

DUP11

OFF LINE SDLC TRANSMITTER
CZDPBC0

AH-8577C-MC

COPYRIGHT © 75-78

FICHE 1 OF 1

DEC 1978

digital

MADE IN USA

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230
231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280
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
331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	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
411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440
441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460
461	462	463	464	465	466	467	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	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	580
581	582	583	584	585	586	587	588	589	590
591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610
611	612	613	614	615	616	617	618	619	620
621	622	623	624	625	626	627	628	629	630
631	632	633	634	635	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	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	748	749	750
751	752	753	754	755	756	757	758	759	760
761	762	763	764	765	766	767	768	769	770
771	772	773	774	775	776	777	778	779	780
781	782	783	784	785	786	787	788	789	790
791	792	793	794	795	796	797	798	799	800
801	802	803	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	860
861	862	863	864	865	866	867	868	869	870
871	872	873	874	875	876	877	878	879	880
881	882	883	884	885	886	887	888	889	890
891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910
911	912	913	914	915	916	917	918	919	920
921	922	923	924	925	926	927	928	929	930
931	932	933	934	935	936	937	938	939	940
941	942	943	944	945	946	947	948	949	950
951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970
971	972	973	974	975	976	977	978	979	980
981	982	983	984	985	986	987	988	989	990
991	992	993	994	995	996	997	998	999	1000

IDENTIFICATION

PRODUCT CODE: AC-8576C-MC
PRODUCT NAME: CZDPBCO OFLNE SDLC YMTR
DATE: APRIL 1978
MAINTAINER: DIAGNOSTICS

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 DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1.0 ABSTRACT

THE FUNCTION OF THE DUP11 DIAGNOSTICS IS TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERIFY THAT THERE ARE NO MALFUNCTIONS AND THAT ALL OPERATIONS OF THE DUP11 ARE CORRECT IN ITS ENVIRONMENT. PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DUP11 CONFIGURATION BY ANSWERING THE PARAMETER DIALOG (LOAD ADDRESS=200, START ADDRESS=1). ALL QUESTIONS SHOULD BE ANSWERED AND THEN EACH DIAGNOSTIC WILL "OVERLAY" THESE PARAMETERS WHICH ARE STORED IN THE "STATUS TABLE" (SEE SECTION 8.4). THE ALTERNATIVE TO THE PARAMETER DIALOG IS DEFAULT PARAMETERS (SEE SECTION 8.5).

THE DIAGNOSTICS WILL RUN UP TO EIGHT CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DUP11'S IN A CHAIN MODE, I.E., RUNNING THE DIAGNOSTIC COMPLETELY FOR ONE DEVICE BEFORE STARTING THE NEXT.

CZDPB DOES READ/WRITE CHECKING, BUS ADDRESS CHECKING, DEVICE AND BUS RESET TESTS ON THE CONTROL AND STATUS REGISTERS. TESTS ARE MADE TO PROVE THERE IS NO INTERACTION WITHIN AND BETWEEN REGISTERS. THE REGISTERS ARE CHECKED BOTH A BIT AT A TIME AND ALL AT ONCE.

IN ADDITION, THE TRANSMITTER SDLC FUNCTIONS ARE CHECKED IN MAINTENANCE INTERNAL MODE, I.E., CLOCKING OF THE DEVICE IS DONE BY THE PROGRAM. THE DEVICE IS SET UP, A SPECIFIC NUMBER OF HALF-CLOCKS ARE DONE, AND A TEST IS MADE FOR A SIGNIFICANT EVENT.

IN CHECKING DATA, THE SOFTWARE EMULATES THE HARDWARE AND USES THE PROCESSOR CARRY BIT AFTER A ROTATE TO DETERMINE WHAT THE OUTPUT SHOULD BE AT THE TRANSMITTER BIT WINDOW A BIT AT A TIME. THE PROGRAM THEN COMPARES THE EMULATED SOFTWARE BIT TO THE HARDWARE BIT OUTPUT. THE PROCESS IS REPEATED UNTIL ALL DATA IS CHECKED.

THE TRANSMITTER BCC IS CHECKED USING THE CRC.CCITT POLYNOMIAL IN THE SAME WAY AS DATA, WITH ONE EXCEPTION--THE BCC IS CALCULATED FIRST BY THE PROGRAM AND THEN COMPARED TO THE OUTPUT OF THE TRANSMITER.

CURRENTLY THERE ARE THREE OFF-LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO ENSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND ESTABLISH THAT DIAGNOSIS OF THE ERROR WILL BE IMMEDIATE TO DISCOVERING THE

PROBLEM.

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE THREE DIAGNOSTICS ARE:

1. CZDPB [REV] BASIC AND OFFLINE TRANSMITTER TESTS
2. CZDPC [REV] OFFLINE RECEIVER AND MODEM CONTROL AND
INTERRUPT TESTS
3. CZDPD [REV] OFFLINE SDLC AND DECMODE DATA AND FUNCTION
TESTS

NOTE: THERE IS A FOURTH PROGRAM, TAPE CZDPE [REV] WHICH IS A
QUICK-VERIFY TAPE THAT REQUIRES ANSWERING A DIALOG.
ITS FUNCTION IS TO ENABLE THE OPERATOR TO QUICKLY
DETERMINE IF THERE IS A PROBLEM WITH THE DEVICE. SEE
THE DOCUMENTATION IN THAT LISTING FOR MORE INFORMATION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (OR EQUIVALENT)
DUP11

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABS AND
BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1560 ARE
ESPECIALLY TO BE NOTED AND LEFT UNTOUCHED BY THE OPERATOR
AFTER THE DUP11 PARAMETER DIALOG HAS BEEN EXECUTED OR AFTER
THE DEFAULT SETUP HAS BEEN DONE.

3.0 LOADING PROCEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE
ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA
SUCH AS DISK, MAGTAPE, DECTAPE, OR CASSETTE FOLLOW
INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT
SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS = **500

MEMORY	SIZE
	(*)=
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER. (ALSO PLACE
'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW
BE LOADING INTO CPU)

4.0 STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR DEFAULT PARAMETERS ESTABLISHED IN THE TAPE (SEE SECTION 8.5.3 FOR FULL EXPLANATION OF DEFAULT PARAMETERS) OR LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS PREVIOUSLY SET UP BY THE DUP11 PARAMETER DIALOG OR A PREVIOUSLY RUN DUP11 DIAGNOSTIC. SET SWR=1 TO GO THROUGH THE PARAMETER DIALOG. (IT IS NOT NECESSARY TO INPUT NEW PARAMETERS FOR EACH TAPE.) (SECTION 7.2, 8.4 AND 8.5 MAY BE HELPFUL)
- D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

'EXAMPLE'

'MAP OF DUP11 STATUS'

1500	160050	CSR OF FIRST DUP11
1502	000300	VECTOR OF FIRST DUP11
1504	140026	STATUS AND SYNC FOR FIRST DUP11
1506	160060	CSR OF SECOND DUP11
1510	000310	VECTOR OF SECOND DUP11
1512	140026	STATUS AND SYNC FOR SECOND DUP11

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADDRESS 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER. FOR INFORMATION ON THE STATUS TABLE SEE SECTION 8.4 FOR HELP.

IT IS POSSIBLE FOR THE OPERATOR TO MANUALLY CHANGE (TOGGLE IN) THE INFORMATION IN THE MAP TO SUIT A SPECIFIC CONFIGURATION OF DEVICES, BUT THE RESPONSIBILITY FOR VERIFYING THAT INFORMATION RESTS WITH THE OPERATOR.

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

SW 15	SET:	HALT ON ERROR
SW 14	SET:	LOOP ON CURRENT TEST
SW 13	SET:	INHIBIT ERROR PRINT OUT
SW 12	SET:	INHIBIT TYPE OUT/BELL ON ERROR.
SW 11	SET:	INHIBIT ITERATIONS. (QUICK PASS)
SW 10	SET:	ESCAPE TO NEXT TEST ON ERROR
SW 09	SET:	LOOP WITH CURRENT DATA
SW 08	SET:	CATCH ERROR AND LOOP ON IT
SW 07	SET:	USE PREVIOUS STATUS TABLE.
SW 06	SET:	RESERVED
SW 05	SET:	RESERVED
SW 04	SET:	RESERVED

SW 03 SET: SELECT DUP11'S DESIRED ACTIVE
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: ENTER PARAMETERS USING MANUAL DIALOG

SWITCHES 8 THROUGH 15 ARE DYNAMIC AND SHOULD BE USED AS NEEDED IN THE DIAGNOSTIC. SWITCHES 0 THROUGH 3 ARE STATIC (ONLY ARE OPERABLE WHEN THE MONITOR PORTION OF THE TAPE IS RUNNING) AND SHOULD BE SET UP PRIOR TO STARTING OR RESTARTING THE DIAGNOSTIC.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 03 RESELECT DUP11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DUP11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS THREE DUP11S BITS 00, 01, 02 WILL BE SET IN LOC 'DUPACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW03) ALTERS THAT LOCATION. THEREFORE, IF THREE DUP11S ARE IN THE SYSTEM ***DO NOT*** SET SWITCHES GREATER THAN SW 02 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DUP11S THAN HAS BEEN GIVEN INFORMATION ABOUT IN THE PARAMETER PROGRAM.

AS EXPLAINED IN SECTION 1.0, DEVICES SHOULD BE CONSECUTIVELY ADDRESSED, AND CAN BE SELECTED OR DESELECTED USING THIS SWITCH.

METHOD: A. LOAD ADDRESS 200
B. START WITH SW 03=1
C. PROGRAM WILL TYPE MESSAGE
D. SET THE BINARY NUMBER OF DUP11S DESIRED ACTIVE. EXAMPLE: 1=1 DUP11; 3=2 DUP11; 7=3 DUP11; 17=4 DUP11 37=5 DUP11 ETC. PRESS CONTINUE.
E. NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
F. SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST. IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE. THE REASON FOR THIS IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS IN THE MONITOR PORTION OF THE PROGRAM. IT IS POSSIBLE TO LD200, AND RAISE SW01, THEN START, PROVIDED PARAMETERS HAVE BEEN PREVIOUSLY SET UP AS DESCRIBED IN SECTION 4.0. ALSO, WHEN A TEST IS SELECTED, ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA. THIS SWITCH WILL ONLY WORK IF CALL 'SCOPI' IS IN THAT TEST. THE REASON IS THAT MOST TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE, THUS KNOWN AS BLOCK DATA--ONE

PATTERN CAN'T BE SINGLED OUT. (SEE SECTION 4.1.3.B.1)
4.1.3 SWITCH REGISTER PRIORITIES

A) ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GOTO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

B) SCOPE SWITCHES

1. SW 09 - (IF ENABLED BY 'SCOP1') ON AN ERROR. IF AN ASTERISK '*' IS PRINTED IN FRONT OF THE TEST NUMBER (EX. *TEST NO. 10), SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS USUALLY THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0).

IF SW09 IS NOT ENABELED AND THERE IS A *HARD* ERROR (CONSTANT ERROR) SW08 IS BEST. (SW14=0, SW10=0, SW09=0, SW08=1).

FOR INTERMITTENT ERRORS, SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NO ERROR. (SW14=1, SW10=0, SW09=0, SW08=1,0)

2. SW 14 - LOOP ON TEST. WILL LOOP ON TEST UNTIL SWITCH IS LOWERED.
3. SW 11 - INHIBIT ITERATIONS (QUICK PASS). ALLOWS ONLY ONE PASS THROUGH A TEST.

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200. THERE ARE NO OTHER STARTING ADDRESSES FOR THE DUP11 DIAGNOSTICS.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER *ALL* AVAILABLE DUP11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5.0 OPERATING PROCEDURE

WHEN THE PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.

5.1 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1) WHENEVER AN ERROR OCCURS.
2. CLEAR SW 15.
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST), TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT, LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT. IN THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED SINCE THE ERROR PC IS THE HLT+2 LOCATION.

IN SOME TESTS, THERE IS A SUBROUTINE CALL THROUGH A REGISTER (E.G., JSR R1,FLAG). THE SUBROUTINE DOES THE DATA CHECKING FOR THE TEST AND WILL REPORT AN ERROR IF ONE OCCURS. THIS MEANS THAT THE FAILING TEST COULD BE IN ONE PART OF THE LISTING WHILE THE SUBROUTINE THAT FOUND THE ERROR IS IN ANOTHER PART. TO DETERMINE THE PC OF THE FAILING TEST, CHECK THE REGISTER USED BY THE SUBROUTINE. IT WILL CONTAIN THE RETURN ADDRESS OF THE FAILING TEST.

6.0 ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.1 ERROR RECOVERY

IF FOR SOME REASON THE DUP11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN LOOK IN LOCATION 'TSTNO' FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. THIS GIVES THE OPERATOR SOME IDEA AS TO WHAT THE DUP11 WAS DOING AT THE TIME OF THE ERROR.

7.0 RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4 (PLEASE). STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW THE PROGRAM WAS STARTED. ALSO, IT IS IMPORTANT TO USE THE LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETELY ISCLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

DUP11 'PARAMETER DIALOG' MUST BE RUN ONLY ONCE PRIOR TO THE FIRST RUNNING OF ANY DUP11 DIAGNOSTIC IF 'DEFAULT PARAMETERS' ARE NOT USED. IF ONLY DUP11 DIAGNOSTICS WERE LOADED AFTER DUP11 PARAMETER SETUP, AND IF CORE MEMORY HAS NOT BEEN CHANGED, I.E., USE OF DIAGNOSTICS OTHER THAN DUP11 DIAGNOSTICS, AND IF THERE WERE NO DUP11 CONFIGURATION CHANGES, THE DUP11 PARAMETER SETUP NEED NEVER BE RUN AGAIN. HOWEVER, IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DUP11 PARAMETER SETUP MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS. UNDER NORMAL OPERATING CONDITIONS IT SHOULD NOT BE NECESSARY TO INPUT NEW PARAMETERS TO SUBSEQUENT DIAGNOSTICS, UNLESS A CHANGE IS REQUIRED.

NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING THE DEFAULT PARAMETERS WHEN THE PROGRAM IS INITIALLY STARTED WITH SWR=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS FOR THE PURPOSE OF RUNNING MULTIPLE DUP11'S IN CHAIN MODE.

1. CSR ADDRESSES MUST BE CONSECUTIVE.
2. VECTORS ARE CONSECUTIVE IF PARAMETER PROGRAM IS USED.
3. ALL JUMPERS ARE ASSUMED TO BE AS SETUP IN PARAMETER DIALOG.
4. PRIORITY LEVEL MUST BE THE SAME FOR ALL DEVICES.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

ALL DUP11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED, THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO VERIFY NO *HARD* ERRORS AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS--EACH TIME PROGRAM IS STARTED--WILL BE A 'QUICK PASS' UNTIL ALL DUP11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS WITH THE NORMAL ITERATION COUNT (ICOUNT=50), THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS CZDPBC CSR:160050 VEC:300 PASSES:000001 ERRORS:000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

8.3 KEY LOCATIONS

RETURN CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DUP11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) /0000000001000000 MEANS THAT DUP11 NO.05 IS THE DUP11 NOW RUNNING.

DUPCR00-DUPCR07
(1500)-(1560) THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 8 (DECIMAL) DUP11S SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUP11.

DUPACTV EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DUP11 WILL BE TESTED IN TURN. EXAMPLE: (DUPACTV) /0000000000011111 MEANS THAT DUP11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DUPACTV) /0000000000010001 MEANS THAT DUP11 NO. 00,04 WILL BE TESTED.

RXCSR CONTAINS THE RECEIVER CSR OF THE CURRENT DUP11
UNDER TEST.

8.4 MORE ON THAT 'STATUS TABLE' (1500-1560)

'MAP OF DUP11 STATUS'

1500	160050
1502	000300
1504	140000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8
DUP11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE).
EXPLANATION:

1500	160050	THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DUP11 IN THE SYSTEM.
1502	000300	THIS IS VECTOR 'A' FOR THE FIRST DUP11 IN THE SYSTEM.
1504	140026	THIS REPRESENTS SYNC AND SOFTWARE STATUS FOR THE FIRST DUP.

THE BITS ARE AS FOLLOWS:

BIT 15	SET:	OPTIONAL CLEAR JUMPER IN
BIT 14	SET:	TURNAROUND CONNECTOR ON
BIT 13	SET:	
BIT 12	SET:	
BIT 11	SET:	
BIT 10	SET:	
BIT 09	SET:	
BIT 08	SET:	
BIT 07-00		SYNC CHARACTER FOR DECMODE TESTS.

THE ABOVE IS REPEATED FOR EACH DUP11 IN THE SYSTEM. THE TABLE
IS FILLED BY DEFAULT PARAMETERS OR BY THE MANUAL PARAMETER
INPUT AS DESCRIBED PREVIOUSLY. ALSO, IF DESIRED BY THE
USER - THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO
SUIT THE SPECIFIC CONFIGURATION, THUS MAKING EACH DEVICE MAP
DIFFERENT. IT IS THE RESPONSIBILITY OF THE OPERATOR TO VERIFY
THE DATA IN THE MAP.

8.5 METHOD OF DEVELOPING DEFAULT PARAMETERS

8.5.1 DEFAULT PARAMETER ASSUMPTIONS

TOO MUCH HARDWARE WOULD HAVE TO BE ANALYZED TO SIZE THE THE
PARAMETERS. THE PROGRAM MUST ASSUME THE VARIATIONS. THE
RESULT, IF NOT TO YOUR SPECIFIC CONFIGURATION, MAY BE ALTERED
BY HAND (TOGGLE IN) AS DESIRED. IN THIS WAY 95% OF THE
PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

- 1) ALL JUMPERS ARE ASSUMED TO BE IN THE FOLLOWING CONFIGURATION.

	IN	OUT
W1=SECONDARY REC ENABLE	X	
W2=SEC REC DISABLE		X
W3=CLEAR OPTION	X	
W4=SEC TX ENABLE	X	
W5=DSC A CONTROL		X
W6=A+B DS CONTROL	X	
W7=BUS GRANT CONTROL	X	

- 2) THE H325 TURN AROUND CONNECTOR IS ASSUMED TO BE ON.
- 3) THE MANUFACTURING OPTION CSR OF 160050 AND VECTOR OF 770 ARE USED.
- 4) THE BR LEVEL IS ASSUMED TO BE 5.

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4 FOR GREATER DETAIL.

INTRODUCTION TO DUP11 DIAGNOSTIC

```

558      ;*CZDPBC0 /<377>/DUP11 OFLNE SDLC XMTR
559      ;*COPYRIGHT(C) 1975,1978, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
560      ;-----
561
562      ;STARTING PROCEDURE
563      ;LOAD PROGRAM
564      ;LOAD ADDRESS 000200
565      ;PRESS START
566      ;PROGRAM WILL TYPE 'CZDPBC0 /<377>/DUP11 OFLNE SDLC XMTR '
567      ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
568      ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
569      ;AND THEN RESUME TESTING
570
571
572      ;SWITCH REGISTER OPTIONS
573      ;-----
574
575      100000      SW15=100000      ;=1,HALT ON ERROR
576      040000      SW14=40000      ;=1,LOOP ON CURRENT TEST
577      020000      SW13=20000      ;=1,INHIBIT ERROR TYPEOUT
578      010000      SW12=10000      ;=1,DELETE TYPEOUT/BELL ON ERROR.
579      004000      SW11=4000       ;=1,INHIBIT ITERATIONS
580      002000      SW10=2000       ;=1,ESCAPE TO NEXT TEST ON ERROR
581      001000      SW09=1000       ;=1,LOOP WITH CURRENT DATA
582      000400      SW08=400        ;=1,LOOP ON ERROR
583      000200      SW07=200
584      000100      SW06=100
585      000040      SW05=40
586      000020      SW04=20
587      000010      SW03=10
588
589      000004      SW02=4
590      000002      SW01=2
591      000001      SW00=1
                    ;SELECT DUP'S DESIRED ACTIVE
                    ;NOTE:THIS MUST NOT EXCEED ORIGINAL COUNT
                    ;LOCK ON TEST SELECT
                    ;RESTART PROGRAM AT SELECTED TEST
                    ;ENTER PARAMETERS
  
```



```
592
593
594      ;REGISTER DEFINITIONS
595      :-----
596
597      000000      R0=%0      ;GENERAL REGISTER
598      000001      R1=%1      ;GENERAL REGISTER
599      000002      R2=%2      ;GENERAL REGISTER
600      000003      R3=%3      ;GENERAL REGISTER
601      000004      R4=%4      ;GENERAL REGISTER
602      000005      R5=%5      ;GENERAL REGISTER
603      000006      SP=%6      ;PROCESSOR STACK POINTER
604      000007      PC=%7      ;PROGRAM COUNTER
605
606      ;LOCATION EQUIVALENCIES
607      :-----
608
609      177776      PS=177776    ;PROCESSOR STATUS WORD
610      001150      STACK=1150  ;START OF PROCESSOR STACK
611
612      ;INSTRUCTION DEFINITIONS
613      :-----
614
615      005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
616      005726      POP1SP=5726  ;INCREMENT PROCESSOR STACK 1 WORD
617      010046      PUSHRO=10046 ;SAVE R0 ON STACK
618      012600      POPRO=12600  ;RESTORE R0 FROM STACK
619      024646      PUSH2SP=24646;DECREMENT STACK TWICE
620      022626      POP2SP=22626 ;INCREMENT STACK TWICE
621      .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
622
623
624      100000      BIT15=100000
625      040000      BIT14=40000
626      020000      BIT13=20000
627      010000      BIT12=10000
628      004000      BIT11=4000
629      002000      BIT10=2000
630      001000      BIT9=1000
631      000400      BIT8=400
632      000200      BIT7=200
633      000100      BIT6=100
634      000040      BIT5=40
635      000020      BIT4=20
636      000010      BIT3=10
637      000004      BIT2=4
638      000002      BIT1=2
639      000001      BIT0=1
640
641
```

```

642      ;*****
643      ;-----
644      ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
645      ;THE STANDARD 'TRAP CATCHER' IS PLACED
646      ;BETWEEN ADDRESS 0 TO ADDRESS 776.
647      ;IT LOOKS LIKE 'PC+2 HALT'.
648      ;-----
649      ;*****
650
651      000000      .=0
652      ;STANDARD INTERRUPT VECTORS
653      ;-----
654
655      000024      .=24
656      000024      005050      .PFAIL      ;POWER FAIL HANDLER
657      000026      000340      340          ;SERVICE AT LEVEL 7
658      000030      004350      .HLT          ;ERROR HANDLER
659      000032      000340      340          ;SERVICE AT LEVEL 7
660      000034      004316      .TRPSRV      ;GENERAL HANDLER DISPATCH SERVICE
661      000036      000340      340          ;SERVICE AT LEVEL 7
662
663      000040      000000      .=40
664      000042      000000      0            ;SAVE FOR ACT-11 OR DDP2
665      000044      000000      0            ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
666      000046      003104      0            ;SAVE FOR ACT-11 OR DDP2
667      000052      000052      $ENDAD      ;FOR USE WITH ACT-11 OR DDP2
668      000052      000000      .=52
669      000052      000000      0            ;ACT-11 PROGRAM CHARACTERISTICS
670
671      000174      000174      .=174
672      000176      000000      DISPREG:0    ;SOFTWARE DISPLAY REGISTER
673      000200      000200      SWREG: 0     ;SOFTWARE SWITCH REGISTER
674      000200      000137      001562      .=200
675      000200      000137      001562      JMP      .START      ;GO TO START OF PROGRAM
676
677      001000      001000      001000      .=1000
678      001000      005377      055103      050104      MTITLE: .ASCIZ <377><12>/CZDPBC0 /<377>/DUP11 OF LNE SDLC XMTR /<377>
679      001200      001200      001200      .=1200
680      001200      001200      001200      ;SWR AND LIGHTS
681      001200      001200      001200      ;-----
682
683      001200      177570      177570      DISPLAY:      ;11/45 CONSOLE LIGHTS
684      001202      177570      177570      SWR:         ;INDIRECT POINTER TO SWITCH REGISTER
685
686      001200      177570      177570      ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
687      001200      177570      177570      ;-----
688
689      001204      177560      177560      TKCSR:        ;TELETYPE KEYBOARD CONTROL REGISTER
690      001206      177562      177562      TKDBR:        ;TELETYPE KEYBOARD DATA BUFFER
691      001210      177564      177564      TPCSR:        ;TELEPRINTER CONTROL REGISTER
692      001212      177566      177566      TPDBR:        ;TELEPRINTER DATA BUFFER
693
694      001212      177566      177566      ;PROGRAM CONTROL PARAMETERS
695      001212      177566      177566      ;-----
696
  
```


PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

697	001214	000000	RETURN: 0	:SCOPE ADDRESS FOR LOOP ON TEST
698	001216	000000	NEXT: 0	:ADDRESS OF NEXT TEST TO BE EXECUTED
699	001220	000000	LOCK: 0	:ADDRESS FOR LOCK ON CURRENT DATA
700	001222	000001	ICOUNT: 1	:NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
701	001224	000000	LPCNT: 0	:NUMBER OF ITERATIONS COMPLETED
702	001226	000000	TSTNO: 0	:NUMBER OF TEST IN PROGRESS
703	001230	000000	PASCNT: 0	:NUMBER OF PASSES COMPLETED
704	001232	000000	ERRCNT: 0	:TOTAL NUMBER OF ERRORS
705	001234	000000	LASTERR: 0	:PC OF LAST ERROR CALL
706				
707			:PROGRAM VARIABLES	
708			:-----	
709				
710	001236	000000	TEMP1: 0	:TEMPORARY STORAGE
711	001240	000000	TEMP2: 0	:TEMPORARY STORAGE
712	001242	000000	TEMP3: 0	:TEMPORARY STORAGE
713	001244	000000	TEMP4: 0	:TEMPORARY STORAGE
714	001246	000000	TEMP5: 0	:TEMPORARY STORAGE
715	001250	000000	SAVR0: 0	:R0 STORAGE
716	001252	000000	SAVR1: 0	:R1 STORAGE
717	001254	000000	SAVR2: 0	:R2 STORAGE
718	001256	000000	SAVR3: 0	:R3 STORAGE
719	001260	000000	SAVR4: 0	:R4 STORAGE
720	001262	000000	SAVR5: 0	:R5 STORAGE
721	001264	000000	SAVSP: 0	:STACK POINTER STORAGE
722	001266	000000	SAVPC: 0	:PROGRAM COUNTER STORAGE
723				
724	001270	000000	SAVR0A: 0	:R0 STORAGE
725	001272	000000	SAVR1A: 0	:R1 STORAGE
726	001274	000000	SAVR2A: 0	:R2 STORAGE
727	001276	000000	SAVR3A: 0	:R3 STORAGE
728	001300	000000	SAVR4A: 0	:R4 STORAGE
729	001302	000000	SAVR5A: 0	:R5 STORAGE
730	001304	000000	SAVSPA: 0	:STACK POINTER STORAGE
731	001306	000000	SAVPCA: 0	:PROGRAM COUNTER STORAGE
732				
733	001310	000001	DUPACTV: .BLKB 1	:DUP11'S SELECTED ACTIVE.
734	001311	000001	DUPNUM: .BLKB 1	:OCTAL NUMBER OF DUP11'S.
735	001312	000001	SAVACT: .BLKB 1	:ORIGINAL ACTV. DEVICES.
736	001313	000001	SAVNUM: .BLKB 1	:WORKABLE NUMBER.
737	001314	000001	RUN: .BLKB 1	:POINTER ONE PAST RUNNING DEVICE.
738		001316	.EVEN	
739	001316	001500	CREAM: DUP.MAP	:TABLE POINTER.

```

740
741                                     :CONTRCL REGISTER DEFINITIONS
742                                     :-----
743                                     :RXCSR BIT DEFINITIONS
744         100000      DSCA=BIT15      :DATA SET CHANGE A
745         040000      RING=BIT14      :RING
746         020000      CTS=BIT13      :CLR TO SEND
747         010000      CARDET=BIT12    :CARRIER DETECT
748         004000      RECACT=BIT11    :REC ACTIVE
749         002000      SRD=BIT10      :SEC REC DATA
750         001000      DSR=BIT9       :DATA SET RDY
751         000400      STPSYN=BIT8     :STRIP SYNC
752         000200      RXDONE=BIT7     :REC DONE
753         000100      RINTEN=BIT6     :REC INTR ENABLE
754         000040      DSINTE=BIT5     :DSC INTR ENABLE
755         000020      RCVEN=BIT4      :REC ENABLE
756         000010      STD=BIT3       :SEC XMIT DATA
757         000004      RTS=BIT2       :REQ TO SEND
758         000002      DTR=BIT1       :DATA TERM RDY
759         000001      DSCB=BIT0      :DATA SET CHANGE B
760                                     :RXDBUF BIT DEFINITIONS
761         100000      RXDERR=BIT15    :REC DATA ERROR
762         040000      OVRUN=BIT14     :OVERRUN ERROR
763         010000      CRCERR=BIT12    :CRC ERROR
764         002000      RABORT=BIT10    :REC ABORT
765         001000      REOM=BIT9      :REC END OF MESSAGE
766         000400      RSOM=BIT8      :REC START OF MESSAGE
767                                     :PARCSR BIT DEFINITIONS
768         100000      DECMOD=BIT15    :DEC MODE (DDCMP)
769         001000      CRCEN=BIT9      :CRC ENABLE
770         010000      PRISEC=BIT12    :PRI/SEC SELECT
771                                     :TXCSR BIT DEFINITIONS
772         100000      TXDLAT=BIT15    :TX DATA LATE
773         040000      MTDATA=BIT14    :MAINT DATA OUT
774         020000      CLK=BIT13      :CLK
775         010000      MMODEB=BIT12    :MAINT MODE B
776         004000      MMODEA=BIT11    :MAINT MODE A
777         002000      BITW=BIT10     :BIT WINDOW INPUT
778         001000      TXACT=BIT9     :TX ACTIVE
779         000400      MRESET=BIT8     :MASTER RESET
780         000200      TXDONE=BIT7     :XMIT DONE
781         000100      TXINTE=BIT6     :XMIT DONE INTR ENABLE
782         000020      SEND=BIT4      :SEND
783         000010      HDXEN=BIT3     :HDX/FDX
784                                     :TXCSR WRD DEFINITIONS
785         000000      USER=0         :USER MODE
786         014000      MMODE=14000    :MAINT INT MODE
787         010000      MEXT=10000     :MAINT EXT MODE
788         004000      SYSTST=4000    :SYSTEM TEST MODE
789
790                                     :TXDBUF BIT DEFINITIONS
791                                     :-----
792         100000      RCRC7T=BIT15
793         040000      RCRCIN=BIT14
794         020000      TCRC7T=BIT13
795         010000      TCRCIN=BIT12
  
