASR      R0          ;INDEX
662 003714' 010302  PASSED: MOV      R3,R2          ;GET READY
663 003716' 064702 174064        ADD      ADDR,R2      ;TO FORM DEVICE ADDRESS
664 003722' 032712 020000        BIT      #020000,(R2) ;SEE IF BUFFER SILO FULL SET
665 003726' 001022        BNE      CONT1        ;CONTINUE IF OK
666 003730' 010267 174144        MOV      R2,CSRA      ;STORE COMMAND REGISTER ADDRESS

```

```

667 003734' 011267 174142      MOV      (R2),ACSR      ;STORE CONTENTS OF COMMAND REGISTER
668 003740' 012767 000011 174140 MOV      #11,ERRTYP    ;ILLEGAL INTERRUPT
669                                     ;*****
670 003746' 104405 000000' 000000 HRDR$,BEGIN,NULL      ;FALSE INTERRUPT
671                                     ;*****
672 003754' 016767 176036 176036 MOV      TIME,TIMER    ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
673 003762' 052777 010000 174110 BIS      #010000,ACSR  ;BETTER ENABLE RECEIVER BEFORE LEAVING
674 003770' 104400 000000'      EXITS$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
675
676 003774' 010304      CONT1: MOV      R3,R4      ;BEGIN TO FORM
677 003776' 006304      ASL      R4              ;A NEW OFFSET
678 004000' 006304      ASL      R4              ;INTO EXPECTED RECEIVE DATA TABLE
679 004002' 010405      MOV      R4,R5          ;BETTER SAVE IT
680 004004' 006304      ASL      R4              ;SO YOU CAN FORM A SILO OFFSET
681 004006' 006304      ASL      R4
682 004010' 060004      ADD      R0,R4          ;OFFSET = 202(OCTAL) TIMES UNIT#
683 004012' 06270' 000606'      ADD      #SILO,R4 ;FORM SILO BASE OFFSET
684 004016' 010401      MOV      R4,R1          ;REMEMBER FIRST SILO WORD ADDRESS
685 004020' 016224 000002      NOTDUN: MOV     2(R2),(R4)+ ;MOVE DATA FROM DJ TO SILO BUFFER
686 004024' 005360 001712'      DEC      RCNT(R0) ;DECREMENT # OF CHARACTERS RECEIVED
687 004030' 003373      BGT      NOTDUN         ;BR IF NOT RECEIVED 65 YET
688 004032' 016760 175746 001712' MOV      CNTSEL,RCNT(R0) ;RESET RECEIVE COUNT
689 004040' 062705 000406'      ADD      #RCVTBL,R5 ;FORM OFFSET INTO EXP RCV DATA TABLE
690 004044' 010567 175722      MOV      R5,I      ;SAVE IT IN 'I' AS AN INDEX
691                                     ;(RCVTBL BASE ADDRESS FOR THIS UNIT)
692 004050' 116105 000001      NXTCHR: MOVB    1(R1),R5 ;GET LINE NUMBER FROM DATA WORD
693 004054' 042705 177760      BIC      #177760,R5 ;MASK SO HAVE ONLY LINE #
694 004060' 006305      ASL      R5              ;MAKE LINE# EVEN WORD BOUNDARY
695 004062' 066705 175704      ADD      I,R5      ;FORM EXPECTED DATA ADDRESS
696 004066' 011567 175734      MOV      (R5),CHAR ;-TEMP HLDING FOR BIT SIZING
697 004072' 146767 175726 175726 BICB     MASK,CHAR ;-CHECK ONLY STRAPPED BITS
698 004100' 026721 175722      CMP      CHAR,(R1)+ ;-COMPARE EXPECTED VS ACTUAL DATA
699 004104' 001066      BNE      DATERR         ;BR IF BAD DATA
700 004106' 105215      DONE:   INCB      (R5)    ;INCREMENT EXPECTED DATA
701 004110' 020104      CMP      R1,R4          ;HAVF YOU FINISHED ENTIRE BUFFER?