```


CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 19

SEQ 0018

CZDPBC.P11 02-MAY-78 08:23

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

796	004000	TIMER=BIT11	;MAINTENANCE TIMER
797	002000	TABORT=BIT10	;TRANSMIT ABORT
798	001000	TEOM=BIT9	;TRANSMIT END OF MESSAGE
799	000400	TSOM=BIT8	;TRANSMIT START OF MESSAGE

800

801

802

;MISC. PROGRAM DEFINITIONS

803 001320 000000

PRIRTY: .WORD 0

804 001322 000001

TCNFLG: .BLKB 1

805 001323 000001

OPCLRJ: .BLKB 1

806 001324 000000

DATA: .WORD 0

807 001326 000000

SHIFTS: .WORD 0

808 001330 000000

MIND: .WORD 0

809 001332 000000

FLAG: .WORD 0

810 001334 000001

STJMFL: .BLKW 1

811 001336 000001

SRJMFL: .BLKW 1

812

813

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
814
815
816           :PROGRAM CONTRCL FLAGS
817           :-----
818 001340      000      INIFLG: .BYTE      0      ;PROGRAM INITIALIZATION FLAG
819 001341      000      ERRFLG: .BYTE      0      ;ERROR OCCURED FLAG
820 001342      000      LOKFLG: .BYTE      0      ;LOCK ON CURRENT TEST FLAG
821 001343      000      QV.FLG: .BYTE      0      ;QUICK VERIFY FLAG.
822                                     ;ON FIRST PASS OF EACH DUP11 ITERATIONS
823                                     ;WILL BE SUPPRESSED
824
825           .EVEN
826           $Y=0
827
828           :DEFINITIONS FOR TRAP SUBROUTINE CALLS
829           :POINTERS TO SUBROUTINES CAN BE FOUND
830           :IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
831
832           :*****
833           :-----
834 001344      104400      .TRPTAB:
835           SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER
836           .SCOPE
837 001344      003160      SCOP1=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
838           .SCOP1
839 001346      003312      TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
840           .TYPE
841 001350      003336      INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
842           .INSTR
843 001352      003412      INSTER=TRAP+4      ;CALL TO INPUT ERROR HANDLER
844           .INSTER
845 001354      003516      PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE
846           .PARAM
847 001356      003536      SAV05=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE
848           .SAV05
849 001360      003736      RES05=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE
850           .RES05
851 001362      003776      CONVRT=TRAP+10      ;CALL TO DATA OUTPUT ROUTINE
852           .CONVRT
853 001364      004030      CNVRT=TRAP+11      ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
854           .CNVRT
855 001366      004034      PKCLK=TRAP+12      ;CALL TO CLOCK ROUTINE
856           .PKCLK
857 001370      005006      SETFLG=TRAP+13      ;CALL TO TELETYPE INPUT ROUTINE
858           .SETFLG
859
860           :-----
861           :*****
```


PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

861                                ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
862
863 001374 000000    DUPRVC: 0      ;POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
864 001376 000000    DUPRPS: 0      ;POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
865 001400 000000    DUPTVC: 0      ;POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
866 001402 000000    DUPTPS: 0      ;POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
867 001404 000000    RXCSR: 0       ;POINTER TO DUP11 RECEIVER STATUS REGISTER
868 001406 000000    RXDBUF: 0      ;POINTER TO DUP11 RECEIVER DATA BUFFER
869 001410 000000    PARCSR: 0      ;POINTER TO DUP11 PARAMETER STATUS REGISTER
870 001412 000000    TXCSR: 0       ;POINTER TO DUP11 TRANSMITTER STATUS REGISTER
871 001414 000000    TXDBUF: 0      ;POINTER TO DUP11 TRANSMITTER DATA BUFFER
872 001416 000000    DUPSEC: 0      ;POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
873 001420 000000    HUPPSR: 0      ;POINTER TO PARAMETER STATUS HIGH BYTE
874 001422 000000    HUPRBF: 0      ;POINTER TO RECEIVER BUFFER HIGH BYTE
875 001424 000000    HUPRCR: 0      ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
876 001426 000000    HUPTBF: 0      ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
877 001430 000000    HUPTCR: 0      ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
878
879
880                                ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
881                                ;-----
882
883 001432 000      MASK.A: .BYTE 000    ;LAST CHAR TO TEST AND PARITY MASK
884
885 001433 010      CLK.A: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
886
887 001434 000000    L00.00: 000000      ;PARAMETERS
888
  
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
889                                     ;DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS
890                                     ;-----
891
892                                     . =1500
893 001500 DUP.MAP:
894 001500 000001 DUPCR0: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
895 001502 000001 DUPTR0: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 0
896 001504 000001 DUPO.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 0
897
898 001506 000001 DUPCR1: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
899 001510 000001 DUPTR1: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 1
900 001512 000001 DUP1.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 1
901
902 001514 000001 DUPCR2: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
903 001516 000001 DUPTR2: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 2
904 001520 000001 DUP2.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 2
905
906 001522 000001 DUPCR3: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
907 001524 000001 DUPTR3: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 3
908 001526 000001 DUP3.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 3
909
910 001530 000001 DUPCR4: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
911 001532 000001 DUPTR4: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 4
912 001534 000001 DUP4.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 4
913
914 001536 000001 DUPCR5: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
915 001540 000001 DUPTR5: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 5
916 001542 000001 DUP5.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 5
917
918 001544 000001 DUPCR6: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
919 001546 000001 DUPTR6: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 6
920 001550 000001 DUP6.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 6
921
922 001552 000001 DUPCR7: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
923 001554 000001 DUPTR7: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 7
924 001556 000001 DUP7.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 7
925
926 001560 000000 DUP.END: 000000
927
928
929
930
931
```


[illegible]

DEFINITIONS

- A- OPTIONAL CLEAR JUMPER IN=1
B- TURNAROUND CONNECTOR GN=1
C-
D-

```

951
952      ;PROGRAM INITIALIZATION
953      ;LOCK OUT INTERRUPTS
954      ;SET UP PROCESSOR STACK
955      ;SET UP POWER FAIL VECTOR
956      ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
957      ;TYPE TITLE MESSAGE
958
959 001562 012737 000340 177776 .START: MOV    #340,PS      ;LOCK OUT INTERRUPTS
960 001570 012706 001150      MOV    #STACK,SP      ;SET UP STACK
961 001574 012737 005050 000024      MOV    #.PFAIL,@#24      ;SET UP POWER FAIL VECTOR
962 001602 113737 001311 001313      MOV    DUPNUM,SAVNUM      ;SAVE NUMBER OF DEVICES IN SYSTEM
963 001610 005037 001230      CLR     PASCNT      ;CLEAR PASS COUNT
964 001614 105037 001341      CLRB   ERRFLG      ;CLEAR ERROR FLAG
965 001620 105037 001343      CLRB   QV.FLG      ;ZERO QUICK VERIFY FLAG
966 001624 012737 001500 001316      MOV    #DUP.MAP,CREAM      ;GET MAP POINTER.
967 001632 112737 000001 001314      MOV    #1,RUN      ;POINT POINTER TO FIRST DEVICE.
968 001640 005037 001232      CLR     ERRCNT      ;CLEAR ERROR COUNT
969 001644 005037 001234      CLR     LSTERR      ;CLEAR LAST ERROR POINTER
970 001650 012737 000001 001226      MOV    #1,TSTNO      ;SET UP FOR TEST 1
971 001656 012737 001562 001214      MOV    #.START,RETURN      ;SET UP FOR POWER FAIL BEFORE
972                                     ;TESTING STARTS
973 001664 013746 000006      MOV    @#6,-(SP)      ;SAVE CURRENT VECTORS
974 001670 013746 000004      MOV    @#4,-(SP)      ;
975 001674 012737 001710 000004      MOV    #12$,@#4      ;SETUP FOR TIMEOUT
976 001702 005777 177274      TST    @SWR      ;REFERENCE HARDWARE SWITCH REG
977 001706 000407      BR     13$      ;BR IF IT EXISTS
978 001710 012737 000176 001202 12$: MOV    #SWREG,SWR      ;POINT TO SOFT SWR
979 001716 012737 000174 001200      MOV    #DISPREG,DISPLAY      ;POINT TO SOFT DISPLAY REG
980 001724 022626      CMP     (SP)+,(SP)+      ;ADJUST STACK
981 001726 012637 000004      MOV    (SP)+,@#4      ;RESTORE VECTORS
982 001732 012637 000006      MOV    (SP)+,@#6      ;
983 001736 105737 001340      TSTB   INIFLG      ;HAS INITIALIZATION BEEN PERFORMED
984 001742 001401      BEQ     11$
985 001744 000410      BR     6$
986 001746 022737 003104 000042 11$: CMP    #SENDAD,@#42      ;IF ACT-11 AUTO MODE,
987 001754 001404      BEQ     6$      ;DON'T TYPE ID
988 001756 104402 001000      TYPE    ,MTITLE      ;TYPE TITLE MESSAGE
989 001762 105137 001340      COMB   INIFLG      ;IF NOT SET FLAG AND DO
990 001766 105777 177210      TSTB   @SWR      ;BIT7=1??
991 001772 100002      BPL     10$
992 001774 000137 002520      JMP     1$
993 002000      10$:
994 002000 032777 000001 177174      BIT    #SW00,@SWR      ;ENTER PARAMETERS
995 002006 001002      BNE     .+6      ;YES
996 002010 000137 002360      JMP     21$      ;NO
997 002014 105137 001332      COMB   FLAG
998 002020 112737 000001 001340      MOV    #1,INIFLG      ;SET TO MANUAL ENTRY
999 002026 012700 001500      MOV    #DUP.MAP,R0      ;CLR MAP
1000 002032 005020      CLR     (R0)+
1001 002034 020027 001560      CMP    R0,#DUP.END      ;DONE WITH MAP?
1002 002040 001374      BNE     68$      ;BR IF NO
1003 002042 104403      INSTR  ;OUTPUT MESSAGE & GET INPUT STRING
1004 002044 005474      MCSR   ;MESSAGE
1005 002046 104405      PARAM  ;CONVERT STRING
1006 002050 160000      160000 ;LOW LIMIT

```


1007	002052	175500		175500	:HIGH LIMIT
1008	002054	001500		DUPCR0	:STORE AT THIS LOCATION
1009	002056	001	.BYTE	1	:MASK
1010	002057	001	.BYTE	1	:HOW MANY TIMES + 2
1011	002060	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1012	002062	005513		MVEC	:MESSAGE
1013	002064	104405		PARAM	:CONVERT STRING
1014	002066	000300		300	:LOW LIMIT
1015	002070	000770		770	:HIGH LIMIT
1016	002072	001502		DUPTR0	:STORE AT THIS LOCATION
1017	002074	001	.BYTE	1	:MASK
1018	002075	001	.BYTE	1	:HOW MANY TIMES + 2
1019	002076	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1020	002100	005703		MPAR	:MESSAGE
1021	002102	104405		PARAM	:CONVERT STRING
1022	002104	000004		4	:LOW LIMIT
1023	002106	000007		7	:HIGH LIMIT
1024	002110	001240		TEMP2	:STORE AT THIS LOCATION
1025	002112	000	.BYTE	0	:MASK
1026	002113	001	.BYTE	1	:HOW MANY TIMES + 2
1027	002114	013737	001240 001320	MOV	TEMP2,PRIITY ;SAVE PRIORITY
1028	002122	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1029	002124	005650		MTOTAL	:MESSAGE
1030	002126	104405		PARAM	:CONVERT STRING
1031	002130	000001		1	:LOW LIMIT
1032	002132	000010		8.	:HIGH LIMIT
1033	002134	001236		TEMP1	:STORE AT THIS LOCATION
1034	002136	000	.BYTE	0	:MASK
1035	002137	001	.BYTE	1	:HOW MANY TIMES + 2
1036	002140	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1037	002142	005526		MJMPR	:MESSAGE
1038	002144	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1039	002146	001323		OPCLRJ	:THIS FLAG
1040	002150	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1041	002152	005601		MTCN	:MESSAGE
1042	002154	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1043	002156	001322		TCNFLG	:THIS FLAG
1044	002160	105737	001322	TSTB	TCNFLG
1045	002164	001410		BEQ	71\$
1046	002166	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1047	002170	005727		MSTJM	:MESSAGE
1048	002172	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1049	002174	001334		STJMFL	:THIS FLAG
1050	002176	104403		INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1051	002200	005762		MSRJM	:MESSAGE
1052	002202	104413		SETFLG	:SET FLAG BASED UPON INPUT STRING
1053	002204	001336		SRJMFL	:THIS FLAG
1054	002206	105737	001323	TSTB	OPCLRJ
1055	002212	001403		BEQ	69\$
1056	002214	052737	100000 001504	BIS	#BIT15,DUP0.A
1057	002222	105737	001322	TSTB	TCNFLG
1058	002226	001403		BEQ	70\$
1059	002230	052737	040000 001504	BIS	#BIT14,DUP0.A
1060	002236	112737	000001 001312	MOVB	#1,SAVACT
1061	002244	113737	001236 001311	MOVB	TEMP1,DUPNUM
1062	002252	113737	001236 001313	MOVB	TEMP1,SAVNUM

CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 26
 CZDPBC.P11 02-MAY-78 08:23 PROGRAM INITIALIZATION AND START UP.

SEQ 0025

1063	002260	005337	001236	65\$:	DEC	TEMP1	
1064	002264	001404			BEQ	64\$	
1065	002266	000261			SEC		
1066	002270	106137	001312		ROLB	SAVACT	
1067	002274	000771			BR	65\$	
1068	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2 ;# OF TIMES
1069	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV
1070	002312	000241				CLC	
1071	002314	106037	001240			RORB	TEMP2
1072	002320	012700	001500			MOV	#DUPCRO,R0
1073	002324	012701	001506			MOV	#DUPCR1,R1
1074	002330	000241			67\$:	CLC	
1075	002332	106037	001240			RORB	TEMP2
1076	002336	103051				BCC	66\$
1077	002340	012011				MOV	(R0)+,(R1)
1078	002342	062721	000010			ADD	#10,(R1)+ ;CSR
1079	002346	012011				MOV	(R0)+,(R1)
1080	002350	062721	000010			ADD	#10,(R1)+ ;VECTOR
1081	002354	012021				MOV	(R0)+,(R1)+ ;PARAMETERS
1082	002356	000764				BR	67\$
1083	002360	012700	001500		21\$:	MOV	#DUP.MAP,R0 ;SETUP TO CLEAR MAP
1084	002364	005020			20\$:	CLR	(R0)+ ;CLEAR
1085	002366	020027	001560			CMP	R0,#DUP.END ;CHECK FOR FINISH
1086	002372	001374				BNE	20\$;BR IF MORE TO GO
1087	002374	012700	001500			MOV	#DUP.MAP,R0 ;SETUP TO DEFAULT
1088	002400	012710	160050			MOV	#160050,(R0) ;LOAD CSR
1089	002404	012760	000770	000002		MOV	#770,2(R0) ;LOAD VECTOR
1090	002412	012760	140026	000004		MOV	#140026,4(R0) ;LOAD PARAMETERS AND SYNC
1091	002420	112737	000005	001320		MOVB	#5,PRIORITY ;LOAD PRIORITY
1092	002426	012700	000001			MOV	#1,R0 ;SAVE CORE THIS WAY
1093	002432	110037	001310			MOVB	R0,DUPACTV ;PRESET PROGRAM CONTROLS
1094	002436	110037	001311			MOVB	R0,DUPNUM ;DITTO
1095	002442	110037	001312			MOVB	R0,SAVACT ;DITTO
1096	002446	110037	001313			MOVB	R0,SAVNUM ;DITTO
1097	002452	110037	001322			MOVB	R0,TCNFLAG ;DITTO
1098	002456	110037	001323			MOVB	R0,OPCLRJ ;DITTO
1099	002462				66\$:		
1100	002462	104402	006015		16\$:	TYPE	,XHEAD ;TYPE HEADER
1101	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1 ;SET POINTER
1102	002474	017737	176536	001240	5\$:	MOV	@TEMP1,TEMP2 ;SET DATA
1103	002502	001406				BEQ	1\$;ALL DONE WITH DATA
1104	002504	104410				CONVRT	
1105	002506	006044				XSTATQ	
1106	002510	062737	000002	001236		ADD	#2,TEMP1 ;UPDATE POINTER
1107	002516	000766				BR	5\$
1108	002520	032777	000001	176454	1\$:	BIT	#SW00,@SWR
1109	002526	001405				BEQ	7\$
1110	002530	005737	001332			TST	FLAG
1111	002534	001002				BNE	7\$
1112	002536	000137	002000			JMP	10\$
1113	002542	005037	001332		7\$:	CLR	FLAG
1114	002546	005737	000042			TST	@#42 ;IS PROGRAM RUNNING UNDER MONITOR
1115	002552	001030				BNE	3\$;BR IF YES
1116	002554	032777	000010	176420		BIT	#SW03,@SWR ;SELECT SPECIFIC DEVICES??
1117	002562	001424				BEQ	3\$;BR IF NO.
1118	002564	104402	005414			TYPE	,MNEW ;TYPE THE MESSAGE.

1119	002570	005000			CLR	R0		;ZERO DATA LIGHTS
1120	002572	000000			HALT			;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1121	002574	127737	176402	001312	CMPB	@SWR,SAVACT		;IS THE NUMBER VALID?
1122	002602	101404			BLOS	2\$		;BR IF NUMBER IS OK.
1123	002604	104402	005255		TYPE	,MERR3		;TELL USER OF INVALID NUMBER.
1124	002610	000000			HALT			;STOP EVERY THING.
1125	002612	000776			BR	.-2		;RESTART THE PROGRAM AGAIN.
1126	002614	117737	176362	001310	2\$:	MOVB	@SWR,DUPACTV	;GET NEW DEVICE PATTERN
1127	002622	113700	001310		MOVB	DUPACTV,R0		;SHOW THE USER WHAT HE SELECTED.
1128	002626	042700	177400		BIC	#^C<377>,R0		;USE ONLY LOW BYTE.
1129	002632	000000			HALT			;CONTINUE DYNAMIC SWITCHES.
1130	002634	012700	000300		3\$:	MOV	#300,R0	;PREPARE TO CLEAR THE FLOATING
1131	002640	012701	000302		MOV	#302,R1		;VECTOR AREA. 300-776
1132	002644	010120			4\$:	MOV	R1,(R0)+	;START PUTTING 'PC+2 - HALT''
1133	002646	005021			CLR	(R1)+		;IN VECTOR AREA.
1134	002650	022021			CMP	(R0)+,(R1)+		;POP POINTERS
1135	002652	022700	001000		CMP	#1000,R0		;ALL DONE??
1136	002656	001372			BNE	4\$		;BR IF NO.
1137								
1138								
1139								
1140								
1141	002660	012737	000340	177776	.BEGIN:	MOV	#340,PS	;LOCK OUT INTERRUPTS
1142	002666	012706	001150		MOV	#STACK,SP		;SET UP STACK
1143	002672	005737	000042		TST	@#42		;IS PROGRAM UNDER MONITOR CONTROL
1144	002676	001023			BNE	2\$		;BR IF YES
1145	002700	032777	000004	176274	BIT	#BIT2,@SWR		;CHECK FOR LOCK ON TEST
1146	002706	001411			BEQ	1\$		;BR IF NO LOCK DESIRED.
1147	002710	104402	005313		TYPE	,MLOCK		;TYPE LOCK SELECTED.
1148	002714	012737	000240	003174	MOV	#NOP,TTST		;ADJUST SCOPE ROUTINE.
1149	002722	012737	000240	003176	MOV	#NOP,TTST+2		;SET UP TO LOCK
1150	002730	000406			BR	2\$		;CONTINUE ALONG.
1151	002732	013737	003306	003174	1\$:	MOV	BRW,TTST	;PREPARE NORMAL SCOPE ROUTINE
1152	002740	013737	003310	003176	MOV	BRX,TTST+2		;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1153	002746	012737	006226	001214	2\$:	MOV	#CYCLE,RETURN	;START AT 'CYCLE' FIND WHICH DEVICE TO TEST
1154	002754	104402	005203		TYPE	,MR		;TYPE R
1155	002760	000177	176230		JMP	@RETURN		;START TESTING

```

1156                                     ;END OF PASS
1157                                     ;TYPE NAME OF TEST
1158                                     ;UPDATE PASS COUNT
1159                                     ;CHECK FOR EXIT TO ACT-11
1160                                     ;RESTART TEST
1161
1162 002764 005037 001234 .EOP: CLR LSTERR ;CLEAR LAST ERROR PC
1163 002770 105037 001341 CLR RB ERRFLG ;CLEAR ERROR FLAG
1164 002774 005237 001230 INC PASCNT ;UPDATE PASS COUNT
1165 003000 013777 001230 176172 MOV PASCNT,@DISPLAY ;DISPLAY PASS COUNT
1166 003006 104402 005157 TYPE ,MEPASS ;TYPE END PASS
1167 003012 104402 005342 TYPE ,MCSRX ;TYPE CSR
1168 003016 104411 003130 CNVRT ,XCSR ;SHOW IT
1169 003022 104402 005350 TYPE ,MVECX ;TYPE VECTOR
1170 003026 104411 003136 CNVRT ,XVEC ;SHOW IT
1171 003032 104402 005356 TYPE ,MPASSX ;TYPE PASSES
1172 003036 104411 003144 CNVRT ,XPASS ;SHOW IT
1173 003042 104402 005367 TYPE ,MERRX ;TYPE ERRORS
1174 003046 104411 003152 CNVRT ,XERR ;SHOW IT
1175 003052 105337 001313 DECB SAVNUM ;ARE ALL DEVICES TESTED?
1176 003056 001017 BNE RESTRT ;BR IF NO.
1177 003060 112737 000377 001343 MOV B #377,QV.FLG ;SET THE QUICK VERIFY FLAG.
1178 003066 113737 001311 001313 MOV B DUPNUM,SAVNUM ;RESTORE THE COUNT
1179 003074 013701 000042 MOV @#42,R1 ;CHECK FOR ACT-11 OR DDP
1180 003100 001406 BEQ RESTRt ;IF NOT, CONTINUE TESTING
1181 003102 000005 RESET ;STOP THE SHOW--CLEAR THE WORLD
1182 003104
1183 003104 004711 $ENDAD: JSR PC,(R1)
1184 003106 000240 NOP
1185 003110 000240 NOP
1186 003112 000240 NOP
1187 003114 000240 NOP
1188 003116 012737 006226 001214 RESTRt: MOV #CYCLE,RETURN
1189 003124 000137 006226 JMP CYCLE
1190 003130 000001 XCSR: 1
1191 003132 006 002 .BYTE 6,2
1192 003134 001404 RXCSR
1193 003136 000001 XVEC: 1
1194 003140 003 002 .BYTE 3,2
1195 003142 001374 DUPRVC
1196 003144 000001 XPASS: 1
1197 003146 006 002 .BYTE 6,2
1198 003150 001230 PASCNT
1199 003152 000001 XERR: 1
1200 003154 006 002 .BYTE 6,2
1201 003156 001232 ERRCNT
1202
1203                                     ;SCOPE LOOP AND INTERATION HANDLER
1204
1205 003160 005037 001234 .SCOPE: CLR LSTERR ;CLEAR LAST ERROR PC
1206 003164 010016 MOV R0,(SP) ;SAVE R0 ON STACK
1207 003166 032777 040000 176006 BIT #BIT14,@SWR ;LOOP ON TEST?
1208 003174 001407 TTST: BEQ 1$ ;BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
1209 003176 000437 BR 3$ ;GO TO 3$ (DITTO)
1210 003200 105777 176000 TSTB @TKCSR ;KYBD DONE?
1211 003204 100034 BPL 3$ ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```



```

1212 003206 017700 175774      MOV      @TKDBR,R0      ;CLR DONE BIT
1213 003212 000415      BR        2$      ;CONTINUE
1214 003214 032777 004000 175760 1$: BIT      #SW11,@SWR      ;DELETE ITERATION (QUICK PASS)?
1215 003222 001011      BNE      2$      ;BR IF YES
1216 003224 105737 001343      TSTB     QV.FLG      ;HAS FIRST PASS BEEN COMPLETED?
1217 003230 001406      BEQ      2$      ;BR IF QUICK VERIFY
1218 003232 005237 001224      INC      LPCNT      ;UPDATE ITERATION COUNTER
1219 003236 023737 001224 001222  CMP      LPCNT,ICOUNT  ;ALL ITERATIONS DONE?
1220 003244 001014      BNE      3$      ;BR IF NOT YET
1221 003246 105037 001341 2$: CLRB     ERRFLG      ;PREPARE FOR NEW TEST
1222 003252 005037 001224      CLR      LPCNT      ;START ICOUNT AT ZERO
1223 003256 005037 001220      CLR      LOCK
1224 003262 012737 000050 001222  MOV      #50,ICOUNT  ;RESET ITERATIONS
1225 003270 013737 001216 001214  MOV      NEXT,RETURN ;GET NEXT TEST
1226 003276 011600 3$: MOV      (SP),R0      ;POP R0 OFF STACK
1227 003300 022626      POP2SP      ;FAKE AN RTI
1228 003302 000177 175706      JMP      @RETURN    ;GO DO THE TEST
1229 003306 001407      BRW:     1407
1230 003310 000437      BRX:     437
1231
1232      ;CHECK FOR FREEZE ON CURRENT DATA
1233      -----
1234
1235 003312 032777 001000 175662 .SCOP1: BIT      #SW09,@SWR      ;IS SW09=1(SET)?
1236 003320 001405      BEQ      1$      ;BR IF NOT SET.
1237 003322 005737 001220      TST      LOCK
1238 003326 001402      BEQ      1$
1239 003330 013716 001220      MOV      LOCK,(SP)  ;GOTO THE ADDRESS IN LOCK.
1240 003334 000002 1$: RTI      ;GO BACK.
1241
1242      ;TELETYPE OUTPUT ROUTINE
1243      -----
1244
1245 003336 010546 .TYPE: MOV      R5,-(SP)      ;SAVE R5 ON THE STACK.
1246 003340 017605 000002      MOV      @2(SP),R5      ;GET ADDRESS OF MESSAGE.
1247 003344 062766 000002 000002  ADD      #2,2(SP)      ;POP OVER ADDRESS.
1248 003352 032777 010000 175622 1$: BIT      #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
1249 003360 001012      BNE      3$      ;BR IF NO PRINT OUT WANTED (SW12=1)
1250 003362 105715      TSTB     (R5)      ;IS NUMBER MINUS? (MSB=1(BIT7))
1251 003364 100002      BPL      2$      ;BR IF NUMBER IS PLUS
1252 003366 104402 005136      TYPE     ,MCRLF      ;TYPE A CR/LF!
1253 003372 105777 175612 2$: TSTB     @TPCSR      ;TTY READY?
1254 003376 100375      BPL      2$      ;BR IF NO.
1255 003400 112577 175606      MOVB     (R5)+,@TPDBR ;PRINT CURRENT CHAR.
1256 003404 001362      BNE      1$      ;IF NOT ZERO KEEP PRINTING!
1257 003406 012605 3$: MOV      (SP)+,R5      ;END OF OUTPUT. RESTORE R5
1258 003410 000002      RTI      ;GO HOME
1259
1260
1261 003412 010346 .INSTR: MOV      R3,-(SP)      ;SAVE R3 ON STACK
1262 003414 010446      MOV      R4,-(SP)      ;SAVE R4 ON STACK
1263 003416 017637 000004 003434      MOV      @4(SP),.MSG
1264 003424 062766 000002 000004  ADD      #2,4(SP)
1265 003432 104402 .INST1: TYPE
1266 003434 000000 .MSG: 0
1267 003436 012704 006162      MOV      #INBUF,R4
  
```

1268	003442	012703	000007		MOV	#7,R3	
1269	003446	105777	175532	1\$:	TSTB	@TKCSR	
1270	003452	100375			BPL	1\$	
1271	003454	117714	175526		MOVB	@TKDBR,(R4)	
1272	003460	142714	000200		BICB	#200,(R4)	
1273	003464	122427	000015		CMPB	(R4)+,#15	
1274	003470	001417			BEQ	INSTR2	
1275	003472	105777	175512	2\$:	TSTB	@TPCSR	
1276	003476	100375			BPL	2\$	
1277	003500	017777	175502 175504		MOV	@TKDBR,@TPDBR	
1278	003506	005303			DEC	R3	
1279	003510	001356			BNE	1\$	
1280	003512	012604			MOV	(SP)+,R4	
1281	003514	012603			MOV	(SP)+,R3	
1282	003516	010346		.INSTE:	MOV	R3,-(SP)	
1283	003520	010446			MOV	R4,-(SP)	
1284	003522	104402	005132		TYPE	,MQM	
1285	003526	000741			BR	.INST1	
1286	003530	012604		INSTR2:	MOV	(SP)+,R4	;RESTORE R4
1287	003532	012603			MOV	(SP)+,R3	;RESTORE R3
1288	003534	000002			RTI		
1289							
1290							
1291							
1292							
1293	003536	010546		.PARAM:	MOV	R5,-(SP)	
1294	003540	010446			MOV	R4,-(SP)	
1295	003542	016605	000004		MOV	4(SP),R5	
1296	003546	012537	003726		MOV	(R5)+,LOLIM	
1297	003552	012537	003730		MOV	(R5)+,HILIM	
1298	003556	012537	003732		MOV	(R5)+,DEVADR	
1299	003562	112537	003734		MOVB	(R5)+,LOBITS	
1300	003566	112537	003735		MOVB	(R5)+,ADRCNT	
1301	003572	010566	000004		MOV	R5,4(SP)	
1302	003576	005005		PARAM1:	CLR	R5	
1303	003600	012704	006162		MOV	#INBUF,R4	
1304	003604	122714	000015		CMPB	#15,(R4)	
1305	003610	001420			BEQ	PARERR	
1306	003612	121427	000060	1\$:	CMPB	(R4),#60	
1307	003616	002415			BLT	PARERR	
1308	003620	121427	000067		CMPB	(R4),#67	
1309	003624	003012			BGT	PARERR	
1310	003626	142714	000060		BICB	#60,(R4)	
1311	003632	152405			BISB	(R4)+,R5	
1312	003634	122714	000015		CMPB	#15,(R4)	
1313	003640	001406			BEQ	LIMITS	
1314	003642	006305			ASL	R5	
1315	003644	006305			ASL	R5	
1316	003646	006305			ASL	R5	
1317	003650	000760			BR	1\$	
1318	003652	104404		PARERR:	INSTR		
1319	003654	000750			BR	PARAM1	
1320							
1321							
1322							
1323							

:CONVERT ASCII STRING TO OCTAL
 :-----

:TEST TO SEE IF NUMBER IS WITHIN LIMITS
 :-----

1324	003656	020537	003730	LIMITS:	CMP	R5,HILIM	
1325	003662	101373			BHI	PARERR	
1326	003664	020537	003726		CMP	R5,LOLIM	
1327	003670	103770			BLO	PARERR	
1328	003672	133705	003734		BITB	LOBITS,R5	
1329	003676	001365			BNE	PARERR	
1330							
1331							
1332							:STORE NUMBER AT SPECIFIED ADDRESS
1333	003700	013704	003732		MOV	DEVADR,R4	
1334	003704	010524		1\$:	MOV	R5,(R4)+	
1335	003706	062705	000002		ADD	#2,R5	
1336	003712	105337	003735		DECB	ADRCNT	
1337	003716	001372			BNE	1\$	
1338	003720	012604			MOV	(SP)+,R4	
1339	003722	012605			MOV	(SP)+,R5	
1340	003724	000002			RTI		
1341	003726	000000		LOLIM:	0		
1342	003730	000000		HILIM:	0		
1343	003732	000000		DEVADR:	0		
1344	003734	000000		LOBITS:	0		
1345		003735		ADRCNT=	LOBITS+1		
1346							
1347							:SAVE PC OF TEST THAT FAILED AND R0-R5
1348							-----
1349							
1350	003736	016637	000004 001266	.SAV05:	MOV	4(SP),SAVPC	:SAVE R7 (PC)
1351							
1352							:SAVE R0-R5
1353							
1354	003744	010537	001262	SV05:	MOV	R5,SAVR5	:SAVE R5
1355	003750	010437	001260		MOV	R4,SAVR4	:SAVE R4
1356	003754	010337	001256		MOV	R3,SAVR3	:SAVE R3
1357	003760	010237	001254		MOV	R2,SAVR2	:SAVE R2
1358	003764	010137	001252		MOV	R1,SAVR1	:SAVE R1
1359	003770	010037	001250		MOV	R0,SAVR0	:SAVE R0
1360	003774	000002			RTI		:LEAVE.
1361							
1362							:RESTORE R0-R5
1363							
1364	003776	013700	001250	.RES05:	MOV	SAVR0,R0	:RESTORE R0
1365	004002	013701	001252		MOV	SAVR1,R1	:RESTORE R1
1366	004006	013702	001254		MOV	SAVR2,R2	:RESTORE R2
1367	004012	013703	001256		MOV	SAVR3,R3	:RESTORE R3
1368	004016	013704	001260		MOV	SAVR4,R4	:RESTORE R4
1369	004022	013705	001262		MOV	SAVR5,R5	:RESTORE R5
1370	004026	000002			RTI		:LEAVE
1371							
1372							
1373							:CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1374							-----
1375							
1376	004030	104402	005136	.CONVR:	TYPE	,MCRLF	
1377	004034	010046		.CNVRT:	MOV	R0,-(SP)	
1378	004036	010146			MOV	R1,-(SP)	
1379	004040	010346			MOV	R3,-(SP)	

1380	004042	010446			MOV	R4,-(SP)
1381	004044	010546			MOV	R5,-(SF)
1382	004046	017601	000012		MOV	@12(SP),R1
1383	004052	062766	000002	000012	ADD	#2,12(SP)
1384	004060	012137	004234		MOV	(R1)+,WRDCNT
1385	004064	112137	004236	1\$:	MOVB	(R1)+,CHRCNT
1386	004070	112137	004237		MOVB	(R1)+,SPACNT
1387	004074	013137	004240		MOV	@(R1)+,BINWRD
1388	004100	013704	004240	2\$:	MOV	BINWRD,R4
1389	004104	113705	004236		MOVB	CHRCNT,R5
1390	004110	012700	006056		MOV	#TEMP,R0
1391	004114	010403		3\$:	MOV	R4,R3
1392	004116	042703	177770		BIC	#177770,R3
1393	004122	062703	000060		ADD	#060,R3
1394	004126	110320			MOVB	R3,(R0)+
1395	004130	000241			CLC	
1396	004132	006004			ROR	R4
1397	004134	000241			CLC	
1398	004136	006004			ROR	R4
1399	004140	000241			CLC	
1400	004142	006004			ROR	R4
1401	004144	005305			DEC	R5
1402	004146	001362			BNE	3\$
1403	004150	012703	006120		MOV	#MDATA,R3
1404	004154	114023		4\$:	MOVB	-(R0),(R3)+
1405	004156	105337	004236		DECB	CHRCNT
1406	004162	001374			BNE	4\$
1407	004164	105737	004237		TSTB	SPACNT
1408	004170	001405			BEQ	6\$
1409	004172	112723	000040	5\$:	MOVB	#040,(R3)+
1410	004176	105337	004237		DECB	SPACNT
1411	004202	001373			BNE	5\$
1412	004204	105013		6\$:	CLRB	(R3)
1413	004206	104402	006120		TYPE	,MDATA
1414	004212	005337	004234		DEC	WRDCNT
1415	004216	001322			BNE	1\$
1416	004220	012605			MOV	(SP)+,R5
1417	004222	012604			MOV	(SP)+,R4
1418	004224	012603			MOV	(SP)+,R3
1419	004226	012601			MOV	(SP)+,R1
1420	004230	012600			MOV	(SP)+,R0
1421	004232	000002			RTI	
1422	004234	000000			WRDCNT:	0
1423	004236	000000			CHRCNT:	0
1424		004237			SPACNT=	CHRCNT+1
1425	004240	000000			BINWRD:	0
1426						
1427						
1428						
1429						
1430						
1431						
1432						
1433	004242	017605	000000		.SETFLG:MOV	@(SP),R5
1434	004246	042737	000040	006162	BIC	#40,INBUF
1435	004254	122737	000116	006162	CMPB	#'N',INBUF ;IS IT 'N'?


```

1436 004262 001002      BNE      1$
1437 004264 105015      CLRB     (R5)      ;000
1438 004266 000406      BR       2$
1439 004270 122737 000131 006162 1$: CMPB     #'Y',INBUF      ;IS IT 'Y' ?
1440 004276 001005      BNE      3$
1441 004300 112715 177777      MOVB     #-1,(R5)      ;377
1442 004304 062716 000002      2$: ADD      #2,(SP)
1443 004310 000002      RTI
1444 004312 104404      3$: INSTER    ;RETRY
1445 004314 000752      BR        .SETFLG
1446
1447
1448      ;TRAP DISPATCH SERVICE
1449      ;ARGUMENT OF TRAP IS EXTRACTED
1450      ;AND USED AS OFFSET TO OBTAIN POINTER
1451      ;TO SELECTED SUBROUTINE
1452
1453 004316 011646      .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
1454 004320 162716 000002      SUB      #2,(SP)      ;=PC OF TRAP
1455 004324 017616 000000      MOV      @ (SP), (SP)      ;GET TRP
1456 004330 006316      TRPOK: ASL      (SP)      ;MULTIPLY TRAP ARG BY 2
1457 004332 042716 177001      BIC      #177001,(SP)      ;CLEAR UNWANTED BITS
1458 004336 062716 001344      ADD      #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
1459 004342 017616 000000      MOV      @ (SP), (SP)      ;SUBROUTINE ADDRESS
1460 004346 000136      JMP      @ (SP)+      ;GO TO SUBROUTINE
1461
1462      ;ERROR HANDLER
1463      ;-----
1464
1465 004350 032777 010000 174624 .HLT: BIT      #SW12,@SWR      ;BELL ON ERROR?
1466 004356 001406      BEQ      XBX      ;BR IF NO BELL
1467 004360 105777 174624      TSTB     @TPCSR      ;TTY READY.
1468 004364 100003      BPL      XBX      ;DON'T WAIT IF TTY NOT READY.
1469 004366 112777 000207 174616      MOVB     #207,@TPDBR      ;PUSH A BELL AT THE TTY.
1470 004374 032777 020000 174600 XBX: BIT      #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1471 004402 001105      BNE      HALTS      ;BR IF NO PRINT OUT WANTED.
1472 004404 021637 001234      CMP      (SP),LSTERR      ;WAS THIS ERROR FOUND LAST TIME?
1473 004410 001404      BEQ      1$      ;BR IF YES
1474 004412 011637 001234      MOV      (SP),LSTERR      ;RECORD BEING HERE
1475 004416 105037 001341      CLRB     ERRFLG      ;PREPARE HEADER
1476 004422 104406      1$: SAVO5      ;SAVE ALL PROC REGISTERS
1477 004424 011605      MOV      (SP),R5      ;GET THE PC OF ERROR
1478 004426 162705 000002      SUB      #2,R5      ;GET ADDRESS OF TRAP CALL
1479 004432 011504      MOV      (R5),R4      ;GET HLT INSTRUCTION
1480 004434 006304      ASL      R4      ;MULT BY TWO
1481 004436 061504      ADD      (R5),R4      ;DOUBLE IT
1482 004440 006304      ASL      R4      ;MULT AGAIN
1483 004442 042704 177001      BIC      #177001,R4      ;CLEAR JUNK
1484 004446 062704 030026      ADD      #.ERRTAB,R4      ;GET POINTER
1485 004452 012437 004566      MOV      (R4)+,ERRMSG      ;GET ERROR MESSAGE
1486 004456 012437 004600      MOV      (R4)+,DATAHD      ;GET DATA HEADRER
1487 004462 011437 004612      MOV      (R4),DATABP      ;GET DATA TABLE
1488 004466 105737 001341      TSTB     ERRFLG      ;TYPE HEADREER
1489 004472 001403      BEQ      TYPMSG      ;BR IF YES
1490 004474 005737 004612      TST      DATABP      ;DOES DATA TABLE EXIST?
1491 004500 001040      BNE      TYPDAT      ;BR IF YES.

```

1492	004502	104402	005136		TYPMSG: TYPE	,MCRLF	
1493	004506	104402	005136		TYPE	,MCRLF	
1494	004512	005737	001220		TST	LOCK	
1495	004516	001402			BEQ	1\$	
1496	004520	104402	005412		TYPE	,MASTEK	
1497	004524	104402	005400		1\$: TYPE	,MTSTN	
1498	004530	104411	005000		CNVRT	,XTSTN	:SHOW IT
1499	004534	104402	005467		TYPE	,MERRPC	:TYPE PC.
1500	004540	104411	004772		CNVRT	,ERTABO	:SHOW IT
1501	004544	104402	005136		TYPE	,MCRLF	:GIVE A CR/LF
1502	004550	112737	177777	001341	MOV8	#-1,ERRFLG	:NO MORE HEADER UNLESS NO DATA TABLE.
1503	004556	005737	004566		TST	ERRMSG	:IS THERE AN ERROR MESSAGE?
1504	004562	001402			BEQ	WRKO.FM	:BR IF NO.
1505	004564	104402			TYPE		:TYPE
1506	004566	000000			ERRMSG: 0		: ERROR MESSAGE
1507	004570				WRKO.FM		
1508	004570	005737	004600		TST	DATAHD	:DATA HEADER?
1509	004574	001402			BEQ	TYPDAT	:BR IF NO
1510	004576	104402			TYPE		:TYPE
1511	004600	000000			DATAHD: 0		: DATA HEADER
1512	004602	005737	004612		TYPDAT: TST	DATABP	:DATA TABLE?
1513	004606	001402			BEQ	RESREG	:BR IF NO.
1514	004610	104410			CONVRT		:SHOW
1515	004612	000000			DATABP: 0		: DATA TABLE
1516	004614	104407			RESREG: RES05		:RESTORE PROC REGISTERS
1517	004616	022737	003104	000042	HALTS: CMP	#\$ENDAD,@#42	:IF ACT-11 AUTO MODE--HALT!!
1518	004624	001403			BEQ	1\$	
1519	004626	005777	174350		TST	@SWR	:HALT ON ERROR?
1520	004632	100035			BPL	EXITER	:BR IF NO HALT ON ERROR
1521	004634	010046			1\$: PUSHRO		:SAVE RO
1522	004636	016600	000002		MOV	2(SP),RO	:SHOW ERROR PC IN DATA LIGHTS
1523	004642	013746	000004		MOV	4,-(SP)	:SAVE OLD TRAP
1524	004646	013746	000006		MOV	6,-(SP)	
1525	004652	012737	004710	000004	MOV	#22\$,4	:FORCE HALT IF TIME-OUT
1526	004660	012737	000340	000006	MOV	#340,6	:WHEN REFERENCING TXCSR
1527	004666	042777	014000	174516	BIC	#\$YSTST!MEXT,@TXCSR	
1528	004674	000000			HALT		:HALT
1529	004676	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1530	004702	012637	000004		MOV	(SP)+,4	
1531	004706	000406			BR	33\$	
1532	004710	000000			22\$: HALT		:HALT
1533	004712	022626			CMP	(SP)+,(SP)+	:POP STACK
1534	004714	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1535	004720	012637	000004		MOV	(SP)+,4	
1536	004724	012600			33\$: POPRO		:GET RO
1537	004726	005237	001232		EXITER: INC	ERRCNT	:UPDATE ERROR COUNT
1538	004732	032777	000400	174242	BIT	#\$W08,@SWR	:GOTO TOP OF TEST?
1539	004740	001007			BNE	1\$	:BR IF YES
1540	004742	032777	0020C0	174232	BIT	#\$W10,@SWR	:GOTO NEXT TEST?
1541	004750	001407			BEQ	2\$	:BR IF NO
1542	004752	013737	001216	001214	MOV	NEXT,RETURN	:SET FOR NEXT TEST
1543	004760	012706	001150		1\$: MOV	#\$STACK,SP	:RESET SP
1544	004764	000177	174224		JMP	@RETURN	:GOTO SPECIFIED TEST
1545	004770	000002			2\$: RTI		:RETURN
1546	004772	000001			ERTABO: 1		
1547	004774	006	002		.BYTE	6,2	


```

1548 004776 001266          SAVPC
1549 005000 000001          XTSTN: 1
1550 005002      003      002      .BYTE 3,2
1551 005004 001226          TSTNO
1552 005006 017600 000000      .PKCLK: MOV @ (SP),R0      ;GET THE # OF TICKS TO POKE
1553 005012 062716 000002      ADD #2,(SP)      ;POP OVER THE #
1554 005016          1$:
1555 005016 052777 020000 174366      BIS #CLK,@TXCSR      ;POKE CLOCK UP
1556 005024 005300          DEC R0      ;ARE WE DONE?
1557 005026 001405          BEQ 2$      ;YES-GO TO 2$
1558 005030 042777 020000 174354      BIC #CLK,@TXCSR      ;POKE CLOCK DOWN
1559 005036 005300          DEC R0      ;ARE WE DONE?
1560 005040 001366          BNE 1$      ;NO-REPEAT
1561 005042 000002      2$: RTI      ;RETURN
1562
1563
1564          ;WAIT ROUTINE
1565 005044 000240      SMALL: NOP      ;STALL
1566 005046 000207          RTS PC      ;RETURN
1567
1568          ;POWER FAIL ROUTINE
1569
1570 005050 012737 005060 000024      .PFAIL: MOV #PWRUP,24      ;LOAD PFAIL VECTOR FOR POWER UP
1571 005056 000000          HALT
1572 005060 000005      PWRUP: RESET      ;WAIT TTY TO COME UP
1573 005062 012706 001150          MOV #STACK,SP      ;REINIT STACK POINTER
1574 005066 012737 005050 000024      MOV #.PFAIL,24      ;LOAD PFAIL VECTOR FOR POWER DOWN
1575 005074 104402          TYPE
1576 005076 005141          MPOWER
1577 005100 000177 174110          JMP @RETURN
1578          ;CLRVEC ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1579
1580 005104 012702 000300      CLRVEC: MOV #300,R2      ;R2 COMM VECTOR AREA ADRS
1581 005110 012701 000302          MOV #302,R1      ;INIT R1 WITH ADRS OF HALT
1582 005114 010122      1$: MOV R1,(R2)+      ;MOV .+2 TO PC
1583 005116 005022          CLR (R2)+      ;MOV HALT TO PC
1584 005120 022121          CMP (R1)+,(R1)+      ;INC TO NEXT VECTOR AREA
1585 005122 022701 000776          CMP #776,R1      ;END OF VECTOR AREA
1586 005126 001372          BNE 1$      ;NO
1587 005130 000207          RTS PC      ;RETURN
1588
1589
1590
1591 005132 020040 000077      MQM: .ASCIIZ / ?/
      (2) 005136 005015      MCRLF: .ASCIIZ <15><12>
      (2) 005141 377 053520 020122      MPOWER: .ASCIIZ <377>/PWR FAILED. /
      (2) 005157 015 042777 042116      MEPASS: .ASCIIZ <15><377>/END PASS CZDPBC /
      (2) 005203 377 000122      MR: .ASCIIZ <377>/R/
      (2) 005206 050377 047522 051107      MERR2: .ASCIIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
      (2) 005255 377 047111 052523      MERR3: .ASCIIZ <377>/INSUFFICIENT DATA!/
      (2) 005301 377 042524 052123      MTSTPC: .ASCIIZ <377>/TEST PC-/
      (2) 005313 377 047514 045503      MLOCK: .ASCIIZ <377>/LOCK ON SELECTED TEST/
      (2) 005342 051503 035122 000040      MCSRX: .ASCIIZ /CSR: /
      (2) 005350 042526 035103 000040      MVECX: .ASCIIZ /VEC: /
      (2) 005356 040520 051523 051505      MPASSX: .ASCIIZ /PASSES: /
      (2) 005367 105 051122 051117      MERRX: .ASCIIZ /ERRORS: /
  
```

CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 36
 CZDPBC.P11 02-MAY-78 08:23 END OF PASS ROUTINE

SEQ 0035

(2)	005400	042524	052123	047040	MTSTN: .ASCIIZ	/TEST NO: /
(2)	005412	000052			MASTEK: .ASCIIZ	/* /
(2)	005414	051777	052105	051440	MNEW: .ASCIIZ	<377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
(2)	005467	120	035103	000040	MERRPC: .ASCIIZ	/PC: /
(2)	005474	051377	041505	041440	MCSR: .ASCIIZ	<377>/REC CSR ADRS /
(2)	005513	377	042526	020103	MVEC: .ASCIIZ	<377>/VEC ADRS /
(2)	005526	044777	020123	044124	MJMPR: .ASCIIZ	<377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
(2)	005601	377	051511	052040	MTCN: .ASCIIZ	<377>/IS THE H325 CONNECTOR ON? (Y OR N) /
(2)	005650	021777	047440	020106	MTOTAL: .ASCIIZ	<377>/# OF DUP'S (IN OCTAL) /
(2)	005703	377	051120	047511	MPAR: .ASCIIZ	<377>/PRIORITY (4 TO 7) /
(2)	005727	377	042523	020103	MSTJM: .ASCIIZ	<377>/SEC TX JMPR IN? (Y OR N) /
(2)	005762	051777	041505	051040	MSRJM: .ASCIIZ	<377>/SEC RX JMPR IN? (Y OR N) /
(2)	006015	377	040515	020120	XHEAD: .ASCIIZ	<377>/MAP OF DUP11 STATUS/<377>
(2)		006044			.EVEN	
(2)	006044	000002			XSTATQ: 2	
1592	006046	006	003		.BYTE	6,3
1593	006050	001236			TEMP1	
1594	006052	006	002		.BYTE	6,2
1595	006054	001240			TEMP2	
1596					.EVEN	
1597						
1598	006056	000000			TEMP: 0	
1599		006120			.=. +40	
1600	006120	000000			MDATA: 0	
1601		006162			.=. +40	
1602	006162	000000			INBUF: 0	
1603		006224			.=. +40	
1604	006224	000001			TRP.PC: .BLKW 1	
1605						


```

1606
1607
1608      ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DUP11'S
1609      ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1610      ;AND RUNS THE SPECIFIED DUP11'S.  THIS ROUTINE *MUST*
1611      ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1612      ;SETUP NECESSARY.
1613
1614
1615      006226 105737 001310      CYCLE:  TSTB      DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1616      006232 001004      BNE      1$      ;BR IF OK.
1617      006234 104402 005206      TYPE      ,MERR2      ;NO DUP11'S SELECTED!!
1618      006240 000000      HALT      ;STOP THE SHOW.
1619      006242 000776      BR      -2      ;DISQUALIFY CONT. SW.
1620      006244 133737 001314 001310 1$:  BITB      RUN,DUPACTV      ;IS THIS ONE 'ACTIVE'
1621      006252 001020      BNE      2$      ;BR IF GOOD ONE FOUND.
1622      006254 000241      CLC      ;CLEAR PROC. CARRY BIT.
1623      006256 106137 001314      ROLB      RUN      ;UPDATE POINTER
1624      006262 105537 001314      ADCB      RUN      ;CATCH CARRY FROM RUN
1625      006266 062737 000006 001316      ADD      #6,CREAM      ;UPDATE ADDRESS POINTER.
1626      006274 022737 001560 001316      CMP      #DUP.END,CREAM
1627      006302 001360      BNE      1$      ;KEEP GOING; NOT ALL TESTED FOR.
1628      006304 012737 001500 001316      MOV      #DUP.MAP,CREAM      ;RESET ADDRESS POINTER.
1629      006312 000754      BR      1$      ;KEEP LOOKING FOR ACTIVE DUP11
1630      006314 000241      CLC      ;CLEAR PROC. CARRY.
1631      006316 106137 001314      ROLB      RUN      ;UPDATE POINTER.
1632      006322 105537 001314      ADCB      RUN      ;CATCH CARRY.
1633      006326 013700 001316      MOV      CREAM,R0      ;GET ADDRESS POINTER.
1634      006332 062737 000006 001316      ADD      #6,CREAM      ;UPDATE.
1635      006340 022737 001560 001316      CMP      #DUP.END,CREAM
1636
1637      006346 001003      BNE      3$      ;ALL DONE?
1638      006350 012737 001500 001316      MOV      #DUP.MAP,CREAM      ;BR IF NO.
1639      006356 012037 001404      MOV      (R0)+,RXCSR      ;RESTORE POINTER.
1640      006362 012037 001374      MOV      (R0)+,DUPRVC      ;LOAD SYSTEM CTRL. REG
1641      006366 012037 001434      MOV      (R0)+,L00.00      ;LOAD VECTOR
1642      006372 012700 000002      MOV      #2,R0      ;GET PARAMETERS
1643      006376 013737 001404 001424      MOV      RXCSR,HUPRCR      ;SAVE CORE THIS WAY!
1644      006404 005237 001424      INC      HUPRCR      ;GET CONTROL REG HIGH BYTE
1645      006410 013737 001424 001406      MOV      HUPRCR,RXDBUF      ;GOT IT
1646      006416 005237 001406      INC      RXDBUF      ;GET RX CONTROL REG BUFFER
1647      006422 013737 001406 001416      MOV      RXDBUF,DUPSEC      ;GOT IT
1648      006430 013737 001406 001410      MOV      RXDBUF,PARCSR      ;GOT SECONDARY REG SELECT REG
1649      006436 013737 001406 001422      MOV      RXDBUF,HUPRBF      ;GOT PARAMETER STATUS REGISTER
1650      006444 005237 001422      INC      HUPRBF      ;GET RX BUFFER HIGH BYTE
1651      006450 013737 001422 001420      MOV      HUPRBF,HUPPSR      ;GOT IT
1652      006456 013737 001420 001412      MOV      HUPPSR,TXCSR      ;GOT PAR STATUS REG HIGH BYTE
1653      006464 005237 001412      INC      TXCSR      ;GET TX CONTROL REGISTER
1654      006470 013737 001412 001430      MOV      TXCSR,HUPTCR      ;GOT IT
1655      006476 005237 001430      INC      HUPTCR      ;GET TX CONTROL REG HIGH BYTE
1656      006502 013737 001430 001414      MOV      HUPTCR,TXDBUF      ;GOT IT
1657      006510 005237 001414      INC      TXDBUF      ;BET TX BUFFER
1658      006514 013737 001414 001426      MOV      TXDBUF,HUPTBF      ;GOT IT
1659      006522 005237 001426      INC      HUPTBF      ;GET TX BUFFER HIGH BYTE
1660
1661      006526 013737 001374 001376      MOV      DUPRVC,DUPRPS      ;GOT IT
                                     ;RX VECTOR

```

1662	006534	060037	001376		ADD	RO,DUPRPS	;RX PRIORITY LEVEL
1663	006540	013737	001376	001400	MOV	DUPRPS,DUPTVC	
1664	006546	060037	001400		ADD	RO,DUPTVC	;TX VECTOR
1665	006552	013737	001400	001402	MOV	DUPTVC,DUPTPS	
1666	006560	060037	001402		ADD	RO,DUPTPS	;TX PRIORITY LEVEL
1667							
1668							
1669	006564	012700	001434		MOV	#L00.00,RO	;LOAD STAU 00-00
1670	006570	012701	001432		MOV	#MASK,A,R1	;PREPARE MASK.
1671	006574	012702	001433		MOV	#CLK,A,R2	;PREPARE CLOCKS
1672	006600	004737	006744		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1673	006604	005737	000042		TST	@#42	
1674	006610	001050			BNE	4\$	
1675	006612	032777	000002	172362	BIT	#SW01,@SWR	;IF SW01=1,GET STARTING TEST #
1676	006620	001444			BEQ	4\$	
1677	006622	104402	005136		7\$: TYPE	,MCRLF	
1678	006626	104403			INSTR	;OUTPUT MESSAGE & GET INPUT STRING	
1679	006630	005400			MTSTN	;MESSAGE	
1680	006632	104405			PARAM	;CONVERT STRING	
1681	006634	000001			1	;LOW LIMIT	
1682	006636	001000			1000	;HIGH LIMIT	
1683	006640	001226			TSTNO	;STORE AT THIS LOCATION	
1684	006642	000			.BYTE 0	;MASK	
1685	006643	001			.BYTE 1	;HOW MANY TIMES + 2	
1686	006644	012700	007160		MOV	#TST1,RO	
1687	006650	022710	012737		5\$: CMP	#12737,(RO)	
1688	006654	001017			BNE	6\$	
1689	006656	023760	001226	000002	CMP	TSTNO,2(RO)	
1690	006664	001013			BNE	6\$	
1691	006666	022760	001226	000004	CMP	#TSTNO,4(RO)	
1692	006674	001007			BNE	6\$	
1693	006676	010037	001214		MOV	RO,RETURN	;SAVE PC
1694	006702	104402	005136		TYPE	,MCRLF	
1695	006706	104402	005203		TYPE	,MR	
1696	006712	000412			BR	8\$	
1697	006714	005720			6\$: TST	(RO)+	
1698	006716	020027	026412		CMP	RO,#TLAST+10	
1699	006722	001352			BNE	5\$	
1700	006724	104402	005132		TYPE	,MQM	
1701	006730	000734			BR	7\$	
1702							
1703	006732	012737	007160	001214	4\$: MOV	#TST1,RETURN	;PREPARE RETURN ADDRESS
1704	006740	000177	172250		8\$: JMP	@RETURN	;GO START TESTING.
1705							
1706	006744	011003			FIX.00: MOV	(RO),R3	;GET PARAMETERS.
1707	006746	000207			5\$: RTS	PC	;