702 004112' 103756      BLO      NXTCHR         ;GO BACK IF NOT
703 004114' 005360 001722'      DEC      ENDCNT(R0)
704 004120' 003011      BGT      1$              ;
705 004122' 012760 000001 001722' MOV      #1,ENDCNT(R0) ;
706 004130' 042712 050405      BIC      #050405,(R2) ;TURN OFF DJ11 ;MK001
707 004134' 005367 175654      DEC      ENDFLG
708 004140' 104400 000000'      EXITS$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
709 004144' 010205      1$:   MOV      R2,R5      ;COPY CSR ADDRESS TO R5
710 004146' 004767 000024      JSR      PC,TBLSNC ;SYNCH RCV AND XMT TABLES
711 004152' 052712 050000      BIS      #50000,(R2) ;NO- ENABLE SILO FULL INT AND XMTR INT
712 004156' 016062 001742' 000004      MOV      XMITEN(R0),4(R2) ;TURN XMITTERS BACK ON
713 004164' 016060 001742' 001752'      MOV      XMITEN(R0),XMITSV(R0) ;SAVE INDICATOR OF LINES ACTIVATED
714 004172' 104400 000000'      EXITS$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
715
716                                     ;SYNCHRONIZE RECEIVER AND TRANSMITTER TABLES FOR ONE UNIT
717                                     ;WHOSE CSR ADDRESS IS IN R5
718 004176' 010546      TBLSNC: MOV      R5,-(SP) ;CSR ADDRESS
719 004200' 166716 173602      SUB      ADDR,(SP)
720 004204' 006316      ASL      (SP)
721 004206' 011646      MOV      (SP),-(SP) ;GET OFFSET (UNIT # TIMES 20 OCTAL)
722 004210' 006316      ASL      (SP)          ;COPY OFFSET
                                           ;OFFSET INTO RCV TABLE (UNIT # TIMES 40)

```

```

723 004212' 062716 000406'      ADD    #RCVTBL,(SP)
724 004216' 062766 000306' 000002  ADD    #XMTTBL,2(SP)
725 004224' 012746 000020      MOV     #16,-(SP)          ;COUNTER
726 004230' 117676 000004 000002 1$:  MOVB   @4(SP),@2(SP)      ;MOVE XMIT TABLE CONTENTS INTO DATA
727                                     ;PORTION OF RCV TABLE
728
729 004236' 005266 000004      INC     4(SP)
730 004242' 062766 000002 000002  ADD     #2,2(SP)
731 004250' 005316      DEC     (SP)
732 004252' 001366      BNE     1$
733 004254' 022626      CMP     (SP)+,(SP)+      ;FIX SP
734 004256' 005726      TST     (SP)+
735 004260' 000207      RTS     PC
736
737                                     ;ERROR HANDLER FOR DATA ERRORS
738 004262' 010267 173612  DATERR: MOV     R2,CSRA          ;STORE COMMAND REGISTER ADDRESS
739 004266' 010567 173610      MOV     R5,SBADR          ;STORE ADDRESS OF TEST DATA
740 004272' 005741      TST     -(R1)          ;RESET R1 BACK TO THE ADDRESS OF THE FAIL DATA
741 004274' 010167 173604      MOV     R1,WASADR          ;STORE ADDRESS OF XMITTED DATA
742 004300' 011567 173602      MOV     (R5),ASB          ;STORE EXPECTED RECEIVE DATA
743 004304' 012167 173600      MOV     (R1)+,AWAS          ;STORE ACTUAL RECEIVED DATA AND RESET R1 TO RIGH
744 004310' 016702 175362      MOV     ERRQ1,R2
745 004314' 146767 175504 173564  BICB    MASK,ASB          ;-MASK OUT STRAPPED OFF BITS
746 004322' 010022      MOV     R0,(R2)+
747 004324' 010122      MOV     R1,(R2)+
748 004326' 010422      MOV     R4,(R2)+
749 004330' 010522      MOV     R5,(R2)+
750 004332' 020227 001676'      CMP     R2,#ERRQ+32.
751 004336' 103402      BLO     1$
752 004340' 012702 001636'      MOV     #ERRQ,R2
753 004344' 010267 175326 1$:  MOV     R2,ERRQ1
754                                     ;*****
755 004350' 104404 000000'      DATERR$,BEGIN          ;DATA ERROR!!!
756                                     ;*****
757 004354' 016767 175436 175436  MOV     TIME,TIMER          ;RESET TIMER DUE TO PRINTOUT DELAY
758 004362' 016702 175312      MOV     ERRQ0,R2
759 004366' 012200      MOV     (R2)+,R0
760 004370' 012201      MOV     (R2)+,R1
761 004372' 012204      MOV     (R2)+,R4
762 004374' 012205      MOV     (R2)+,R5
763 004376' 020227 001676'      CMP     R2,#ERRQ+32.