```

1708
1709
1710
1711 006750 012104 ACC: MOV (R1)+,R4 ;GET THE FLAG FOR # OF CLOCK TICKS
1712 006752 104412 000002 1$: PKCLK ,2
1713 006756 000241 CLC ;PUT CARRY IN A KNOWN STATE
1714 006760 032777 040000 172424 BIT #MTDATA,@TXCSR ;FIND OUT IF BIT IS A 1 OR 0
1715 006766 001401 BEQ ,+4 ;BR IF 0
1716 006770 000261 SEC ;SET THE BIT
1717 006772 006004 ROR R4 ;PICK UP CARRY AND PUSH INTO R4
1718 006774 103366 BCC 1$ ;BRANCH IF MORE TO GO
1719 006776 000201 RTS R1
1720 007000 005037 001246 ABRT: CLR TEMP5
1721 007004 012137 001244 MOV (R1)+,TEMP4 ;GET THE # OF ABORTS TO DO
1722 007010 104412 000002 1$: PKCLK ,2 ;POKE OUT A BIT
1723 007014 032777 040000 172370 BIT #MTDATA,@TXCSR ;CHECK MAINT DATA OUT
1724 007022 001001 BNE 2$ ;BRANCH IF IT IS A ONE
1725 007024 104013 HLT 13 ;OUTPUT WAS A ZERO--NG
1726 007026 005237 001246 2$: INC TEMP5 ;INC THE # OF BITS OUTPUT
1727 007032 022737 000010 001246 CMP #8.,TEMP5 ;IS THE CHARACTER DONE?
1728 007040 001363 BNE 1$ ;BRANCH IF NOT DONE
1729 007042 005337 001244 DEC TEMP4 ;LOWER THE #TO DO
1730 007046 001360 BNE 1$ ;BRANCH IF NOT DONE
1731 007050 000201 RTS R1
1732
1733 007052 012137 001244 FLG: MOV (R1)+,TEMP4 ;GET THE # OF FLAGS
1734 007056 104412 000002 64$: PKCLK ,2 ;POKE OUT THE FIRST BIT OF THE FLAG
1735 007062 032777 040000 172322 BIT #MTDATA,@TXCSR ;CHECK MAINT DATA OUT
1736 007070 001401 BEQ 65$ ;BRANCH IF 0
1737 007072 104012 HLT 12 ;DUP FAILED TO CLOCK OUT FIRST BIT
1738 007074 005037 001246 65$: CLR TEMP5 ;SETUP FOR 1'S OUTPUT
1739 007100 104412 000002 1$: PKCLK ,2 ;CONTINUE TO POKE OUT BITS
1740 007104 032777 040000 172300 BIT #MTDATA,@TXCSR ;TEST MAINT DATA OUT
1741 007112 001001 BNE 2$ ;BRANCH IF A 1
1742 007114 104013 HLT 13 ;DUP FAILED TO CLOCK A ONE
1743 007116 005237 001246 2$: INC TEMP5 ;KEEP UP WITH THE # OF 1'S OUTPUT
1744 007122 022737 000006 001246 CMP #6.,TEMP5 ;ARE WE DONE WITH SIX ONES?
1745 007130 001363 BNE 1$ ;NO BRANCH BACK
1746 007132 104412 000002 PKCLK ,2 ;YES,OUTPUT THE LAST 0
1747 007136 032777 040000 172246 BIT #MTDATA,@TXCSR ;CHECK IT
1748 007144 001401 BEQ 3$ ;BRANCH IF A 0
1749 007146 104012 HLT 12 ;LAST BIT OF FLAG WAS NOT CORECT
1750 007150 005337 001244 3$: DEC TEMP4 ;ARE WE DONE WITH FLAGS?
1751 007154 001340 BNE 64$ ;BR IF NO
1752 007156 000201 RTS R1
1753

```

CONTROL REGISTERS ADDRESS TIMEOUT TEST

```
1754                                     :***** TEST 1 *****
1755                                     :*VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE
1756                                     :*A TIME-OUT TRAP.
1757                                     :*****
1758
1759                                     :*****
1760                                     :*
1761                                     : TEST 1
1762                                     :*
1763                                     :*****
1764                                     :*****
1765 007160 012737 000001 001226 TST1: MOV #1,@#TSTNO
1766 007166 012737 007272 001216      MOV #TST2,NEXT
1767 007174 012737 007226 001220      MOV #1$,LOCK
1768 007202 012700 000004          MOV #4,R0          ;SET FOR MAX. 4 PRI. REGISTERS
1769 007206 013701 001404          MOV RXCSR,R1        ;GET FIRST PRI. ADDRESS
1770 007212 012737 007264 000004      MOV #2$,4        ;SET FOR TIME-OUT TRAP.
1771 007220 012737 000340 000006      MOV #340,6        ;SAFE GUARD.
1772 007226 005711          1$: TST (R1)                ;REFERENCE THE ADDRESS.
1773 007230 105711          TSTB (R1)                  ;REFERENCE BYTE ADDRESS.
1774 007232 105761 000001          TSTB 1(R1)
1775 007236 104401          SCOP1          ;IF SW09=1; GOTO 1$
1776 007240 062701 000002          ADD #2,R1          ;UPDATE TO NEXT ADDRESS.
1777 007244 005300          DEC R0          ;ARE ALL ADDRESS CHECKED?
1778 007246 001367          BNE 1$          ;BR IF NO.
1779 007250 012737 000006 000004      MOV #6,4        ;RESET TRAP ZONE.
1780 007256 005037 000006          CLR @#6
1781 007262 104400          SCOPE          ;SCOPE THIS TEST
1782 007264 011602          2$: MOV (SP),R2          ;SAVE THE TRAP PC
1783 007266 104001          HLT 1          ;REPORT TIME-OUT TRAP
1784 007270 000002          RTI          ;RETURN TO MAIN PROGRAM
1785
1786                                     :***** TEST 2 *****
1787                                     :*PRIMARY REGISTER ADDRESSING TEST
1788                                     :*LOAD EACH PRIMARY REGISTER WITH A
1789                                     :*DIFFERENT NUMBER AND VERIFY EACH
1790                                     :*WAS INDIVIDUALLY ADDRESSED
1791                                     :*****
1792
1793                                     :*****
1794                                     :*
1795                                     : TEST 2
1796                                     :*
1797                                     :*****
1798                                     :*****
1799 007272 012737 000002 001226 TST2: MOV #2,@#TSTNO
1800 007300 012737 007536 001216      MOV #TST3,NEXT
1801 007306 012737 007332 001220      MOV #1$,LOCK
1802 007314 012700 007440          MOV #3$,R0          ;SET THE TABLE POINTER
1803 007320 013703 001404          MOV RXCSR,R3        ;SET THE DUP HARDWARE POINTER
1804 007324 005013          CLR (R3)          ;CLR THE REGISTER BEFORE STARTING
1805 007326 012702 000004          MOV #4,R2          ;SET FOR 4 PRIMARY REGISTERS
1806 007332 011005          1$: MOV (R0),R5          ;SET 'EXPECTED'
1807 007334 010513          MOV R5,(R3)          ;WRITE 'EXPECTED' TO THE REGISTER
1808 007336 011304          MOV (R3),R4          ;READ THE REGISTER BACK
1809 007340 042704 000200          BIC #BIT7,R4        ;CLR UNWANTED BIT
```


CONTROL REGISTERS DUAL ADDRESSING TEST

```
1810 007344 042704 000200      BIC    #BIT7,R4      ;CLR UNWANTED BITS
1811 007350 020504              CMP    R5,R4      ;DOES EXPECTED=RECEIVED?
1812 007352 001401              BEQ    2$      ;BR IF YES
1813 007354 104003              HLT    3      ;THIS IS A DATA ERROR. IT IS **NOT**
1814                                ;A DUAL ADDRESSING ERROR!!!!!!
1815 007356 104401      2$:     SCOP1      ;SW09=1?
1816 007360 022023              CMP    (R0)+,(R3)+      ;POP DATA AND HARDWARE POINTERS
1817 007362 005302              DEC    R2      ;UPDATE THE REGISTER COUNTER
1818 007364 001362              BNE    1$      ;BRANCH IF MORE TO GO
1819                                ;:NOW CHECK FOR DUAL ADDRESSING
1820 007366 012700 007440      MOV    #3$,R0      ;SET THE TABLE POINTER
1821 007372 013703 001404      MOV    RXCSR,R3      ;SET THE DUP HARDWARE POINTER
1822 007376 012737 007410 001220      MOV    #4$,LOCK      ;SET FOR SW09=1
1823 007404 012702 000004      MOV    #4,R2      ;SET FOR 4 PRIMARY REGISTERS
1824 007410 011005      4$:     MOV    (R0),R5      ;SET 'EXPECTED'
1825 007412 011304      MOV    (R3),R4      ;READ THE REGISTER BACK
1826 007414 042704 000200      BIC    #BIT7,R4      ;CLR UNWANTED BITS
1827 007420 020504              CMP    R5,R4
1828 007422 001401              BEQ    5$      ;BRANCH IF OK
1829 007424 104003              HLT    3      ;THIS IS A DUAL ADDRESSING ERROR
1830 007426 104401      5$:     SCOP1      ;SW09=1?
1831 007430 022023              CMP    (R0)+,(R3)+      ;POP POINTERS
1832 007432 005302              DEC    R2      ;UPDATE THE REGISTER COUNTER
1833 007434 001365              BNE    4$      ;BRANCH IF MORE TO GO
1834 007436 000404              BR     6$
1835 007440 000020      3$:     .WORD 00020      ;RXCSR
1836 007442 000000      .WORD 00000      ;RXDBUF AND PARCSR
1837 007444 000010      .WORD 00010      ;TXCSR
1838 007446 000100      .WORD 00100      ;TXDBUF
1839                                ;////////////////////////////////////
1840                                ;:NOW CHECK FOR BYTE ADDRESSING
1841 007450 012737 007474 001220      6$:     MOV    #11$,LOCK
1842 007456 012700 007526      MOV    #12$,R0      ;SET TABLE POINTER
1843 007462 013703 001404      MOV    RXCSR,R3      ;SET DUP HARDWARE POINTER
1844 007466 005013              CLR    (R3)      ;CLEAR REG
1845 007470 012702 000010      MOV    #10,R2      ;SET 8 BYTE ADDRESSES
1846 007474 111005      11$:    MOVB   (R0),R5      ;SET EXPECTED
1847 007476 110513      MOVB   R5,(R3)      ;WRITE EXPECTED TO REG
1848 007500 111304      MOVB   (R3),R4      ;READ REG BACK
1849 007502 142704 000200      BICB   #BIT7,R4      ;CLR UNWANTED BITS
1850 007506 120504      CMPB   R5,R4      ;DOES EXPECTED=RECEIVED?
1851 007510 001401              BEQ    10$      ;IF YES,BR
1852 007512 104003              HLT    3      ;DATA ERROR
1853 007514 104401      10$:    SCOP1
1854 007516 122023      CMPB   (R0)+,(R3)+      ;POP POINTERS
1855 007520 005302              DEC    R2      ;UPDATE REG CNTR
1856 007522 001364              BNE    11$      ;BR IF MORE TO GO
1857 007524 104400      SCOPE
1858 007526      020      12$:    .BYTE 020      ;LO BYTE RXCSR
1859 007527      001      .BYTE 001      ;HI BYTE RXCSR
1860 007530      000      .BYTE 000      ;LO BYTE RXDBUF AND PARCSR
1861 007531      000      .BYTE 000      ;HI BYTE RXDBUF AND PARCSR
1862 007532      010      .BYTE 010      ;LO BYTE TXCSR
1863 007533      000      .BYTE 000      ;HI BYTE TXCSR
1864 007534      100      .BYTE 100      ;LO BYTE TXDBUF
1865 007535      002      .BYTE 002      ;HI BYTE TXDBUF
```

CONTROL REGISTERS DUAL ADDRESSING TEST

://

```
***** TEST 3 *****
*RECEIVER CONTROL REGISTER RESET TEST. TEST THAT AFTER
*RECEIVER CONTROL REGISTER IS WRITTEN AND A BUS RESET IS
*DONE THAT RECEIVER CONTROL REGISTER IS CLEARED.
*****
```

:*****

:*

: TEST 3

:*

:*****

:*****

```
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881 007536 012737 000003 001226
1882 007544 012737 007600 001216
1883 007552 005005
1884 007554 013703 001404
1885 007560 012713 177777
1886 007564 000005
1887 007566 011304
1888 007570 020504
1889 007572 001401
1890 007574 104002
1891 007576 104400
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905 007600 012737 000004 001226
1906 007606 012737 007636 001216
1907 007614 005005
1908 007616 013703 001406
1909 007622 000005
1910 007624 011304
1911 007626 020504
1912 007630 001401
1913 007632 104002
1914 007634 104400
1915
1916
1917
1918
1919
1920
1921
```

```
TST3:  MOV    #3,@#TSTNO
        MOV    #TST4,NEXT
        CLR     R5                ;SET 'EXPECTED'
        MOV    RXCSR,R3          ;GET THE RECEIVER CONTROL REGISTER
        MOV    #-1,(R3)         ;LOAD RECEIVER CONTROL REGISTER WITH ALL ONES
        RESET
        MOV     (R3),R4          ;READ THE RECEIVER CONTROL REGISTER
        CMP     R5,R4           ;R5=GOOD,R4= ?
        BEQ     1$              ;BR IF OK
        HLT     2               ;COMPARISON ERROR
1$:     SCOPE                    ;SCOPE THIS TEST
```

```
***** TEST 4 *****
*RECEIVER BUFFER REGISTER RESET TEST. TEST THAT AFTER A BUS
*RESET IS DONE THAT RECEIVER BUFFER REGISTER IS CLEARED.
*****
```

:*****

:*

: TEST 4

:*

:*****

:*****

```
1905 007600 012737 000004 001226
1906 007606 012737 007636 001216
1907 007614 005005
1908 007616 013703 001406
1909 007622 000005
1910 007624 011304
1911 007626 020504
1912 007630 001401
1913 007632 104002
1914 007634 104400
1915
1916
1917
1918
1919
1920
1921
```

```
TST4:  MOV    #4,@#TSTNO
        MOV    #TST5,NEXT
        CLR     R5                ;SET 'EXPECTED'
        MOV    RXDBUF,R3         ;GET THE RECEIVER BUFFER REGISTER
        RESET
        MOV     (R3),R4          ;READ THE RECEIVER BUFFER REGISTER
        CMP     R5,R4           ;R5=GOOD,R4= ?
        BEQ     1$              ;BR IF OK
        HLT     2               ;COMPARISON ERROR
1$:     SCOPE                    ;SCOPE THIS TEST
```

```
***** TEST 5 *****
*PARAMETER STATUS REGISTER RESET TEST. TEST THAT AFTER
*PARAMETER STATUS REGISTER IS WRITTEN AND A BUS RESET IS
*DONE THAT PARAMETER STATUS REGISTER IS CLEARED.
*****
```


1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977

007636	012737	000005	001226
007644	012737	007700	001216
007652	005005		
007654	013703	001410	
007660	012713	177777	
007664	000005		
007666	011304		
007670	020504		
007672	001401		
007674	104002		
007676	104400		
007700	012737	000006	001226
007706	012737	007746	001216
007714	005005		
007716	013703	001412	
007722	012713	177777	
007726	000005		
007730	011304		
007732	042704	000200	
007736	020504		
007740	001401		
007742	104002		
007744	104400		

```

:*****
:
: TEST 5
:
:*****
:*****
TST5:  MOV    #5, @#TSTNO
      MOV    #TST6, NEXT
      CLR    R5
      MOV    PARCSR, R3
      MOV    #-1, (R3)
      RESET
      MOV    (R3), R4
      CMP    R5, R4
      BEQ    1$
      HLT    2
1$:    SCOPE
      :SET 'EXPECTED'
      :GET  THE PARAMETER STATUS REGISTER
      :LOAD PARAMETER STATUS REGISTER WITH ALL ONES
      :READ THE PARAMETER STATUS REGISTER
      :R5=GOOD, R4= ?
      :BR IF OK
      :COMPARISON ERROR
      :SCOPE THIS TEST
  
```

```

:***** TEST 6 *****
: *TRANSMITTER CONTROL REGISTER RESET TEST. TEST THAT AFTER
: *TRANSMITTER CONTROL REGISTER IS WRITTEN AND A BUS RESET IS
: *DONE THAT TRANSMITTER CONTROL REGISTER IS CLEARED.
:*****
  
```

```

:*****
:
: TEST 6
:
:*****
:*****
TST6:  MOV    #6, @#TSTNO
      MOV    #TST7, NEXT
      CLR    R5
      MOV    TXCSR, R3
      MOV    #-1, (R3)
      RESET
      MOV    (R3), R4
      BIC    #BIT7, R4
      CMP    R5, R4
      BEQ    1$
      HLT    2
1$:    SCOPE
      :SET 'EXPECTED'
      :GET  THE TRANSMITTER CONTROL REGISTER
      :LOAD TRANSMITTER CONTROL REGISTER WITH ALL ONES
      :READ THE TRANSMITTER CONTROL REGISTER
      :CLR UNWANTED BITS
      :R5=GOOD, R4= ?
      :BR IF OK
      :COMPARISON ERROR
      :SCOPE THIS TEST
  
```

```

:***** TEST 7 *****
: *TRANSMITTER BUFFER REGISTER RESET TEST. TEST THAT AFTER
: *TRANSMITTER BUFFER REGISTER IS WRITTEN AND A BUS RESET IS
: *DONE THAT TRANSMITTER BUFFER REGISTER IS CLEARED.
:*****
  
```

```

:*****
:
: TEST 7
:
:*****
  
```

```

1978
1979
1980 007746 012737 000007 001226 TST7: MOV #7,@TSTNO
1981 007754 012737 010010 001216 MOV #TST10,NEXT
1982 007762 005005 CLR R5 ;SET 'EXPECTED'
1983 007764 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
1984 007770 012713 177777 MOV #-1,(R3) ;LOAD TRANSMITTER BUFFER REGISTER WITH ALL ONES
1985 007774 000005 RESET
1986 007776 011304 MOV (R3),R4 ;READ THE TRANSMITTER BUFFER REGISTER
1987 010000 020504 CMP R5,R4 ;R5=GOOD,R4=?
1988 010002 001401 BEQ 1$ ;BR IF OK
1989 010004 104002 HLT 2 ;COMPARISON ERROR
1990 010006 104400 1$: SCOPE ;SCOPE THIS TEST
1991
1992
1993
1994
1995
1996
1997
1998
1999

```

```

***** TEST 10 *****
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
*SET BIT3, VERIFY BIT3 WAS SET.
*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
*****

```

```

2000
2001
2002
2003
2004
2005 010010 012737 000010 001226 TST10: MOV #10,@TSTNO
2006 010016 012737 010074 001216 MOV #TST11,NEXT
2007 010024 013703 001412 MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
2008 010030 012705 000010 MOV #BIT3,R5 ;SET 'EXPECTED'
2009 010034 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2010 010036 011304 MOV (R3),R4 ;READ THE REGISTER.
2011 010040 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
2012 010044 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2013 010046 001401 BEQ 1$ ;ARE THEY THE SAME?
2014 010050 104003 HLT 3 ;COMPARISON ERROR.
2015 010052 040513 1$: BIC R5,(R3) ;CLEAR BIT3
2016 010054 011304 MOV (R3),R4 ;READ THE REGISTER.
2017 010056 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
2018 010062 005005 CLR R5 ;SET 'EXPECTED'
2019 010064 020504 CMP R5,R4 ;R5=GOOD; R4=?
2020 010066 001401 BEQ 2$ ;BR IF OK
2021 010070 104003 HLT 3 ;COMPARISON ERROR
2022 010072 104400 2$: SCOPE ;SCOPE THIS TEST
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033

```

```

***** TEST 11 *****
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
*SET BIT4, VERIFY BIT4 WAS SET.
*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
*****

```

```

*****
*
TEST 11

```



```

2034
2035
2036
2037 010074 012737 000011 001226 TST11: MOV #11, @TSTNO
2038 010102 012737 010160 001216 MOV #TST12, NEXT
2039 010110 013703 001412 MOV TXCSR, R3 ;SET REGISTER TO BE TESTED.
2040 010114 012705 000020 MOV #BIT4, R5 ;SET 'EXPECTED'
2041 010120 010513 MOV R5, (R3) ;WRITE THE REGISTER.
2042 010122 011304 MOV (R3), R4 ;READ THE REGISTER.
2043 010124 042704 000200 BIC #BIT7, R4 ;CLEAR UNWANTED BITS
2044 010130 020504 CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
2045 010132 001401 BEQ 1$ ;ARE THEY THE SAME?
2046 010134 104003 HLT 3 ;COMPARISON ERROR.
2047 010136 040513 1$: BIC R5, (R3) ;CLEAR BIT4
2048 010140 011304 MOV (R3), R4 ;READ THE REGISTER.
2049 010142 042704 000200 BIC #BIT7, R4 ;CLEAR UNWANTED BITS
2050 010146 005005 CLR R5 ;SET 'EXPECTED'
2051 010150 020504 CMP R5, R4 ;R5=GOOD; R4=?
2052 010152 001401 BEQ 2$ ;BR IF OK
2053 010154 104003 HLT 3 ;COMPARISON ERROR
2054 010156 104400 2$: SCOPE ;SCOPE THIS TEST
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068

```

```

***** TEST 12 *****
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
*SET BIT10, VERIFY BIT10 WAS SET.
*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
*****

```

```

2069 010160 012737 000012 001226 TST12: MOV #12, @TSTNO
2070 010166 012737 010244 001216 MOV #TST13, NEXT
2071 010174 013703 001412 MOV TXCSR, R3 ;SET REGISTER TO BE TESTED.
2072 010200 012705 002000 MOV #BIT10, R5 ;SET 'EXPECTED'
2073 010204 010513 MOV R5, (R3) ;WRITE THE REGISTER.
2074 010206 011304 MOV (R3), R4 ;READ THE REGISTER.
2075 010210 042704 000200 BIC #BIT7, R4 ;CLEAR UNWANTED BITS
2076 010214 020504 CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
2077 010216 001401 BEQ 1$ ;ARE THEY THE SAME?
2078 010220 104003 HLT 3 ;COMPARISON ERROR.
2079 010222 040513 1$: BIC R5, (R3) ;CLEAR BIT10
2080 010224 011304 MOV (R3), R4 ;READ THE REGISTER.
2081 010226 042704 000200 BIC #BIT7, R4 ;CLEAR UNWANTED BITS
2082 010232 005005 CLR R5 ;SET 'EXPECTED'
2083 010234 020504 CMP R5, R4 ;R5=GOOD; R4=?
2084 010236 001401 BEQ 2$ ;BR IF OK
2085 010240 104003 HLT 3 ;COMPARISON ERROR
2086 010242 104400 2$: SCOPE ;SCOPE THIS TEST
2087
2088
2089

```

```

***** TEST 13 *****

```

TRANSMITTER CONTROL REGISTER READ/WRITE TEST BIT 11

```
2090      ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
2091      ;*SET BIT11, VERIFY BIT11 WAS SET.
2092      ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
2093      ;*****
2094
2095      ;*****
2096      ;
2097      ; TEST 13
2098      ;
2099      ;*****
2100      ;*****
```

```
2101 010244 012737 000013 001226 TST13: MOV #13,@#TSTNO
2102 010252 012737 010330 001216 MOV #TST14,NEXT
2103 010260 013703 001412 MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
2104 010264 012705 004000 MOV #BIT11,R5 ;SET 'EXPECTED'
2105 010270 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2106 010272 011304 MOV (R3),R4 ;READ THE REGISTER.
2107 010274 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
2108 010300 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2109 010302 001401 BEQ 1$ ;ARE THEY THE SAME?
2110 010304 104003 HLT 3 ;COMPARISON ERROR.
2111 010306 040513 1$: BIC R5,(R3) ;CLEAR BIT11
2112 010310 011304 MOV (R3),R4 ;READ THE REGISTER.
2113 010312 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
2114 010316 005005 CLR R5 ;SET 'EXPECTED'
2115 010320 020504 CMP R5,R4 ;R5=GOOD; R4=?
2116 010322 001401 BEQ 2$ ;BR IF OK
2117 010324 104003 HLT 3 ;COMPARISON ERROR
2118 010326 104400 2$: SCOPE ;SCOPE THIS TEST
2119
2120
2121
```

```
2122      ;***** TEST 14 *****
2123      ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
2124      ;*SET BIT12, VERIFY BIT12 WAS SET.
2125      ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
2126      ;*****
2127
2128      ;*****
2129      ;
2130      ; TEST 14
2131      ;
2132      ;*****
2133      ;*****
```

```
2133 010330 012737 000014 001226 TST14: MOV #14,@#TSTNO
2134 010336 012737 010414 001216 MOV #TST15,NEXT
2135 010344 013703 001412 MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
2136 010350 012705 010000 MOV #BIT12,R5 ;SET 'EXPECTED'
2137 010354 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2138 010356 011304 MOV (R3),R4 ;READ THE REGISTER.
2139 010360 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
2140 010364 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2141 010366 001401 BEQ 1$ ;ARE THEY THE SAME?
2142 010370 104003 HLT 3 ;COMPARISON ERROR.
2143 010372 040513 1$: BIC R5,(R3) ;CLEAR BIT12
2144 010374 011304 MOV (R3),R4 ;READ THE REGISTER.
2145 010376 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
```


TRANSMITTER CONTROL REGISTER READ/WRITE TEST BIT 12

```
2146 010402 005005          CLR    R5          ;SET 'EXPECTED'
2147 010404 020504          CMP    R5,R4        ;R5=GOOD; R4=?
2148 010406 001401          BEQ    2$          ;BR IF OK
2149 010410 104003          HLT    3          ;COMPARISON ERROR
2150 010412 104400          2$: SCOPE          ;SCOPE THIS TEST
```

```
***** TEST 15 *****
;TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
;SET BIT13, VERIFY BIT13 WAS SET.
;CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
*****
```

```
*****
;
; TEST 15
;
*****
```

```
*****
;*****
TST15: MOV    #15,@TSTNO
        MOV    #TST16,NEXT
        MOV    TXCSR,R3          ;SET REGISTER TO BE TESTED.
        MOV    #BIT13,R5        ;SET 'EXPECTED'
        MOV    R5,(R3)          ;WRITE THE REGISTER.
        MOV    (R3),R4          ;READ THE REGISTER.
        BIC    #BIT7,R4        ;CLEAR UNWANTED BITS
        CMP    R5,R4            ;R5=GOOD; R4=UNKNOWN.
        BEQ    1$              ;ARE THEY THE SAME?
        HLT    3                ;COMPARISON ERROR.
        1$: BIC    R5,(R3)        ;CLEAR BIT13
        MOV    (R3),R4          ;READ THE REGISTER.
        BIC    #BIT7,R4        ;CLEAR UNWANTED BITS
        CLR    R5              ;SET 'EXPECTED'
        CMP    R5,R4            ;R5=GOOD; R4=?
        BEQ    2$              ;BR IF OK
        HLT    3                ;COMPARISON ERROR
        2$: SCOPE              ;SCOPE THIS TEST
```

```
***** TEST 16 *****
;TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;SET BIT0, VERIFY BIT0 WAS SET.
;CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
*****
```

```
*****
;
; TEST 16
;
*****
```

```
*****
;*****
TST16: MOV    #16,@TSTNO
        MOV    #TST17,NEXT
        MOV    TXDBUF,R3        ;SET REGISTER TO BE TESTED.
        MOV    #BIT0,R5        ;SET 'EXPECTED'
        MOV    R5,(R3)          ;WRITE THE REGISTER.
```

TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 0

```

2202 010526 011304      MOV      (R3),R4      ;READ THE REGISTER.
2203 010530 020504      CMP      R5,R4      ;R5=GOOD; R4=UNKNOWN.
2204 010532 001401      BEQ      1$          ;ARE THEY THE SAME?
2205 010534 104003      HLT      3          ;COMPARISON ERROR.
2206 010536 040513      1$: BIC      R5,(R3)      ;CLEAR BIT0
2207 010540 011304      MOV      (R3),R4      ;READ THE REGISTER.
2208 010542 005005      CLR      R5          ;SET 'EXPECTED'
2209 010544 020504      CMP      R5,R4      ;R5=GOOD; R4=?
2210 010546 001401      BEQ      2$          ;BR IF OK
2211 010550 104003      HLT      3          ;COMPARISON ERROR
2212 010552 104400      2$: SCOPE      ;SCOPE THIS TEST
  
```

```

;***** TEST 17 *****
;*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;*SET BIT1, VERIFY BIT1 WAS SET.
;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
;*****
  
```

```

;*****
;:
;: TEST 17
;:
;:
;*****
  
```

```

2227 010554 012737 000017 001226 TST17: MOV      #17,@TSTNO
2228 010562 012737 010630 001216 MOV      #TST20,NEXT
2229 010570 013703 001414      MOV      TXDBUF,R3      ;SET REGISTER TO BE TESTED.
2230 010574 012705 000002      MOV      #BIT1,R5      ;SET 'EXPECTED'
2231 010600 010513      MOV      R5,(R3)      ;WRITE THE REGISTER.
2232 010602 011304      MOV      (R3),R4      ;READ THE REGISTER.
2233 010604 020504      CMP      R5,R4      ;R5=GOOD; R4=UNKNOWN.
2234 010606 001401      BEQ      1$          ;ARE THEY THE SAME?
2235 010610 104003      HLT      3          ;COMPARISON ERROR.
2236 010612 040513      1$: BIC      R5,(R3)      ;CLEAR BIT1
2237 010614 011304      MOV      (R3),R4      ;READ THE REGISTER.
2238 010616 005005      CLR      R5          ;SET 'EXPECTED'
2239 010620 020504      CMP      R5,R4      ;R5=GOOD; R4=?
2240 010622 001401      BEQ      2$          ;BR IF OK
2241 010624 104003      HLT      3          ;COMPARISON ERROR
2242 010626 104400      2$: SCOPE      ;SCOPE THIS TEST
  
```

```

;***** TEST 20 *****
;*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;*SET BIT2, VERIFY BIT2 WAS SET.
;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
;*****
  
```

```

;*****
;:
;: TEST 20
;:
;:
;*****
  
```

```

2257 010630 012737 000020 001226 TST20: MOV      #20,@TSTNO
  
```


TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 2

```
2258 010636 012737 010704 001216      MOV    #TST21,NEXT
2259 010644 013703 001414      MOV    TXDBUF,R3          ;SET REGISTER TO BE TESTED.
2260 010650 012705 000004      MOV    #BIT2,R5          ;SET 'EXPECTED'
2261 010654 010513      MOV    R5,(R3)          ;WRITE THE REGISTER.
2262 010656 011304      MOV    (R3),R4          ;READ THE REGISTER.
2263 010660 020504      CMP     R5,R4          ;R5=GOOD; R4=UNKNOWN.
2264 010662 001401      BEQ    1$              ;ARE THEY THE SAME?
2265 010664 104003      HLT     3              ;COMPARISON ERROR.
2266 010666 040513      1$:    BIC     R5,(R3)          ;CLEAR BIT2
2267 010670 011304      MOV    (R3),R4          ;READ THE REGISTER.
2268 010672 005005      CLR     R5              ;SET 'EXPECTED'
2269 010674 020504      CMP     R5,R4          ;R5=GOOD; R4=?
2270 010676 001401      BEQ    2$              ;BR IF OK
2271 010700 104003      HLT     3              ;COMPARISON ERROR
2272 010702 104400      2$:    SCOPE          ;SCOPE THIS TEST
```

```
***** TEST 21 *****
;TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;SET BIT3, VERIFY BIT3 WAS SET.
;CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
*****
```

```
*****
;
; TEST 21
;
*****
```

```
2286 *****
2287 010704 012737 000021 001226 TST21: MOV    #21,@TSTNO
2288 010712 012737 010760 001216      MOV    #TST22,NEXT
2289 010720 013703 001414      MOV    TXDBUF,R3          ;SET REGISTER TO BE TESTED.
2290 010724 012705 000010      MOV    #BIT3,R5          ;SET 'EXPECTED'
2291 010730 010513      MOV    R5,(R3)          ;WRITE THE REGISTER.
2292 010732 011304      MOV    (R3),R4          ;READ THE REGISTER.
2293 010734 020504      CMP     R5,R4          ;R5=GOOD; R4=UNKNOWN.
2294 010736 001401      BEQ    1$              ;ARE THEY THE SAME?
2295 010740 104003      HLT     3              ;COMPARISON ERROR.
2296 010742 040513      1$:    BIC     R5,(R3)          ;CLEAR BIT3
2297 010744 011304      MOV    (R3),R4          ;READ THE REGISTER.
2298 010746 005005      CLR     R5              ;SET 'EXPECTED'
2299 010750 020504      CMP     R5,R4          ;R5=GOOD; R4=?
2300 010752 001401      BEQ    2$              ;BR IF OK
2301 010754 104003      HLT     3              ;COMPARISON ERROR
2302 010756 104400      2$:    SCOPE          ;SCOPE THIS TEST
```

```
***** TEST 22 *****
;TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;SET BIT4, VERIFY BIT4 WAS SET.
;CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
*****
```

```
*****
;
; TEST 22
;
```

TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 4

```
2314
2315
2316
2317 010760 012737 000022 001226 TST22: MOV #22,@TSTNO
2318 010766 012737 011034 001216 MOV #TST23,NEXT
2319 010774 013703 001414 MOV TXDBUF,R3 ;SET REGISTER TO BE TESTED.
2320 011000 012705 000020 MOV #BIT4,R5 ;SET 'EXPECTED'
2321 011004 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2322 011006 011304 MOV (R3),R4 ;READ THE REGISTER.
2323 011010 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2324 011012 001401 BEQ 1$ ;ARE THEY THE SAME?
2325 011014 104003 HLT 3 ;COMPARISON ERROR.
2326 011016 040513 1$: BIC R5,(R3) ;CLEAR BIT4
2327 011020 011304 MOV (R3),R4 ;READ THE REGISTER.
2328 011022 005005 CLR R5 ;SET 'EXPECTED'
2329 011024 020504 CMP R5,R4 ;R5=GOOD; R4=?
2330 011026 001401 BEQ 2$ ;BR IF OK
2331 011030 104003 HLT 3 ;COMPARISON ERROR
2332 011032 104400 2$: SCOPE ;SCOPE THIS TEST
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
```

```
***** TEST 23 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BITS, VERIFY BITS WAS SET.
*CLEAR BITS, VERIFY BITS WAS CLEARED.
*****
```

```
2347 011034 012737 000023 001226 TST23: MOV #23,@TSTNO
2348 011042 012737 011110 001216 MOV #TST24,NEXT
2349 011050 013703 001414 MOV TXDBUF,R3 ;SET REGISTER TO BE TESTED.
2350 011054 012705 000040 MOV #BIT5,R5 ;SET 'EXPECTED'
2351 011060 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2352 011062 011304 MOV (R3),R4 ;READ THE REGISTER.
2353 011064 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2354 011066 001401 BEQ 1$ ;ARE THEY THE SAME?
2355 011070 104003 HLT 3 ;COMPARISON ERROR.
2356 011072 040513 1$: BIC R5,(R3) ;CLEAR BITS
2357 011074 011304 MOV (R3),R4 ;READ THE REGISTER.
2358 011076 005005 CLR R5 ;SET 'EXPECTED'
2359 011100 020504 CMP R5,R4 ;R5=GOOD; R4=?
2360 011102 001401 BEQ 2$ ;BR IF OK
2361 011104 104003 HLT 3 ;COMPARISON ERROR
2362 011106 104400 2$: SCOPE ;SCOPE THIS TEST
2363
2364
2365
2366
2367
2368
2369
```

```
***** TEST 24 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT6, VERIFY BIT6 WAS SET.
*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
*****
```


TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 6

2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425:*****
: *
: TEST 24
: *

:*****

:*****

```
TST24:  MOV    #24,@TSTNO
        MOV    #TST25,NEXT
        MOV    TXDBUF,R3      ;SET REGISTER TO BE TESTED.
        MOV    #BIT6,R5      ;SET 'EXPECTED'
        MOV    R5,(R3)        ;WRITE THE REGISTER.
        MOV    (R3),R4        ;READ THE REGISTER.
        CMP    R5,R4          ;R5=GOOD; R4=UNKNOWN.
        BEQ    1$             ;ARE THEY THE SAME?
        HLT    3              ;COMPARISON ERROR.
1$:      BIC    R5,(R3)        ;CLEAR BIT6
        MOV    (R3),R4        ;READ THE REGISTER.
        CLR    R5             ;SET 'EXPECTED'
        CMP    R5,R4          ;R5=GOOD; R4=?
        BEQ    2$             ;BR IF OK
        HLT    3              ;COMPARISON ERROR
2$:      SCOPE                 ;SCOPE THIS TEST
```

```
:***** TEST 25 *****
;*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
;*SET BIT7, VERIFY BIT7 WAS SET.
;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED.
:*****
```

:*****
: *
: TEST 25
: *

:*****

:*****

```
TST25:  MOV    #25,@TSTNO
        MOV    #TST26,NEXT
        MOV    TXDBUF,R3      ;SET REGISTER TO BE TESTED.
        MOV    #BIT7,R5      ;SET 'EXPECTED'
        MOV    R5,(R3)        ;WRITE THE REGISTER.
        MOV    (R3),R4        ;READ THE REGISTER.
        CMP    R5,R4          ;R5=GOOD; R4=UNKNOWN.
        BEQ    1$             ;ARE THEY THE SAME?
        HLT    3              ;COMPARISON ERROR.
1$:      BIC    R5,(R3)        ;CLEAR BIT7
        MOV    (R3),R4        ;READ THE REGISTER.
        CLR    R5             ;SET 'EXPECTED'
        CMP    R5,R4          ;R5=GOOD; R4=?
        BEQ    2$             ;BR IF OK
        HLT    3              ;COMPARISON ERROR
2$:      SCOPE                 ;SCOPE THIS TEST
```

:***** TEST 26 *****

TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 8

```
2426 ;*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
2427 ;*SET BIT8, VERIFY BIT8 WAS SET.
2428 ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
2429 ;:*****
2430
2431 ;:*****
2432 ;*
2433 ;: TEST 26
2434 ;*
2435 ;:*****
2436 ;:*****
2437 011240 012737 000026 001226 TST26: MOV #26,@TSTNO
2438 011246 012737 011314 001216 MOV #TST27,NEXT
2439 011254 013703 001414 MOV TXDBUF,R3 ;SET REGISTER TO BE TESTED.
2440 011260 012705 000400 MOV #BIT8,R5 ;SET 'EXPECTED'
2441 011264 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2442 011266 011304 MOV (R3),R4 ;READ THE REGISTER.
2443 011270 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2444 011272 001401 BEQ 1$ ;ARE THEY THE SAME?
2445 011274 104003 HLT 3 ;COMPARISON ERROR.
2446 011276 040513 1$: BIC R5,(R3) ;CLEAR BIT8
2447 011300 011304 MOV (R3),R4 ;READ THE REGISTER.
2448 011302 005005 CLR R5 ;SET 'EXPECTED'
2449 011304 020504 CMP R5,R4 ;R5=GOOD; R4=?
2450 011306 001401 BEQ 2$ ;BR IF OK
2451 011310 104003 HLT 3 ;COMPARISON ERROR
2452 011312 104400 2$: SCOPE ;SCOPE THIS TEST
2453
2454
2455 ;***** TEST 27 *****
2456 ;*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
2457 ;*SET BIT9, VERIFY BIT9 WAS SET.
2458 ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
2459 ;:*****
2460
2461 ;:*****
2462 ;*
2463 ;: TEST 27
2464 ;*
2465 ;:*****
2466 ;:*****
2467 011314 012737 000027 001226 TST27: MOV #27,@TSTNO
2468 011322 012737 011370 001216 MOV #TST30,NEXT
2469 011330 013703 001414 MOV TXDBUF,R3 ;SET REGISTER TO BE TESTED.
2470 011334 012705 001000 MOV #BIT9,R5 ;SET 'EXPECTED'
2471 011340 010513 MOV R5,(R3) ;WRITE THE REGISTER.
2472 011342 011304 MOV (R3),R4 ;READ THE REGISTER.
2473 011344 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
2474 011346 001401 BEQ 1$ ;ARE THEY THE SAME?
2475 011350 104003 HLT 3 ;COMPARISON ERROR.
2476 011352 040513 1$: BIC R5,(R3) ;CLEAR BIT9
2477 011354 011304 MOV (R3),R4 ;READ THE REGISTER.
2478 011356 005005 CLR R5 ;SET 'EXPECTED'
2479 011360 020504 CMP R5,R4 ;R5=GOOD; R4=?
2480 011362 001401 BEQ 2$ ;BR IF OK
2481 011364 104003 HLT 3 ;COMPARISON ERROR
```


TRANSMITTER BUFFER REGISTER READ/WRITE TEST BIT 9

2482 011366 104400

2\$: SCOPE

;SCOPE THIS TEST

2483

2484

2485

2486

2487

2488

2489

2490

2491

2492

2493

2494

2495

2496

2497

2498

2499

2500

2501

2502

2503

2504

2505

2506

2507

2508

2509

2510

2511

2512

2513

2514

2515

2516

2517

2518

2519

2520

2521

2522

2523

2524

2525

2526

2527

2528

2529

2530

2531

2532

2533

2534

2535

2536

2537

011370 012737 000030 001226
011376 012737 011444 001216
011404 013703 001414
011410 012705 002000
011414 010513
011416 011304
011420 020504
011422 001401
011424 104003
011426 040513
011430 011304
011432 005005
011434 020504
011436 001401
011440 104003
011442 104400

***** TEST 30 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT10, VERIFY BIT10 WAS SET.
*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.

:*****

:*

:TEST 30

:*

:*****

:*****

TST30: MOV #30,@TSTNO
MOV #TST31,NEXT
MOV TXDBUF,R3 ;SET REGISTER TO BE TESTED.
MOV #BIT10,R5 ;SET 'EXPECTED'
MOV R5,(R3) ;WRITE THE REGISTER.
MOV (R3),R4 ;READ THE REGISTER.
CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
BEQ 1\$;ARE THEY THE SAME?
HLT 3 ;COMPARISON ERROR.
1\$: BIC R5,(R3) ;CLEAR BIT10
MOV (R3),R4 ;READ THE REGISTER.
CLR R5 ;SET 'EXPECTED'
CMP R5,R4 ;R5=GOOD; R4=?
BEQ 2\$;BR IF OK
HLT 3 ;COMPARISON ERROR
2\$: SCOPE ;SCOPE THIS TEST

***** TEST 31 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 1 RESET AND CLEAR TEST
*WRITE BIT 1,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION

:*****

:*

:TEST 31

:*

:*****

:*****

011444 012737 000031 001226
011452 012737 011540 001216
011460 013703 001404
011464 052713 000002
011470 005005
011472 052777 000400 167712
011500 004737 005044
011504 032713 000002
011510 001402
011512 011304
011514 104003

TST31: MOV #31,@TSTNO
MOV #TST32,NEXT
MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
BIS #BIT1,(R3) ;SET BIT 1 AT RECEIVER CONTROL REGISTER
CLR R5 ;SET 'EXPECTED'
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT1,(R3) ;TEST BIT 1
BEQ 1\$;BIT 1 IS CLEARED
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 1 IS SET AND SHOULDN'T BE

RECEIVER CONTROL REGISTER RESET AND CLEAR TEST BIT 1

```
2538 011516 052713 000002 1$: BIS #BIT1,(R3) ;SET BIT 1 AGAIN
2539 011522 005013 CLR (R3) ;CLEAR THE RECEIVER CONTROL REGISTER
2540 011524 032713 000002 BIT #BIT1,(R3) ;TEST TO SEE IF BIT 1 CLEARED
2541 011530 001402 BEQ 2$ ;BIT 1 IS OK
2542 011532 011304 MOV (R3),R4 ;LOAD 'FOUND'
2543 011534 104003 HLT 3 ;BIT 1 FAILED TO CLEAR
2544 011536 104400 2$: SCOPE ;SCOPE THIS TEST
2545
2546
2547
```

```
***** TEST 32 *****
;*RECEIVER CONTROL REGISTER READ/WRITE BIT 2 RESET AND CLEAR TEST
;*WRITE BIT 2,AND TEST THAT IT WILL BE CLEARED AFTER A
;*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

```
*****
```

```
TEST 32
```

```
*****
```

```
*****
```

```
2559 011540 012737 000032 001226 TST32: MOV #32,@#TSTNO
2560 011546 012737 011634 001216 MOV #TST33,NEXT
2561 011554 013703 001404 MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
2562 011560 052713 000004 BIS #BIT2,(R3) ;SET BIT 2 AT RECEIVER CONTROL REGISTER
2563 011564 005005 CLR R5 ;SET 'EXPECTED'
2564 011566 052777 000400 167616 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2565 011574 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2566 011600 032713 000004 BIT #BIT2,(R3) ;TEST BIT 2
2567 011604 001402 BEQ 1$ ;BIT 2 IS CLEARED
2568 011606 011304 MOV (R3),R4 ;LOAD 'FOUND'
2569 011610 104003 HLT 3 ;BIT 2 IS SET AND SHOULDN'T BE
2570 011612 052713 000004 1$: BIS #BIT2,(R3) ;SET BIT 2 AGAIN
2571 011616 005013 CLR (R3) ;CLEAR THE RECEIVER CONTROL REGISTER
2572 011620 032713 000004 BIT #BIT2,(R3) ;TEST TO SEE IF BIT 2 CLEARED
2573 011624 001402 BEQ 2$ ;BIT 2 IS OK
2574 011626 011304 MOV (R3),R4 ;LOAD 'FOUND'
2575 011630 104003 HLT 3 ;BIT 2 FAILED TO CLEAR
2576 011632 104400 2$: SCOPE ;SCOPE THIS TEST
2577
2578
```

```
***** TEST 33 *****
;*RECEIVER CONTROL REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST
;*WRITE BIT 3,AND TEST THAT IT WILL BE CLEARED AFTER A
;*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

```
*****
```

```
TEST 33
```

```
*****
```

```
*****
```

```
2591 011634 012737 000033 001226 TST33: MOV #33,@#TSTNO
2592 011642 012737 011730 001216 MOV #TST34,NEXT
2593 011650 013703 001404 MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
```


RECEIVER CONTROL REGISTER RESET AND CLEAR TEST BIT 3

```
2594 011654 052713 000010      BIS    #BIT3,(R3)      ;SET BIT 3 AT RECEIVER CONTROL REGISTER
2595 011660 005005      CLR    R5              ;SET 'EXPECTED'
2596 011662 052777 000400 167522  BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2597 011670 004737 005044      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2598 011674 032713 000010      BIT    #BIT3,(R3)      ;TEST BIT 3
2599 011700 001402      BEQ    1$              ;BIT 3 IS CLEARED
2600 011702 011304      MOV    (R3),R4        ;LOAD 'FOUND'
2601 011704 104003      HLT    3              ;BIT 3 IS SET AND SHOULDN'T BE
2602 011706 052713 000010 1$:    BIS    #BIT3,(R3)      ;SET BIT 3 AGAIN
2603 011712 005013      CLR    (R3)          ;CLEAR THE RECEIVER CONTROL REGISTER
2604 011714 032713 000010      BIT    #BIT3,(R3)      ;TEST TO SEE IF BIT 3 CLEARED
2605 011720 001402      BEQ    2$              ;BIT 3 IS OK
2606 011722 011304      MOV    (R3),R4        ;LOAD 'FOUND'
2607 011724 104003      HLT    3              ;BIT 3 FAILED TO CLEAR
2608 011726 104400 2$:    SCOPE              ;SCOPE THIS TEST
```

```
***** TEST 34 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST
*WRITE BIT 4,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

```
*****
*
* TEST 34
*
*****
```

```
*****
*****
2623 011730 012737 000034 001226 TST34: MOV    #34,@TSTNO
2624 011736 012737 012024 001216  MOV    #TST35,NEXT
2625 011744 013703 001404      MOV    RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
2626 011750 052713 000020      BIS    #BIT4,(R3)      ;SET BIT 4 AT RECEIVER CONTROL REGISTER
2627 011754 005005      CLR    R5              ;SET 'EXPECTED'
2628 011756 052777 000400 167426  BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
2629 011764 004737 005044      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
2630 011770 032713 000020      BIT    #BIT4,(R3)      ;TEST BIT 4
2631 011774 001402      BEQ    1$              ;BIT 4 IS CLEARED
2632 011776 011304      MOV    (R3),R4        ;LOAD 'FOUND'
2633 012000 104003      HLT    3              ;BIT 4 IS SET AND SHOULDN'T BE
2634 012002 052713 000020 1$:    BIS    #BIT4,(R3)      ;SET BIT 4 AGAIN
2635 012006 005013      CLR    (R3)          ;CLEAR THE RECEIVER CONTROL REGISTER
2636 012010 032713 000020      BIT    #BIT4,(R3)      ;TEST TO SEE IF BIT 4 CLEARED
2637 012014 001402      BEQ    2$              ;BIT 4 IS OK
2638 012016 011304      MOV    (R3),R4        ;LOAD 'FOUND'
2639 012020 104003      HLT    3              ;BIT 4 FAILED TO CLEAR
2640 012022 104400 2$:    SCOPE              ;SCOPE THIS TEST
```

```
***** TEST 35 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 8 RESET AND CLEAR TEST
*WRITE BIT 8,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

```
*****
```

RECEIVER CONTROL REGISTER RESET AND CLEAR TEST BIT 8

```
2650
2651      ; TEST 35
2652      ;
2653      ;*****
2654      ;*****
2655 012024 012737 000035 001226 TST35: MOV #35,@TSTNO
2656 012032 012737 012120 001216 MOV #TST36,NEXT
2657 012040 013703 001404 MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
2658 012044 052713 000400 BIS #BIT8,(R3) ;SET BIT 8 AT RECEIVER CONTROL REGISTER
2659 012050 005005 CLR R5 ;SET 'EXPECTED'
2660 012052 052777 000400 167332 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2661 012060 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2662 012064 032713 000400 BIT #BIT8,(R3) ;TEST BIT 8
2663 012070 001402 BEQ 1$ ;BIT 8 IS CLEARED
2664 012072 011304 MOV (R3),R4 ;LOAD 'FOUND'
2665 012074 104003 HLT 3 ;BIT 8 IS SET AND SHOULDN'T BE
2666 012076 052713 000400 1$: BIS #BIT8,(R3) ;SET BIT 8 AGAIN
2667 012102 005013 CLR (R3) ;CLEAR THE RECEIVER CONTROL REGISTER
2668 012104 032713 000400 BIT #BIT8,(R3) ;TEST TO SEE IF BIT 8 CLEARED
2669 012110 001402 BEQ 2$ ;BIT 8 IS OK
2670 012112 011304 MOV (R3),R4 ;LOAD 'FOUND'
2671 012114 104003 HLT 3 ;BIT 8 FAILED TO CLEAR
2672 012116 104400 2$: SCOPE ;SCOPE THIS TEST
2673
2674
2675      ;***** TEST 36 *****
2676      ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST
2677      ;*WRITE BIT 3,AND TEST THAT IT WILL BE CLEARED AFTER A
2678      ;*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
2679      ;*****
2680
2681      ;*****
2682      ;
2683      ; TEST 36
2684      ;
2685      ;*****
2686      ;*****
2687 012120 012737 000036 001226 TST36: MOV #36,@TSTNO
2688 012126 012737 012214 001216 MOV #TST37,NEXT
2689 012134 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
2690 012140 052713 000010 BIS #BIT3,(R3) ;SET BIT 3 AT TRANSMITTER CONTROL REGISTER
2691 012144 005005 CLR R5 ;SET 'EXPECTED'
2692 012146 052777 000400 167236 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2693 012154 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2694 012160 032713 000010 BIT #BIT3,(R3) ;TEST BIT 3
2695 012164 001402 BEQ 1$ ;BIT 3 IS CLEARED
2696 012166 011304 MOV (R3),R4 ;LOAD 'FOUND'
2697 012170 104003 HLT 3 ;BIT 3 IS SET AND SHOULDN'T BE
2698 012172 052713 000010 1$: BIS #BIT3,(R3) ;SET BIT 3 AGAIN
2699 012176 005013 CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER
2700 012200 032713 000010 BIT #BIT3,(R3) ;TEST TO SEE IF BIT 3 CLEARED
2701 012204 001402 BEQ 2$ ;BIT 3 IS OK
2702 012206 011304 MOV (R3),R4 ;LOAD 'FOUND'
2703 012210 104003 HLT 3 ;BIT 3 FAILED TO CLEAR
2704 012212 104400 2$: SCOPE ;SCOPE THIS TEST
2705
```


TRANSMITTER CONTROL REGISTER RESET AND CLEAR TEST BIT 4

2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761

012214	012737	000037	001226
012222	012737	012310	001216
012230	013703	001412	
012234	052713	000020	
012240	005005		
012242	052777	000400	167142
012250	004737	005044	
012254	032713	000020	
012260	001402		
012262	011304		
012264	104003		
012266	052713	000020	
012272	005013		
012274	032713	000020	
012300	001402		
012302	011304		
012304	104003		
012306	104400		