764 004402' 103402      BLO     2$
765 004404' 012702 001636'      MOV     #ERRQ,R2
766 004410' 010267 175264 2$:  MOV     R2,ERRQ0
767 004414' 016702 173460      MOV     CSRA,R2
768 004420' 000632      BR      DONE          ;NOW CONTINUE CHECKING DATA
769
770 004422' 004432'      DROPM: DROP
771 004424' 177777      -1
772 004426' 004447'      HUNG:  HUNG
773 004430' 177777      -1
774
775 004432' 047125 052111 042040 DROP:  .ASCIIZ /UNIT DROPPED/
776 004440' 047522 050120 042105
777 004446' 000
778 004447' 125 044516 020124 HUNG:  .ASCIIZ /UNIT HUNG/

```

779	004454'	052510	043516	000
780	004461'	045	045104	020101
781	004466'	047515	052504	042514
782	004474'	020072	042526	052103
783	004502'	051117	025440	040440
784	004510'	042104	020122	046040
785	004516'	041517	027123	044040
786	004524'	053101	020105	047516
787	004532'	020124	042502	047105
788	004540'	046040	040517	042504
789	004546'	027104	000045	
790				
791		000001		

NOVEC: .ASCIZ /%DJA MODULE: VECTOR + ADDR LOCS. HAVE NOT BEEN LOADED.%/

.EVEN
.END

ACSP	000102R	263#	510*	596*	667*										
ADDR	000006R	229#	480	529	559	573	592	641	663	719					
ADCR22=	001000	281#													
APTPRE=	000200	281#													
ASB	000106R	267#	742*	745*											
ASTAT	000104R	265#													
AUTO =	000010	281#													
AWAS	000110R	268#	743*												
BEGIN	000000R	226#	440	469	483	484	501	502	514	516	521	547	548	551	
		558	568	583	599	603	634	636	651	670	674	708	714	755	
BIT0 =	000001	281#													
BIT1 =	000002	281#													
BIT10 =	002000	281#													
BIT11 =	004000	281#													
BIT12 =	010000	281#													
BIT13 =	020000	281#													
BIT14 =	040000	281#													
BIT15 =	100000	281#													
BIT2 =	000004	281#													
BIT3 =	000010	281#													
BIT4 =	000020	281#													
BIT5 =	000040	281#													
BIT6 =	000100	281#													
BIT7 =	000200	281#													
BIT8 =	000400	281#													
BIT9 =	001000	281#													
BREAK\$=	104407	281#	501	502	547	548									
BR1	000012R	231#	489	493											
BR2	000013R	232#													
BTOD\$ =	104421	281#													
CAPRES=	000004	281#													
CDATAS=	104412	281#													
CHAR	002026R	378#	455*	460*	463	612*	613*	614	696*	697*	698				
CKHNG\$=	000001	281#													
CLKPRE=	000001	281#													
CLKSP\$=	104422	281#													
CNTSEL	002004R	368#	424	425	633	688									
CONFIG	000056R	251#													
CONT1	003774R	665	676#												
CSRA	000100R	261#	509*	595*	602*	666*	673*	738*	767						
DATCK\$=	104411	281#													
DATERR	004262R	699	738#												
DATERS\$=	104404	281#	755												
DECCNT	003536R	616	625	627#											
DJLINK	000226R	294#	477												
DONE	004106R	700#	768												
DROP	004432R	770	775#												
DROPM	004422R	516	770#												
DVID1	000014R	233#	402	417	438										
ECCMEM=	000100	281#													
ENDCNT	001722R	337#	426*	703*	705*										
ENDFLG	002014R	373#	476*	535*	549	707*									
ENDITS\$=	104413	281#	551												
ENDSEL	002002R	367#	416*	420*	432	443									
END\$ =	104410	281#	440	469	484	521	568								
ERRQ	001636R	323#	474	475	750	752	763	765							

[illegible]

TIME1	002022R	376#	545*	554*					
TMET6L	002030R	380#	410						
TRPDFD=	000023	281#							
USTACK=	000001	281#							
VCT	002006R	370#	506*	523					
VECTOR	000010R	230#	478						
WASADR	000104R	264#	741*						
WDFR	000116R	271#	401*	446*					
WDTO	000114R	270#	400*	445*					
XCNT	001702R	327#	425*	627*	633*				
XFLAG	000005R	228#							
XMITEN	001742R	344#	532*	533	534	624*	626*	712	713
XMITSV	001752R	349#	533*	713*					
XMTINT	003244R	296	300	304	308	571#			
XMTOK	003430R	594	605#						
XMTOUT	003570R	631	635#						
XMTRTN	003322R	583	586#						
XMTSEL	001732R	341#	532	626					
XMTTBL	000306R	313#	404	612	615*	724			
XPNTR	001764R	358#	471*	586	587*	588	590*		
XPNTW	001762R	357#	470*	575*	576*	577	579*		
XQ	001616R	319#	357	358	470	471	577	579	588
.	= 004552R	313#	315#	317#	319#	321#	323#		590

. ABS. 000000 000
 004552 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DE2:XDJALO,DE2:XDJALO.SEQ/NL:TOC/SOL/CRF/DOC=DDXCOM,XDJALO
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 12/4=2.5
 CORE USED: 10K (20 PAGES)

DOCUMENT PAGES: 20