```
***** TEST 37 *****
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST
*WRITE BIT 4, AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
*****

:*****
:
: TEST 37
:
:*****
:*****
TST37: MOV #37,@TSTNO
      MOV #TST40,NEXT
      MOV TXCSR,R3      ;GET THE TRANSMITTER CONTROL REGISTER
      BIS #BIT4,(R3)    ;SET BIT 4 AT TRANSMITTER CONTROL REGISTER
      CLR R5            ;SET 'EXPECTED'
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
      BIT #BIT4,(R3)    ;TEST BIT 4
      BEQ 1$           ;BIT 4 IS CLEARED
      MOV (R3),R4       ;LOAD 'FOUND'
      HLT 3            ;BIT 4 IS SET AND SHOULDN'T BE
1$:    BIS #BIT4,(R3)    ;SET BIT 4 AGAIN
      CLR (R3)          ;CLEAR THE TRANSMITTER CONTROL REGISTER
      BIT #BIT4,(R3)    ;TEST TO SEE IF BIT 4 CLEARED
      BEQ 2$           ;BIT 4 IS OK
      MOV (R3),R4       ;LOAD 'FOUND'
      HLT 3            ;BIT 4 FAILED TO CLEAR
2$:    SCOPE            ;SCOPE THIS TEST
```

```
***** TEST 40 *****
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 10 RESET AND CLEAR TEST
*WRITE BIT 10, AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
*****
```

```
*****
:
: TEST 40
:
:*****
:*****
TST40: MOV #40,@TSTNO
      MOV #TST41,NEXT
      MOV TXCSR,R3      ;GET THE TRANSMITTER CONTROL REGISTER
      BIS #BIT10,(R3)   ;SET BIT 10 AT TRANSMITTER CONTROL REGISTER
      CLR R5            ;SET 'EXPECTED'
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
      BIT #BIT10,(R3)   ;TEST BIT 10
      BEQ 1$           ;BIT 10 IS CLEARED
      MOV (R3),R4       ;LOAD 'FOUND'
      HLT 3            ;BIT 10 IS SET AND SHOULDN'T BE
```

TRANSMITTER CONTROL REGISTER RESET AND CLEAR TEST BIT 10

```
2762 012362 052713 002000 1$: BIS #BIT10,(R3) ;SET BIT 10 AGAIN
2763 012366 005013 CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER
2764 012370 032713 002000 BIT #BIT10,(R3) ;TEST TO SEE IF BIT 10 CLEARED
2765 012374 001402 BEQ 2$ ;BIT 10 IS OK
2766 012376 011304 MOV (R3),R4 ;LOAD 'FOUND'
2767 012400 104003 HLT 3 ;BIT 10 FAILED TO CLEAR
2768 012402 104400 2$: SCOPE ;SCOPE THIS TEST
2769
2770
2771 ;***** TEST 41 *****
2772 ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 11 RESET AND CLEAR TEST
2773 ;*WRITE BIT 11,AND TEST THAT IT WILL BE CLEARED AFTER A
2774 ;*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
2775 ;*****
2776
2777 ;*****
2778 ;*
2779 ; TEST 41
2780 ;*
2781 ;*****
2782 ;*****
2783 012404 012737 000041 001226 TST41: MOV #41,@TSTNO
2784 012412 012737 012500 001216 MOV #TST42,NEXT
2785 012420 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
2786 012424 052713 004000 BIS #BIT11,(R3) ;SET BIT 11 AT TRANSMITTER CONTROL REGISTER
2787 012430 005005 CLR R5 ;SET 'EXPECTED'
2788 012432 052777 000400 166752 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2789 012440 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2790 012444 032713 004000 BIT #BIT11,(R3) ;TEST BIT 11
2791 012450 001402 BEQ 1$ ;BIT 11 IS CLEARED
2792 012452 011304 MOV (R3),R4 ;LOAD 'FOUND'
2793 012454 104003 HLT 3 ;BIT 11 IS SET AND SHOULDN'T BE
2794 012456 052713 004000 1$: BIS #BIT11,(R3) ;SET BIT 11 AGAIN
2795 012462 005013 CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER
2796 012464 032713 004000 BIT #BIT11,(R3) ;TEST TO SEE IF BIT 11 CLEARED
2797 012470 001402 BEQ 2$ ;BIT 11 IS OK
2798 012472 011304 MOV (R3),R4 ;LOAD 'FOUND'
2799 012474 104003 HLT 3 ;BIT 11 FAILED TO CLEAR
2800 012476 104400 2$: SCOPE ;SCOPE THIS TEST
2801
2802
2803 ;***** TEST 42 *****
2804 ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 12 RESET AND CLEAR TEST
2805 ;*WRITE BIT 12,AND TEST THAT IT WILL BE CLEARED AFTER A
2806 ;*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
2807 ;*****
2808
2809 ;*****
2810 ;*
2811 ; TEST 42
2812 ;*
2813 ;*****
2814 ;*****
2815 012500 012737 000042 001226 TST42: MOV #42,@TSTNO
2816 012506 012737 012574 001216 MOV #TST43,NEXT
2817 012514 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
```


TRANSMITTER CONTROL REGISTER RESET AND CLEAR TEST BIT 12

```
2818 012520 052713 010000      BIS    #BIT12,(R3)    ;SET BIT 12 AT TRANSMITTER CONTROL REGISTER
2819 012524 005005              CLR    R5                ;SET 'EXPECTED'
2820 012526 052777 000400 166656 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2821 012534 004737 005044      JSR    PC,SMALL          ;WAIT FOR RESET TO FINISH
2822 012540 032713 010000      BIT     #BIT12,(R3)    ;TEST BIT 12
2823 012544 001402              BEQ     1$                ;BIT 12 IS CLEARED
2824 012546 011304              MOV     (R3),R4            ;LOAD 'FOUND'
2825 012550 104003              HLT     3                  ;BIT 12 IS SET AND SHOULDN'T BE
2826 012552 052713 010000 1$:  BIS    #BIT12,(R3)    ;SET BIT 12 AGAIN
2827 012556 005013              CLR     (R3)              ;CLEAR THE TRANSMITTER CONTROL REGISTER
2828 012560 032713 010000      BIT     #BIT12,(R3)    ;TEST TO SEE IF BIT 12 CLEARED
2829 012564 001402              BEQ     2$                ;BIT 12 IS OK
2830 012566 011304              MOV     (R3),R4            ;LOAD 'FOUND'
2831 012570 104003              HLT     3                  ;BIT 12 FAILED TO CLEAR
2832 012572 104400 2$:      SCOPE                      ;SCOPE THIS TEST
```

```
***** TEST 43 *****
;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 13 RESET AND CLEAR TEST
;*WRITE BIT 13,AND TEST THAT IT WILL BE CLEARED AFTER A
;*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
;*****
```

```
*****
;*****
; TEST 43
;*****
```

```
2847 012574 012737 000043 001226 TST43: MOV    #43,@TSTNO
2848 012602 012737 012670 001216 MOV    #TST44,NEXT
2849 012610 013703 001412      MOV     TXCSR,R3          ;GET THE TRANSMITTER CONTROL REGISTER
2850 012614 052713 020000      BIS     #BIT13,(R3)    ;SET BIT 13 AT TRANSMITTER CONTROL REGISTER
2851 012620 005005              CLR     R5                ;SET 'EXPECTED'
2852 012622 052777 000400 166562 BIS     #MRESET,@TXCSR ;RESET THE DEVICE
2853 012630 004737 005044      JSR     PC,SMALL          ;WAIT FOR RESET TO FINISH
2854 012634 032713 020000      BIT      #BIT13,(R3)    ;TEST BIT 13
2855 012640 001402              BEQ     1$                ;BIT 13 IS CLEARED
2856 012642 011304              MOV     (R3),R4            ;LOAD 'FOUND'
2857 012644 104003              HLT     3                  ;BIT 13 IS SET AND SHOULDN'T BE
2858 012646 052713 020000 1$:  BIS     #BIT13,(R3)    ;SET BIT 13 AGAIN
2859 012652 005013              CLR     (R3)              ;CLEAR THE TRANSMITTER CONTROL REGISTER
2860 012654 032713 020000      BIT      #BIT13,(R3)    ;TEST TO SEE IF BIT 13 CLEARED
2861 012660 001402              BEQ     2$                ;BIT 13 IS OK
2862 012662 011304              MOV     (R3),R4            ;LOAD 'FOUND'
2863 012664 104003              HLT     3                  ;BIT 13 FAILED TO CLEAR
2864 012666 104400 2$:      SCOPE                      ;SCOPE THIS TEST
```

```
***** TEST 44 *****
;*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 0 RESET AND CLEAR TEST
;*WRITE BIT 0,AND TEST THAT IT WILL BE CLEARED AFTER A
;*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
;*****
```

```
*****
```

TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 0

```
2874
2875
2876
2877
2878
2879 012670 012737 000044 001226
2880 012676 012737 012764 001216
2881 012704 013703 001414
2882 012710 052713 000001
2883 012714 005005
2884 012716 052777 000400 166466
2885 012724 004737 005044
2886 012730 032713 000001
2887 012734 001402
2888 012736 011304
2889 012740 104003
2890 012742 052713 000001
2891 012746 005013
2892 012750 032713 000001
2893 012754 001402
2894 012756 011304
2895 012760 104003
2896 012762 104400
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911 012764 012737 000045 001226
2912 012772 012737 013060 001216
2913 013000 013703 001414
2914 013004 052713 000002
2915 013010 005005
2916 013012 052777 000400 166372
2917 013020 004737 005044
2918 013024 032713 000002
2919 013030 001402
2920 013032 011304
2921 013034 104003
2922 013036 052713 0000C2
2923 013042 005013
2924 013044 032713 000002
2925 013050 001402
2926 013052 011304
2927 013054 104003
2928 013056 104400
2929
```

```

:
: TEST 44
:
:*****
:*****
TST44: MOV #44,@TSTNO
      MOV #TST45,NEXT
      MOV TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
      BIS #BIT0,(R3)     ;SET BIT 0 AT TRANSMITTER BUFFER REGISTER
      CLR R5             ;SET 'EXPECTED'
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL       ;WAIT FOR RESET TO FINISH
      BIT #BIT0,(R3)     ;TEST BIT 0
      BEQ 1$             ;BIT 0 IS CLEARED
      MOV (R3),R4        ;LOAD 'FOUND'
      HLT 3              ;BIT 0 IS SET AND SHOULDN'T BE
1$:    BIS #BIT0,(R3)     ;SET BIT 0 AGAIN
      CLR (R3)           ;CLEAR THE TRANSMITTER BUFFER REGISTER
      BIT #BIT0,(R3)     ;TEST TO SEE IF BIT 0 CLEARED
      BEQ 2$             ;BIT 0 IS OK
      MOV (R3),R4        ;LOAD 'FOUND'
      HLT 3              ;BIT 0 FAILED TO CLEAR
2$:    SCOPE             ;SCOPE THIS TEST

:***** TEST 45 *****
: *TRANSMITTER BUFFER REGISTER READ/WRITE BIT 1 RESET AND CLEAR TEST
: *WRITE BIT 1,AND TEST THAT IT WILL BE CLEARED AFTER A
: *DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
:*****
:*****
:
: TEST 45
:
:*****
:*****
TST45: MOV #45,@TSTNO
      MOV #TST46,NEXT
      MOV TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
      BIS #BIT1,(R3)     ;SET BIT 1 AT TRANSMITTER BUFFER REGISTER
      CLR R5             ;SET 'EXPECTED'
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL       ;WAIT FOR RESET TO FINISH
      BIT #BIT1,(R3)     ;TEST BIT 1
      BEQ 1$             ;BIT 1 IS CLEARED
      MOV (R3),R4        ;LOAD 'FOUND'
      HLT 3              ;BIT 1 IS SET AND SHOULDN'T BE
1$:    BIS #BIT1,(R3)     ;SET BIT 1 AGAIN
      CLR (R3)           ;CLEAR THE TRANSMITTER BUFFER REGISTER
      BIT #BIT1,(R3)     ;TEST TO SEE IF BIT 1 CLEARED
      BEQ 2$             ;BIT 1 IS OK
      MOV (R3),R4        ;LOAD 'FOUND'
      HLT 3              ;BIT 1 FAILED TO CLEAR
2$:    SCOPE             ;SCOPE THIS TEST
```


TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 2

2930
 2931
 2932
 2933
 2934
 2935
 2936
 2937
 2938
 2939
 2940
 2941
 2942
 2943
 2944
 2945
 2946
 2947
 2948
 2949
 2950
 2951
 2952
 2953
 2954
 2955
 2956
 2957
 2958
 2959
 2960
 2961
 2962
 2963
 2964
 2965
 2966
 2967
 2968
 2969
 2970
 2971
 2972
 2973
 2974
 2975
 2976
 2977
 2978
 2979
 2980
 2981
 2982
 2983
 2984
 2985

013060	012737	000046	001226
013066	012737	013154	001216
013074	013703	001414	
013100	052713	000004	
013104	005005		
013106	052777	000400	166276
013114	004737	005044	
013120	032713	000004	
013124	001402		
013126	011304		
013130	104003		
013132	052713	000004	
013136	005013		
013140	032713	000004	
013144	001402		
013146	011304		
013150	104003		
013152	104400		

```

***** TEST 46 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 2 RESET AND CLEAR TEST
*WRITE BIT 2,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****

*****
*
TEST 46
*
*****
*****
TST46: MOV #46,@TSTNO
MOV #TST47,NEXT
MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
BIS #BIT2,(R3) ;SET BIT 2 AT TRANSMITTER BUFFER REGISTER
CLR R5 ;SET 'EXPECTED'
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT2,(R3) ;TEST BIT 2
BEQ 1$ ;BIT 2 IS CLEARED
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 2 IS SET AND SHOULDN'T BE
1$: BIS #BIT2,(R3) ;SET BIT 2 AGAIN
CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
BIT #BIT2,(R3) ;TEST TO SEE IF BIT 2 CLEARED
BEQ 2$ ;BIT 2 IS OK
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 2 FAILED TO CLEAR
2$: SCOPE ;SCOPE THIS TEST
  
```

```

***** TEST 47 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST
*WRITE BIT 3,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****
  
```

013154	012737	000047	001226
013162	012737	013250	001216
013170	013703	001414	
013174	052713	000010	
013200	005005		
013202	052777	000400	166202
013210	004737	005044	
013214	032713	000010	
013220	001402		
013222	011304		
013224	104003		

```

*****
*
TEST 47
*
*****
*****
TST47: MOV #47,@TSTNO
MOV #TST50,NEXT
MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
BIS #BIT3,(R3) ;SET BIT 3 AT TRANSMITTER BUFFER REGISTER
CLR R5 ;SET 'EXPECTED'
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT3,(R3) ;TEST BIT 3
BEQ 1$ ;BIT 3 IS CLEARED
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 3 IS SET AND SHOULDN'T BE
  
```

TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 3

```
2986 013226 052713 000010 1$: BIS #BIT3,(R3) ;SET BIT 3 AGAIN
2987 013232 005013 CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
2988 013234 032713 000010 BIT #BIT3,(R3) ;TEST TO SEE IF BIT 3 CLEARED
2989 013240 001402 BEQ 2$ ;BIT 3 IS OK
2990 013242 011304 MOV (R3),R4 ;LOAD 'FOUND'
2991 013244 104003 HLT 3 ;BIT 3 FAILED TO CLEAR
2992 013246 104400 2$: SCOPE ;SCOPE THIS TEST
2993
2994
2995 ;***** TEST 50 *****
2996 ;*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST
2997 ;*WRITE BIT 4,AND TEST THAT IT WILL BE CLEARED AFTER A
2998 ;*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
2999 ;*****
3000
3001 ;*****
3002 ;*
3003 ;* TEST 50
3004 ;*
3005 ;*****
3006 ;*****
3007 013250 012737 000050 001226 TST50: MOV #50,@TSTNO
3008 013256 012737 013344 001216 MOV #TST51,NEXT
3009 013264 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
3010 013270 052713 000020 BIS #BIT4,(R3) ;SET BIT 4 AT TRANSMITTER BUFFER REGISTER
3011 013274 005005 CLR R5 ;SET 'EXPECTED'
3012 013276 052777 000400 166106 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3013 013304 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3014 013310 032713 000020 BIT #BIT4,(R3) ;TEST BIT 4
3015 013314 001402 BEQ 1$ ;BIT 4 IS CLEARED
3016 013316 011304 MOV (R3),R4 ;LOAD 'FOUND'
3017 013320 104003 HLT 3 ;BIT 4 IS SET AND SHOULDN'T BE
3018 013322 052713 000020 1$: BIS #BIT4,(R3) ;SET BIT 4 AGAIN
3019 013326 005013 CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
3020 013330 032713 000020 BIT #BIT4,(R3) ;TEST TO SEE IF BIT 4 CLEARED
3021 013334 001402 BEQ 2$ ;BIT 4 IS OK
3022 013336 011304 MOV (R3),R4 ;LOAD 'FOUND'
3023 013340 104003 HLT 3 ;BIT 4 FAILED TO CLEAR
3024 013342 104400 2$: SCOPE ;SCOPE THIS TEST
3025
3026
3027 ;***** TEST 51 *****
3028 ;*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 5 RESET AND CLEAR TEST
3029 ;*WRITE BIT 5,AND TEST THAT IT WILL BE CLEARED AFTER A
3030 ;*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
3031 ;*****
3032
3033 ;*****
3034 ;*
3035 ;* TEST 51
3036 ;*
3037 ;*****
3038 ;*****
3039 013344 012737 000051 001226 TST51: MOV #51,@TSTNO
3040 013352 012737 013440 001216 MOV #TST52,NEXT
3041 013360 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
```


TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 5

```
3042 013364 052713 000040      BIS    #BIT5,(R3)      ;SET BIT 5 AT TRANSMITTER BUFFER REGISTER
3043 013370 005005      CLR    R5              ;SET 'EXPECTED'
3044 013372 052777 000400 166012  BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
3045 013400 004737 005044      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
3046 013404 032713 000040      BIT    #BIT5,(R3)      ;TEST BIT 5
3047 013410 001402      BEQ    1$              ;BIT 5 IS CLEARED
3048 013412 011304      MOV    (R3),R4        ;LOAD 'FOUND'
3049 013414 104003      HLT    3              ;BIT 5 IS SET AND SHOULDN'T BE
3050 013416 052713 000040 1$:    BIS    #BIT5,(R3)      ;SET BIT 5 AGAIN
3051 013422 005013      CLR    (R3)          ;CLEAR THE TRANSMITTER BUFFER REGISTER
3052 013424 032713 000040      BIT    #BIT5,(R3)      ;TEST TO SEE IF BIT 5 CLEARED
3053 013430 001402      BEQ    2$              ;BIT 5 IS OK
3054 013432 011304      MOV    (R3),R4        ;LOAD 'FOUND'
3055 013434 104003      HLT    3              ;BIT 5 FAILED TO CLEAR
3056 013436 104400 2$:    SCOPE              ;SCOPE THIS TEST
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
```

```
***** TEST 52 *****
;TRANSMITTER BUFFER REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
;WRITE BIT 6,AND TEST THAT IT WILL BE CLEARED AFTER A
;DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
;*****
```

```
*****
:
:
: TEST 52
:
:
: *****
```

```
3071 013440 012737 000052 001226 TST52: MOV    #52,@TSTNO
3072 013446 012737 013534 001216  MOV    #TST53,NEXT
3073 013454 013703 001414      MOV    TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
3074 013460 052713 000100      BIS    #BIT6,(R3)      ;SET BIT 6 AT TRANSMITTER BUFFER REGISTER
3075 013464 005005      CLR    R5              ;SET 'EXPECTED'
3076 013466 052777 000400 165716  BIS    #MRESET,@TXCSR  ;RESET THE DEVICE
3077 013474 004737 005044      JSR    PC,SMALL        ;WAIT FOR RESET TO FINISH
3078 013500 032713 000100      BIT    #BIT6,(R3)      ;TEST BIT 6
3079 013504 001402      BEQ    1$              ;BIT 6 IS CLEARED
3080 013506 011304      MOV    (R3),R4        ;LOAD 'FOUND'
3081 013510 104003      HLT    3              ;BIT 6 IS SET AND SHOULDN'T BE
3082 013512 052713 000100 1$:    BIS    #BIT6,(R3)      ;SET BIT 6 AGAIN
3083 013516 005013      CLR    (R3)          ;CLEAR THE TRANSMITTER BUFFER REGISTER
3084 013520 032713 000100      BIT    #BIT6,(R3)      ;TEST TO SEE IF BIT 6 CLEARED
3085 013524 001402      BEQ    2$              ;BIT 6 IS OK
3086 013526 011304      MOV    (R3),R4        ;LOAD 'FOUND'
3087 013530 104003      HLT    3              ;BIT 6 FAILED TO CLEAR
3088 013532 104400 2$:    SCOPE              ;SCOPE THIS TEST
3089
3090
3091
3092
3093
3094
3095
3096
3097
```

```
***** TEST 53 *****
;TRANSMITTER BUFFER REGISTER READ/WRITE BIT 7 RESET AND CLEAR TEST
;WRITE BIT 7,AND TEST THAT IT WILL BE CLEARED AFTER A
;DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
;*****
```

```
*****
```

TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 7

```
3098
3099
3100
3101
3102
3103 013534 012737 000053 001226
3104 013542 012737 013630 001216
3105 013550 013703 001414
3106 013554 052713 000200
3107 013560 005005
3108 013562 052777 000400 165622
3109 013570 004737 005044
3110 013574 032713 000200
3111 013600 001402
3112 013602 011304
3113 013604 104003
3114 013606 052713 000200
3115 013612 005013
3116 013614 032713 000200
3117 013620 001402
3118 013622 011304
3119 013624 104003
3120 013626 104400
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135 013630 012737 000054 001226
3136 013636 012737 013724 001216
3137 013644 013703 001414
3138 013650 052713 000400
3139 013654 005005
3140 013656 052777 000400 165526
3141 013664 004737 005044
3142 013670 032713 000400
3143 013674 001402
3144 013676 011304
3145 013700 104003
3146 013702 052713 000400
3147 013706 005013
3148 013710 032713 000400
3149 013714 001402
3150 013716 011304
3151 013720 104003
3152 013722 104400
3153
```

```

:
: TEST 53
:
: *****
: *****
TST53: MOV #53,@TSTNO
MOV #TST54,NEXT
MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
BIS #BIT7,(R3) ;SET BIT 7 AT TRANSMITTER BUFFER REGISTER
CLR R5 ;SET 'EXPECTED'
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT7,(R3) ;TEST BIT 7
BEQ 1$ ;BIT 7 IS CLEARED
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 7 IS SET AND SHOULDN'T BE
1$: BIS #BIT7,(R3) ;SET BIT 7 AGAIN
CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
BIT #BIT7,(R3) ;TEST TO SEE IF BIT 7 CLEARED
BEQ 2$ ;BIT 7 IS OK
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 7 FAILED TO CLEAR
2$: SCOPE ;SCOPE THIS TEST

: ***** TEST 54 *****
: *TRANSMITTER BUFFER REGISTER READ/WRITE BIT 8 RESET AND CLEAR TEST
: *WRITE BIT 8,AND TEST THAT IT WILL BE CLEARED AFTER A
: *DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
: *****
: *****
:
: TEST 54
:
: *****
: *****
TST54: MOV #54,@TSTNO
MOV #TST55,NEXT
MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
BIS #BIT8,(R3) ;SET BIT 8 AT TRANSMITTER BUFFER REGISTER
CLR R5 ;SET 'EXPECTED'
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT8,(R3) ;TEST BIT 8
BEQ 1$ ;BIT 8 IS CLEARED
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 8 IS SET AND SHOULDN'T BE
1$: BIS #BIT8,(R3) ;SET BIT 8 AGAIN
CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
BIT #BIT8,(R3) ;TEST TO SEE IF BIT 8 CLEARED
BEQ 2$ ;BIT 8 IS OK
MOV (R3),R4 ;LOAD 'FOUND'
HLT 3 ;BIT 8 FAILED TO CLEAR
2$: SCOPE ;SCOPE THIS TEST
```


TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 9

```
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167 013724 012737 000055 001226
3168 013732 012737 014020 001216
3169 013740 013703 001414
3170 013744 052713 001000
3171 013750 005005
3172 013752 052777 000400 165432
3173 013760 004737 005044
3174 013764 032713 001000
3175 013770 001402
3176 013772 011304
3177 013774 104003
3178 013776 052713 001000
3179 014002 005013
3180 014004 032713 001000
3181 014010 001402
3182 014012 011304
3183 014014 104003
3184 014016 104400
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199 014020 012737 000056 001226
3200 014026 012737 014114 001216
3201 014034 013703 001414
3202 014040 052713 002000
3203 014044 005005
3204 014046 052777 000400 165336
3205 014054 004737 005044
3206 014060 032713 002000
3207 014064 001402
3208 014066 011304
3209 014070 104003
```

```
***** TEST 55 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 9 RESET AND CLEAR TEST
*WRITE BIT 9, AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****

*****
: TEST 55
*****
*****
TST55: MOV #55, @TSTNO
      MOV #TST56, NEXT
      MOV TXDBUF, R3      ;GET THE TRANSMITTER BUFFER REGISTER
      BIS #BIT9, (R3)     ;SET BIT 9 AT TRANSMITTER BUFFER REGISTER
      CLR R5              ;SET 'EXPECTED'
      BIS #MRESET, @TXCSR ;RESET THE DEVICE
      JSR PC, SMALL       ;WAIT FOR RESET TO FINISH
      BIT #BIT9, (R3)     ;TEST BIT 9
      BEQ 1$              ;BIT 9 IS CLEARED
      MOV (R3), R4         ;LOAD 'FOUND'
      HLT 3                ;BIT 9 IS SET AND SHOULDN'T BE
1$:    BIS #BIT9, (R3)     ;SET BIT 9 AGAIN
      CLR (R3)             ;CLEAR THE TRANSMITTER BUFFER REGISTER
      BIT #BIT9, (R3)     ;TEST TO SEE IF BIT 9 CLEARED
      BEQ 2$              ;BIT 9 IS OK
      MOV (R3), R4         ;LOAD 'FOUND'
      HLT 3                ;BIT 9 FAILED TO CLEAR
2$:    SCOPE               ;SCOPE THIS TEST
```

```
***** TEST 56 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 10 RESET AND CLEAR TEST
*WRITE BIT 10, AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****
```

```
*****
: TEST 56
*****
*****
TST56: MOV #56, @TSTNO
      MOV #TST57, NEXT
      MOV TXDBUF, R3      ;GET THE TRANSMITTER BUFFER REGISTER
      BIS #BIT10, (R3)    ;SET BIT 10 AT TRANSMITTER BUFFER REGISTER
      CLR R5              ;SET 'EXPECTED'
      BIS #MRESET, @TXCSR ;RESET THE DEVICE
      JSR PC, SMALL       ;WAIT FOR RESET TO FINISH
      BIT #BIT10, (R3)    ;TEST BIT 10
      BEQ 1$              ;BIT 10 IS CLEARED
      MOV (R3), R4         ;LOAD 'FOUND'
      HLT 3                ;BIT 10 IS SET AND SHOULDN'T BE
```

TRANSMITTER BUFFER REGISTER RESET AND CLEAR TEST BIT 10

3210	014072	052713	002000	1\$:	BIS	#BIT10,(R3)	;SET BIT 10 AGAIN
3211	014076	005013			CLR	(R3)	;CLEAR THE TRANSMITTER BUFFER REGISTER
3212	014100	032713	002000		BIT	#BIT10,(R3)	;TEST TO SEE IF BIT 10 CLEARED
3213	014104	001402			BEQ	2\$	;BIT 10 IS OK
3214	014106	011304			MOV	(R3),R4	;LOAD 'FOUND'
3215	014110	104003			HLT	3	;BIT 10 FAILED TO CLEAR
3216	014112	104400		2\$:	SCOPE		;SCOPE THIS TEST

***** TEST 57 *****
 ;*RECEIVER BUFFER REGISTER TEST
 ;*TEST THAT RECEIVER BUFFER REGISTER CANNOT BE WRITTEN.
 ;*WRITE RECEIVER BUFFER REGISTER WITH ALL 1'S
 ;*AND VERIFY THAT ALL 0'S ARE READ BACK.
 ;*****

 ;
 ; TEST 57
 ;

3220							
3221							
3222							
3223							
3224							
3225							
3226							
3227							
3228							
3229							
3230							
3231							
3232	014114	012737	000057	001226	TST57:	MOV	#57,@#TSTNO
3233	014122	012737	014154	001216		MOV	#TST60,NEXT
3234	014130	013703	001406			MOV	RXDBUF,R3
3235							;GET THE REGISTER.
3236	014134	005005				CLR	R5
3237	014136	012713	177777			MOV	#-1,(R3)
3238	014142	011304				MOV	(R3),R4
3239	014144	020504				CMP	R5,R4
3240	014146	001401				BEQ	1\$
3241	014150	104003				HLT	3
3242	014152	104400			1\$:	SCOPE	;SCOPE THIS TEST.

***** TEST 60 *****
 ;*PARAMETER STATUS REGISTER TEST
 ;*TEST THAT PARAMETER STATUS REGISTER CANNOT BE WRITTEN
 ;*READ THE PARAMETER STATUS REGISTER AND STORE THE DATA;
 ;*COMPLEMENT THE DATA AND WRITE THE PARAMETER STATUS REGISTER
 ;*VERIFING THAT THE PARAMETER STATUS REGISTER DID NOT CHANGE.
 ;*****

 ;
 ; TEST 60
 ;

3243							
3244							
3245							
3246							
3247							
3248							
3249							
3250							
3251							
3252							
3253							
3254							
3255							
3256							
3257							
3258							
3259							
3260	014154	012737	000060	001226	TST60:	MOV	#60,@#TSTNO
3261	014162	012737	014216	001216		MOV	#TST61,NEXT
3262	014170	013703	001410			MOV	PARCSR,R3
3263							;GET THE REGISTER.
3264	014174	011305				MOV	(R3),R5
3265	014176	010504				MOV	R5,R4

;READ THE REGISTER INTO R5
 ;SAVE REG INTO R4


```

3266 014200 005104      COM      R4      ;MAKE R4 OPPOSITE TO REGISTER
3267 014202 010413      MOV      R4,(R3) ;WRITE THE REGISTER WITH THE COMPLIMENT
3268 014204 011304      MOV      (R3),R4 ;READ THE REGISTER.
3269 014206 020504      CMP      R5,R4   ;IS THE REGISTER EQUAL TO ZERO.
3270 014210 001401      BEQ      1$      ;BR IF OK.
3271 014212 104003      HLT      3       ;REGISTER NOT ZERO.
3272 014214 104400      1$:      SCOPE    ;SCOPE THIS TEST.
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286

```

```

;***** TEST 61 *****
;*RECEIVER CONTROL REGISTER BIT 0 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 0 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

```

```

;*****
;:*****
;:
;: TEST 61
;:
;:*****

```

```

3287 014216 012737 000061 001226 TST61: MOV      #61,@TSTNO
3288 014224 012737 014330 001216      MOV      #TST62,NEXT
3289 014232 013703 001404      MOV      RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3290 014236 012705 000001      MOV      #BIT0,R5 ;GET BIT 0
3291 014242 010513      MOV      R5,(R3) ;WRITE BIT 0 TO RECEIVER CONTROL REGISTER
3292 014244 011304      MOV      (R3),R4 ;READ RECEIVER CONTROL REGISTER BACK
3293 014246 005005      CLR      R5 ;SET 'EXPECTED'
3294 014250 020504      CMP      R5,R4   ;R5=GOOD,R4= ?
3295 014252 001401      BEQ      5$      ;BIT 0 IS OK
3296 014254 104003      HLT      3       ;BIT FAILED TO CLR
3297 014256 012705 000001      5$:      MOV      #BIT0,R5 ;RELOAD THE BIT
3298 014262 010513      MOV      R5,(R3) ;WRITE BIT 0 TO THE REG
3299 014264 052777 000400 165120      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3300 014272 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3301 014276 011304      MOV      (R3),R4 ;GET BIT 0
3302 014300 032704 000001      BIT      #BIT0,R4 ;TEST BIT 0 FOR RESET CLR
3303 014304 001401      BEQ      7$      ;BIT 0 IS OK
3304 014306 104003      HLT      3       ;BIT FAILED TO CLEAR
3305 014310 010513      7$:      MOV      R5,(R3) ;SET BIT 0
3306 014312 005013      CLR      (R3) ;CLR RECEIVER CONTROL REGISTER
3307 014314 011304      MOV      (R3),R4 ;READ THE RECEIVER CONTROL REGISTER
3308 014316 005005      CLR      R5 ;SET 'EXPECTED'
3309 014320 020504      CMP      R5,R4   ;R5=GOOD,R4= ?
3310 014322 001401      BEQ      10$     ;BIT 0 IS OK
3311 014324 104003      HLT      3       ;BIT FAILED TO CLEAR
3312 014326 104400      10$:      SCOPE    ;SCOPE THIS TEST
3313
3314
3315
3316
3317
3318
3319
3320
3321

```

```

;***** TEST 62 *****
;*RECEIVER CONTROL REGISTER BIT 7 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 7 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

```

```

;*****

```

```

3322      :
3323      : TEST 62
3324      :
3325      : *****
3326      : *****
3327 014330 012737 000062 001226 TST62: MOV #62,@TSTNO
3328 014336 012737 014442 001216 MOV #TST63,NEXT
3329 014344 013703 001404 MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3330 014350 012705 000200 MOV #BIT7,R5 ;GET BIT 7
3331 014354 010513 MOV R5,(R3) ;WRITE BIT 7 TO RECEIVER CONTROL REGISTER
3332 014356 011304 MOV (R3),R4 ;READ RECEIVER CONTROL REGISTER BACK
3333 014360 005005 CLR R5 ;SET 'EXPECTED'
3334 014362 020504 CMP R5,R4 ;R5=GOOD,R4= ?
3335 014364 001401 BEQ 5$ ;BIT 7 IS OK
3336 014366 104003 HLT 3 ;BIT FAILED TO CLR
3337 014370 012705 000200 5$: MOV #BIT7,R5 ;RELOAD THE BIT
3338 014374 010513 MOV R5,(R3) ;WRITE BIT 7 TO THE REG
3339 014376 052777 000400 165006 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3340 014404 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3341 014410 011304 MOV (R3),R4 ;GET BIT 7
3342 014412 032704 000200 BIT #BIT7,R4 ;TEST BIT 7 FOR RESET CLR
3343 014416 001401 BEQ 7$ ;BIT 7 IS OK
3344 014420 104003 HLT 3 ;BIT FAILED TO CLEAR
3345 014422 010513 7$: MOV R5,(R3) ;SET BIT 7
3346 014424 005013 CLR (R3) ;CLR RECEIVER CONTROL REGISTER
3347 014426 011304 MOV (R3),R4 ;READ THE RECEIVER CONTROL REGISTER
3348 014430 005005 CLR R5 ;SET 'EXPECTED'
3349 014432 020504 CMP R5,R4 ;R5=GOOD,R4= ?
3350 014434 001401 BEQ 10$ ;BIT 7 IS OK
3351 014436 104003 HLT 3 ;BIT FAILED TO CLEAR
3352 014440 104400 10$: SCOPE ;SCOPE THIS TEST
3353
3354
3355      : ***** TEST 63 *****
3356      : *RECEIVER CONTROL REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST
3357      : *WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK
3358      : *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
3359      : *****
3360
3361      : *****
3362      :
3363      : TEST 63
3364      :
3365      : *****
3366      : *****
3367 014442 012737 000063 001226 TST63: MOV #63,@TSTNO
3368 014450 012737 014554 001216 MOV #TST64,NEXT
3369 014456 013703 001404 MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3370 014462 012705 001000 MOV #BIT9,R5 ;GET BIT 9
3371 014466 010513 MOV R5,(R3) ;WRITE BIT 9 TO RECEIVER CONTROL REGISTER
3372 014470 011304 MOV (R3),R4 ;READ RECEIVER CONTROL REGISTER BACK
3373 014472 005005 CLR R5 ;SET 'EXPECTED'
3374 014474 020504 CMP R5,R4 ;R5=GOOD,R4= ?
3375 014476 001401 BEQ 5$ ;BIT 9 IS OK
3376 014500 104003 HLT 3 ;BIT FAILED TO CLR
3377 014502 012705 001000 5$: MOV #BIT9,R5 ;RELOAD THE BIT

```


CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 69
 CZDPBC.P11 02-MAY-78 08:23

SEQ 0068

RECEIVER CONTROL REGISTER READ ONLY BIT 9 TEST

```

3378 014506 010513      MOV      R5,(R3)      ;WRITE BIT 9 TO THE REG
3379 014510 052777 000400 164674  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3380 014516 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3381 014522 011304      MOV      (R3),R4 ;GET BIT 9
3382 014524 032704 001000  BIT      #BIT9,R4 ;TEST BIT 9 FOR RESET CLR
3383 014530 001401      BEQ      7$ ;BIT 9 IS OK
3384 014532 104003      HLT      3 ;BIT FAILED TO CLEAR
3385 014534 010513 7$:      MOV      R5,(R3) ;SET BIT 9
3386 014536 005013      CLR      (R3) ;CLR RECEIVER CONTROL REGISTER
3387 014540 011304      MOV      (R3),R4 ;READ THE RECEIVER CONTROL REGISTER
3388 014542 005005      CLR      R5 ;SET 'EXPECTED'
3389 014544 020504      CMP      R5,R4 ;R5=GOOD,R4= ?
3390 014546 001401      BEQ      10$ ;BIT 9 IS OK
3391 014550 104003      HLT      3 ;BIT FAILED TO CLEAR
3392 014552 104400 10$:    SCOPE ;SCOPE THIS TEST
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406

```

```

:***** TEST 64 *****
:*RECEIVER CONTROL REGISTER BIT 10 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 10 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****

```

```

:*****
: *
: TEST 64
: *
:*****

```

```

3407 014554 012737 000064 001226 TST64: MOV      #64,@TSTNO
3408 014562 012737 014666 001216  MOV      #TST65,NEXT
3409 014570 013703 001404      MOV      RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3410 014574 012705 002000      MOV      #BIT10,R5 ;GET BIT 10
3411 014600 010513      MOV      R5,(R3) ;WRITE BIT 10 TO RECEIVER CONTROL REGISTER
3412 014602 011304      MOV      (R3),R4 ;READ RECEIVER CONTROL REGISTER BACK
3413 014604 005005      CLR      R5 ;SET 'EXPECTED'
3414 014606 020504      CMP      R5,R4 ;R5=GOOD,R4= ?
3415 014610 001401      BEQ      5$ ;BIT 10 IS OK
3416 014612 104003      HLT      3 ;BIT FAILED TO CLR
3417 014614 012705 002000 5$:      MOV      #BIT10,R5 ;RELOAD THE BIT
3418 014620 010513      MOV      R5,(R3) ;WRITE BIT 10 TO THE REG
3419 014622 052777 000400 164562  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3420 014630 004737 005044      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3421 014634 011304      MOV      (R3),R4 ;GET BIT 10
3422 014636 032704 002000  BIT      #BIT10,R4 ;TEST BIT 10 FOR RESET CLR
3423 014642 001401      BEQ      7$ ;BIT 10 IS OK
3424 014644 104003      HLT      3 ;BIT FAILED TO CLEAR
3425 014646 010513 7$:      MOV      R5,(R3) ;SET BIT 10
3426 014650 005013      CLR      (R3) ;CLR RECEIVER CONTROL REGISTER
3427 014652 011304      MOV      (R3),R4 ;READ THE RECEIVER CONTROL REGISTER
3428 014654 005005      CLR      R5 ;SET 'EXPECTED'
3429 014656 020504      CMP      R5,R4 ;R5=GOOD,R4= ?
3430 014660 001401      BEQ      10$ ;BIT 10 IS OK
3431 014662 104003      HLT      3 ;BIT FAILED TO CLEAR
3432 014664 104400 10$:    SCOPE ;SCOPE THIS TEST
3433

```

3434
 3435
 3436
 3437
 3438
 3439
 3440
 3441
 3442
 3443
 3444
 3445
 3446
 3447
 3448
 3449
 3450
 3451
 3452
 3453
 3454
 3455
 3456
 3457
 3458
 3459
 3460
 3461
 3462
 3463
 3464
 3465
 3466
 3467
 3468
 3469
 3470
 3471
 3472
 3473
 3474
 3475
 3476
 3477
 3478
 3479
 3480
 3481
 3482
 3483
 3484
 3485
 3486
 3487
 3488
 3489

014666	012737	000065	001226
014674	012737	015000	001216
014702	013703	001404	
014706	012705	004000	
014712	010513		
014714	011304		
014716	005005		
014720	020504		
014722	001401		
014724	104003		
014726	012705	004000	
014732	010513		
014734	052777	000400	164450
014742	004737	005044	
014746	011304		
014750	032704	004000	
014754	001401		
014756	104003		
014760	010513		
014762	005013		
014764	011304		
014766	005005		
014770	020504		
014772	001401		
014774	104003		
014776	104400		

```

***** TEST 65 *****
*RECEIVER CONTROL REGISTER BIT 11 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 11 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

*****
:
: TEST 65
:
: *****
: *****
TST65: MOV #65,@TSTNO
      MOV #TST66,NEXT
      MOV RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
      MOV #BIT11,R5     ;GET BIT 11
      MOV R5,(R3)       ;WRITE BIT 11 TO RECEIVER CONTROL REGISTER
      MOV (R3),R4       ;READ RECEIVER CONTROL REGISTER BACK
      CLR R5            ;SET 'EXPECTED'
      CMP R5,R4         ;R5=GOOD,R4= ?
      BEQ 5$           ;BIT 11 IS OK
      HLT 3            ;BIT FAILED TO CLR
5$:    MOV #BIT11,R5    ;RELOAD THE BIT
      MOV R5,(R3)       ;WRITE BIT 11 TO THE REG
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
      MOV (R3),R4       ;GET BIT 11
      BIT #BIT11,R4     ;TEST BIT 11 FOR RESET CLR
      BEQ 7$           ;BIT 11 IS OK
      HLT 3            ;BIT FAILED TO CLEAR
7$:    MOV R5,(R3)      ;SET BIT 11
      CLR (R3)          ;CLR RECEIVER CONTROL REGISTER
      MOV (R3),R4       ;READ THE RECEIVER CONTROL REGISTER
      CLR R5            ;SET 'EXPECTED'
      CMP R5,R4         ;R5=GOOD,R4= ?
      BEQ 10$          ;BIT 11 IS OK
      HLT 3            ;BIT FAILED TO CLEAR
10$:   SCOPE           ;SCOPE THIS TEST

```

```

***** TEST 66 *****
*RECEIVER CONTROL REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
:
: TEST 66
:
: *****
: *****
TST66: MOV #66,@TSTNO
      MOV #TST67,NEXT
      MOV RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER

```



```

3490 015020 012705 010000      MOV    #BIT12,R5      ;GET BIT 12
3491 015024 010513      MOV    R5,(R3)      ;WRITE BIT 12 TO RECEIVER CONTROL REGISTER
3492 015026 011304      MOV    (R3),R4      ;READ RECEIVER CONTROL REGISTER BACK
3493 015030 005005      CLR    R5          ;SET 'EXPECTED'
3494 015032 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3495 015034 0C1401      BEQ    5$          ;BIT 12 IS OK
3496 015036 104003      HLT    3           ;BIT FAILED TO CLR
3497 015040 012705 010000 5$:  MOV    #BIT12,R5      ;RELOAD THE BIT
3498 015044 010513      MOV    R5,(R3)      ;WRITE BIT 12 TO THE REG
3499 015046 052777 000400 164336 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3500 015054 004737 005044      JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
3501 015060 011304      MOV    (R3),R4      ;GET BIT 12
3502 015062 032704 010000      BIT    #BIT12,R4    ;TEST BIT 12 FOR RESET CLR
3503 015066 001401      BEQ    7$          ;BIT 12 IS OK
3504 015070 104003      HLT    3           ;BIT FAILED TO CLEAR
3505 015072 010513      7$:  MOV    R5,(R3)      ;SET BIT 12
3506 015074 005013      CLR    (R3)        ;CLR RECEIVER CONTROL REGISTER
3507 015076 011304      MOV    (R3),R4      ;READ THE RECEIVER CONTROL REGISTER
3508 015100 005005      CLR    R5          ;SET 'EXPECTED'
3509 015102 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3510 015104 001401      BEQ    10$         ;BIT 12 IS OK
3511 015106 104003      HLT    3           ;BIT FAILED TO CLEAR
3512 015110 104400      10$: SCOPE        ;SCOPE THIS TEST
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526

```

***** TEST 67 *****
 *RECEIVER CONTROL REGISTER BIT 13 READ ONLY DEVICE RESET AND CLEAR TEST
 *WRITE BIT 13 A ONE AND VERIFY A ZERO IS READ BACK
 *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.

 *
 : TEST 67
 *

```

3527 015112 012737 000067 001226 TST67: MOV    #67,@TSTNO
3528 015120 012737 015224 001216      MOV    #TST70,NEXT
3529 015126 013703 001404      MOV    RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
3530 015132 012705 020000      MOV    #BIT13,R5    ;GET BIT 13
3531 015136 010513      MOV    R5,(R3)      ;WRITE BIT 13 TO RECEIVER CONTROL REGISTER
3532 015140 011304      MOV    (R3),R4      ;READ RECEIVER CONTROL REGISTER BACK
3533 015142 005005      CLR    R5          ;SET 'EXPECTED'
3534 015144 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3535 015146 001401      BEQ    5$          ;BIT 13 IS OK
3536 015150 104003      HLT    3           ;BIT FAILED TO CLR
3537 015152 012705 020000 5$:  MOV    #BIT13,R5    ;RELOAD THE BIT
3538 015156 010513      MOV    R5,(R3)      ;WRITE BIT 13 TO THE REG
3539 015160 052777 000400 164224 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3540 015166 004737 005044      JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
3541 015172 011304      MOV    (R3),R4      ;GET BIT 13
3542 015174 032704 020000      BIT    #BIT13,R4    ;TEST BIT 13 FOR RESET CLR
3543 015200 001401      BEQ    7$          ;BIT 13 IS OK
3544 015202 104003      HLT    3           ;BIT FAILED TO CLEAR
3545 015204 010513      7$:  MOV    R5,(R3)      ;SET BIT 13

```

RECEIVER CONTROL REGISTER READ ONLY BIT 13 TEST

```
3546 015206 005013      CLR      (R3)          ;CLR RECEIVER CONTROL REGISTER
3547 015210 011304      MOV      (R3),R4      ;READ THE RECEIVER CONTROL REGISTER
3548 015212 005005      CLR      R5          ;SET 'EXPECTED'
3549 015214 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
3550 015216 001401      BEQ      10$      ;BIT 13 IS OK
3551 015220 104003      HLT      3          ;BIT FAILED TO CLEAR
3552 015222 104400      10$: SCOPE      ;SCOPE THIS TEST
3553
3554
3555
```

```
***** TEST 70 *****
;RECEIVER CONTROL REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST
;WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK
;REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
*****
;
; TEST 70
;
*****
```

```
3566
3567 015224 012737 000070 001226 TST70: MOV      #70,@TSTNO
3568 015232 012737 015336 001216      MOV      #TST71,NEXT
3569 015240 013703 001404          MOV      RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
3570 015244 012705 040000          MOV      #BIT14,R5      ;GET BIT 14
3571 015250 010513          MOV      R5,(R3)      ;WRITE BIT 14 TO RECEIVER CONTROL REGISTER
3572 015252 011304          MOV      (R3),R4      ;READ RECEIVER CONTROL REGISTER BACK
3573 015254 005005          CLR      R5          ;SET 'EXPECTED'
3574 015256 020504          CMP      R5,R4      ;R5=GOOD,R4= ?
3575 015260 001401          BEQ      5$          ;BIT 14 IS OK
3576 015262 104003          HLT      3          ;BIT FAILED TO CLR
3577 015264 012705 040000      5$: MOV      #BIT14,R5      ;RELOAD THE BIT
3578 015270 010513          MOV      R5,(R3)      ;WRITE BIT 14 TO THE REG
3579 015272 052777 000400 164112      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3580 015300 004737 005044          JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
3581 015304 011304          MOV      (R3),R4      ;GET BIT 14
3582 015306 032704 040000      BIT      #BIT14,R4      ;TEST BIT 14 FOR RESET CLR
3583 015312 001401          BEQ      7$          ;BIT 14 IS OK
3584 015314 104003          HLT      3          ;BIT FAILED TO CLEAR
3585 015316 010513      7$: MOV      R5,(R3)      ;SET BIT 14
3586 015320 005013          CLR      (R3)      ;CLR RECEIVER CONTROL REGISTER
3587 015322 011304          MOV      (R3),R4      ;READ THE RECEIVER CONTROL REGISTER
3588 015324 005005          CLR      R5          ;SET 'EXPECTED'
3589 015326 020504          CMP      R5,R4      ;R5=GOOD,R4= ?
3590 015330 001401          BEQ      10$      ;BIT 14 IS OK
3591 015332 104003          HLT      3          ;BIT FAILED TO CLEAR
3592 015334 104400      10$: SCOPE      ;SCOPE THIS TEST
3593
3594
```

```
***** TEST 71 *****
;RECEIVER CONTROL REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
;WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
;REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
*****
```



```

3602
3603      : TEST 71
3604      :
3605      : *****
3606      : *****
3607 015336 012737 000071 001226 TST71: MOV #71,@TSTNO
3608 015344 012737 015450 001216      MOV #TST72,NEXT
3609 015352 013703 001404      MOV RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
3610 015356 012705 100000      MOV #BIT15,R5      ;GET BIT 15
3611 015362 010513      MOV R5,(R3)      ;WRITE BIT 15 TO RECEIVER CONTROL REGISTER
3612 015364 011304      MOV (R3),R4      ;READ RECEIVER CONTROL REGISTER BACK
3613 015366 005005      CLR R5      ;SET 'EXPECTED'
3614 015370 020504      CMP R5,R4      ;R5=GOOD,R4= ?
3615 015372 001401      BEQ 5$      ;BIT 15 IS OK
3616 015374 104003      HLT 3      ;BIT FAILED TO CLR
3617 015376 012705 100000 5$: MOV #BIT15,R5      ;RELOAD THE BIT
3618 015402 010513      MOV R5,(R3)      ;WRITE BIT 15 TO THE REG
3619 015404 052777 000400 164000 BIS #MRESET,@TXCSR      ;RESET THE DEVICE
3620 015412 004737 005044      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
3621 015416 011304      MOV (R3),R4      ;GET BIT 15
3622 015420 032704 100000      BIT #BIT15,R4      ;TEST BIT 15 FOR RESET CLR
3623 015424 001401      BEQ 7$      ;BIT 15 IS OK
3624 015426 104003      HLT 3      ;BIT FAILED TO CLEAR
3625 015430 010513 7$: MOV R5,(R3)      ;SET BIT 15
3626 015432 005013      CLR (R3)      ;CLR RECEIVER CONTROL REGISTER
3627 015434 011304      MOV (R3),R4      ;READ THE RECEIVER CONTROL REGISTER
3628 015436 005005      CLR R5      ;SET 'EXPECTED'
3629 015440 020504      CMP R5,R4      ;R5=GOOD,R4= ?
3630 015442 001401      BEQ 10$      ;BIT 15 IS OK
3631 015444 104003      HLT 3      ;BIT FAILED TO CLEAR
3632 015446 104400 10$: SCOPE      ;SCOPE THIS TEST
3633
3634
3635      : ***** TEST 72 *****
3636      : *RECEIVER BUFFER REGISTER BIT 0 READ ONLY DEVICE RESET AND CLEAR TEST
3637      : *WRITE BIT 0 A ONE AND VERIFY A ZERO IS READ BACK
3638      : *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
3639      : *****
3640
3641      : *****
3642      :
3643      : TEST 72
3644      :
3645      : *****
3646      : *****
3647 015450 012737 000072 001226 TST72: MOV #72,@TSTNO
3648 015456 012737 015562 001216      MOV #TST73,NEXT
3649 015464 013703 001406      MOV RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
3650 015470 012705 000001      MOV #BIT0,R5      ;GET BIT 0
3651 015474 010513      MOV R5,(R3)      ;WRITE BIT 0 TO RECEIVER BUFFER REGISTER
3652 015476 011304      MOV (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
3653 015500 005005      CLR R5      ;SET 'EXPECTED'
3654 015502 020504      CMP R5,R4      ;R5=GOOD,R4= ?
3655 015504 001401      BEQ 5$      ;BIT 0 IS OK
3656 015506 104003      HLT 3      ;BIT FAILED TO CLR
3657 015510 012705 000001 5$: MOV #BIT0,R5      ;RELOAD THE BIT

```

3658	015514	010513			MOV	R5,(R3)	;WRITE BIT 0 TO THE REG
3659	015516	052777	000400	163666	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
3660	015524	004737	005044		JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
3661	015530	011304			MOV	(R3),R4	;GET BIT 0
3662	015532	032704	000001		BIT	#BIT0,R4	;TEST BIT 0 FOR RESET CLR
3663	015536	001401			BEQ	7\$	;BIT 0 IS OK
3664	015540	104003			HLT	3	;BIT FAILED TO CLEAR
3665	015542	010513			7\$: MOV	R5,(R3)	;SET BIT 0
3666	015544	005013			CLR	(R3)	;CLR RECEIVER BUFFER REGISTER
3667	015546	011304			MOV	(R3),R4	;READ THE RECEIVER BUFFER REGISTER
3668	015550	005005			CLR	R5	;SET 'EXPECTED'
3669	015552	020504			CMP	R5,R4	;R5=GOOD,R4= ?
3670	015554	001401			BEQ	10\$	;BIT 0 IS OK
3671	015556	104003			HLT	3	;BIT FAILED TO CLEAR
3672	015560	104400			10\$: SCOPE		;SCOPE THIS TEST

***** TEST 73 *****
 *RECEIVER BUFFER REGISTER BIT 1 READ ONLY DEVICE RESET AND CLEAR TEST
 *WRITE BIT 1 A ONE AND VERIFY A ZERO IS READ BACK
 *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.

 *
 : TEST 73
 *

3687	015562	012737	000073	001226	TST73: MOV	#73,@TSTNO	
3688	015570	012737	015674	001216	MOV	#TST74,NEXT	
3689	015576	013703	001406		MOV	RXDBUF,R3	;GET THE RECEIVER BUFFER REGISTER
3690	015602	012705	000002		MOV	#BIT1,R5	;GET BIT 1
3691	015606	010513			MOV	R5,(R3)	;WRITE BIT 1 TO RECEIVER BUFFER REGISTER
3692	015610	011304			MOV	(R3),R4	;READ RECEIVER BUFFER REGISTER BACK
3693	015612	005005			CLR	R5	;SET 'EXPECTED'
3694	015614	020504			CMP	R5,R4	;R5=GOOD,R4= ?
3695	015616	001401			BEQ	5\$	;BIT 1 IS OK
3696	015620	104003			HLT	3	;BIT FAILED TO CLR
3697	015622	012705	000002		5\$: MOV	#BIT1,R5	;RELOAD THE BIT
3698	015626	010513			MOV	R5,(R3)	;WRITE BIT 1 TO THE REG
3699	015630	052777	000400	163554	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
3700	015636	004737	005044		JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
3701	015642	011304			MOV	(R3),R4	;GET BIT 1
3702	015644	032704	000002		BIT	#BIT1,R4	;TEST BIT 1 FOR RESET CLR
3703	015650	001401			BEQ	7\$	;BIT 1 IS OK
3704	015652	104003			HLT	3	;BIT FAILED TO CLEAR
3705	015654	010513			7\$: MOV	R5,(R3)	;SET BIT 1
3706	015656	005013			CLR	(R3)	;CLR RECEIVER BUFFER REGISTER
3707	015660	011304			MOV	(R3),R4	;READ THE RECEIVER BUFFER REGISTER
3708	015662	005005			CLR	R5	;SET 'EXPECTED'
3709	015664	020504			CMP	R5,R4	;R5=GOOD,R4= ?
3710	015666	001401			BEQ	10\$	;BIT 1 IS OK
3711	015670	104003			HLT	3	;BIT FAILED TO CLEAR
3712	015672	104400			10\$: SCOPE		;SCOPE THIS TEST


```

3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727 015674 012737 000074 001226
3728 015702 012737 016006 001216
3729 015710 013703 001406
3730 015714 012705 000004
3731 015720 010513
3732 015722 011304
3733 015724 005005
3734 015726 020504
3735 015730 001401
3736 015732 104003
3737 015734 012705 000004
3738 015740 010513
3739 015742 052777 000400 163442
3740 015750 004737 005044
3741 015754 011304
3742 015756 032704 000004
3743 015762 001401
3744 015764 104003
3745 015766 010513
3746 015770 005013
3747 015772 011304
3748 015774 005005
3749 015776 020504
3750 016000 001401
3751 016002 104003
3752 016004 104400
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767 016006 012737 000075 001226
3768 016014 012737 016120 001216
3769 016022 013703 001406

;***** TEST 74 *****
;*RECEIVER BUFFER REGISTER BIT 2 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 2 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

;*****
;
; TEST 74
;
;*****
;*****
TST74: MOV #74,@TSTNO
MOV #TST75,NEXT
MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
MOV #BIT2,R5 ;GET BIT 2
MOV R5,(R3) ;WRITE BIT 2 TO RECEIVER BUFFER REGISTER
MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
CLR R5 ;SET 'EXPECTED'
CMP R5,R4 ;R5=GOOD,R4= ?
BEQ 5$ ;BIT 2 IS OK
HLT 3 ;BIT FAILED TO CLR
5$: MOV #BIT2,R5 ;RELOAD THE BIT
MOV R5,(R3) ;WRITE BIT 2 TO THE REG
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
MOV (R3),R4 ;GET BIT 2
BIT #BIT2,R4 ;TEST BIT 2 FOR RESET CLR
BEQ 7$ ;BIT 2 IS OK
HLT 3 ;BIT FAILED TO CLEAR
7$: MOV R5,(R3) ;SET BIT 2
CLR (R3) ;CLR RECEIVER BUFFER REGISTER
MOV (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
CLR R5 ;SET 'EXPECTED'
CMP R5,R4 ;R5=GOOD,R4= ?
BEQ 10$ ;BIT 2 IS OK
HLT 3 ;BIT FAILED TO CLEAR
10$: SCOPE ;SCOPE THIS TEST

;***** TEST 75 *****
;*RECEIVER BUFFER REGISTER BIT 3 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 3 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

;*****
;
; TEST 75
;
;*****
;*****
TST75: MOV #75,@TSTNO
MOV #TST76,NEXT
MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER

```

```

3770 016026 012705 000010      MOV    #BIT3,R5      ;GET BIT 3
3771 016032 010513      MOV    R5,(R3)      ;WRITE BIT 3 TO RECEIVER BUFFER REGISTER
3772 016034 011304      MOV    (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
3773 016036 005005      CLR    R5          ;SET 'EXPECTED'
3774 016040 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3775 016042 001401      BEQ    5$          ;BIT 3 IS OK
3776 016044 104003      HLT    3           ;BIT FAILED TO CLR
3777 016046 012705 000010 5$: MOV    #BIT3,R5      ;RELOAD THE BIT
3778 016052 010513      MOV    R5,(R3)      ;WRITE BIT 3 TO THE REG
3779 016054 052777 000400 163330 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3780 016062 004737 005044      JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
3781 016066 011304      MOV    (R3),R4      ;GET BIT 3
3782 016070 032704 000010      BIT    #BIT3,R4      ;TEST BIT 3 FOR RESET CLR
3783 016074 001401      BEQ    7$          ;BIT 3 IS OK
3784 016076 104003      HLT    3           ;BIT FAILED TO CLEAR
3785 016100 010513      7$: MOV    R5,(R3)      ;SET BIT 3
3786 016102 005013      CLR    (R3)        ;CLR RECEIVER BUFFER REGISTER
3787 016104 011304      MOV    (R3),R4      ;READ THE RECEIVER BUFFER REGISTER
3788 016106 005005      CLR    R5          ;SET 'EXPECTED'
3789 016110 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3790 016112 001401      BEQ    10$         ;BIT 3 IS OK
3791 016114 104003      HLT    3           ;BIT FAILED TO CLEAR
3792 016116 104400      10$: SCOPE      ;SCOPE THIS TEST
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806

```

***** TEST 76 *****
 *RECEIVER BUFFER REGISTER BIT 4 READ ONLY DEVICE RESET AND CLEAR TEST
 *WRITE BIT 4 A ONE AND VERIFY A ZERO IS READ BACK
 *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.

 *
 : TEST 76
 *

```

3807 016120 012737 000076 001226 TST76: MOV    #76,@TSTNO
3808 016126 012737 016232 001216      MOV    #TST77,NEXT
3809 016134 013703 001406      MOV    RXDBUF,R3    ;GET THE RECEIVER BUFFER REGISTER
3810 016140 012705 000020      MOV    #BIT4,R5      ;GET BIT 4
3811 016144 010513      MOV    R5,(R3)      ;WRITE BIT 4 TO RECEIVER BUFFER REGISTER
3812 016146 011304      MOV    (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
3813 016150 005005      CLR    R5          ;SET 'EXPECTED'
3814 016152 020504      CMP    R5,R4        ;R5=GOOD,R4= ?
3815 016154 001401      BEQ    5$          ;BIT 4 IS OK
3816 016156 104003      HLT    3           ;BIT FAILED TO CLR
3817 016160 012705 000020 5$: MOV    #BIT4,R5      ;RELOAD THE BIT
3818 016164 010513      MOV    R5,(R3)      ;WRITE BIT 4 TO THE REG
3819 016166 052777 000400 163216 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3820 016174 004737 005044      JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
3821 016200 011304      MOV    (R3),R4      ;GET BIT 4
3822 016202 032704 000020      BIT    #BIT4,R4      ;TEST BIT 4 FOR RESET CLR
3823 016206 001401      BEQ    7$          ;BIT 4 IS OK
3824 016210 104003      HLT    3           ;BIT FAILED TO CLEAR
3825 016212 010513      7$: MOV    R5,(R3)      ;SET BIT 4

```



```

3826 016214 005013          CLR      (R3)          ;CLR RECEIVER BUFFER REGISTER
3827 016216 011304          MOV      (R3),R4        ;READ THE RECEIVER BUFFER REGISTER
3828 016220 005005          CLR      R5            ;SET 'EXPECTED'
3829 016222 020504          CMP      R5,R4          ;R5=GOOD,R4= ?
3830 016224 001401          BEQ      10$           ;BIT 4 IS OK
3831 016226 104003          HLT      3              ;BIT FAILED TO CLEAR
3832 016230 104400          10$: SCOPE             ;SCOPE THIS TEST
3833
3834
3835          ;***** TEST 77 *****
3836          ;*RECEIVER BUFFER REGISTER BIT 5 READ ONLY DEVICE RESET AND CLEAR TEST
3837          ;*WRITE BIT 5 A ONE AND VERIFY A ZERO IS READ BACK
3838          ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
3839          ;*****
3840
3841          ;*****
3842          ;*
3843          ;* TEST 77
3844          ;*
3845          ;*****
3846          ;*****
3847 016232 012737 000077 001226 TST77: MOV      #77,@TSTNO
3848 016240 012737 016344 001216      MOV      #TST100,NEXT
3849 016246 013703 001406              MOV      RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
3850 016252 012705 000040              MOV      #BIT5,R5      ;GET BIT 5
3851 016256 010513              MOV      R5,(R3)          ;WRITE BIT 5 TO RECEIVER BUFFER REGISTER
3852 016260 011304              MOV      (R3),R4          ;READ RECEIVER BUFFER REGISTER BACK
3853 016262 005005              CLR      R5            ;SET 'EXPECTED'
3854 016264 020504              CMP      R5,R4          ;R5=GOOD,R4= ?
3855 016266 001401              BEQ      5$            ;BIT 5 IS OK
3856 016270 104003              HLT      3              ;BIT FAILED TO CLR
3857 016272 012705 000040          5$: MOV      #BIT5,R5      ;RELOAD THE BIT
3858 016276 010513              MOV      R5,(R3)          ;WRITE BIT 5 TO THE REG
3859 016300 052777 000400 163104      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3860 016306 004737 005044              JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
3861 016312 011304              MOV      (R3),R4          ;GET BIT 5
3862 016314 032704 000040          BIT      #BIT5,R4          ;TEST BIT 5 FOR RESET CLR
3863 016320 001401              BEQ      7$            ;BIT 5 IS OK
3864 016322 104003              HLT      3              ;BIT FAILED TO CLEAR
3865 016324 010513          7$: MOV      R5,(R3)          ;SET BIT 5
3866 016326 005013              CLR      (R3)          ;CLR RECEIVER BUFFER REGISTER
3867 016330 011304              MOV      (R3),R4          ;READ THE RECEIVER BUFFER REGISTER
3868 016332 005005              CLR      R5            ;SET 'EXPECTED'
3869 016334 020504              CMP      R5,R4          ;R5=GOOD,R4= ?
3870 016336 001401              BEQ      10$           ;BIT 5 IS OK
3871 016340 104003              HLT      3              ;BIT FAILED TO CLEAR
3872 016342 104400          10$: SCOPE             ;SCOPE THIS TEST
3873
3874
3875          ;***** TEST 100 *****
3876          ;*RECEIVER BUFFER REGISTER BIT 6 READ ONLY DEVICE RESET AND CLEAR TEST
3877          ;*WRITE BIT 6 A ONE AND VERIFY A ZERO IS READ BACK
3878          ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
3879          ;*****
3880
3881          ;*****

```

```

3882      :
3883      : TEST 100
3884      :
3885      : *****
3886      : *****
3887 016344 012737 000100 001226 TST100: MOV #100,@TSTNO
3888 016352 012737 016456 001216      MOV #TST101,NEXT
3889 016360 013703 001406      MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
3890 016364 012705 000100      MOV #BIT6,R5 ;GET BIT 6
3891 016370 010513      MOV R5,(R3) ;WRITE BIT 6 TO RECEIVER BUFFER REGISTER
3892 016372 011304      MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
3893 016374 005005      CLR R5 ;SET 'EXPECTED'
3894 016376 020504      CMP R5,R4 ;R5=GOOD,R4= ?
3895 016400 001401      BEQ 5$ ;BIT 6 IS OK
3896 016402 104003      HLT 3 ;BIT FAILED TO CLR
3897 016404 012705 000100 5$: MOV #BIT6,R5 ;RELOAD THE BIT
3898 016410 010513      MOV R5,(R3) ;WRITE BIT 6 TO THE REG
3899 016412 052777 000400 162772 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3900 016420 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3901 016424 011304      MOV (R3),R4 ;GET BIT 6
3902 016426 032704 000100      BIT #BIT6,R4 ;TEST BIT 6 FOR RESET CLR
3903 016432 001401      BEQ 7$ ;BIT 6 IS OK
3904 016434 104003      HLT 3 ;BIT FAILED TO CLEAR
3905 016436 010513 7$: MOV R5,(R3) ;SET BIT 6
3906 016440 005013      CLR (R3) ;CLR RECEIVER BUFFER REGISTER
3907 016442 011304      MOV (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
3908 016444 005005      CLR R5 ;SET 'EXPECTED'
3909 016446 020504      CMP R5,R4 ;R5=GOOD,R4= ?
3910 016450 001401      BEQ 10$ ;BIT 6 IS OK
3911 016452 104003      HLT 3 ;BIT FAILED TO CLEAR
3912 016454 104400 10$: SCOPE ;SCOPE THIS TEST
3913
3914
3915      : ***** TEST 101 *****
3916      : *RECEIVER BUFFER REGISTER BIT 7 READ ONLY DEVICE RESET AND CLEAR TEST
3917      : *WRITE BIT 7 A ONE AND VERIFY A ZERO IS READ BACK
3918      : *REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
3919      : *****
3920
3921      : *****
3922      :
3923      : TEST 101
3924      :
3925      : *****
3926      : *****
3927 016456 012737 000101 001226 TST101: MOV #101,@TSTNO
3928 016464 012737 016570 001216      MOV #TST102,NEXT
3929 016472 013703 001406      MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
3930 016476 012705 000200      MOV #BIT7,R5 ;GET BIT 7
3931 016502 010513      MOV R5,(R3) ;WRITE BIT 7 TO RECEIVER BUFFER REGISTER
3932 016504 011304      MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
3933 016506 005005      CLR R5 ;SET 'EXPECTED'
3934 016510 020504      CMP R5,R4 ;R5=GOOD,R4= ?
3935 016512 001401      BEQ 5$ ;BIT 7 IS OK
3936 016514 104003      HLT 3 ;BIT FAILED TO CLR
3937 016516 012705 000200 5$: MOV #BIT7,R5 ;RELOAD THE BIT

```


RECEIVER BUFFER REGISTER READ ONLY BIT 7 TEST

```
3938 016522 010513      MOV      R5,(R3)      ;WRITE BIT 7 TO THE REG
3939 016524 052777 000400 162660  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3940 016532 004737 005044      JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3941 016536 011304      MOV      (R3),R4      ;GET BIT 7
3942 016540 032704 000200      BIT      #BIT7,R4      ;TEST BIT 7 FOR RESET CLR
3943 016544 001401      BEQ      7$      ;BIT 7 IS OK
3944 016546 104003      HLT      3      ;BIT FAILED TO CLEAR
3945 016550 010513 7$:      MOV      R5,(R3)      ;SET BIT 7
3946 016552 005013      CLR      (R3)      ;CLR RECEIVER BUFFER REGISTER
3947 016554 011304      MOV      (R3),R4      ;READ THE RECEIVER BUFFER REGISTER
3948 016556 005005      CLR      R5      ;SET 'EXPECTED'
3949 016560 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
3950 016562 001401      BEQ      10$     ;BIT 7 IS OK
3951 016564 104003      HLT      3      ;BIT FAILED TO CLEAR
3952 016566 104400 10$:      SCOPE      ;SCOPE THIS TEST
3953
3954
3955
```

```
***** TEST 102 *****
*RECEIVER BUFFER REGISTER BIT 8 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 8 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967 016570 012737 000102 001226 TST102: MOV      #102,@TSTNO
3968 016576 012737 016702 001216      MOV      #TST103,NEXT
3969 016604 013703 001406      MOV      RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
3970 016610 012705 000400      MOV      #BIT8,R5      ;GET BIT 8
3971 016614 010513      MOV      R5,(R3)      ;WRITE BIT 8 TO RECEIVER BUFFER REGISTER
3972 016616 011304      MOV      (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
3973 016620 005005      CLR      R5      ;SET 'EXPECTED'
3974 016622 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
3975 016624 001401      BEQ      5$      ;BIT 8 IS OK
3976 016626 104003      HLT      3      ;BIT FAILED TO CLR
3977 016630 012705 000400 5$:      MOV      #BIT8,R5      ;RELOAD THE BIT
3978 016634 010513      MOV      R5,(R3)      ;WRITE BIT 8 TO THE REG
3979 016636 052777 000400 162546  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3980 016644 004737 005044      JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3981 016650 011304      MOV      (R3),R4      ;GET BIT 8
3982 016652 032704 000400      BIT      #BIT8,R4      ;TEST BIT 8 FOR RESET CLR
3983 016656 001401      BEQ      7$      ;BIT 8 IS OK
3984 016660 104003      HLT      3      ;BIT FAILED TO CLEAR
3985 016662 010513 7$:      MOV      R5,(R3)      ;SET BIT 8
3986 016664 005013      CLR      (R3)      ;CLR RECEIVER BUFFER REGISTER
3987 016666 011304      MOV      (R3),R4      ;READ THE RECEIVER BUFFER REGISTER
3988 016670 005005      CLR      R5      ;SET 'EXPECTED'
3989 016672 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
3990 016674 001401      BEQ      10$     ;BIT 8 IS OK
3991 016676 104003      HLT      3      ;BIT FAILED TO CLEAR
3992 016700 104400 10$:      SCOPE      ;SCOPE THIS TEST
3993
```

3994
 3995
 3996
 3997
 3998
 3999
 4000
 4001
 4002
 4003
 4004
 4005
 4006
 4007
 4008
 4009
 4010
 4011
 4012
 4013
 4014
 4015
 4016
 4017
 4018
 4019
 4020
 4021
 4022
 4023
 4024
 4025
 4026
 4027
 4028
 4029
 4030
 4031
 4032
 4033
 4034
 4035
 4036
 4037
 4038
 4039
 4040
 4041
 4042
 4043
 4044
 4045
 4046
 4047
 4048
 4049

016702 012737 000103 001226
 016710 012737 017014 001216
 016716 013703 001406
 016722 012705 001000
 016726 010513
 016730 011304
 016732 005005
 016734 020504
 016736 001401
 016740 104003
 016742 012705 001000
 016746 010513
 016750 052777 000400 162434
 016756 004737 005044
 016762 011304
 016764 032704 001000
 016770 001401
 016772 104003
 016774 010513
 016776 005013
 017000 011304
 017002 005005
 017004 020504
 017006 001401
 017010 104003
 017012 104400

```

***** TEST 103 *****
*RECEIVER BUFFER REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

*****
*
TEST 103
*
*****
*****
TST103: MOV #103,@TSTNO
        MOV #TST104,NEXT
        MOV RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
        MOV #BIT9,R5       ;GET BIT 9
        MOV R5,(R3)        ;WRITE BIT 9 TO RECEIVER BUFFER REGISTER
        MOV (R3),R4        ;READ RECEIVER BUFFER REGISTER BACK
        CLR R5             ;SET 'EXPECTED'
        CMP R5,R4          ;R5=GOOD,R4= ?
        BEQ 5$             ;BIT 9 IS OK
        HLT 3              ;BIT FAILED TO CLR
5$:      MOV #BIT9,R5       ;RELOAD THE BIT
        MOV R5,(R3)        ;WRITE BIT 9 TO THE REG
        BIS #MRESET,@TXCSR ;RESET THE DEVICE
        JSR PC,SMALL       ;WAIT FOR RESET TO FINISH
        MOV (R3),R4        ;GET BIT 9
        BIT #BIT9,R4       ;TEST BIT 9 FOR RESET CLR
        BEQ 7$             ;BIT 9 IS OK
        HLT 3              ;BIT FAILED TO CLEAR
7$:      MOV R5,(R3)        ;SET BIT 9
        CLR (R3)           ;CLR RECEIVER BUFFER REGISTER
        MOV (R3),R4        ;READ THE RECEIVER BUFFER REGISTER
        CLR R5             ;SET 'EXPECTED'
        CMP R5,R4          ;R5=GOOD,R4= ?
        BEQ 10$            ;BIT 9 IS OK
        HLT 3              ;BIT FAILED TO CLEAR
10$:     SCOPE              ;SCOPE THIS TEST
  
```

```

***** TEST 104 *****
*RECEIVER BUFFER REGISTER BIT 10 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 10 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
TEST 104
*
*****
*****
TST104: MOV #104,@TSTNO
        MOV #TST105,NEXT
        MOV RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
  
```



```

4050 017034 012705 002000      MOV    #BIT10,R5      ;GET BIT 10
4051 017040 010513      MOV    R5,(R3)      ;WRITE BIT 10 TO RECEIVER BUFFER REGISTER
4052 017042 011304      MOV    (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
4053 017044 005005      CLR    R5      ;SET 'EXPECTED'
4054 017046 020504      CMP    R5,R4      ;R5=GOOD,R4= ?
4055 017050 001401      BEQ    5$      ;BIT 10 IS OK
4056 017052 104003      HLT    3      ;BIT FAILED TO CLR
4057 017054 012705 002000 5$:  MOV    #BIT10,R5      ;RELOAD THE BIT
4058 017060 010513      MOV    R5,(R3)      ;WRITE BIT 10 TO THE REG
4059 017062 052777 000400 162322 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
4060 017070 004737 005044      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
4061 017074 011304      MOV    (R3),R4      ;GET BIT 10
4062 017076 032704 002000      BIT    #BIT10,R4      ;TEST BIT 10 FOR RESET CLR
4063 017102 001401      BEQ    7$      ;BIT 10 IS OK
4064 017104 104003      HLT    3      ;BIT FAILED TO CLEAR
4065 017106 010513 7$:  MOV    R5,(R3)      ;SET BIT 10
4066 017110 005013      CLR    (R3)      ;CLR RECEIVER BUFFER REGISTER
4067 017112 011304      MOV    (R3),R4      ;READ THE RECEIVER BUFFER REGISTER
4068 017114 005005      CLR    R5      ;SET 'EXPECTED'
4069 017116 020504      CMP    R5,R4      ;R5=GOOD,R4= ?
4070 017120 001401      BEQ    10$     ;BIT 10 IS OK
4071 017122 104003      HLT    3      ;BIT FAILED TO CLEAR
4072 017124 104400 10$:  SCOPE      ;SCOPE THIS TEST

```

```

4073
4074
4075      ;***** TEST 105 *****
4076      ;*RECEIVER BUFFER REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
4077      ;*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
4078      ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
4079      ;*****

```

```

4080
4081      ;*****
4082      ;*
4083      ;* TEST 105
4084      ;*
4085      ;*****
4086      ;*****

```

```

4087 017126 012737 000105 001226 TST105: MOV    #105,@#TSTNO
4088 017134 012737 017240 001216      MOV    #TST106,NEXT
4089 017142 013703 001406      MOV    RXDBUF,R3      ;GET THE RECEIVER BUFFER REGISTER
4090 017146 012705 010000      MOV    #BIT12,R5      ;GET BIT 12
4091 017152 010513      MOV    R5,(R3)      ;WRITE BIT 12 TO RECEIVER BUFFER REGISTER
4092 017154 011304      MOV    (R3),R4      ;READ RECEIVER BUFFER REGISTER BACK
4093 017156 005005      CLR    R5      ;SET 'EXPECTED'
4094 017160 020504      CMP    R5,R4      ;R5=GOOD,R4= ?
4095 017162 001401      BEQ    5$      ;BIT 12 IS OK
4096 017164 104003      HLT    3      ;BIT FAILED TO CLR
4097 017166 012705 010000 5$:  MOV    #BIT12,R5      ;RELOAD THE BIT
4098 017172 010513      MOV    R5,(R3)      ;WRITE BIT 12 TO THE REG
4099 017174 052777 000400 162210 BIS    #MRESET,@TXCSR ;RESET THE DEVICE
4100 017202 004737 005044      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
4101 017206 011304      MOV    (R3),R4      ;GET BIT 12
4102 017210 032704 010000      BIT    #BIT12,R4      ;TEST BIT 12 FOR RESET CLR
4103 017214 001401      BEQ    7$      ;BIT 12 IS OK
4104 017216 104003      HLT    3      ;BIT FAILED TO CLEAR
4105 017220 010513 7$:  MOV    R5,(R3)      ;SET BIT 12

```

```

4106 017222 005013      CLR      (R3)      ;CLR RECEIVER BUFFER REGISTER
4107 017224 011304      MOV      (R3),R4    ;READ THE RECEIVER BUFFER REGISTER
4108 017226 005005      CLR      R5      ;SET 'EXPECTED'
4109 017230 020504      CMP      R5,R4    ;R5=GOOD,R4= ?
4110 017232 001401      BEQ      10$     ;BIT 12 IS OK
4111 017234 104003      HLT      3       ;BIT FAILED TO CLEAR
4112 017236 104400      10$: SCOPE    ;SCOPE THIS TEST
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126

```

```

;***** TEST 106 *****
;*RECEIVER BUFFER REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

```

```

;*****
;:*****
;: TEST 106
;:*****
;:*****

```

```

4127 017240 012737 000106 001226 TST106: MOV      #106,@TSTNO
4128 017246 012737 017352 001216      MOV      #TST107,NEXT
4129 017254 013703 001406          MOV      RXDBUF,R3    ;GET THE RECEIVER BUFFER REGISTER
4130 017260 012705 040000          MOV      #BIT14,R5    ;GET BIT 14
4131 017264 010513          MOV      R5,(R3)    ;WRITE BIT 14 TO RECEIVER BUFFER REGISTER
4132 017266 011304          MOV      (R3),R4    ;READ RECEIVER BUFFER REGISTER BACK
4133 017270 005005          CLR      R5      ;SET 'EXPECTED'
4134 017272 020504          CMP      R5,R4    ;R5=GOOD,R4= ?
4135 017274 001401          BEQ      5$      ;BIT 14 IS OK
4136 017276 104003          HLT      3       ;BIT FAILED TO CLR
4137 017300 012705 040000      5$: MOV      #BIT14,R5    ;RELOAD THE BIT
4138 017304 010513          MOV      R5,(R3)    ;WRITE BIT 14 TO THE REG
4139 017306 052777 000400 162076      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4140 017314 004737 005044          JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
4141 017320 011304          MOV      (R3),R4    ;GET BIT 14
4142 017322 032704 040000      BIT      #BIT14,R4    ;TEST BIT 14 FOR RESET CLR
4143 017326 001401          BEQ      7$      ;BIT 14 IS OK
4144 017330 104003          HLT      3       ;BIT FAILED TO CLEAR
4145 017332 010513      7$: MOV      R5,(R3)    ;SET BIT 14
4146 017334 005013          CLR      (R3)    ;CLR RECEIVER BUFFER REGISTER
4147 017336 011304          MOV      (R3),R4    ;READ THE RECEIVER BUFFER REGISTER
4148 017340 005005          CLR      R5      ;SET 'EXPECTED'
4149 017342 020504          CMP      R5,R4    ;R5=GOOD,R4= ?
4150 017344 001401          BEQ      10$     ;BIT 14 IS OK
4151 017346 104003          HLT      3       ;BIT FAILED TO CLEAR
4152 017350 104400      10$: SCOPE    ;SCOPE THIS TEST
4153
4154
4155
4156
4157
4158
4159
4160
4161

```

```

;***** TEST 107 *****
;*RECEIVER BUFFER REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
;*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

```

```

;*****

```



```

4162      :
4163      : TEST 107
4164      :
4165      : *****
4166      : *****
4167 017352 012737 000107 001226 TST107: MOV #107,@#TSTNO
4168 017360 012737 017464 001216 MOV #TST110,NEXT
4169 017366 013703 001406 MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
4170 017372 012705 100000 MOV #BIT15,R5 ;GET BIT 15
4171 017376 010513 MOV R5,(R3) ;WRITE BIT 15 TO RECEIVER BUFFER REGISTER
4172 017400 011304 MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
4173 017402 005005 CLR R5 ;SET 'EXPECTED'
4174 017404 020504 CMP R5,R4 ;R5=GOOD,R4=?
4175 017406 001401 BEQ 5$ ;BIT 15 IS OK
4176 017410 104003 HLT 3 ;BIT FAILED TO CLR
4177 017412 012705 100000 5$: MOV #BIT15,R5 ;RELOAD THE BIT
4178 017416 010513 MOV R5,(R3) ;WRITE BIT 15 TO THE REG
4179 017420 052777 000400 161764 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4180 017426 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4181 017432 011304 MOV (R3),R4 ;GET BIT 15
4182 017434 032704 100000 BIT #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
4183 017440 001401 BEQ 7$ ;BIT 15 IS OK
4184 017442 104003 HLT 3 ;BIT FAILED TO CLEAR
4185 017444 010513 7$: MOV R5,(R3) ;SET BIT 15
4186 017446 005013 CLR (R3) ;CLR RECEIVER BUFFER REGISTER
4187 017450 011304 MOV (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
4188 017452 005005 CLR R5 ;SET 'EXPECTED'
4189 017454 020504 CMP R5,R4 ;R5=GOOD,R4=?
4190 017456 001401 BEQ 10$ ;BIT 15 IS OK
4191 017460 104003 HLT 3 ;BIT FAILED TO CLEAR
4192 017462 104400 10$: SCOPE ;SCOPE THIS TEST
4193
4194      : ***** TEST 110 *****
4195      : *THIS TEST VERIFIES BIT7 OF THE TRANSMITTER CONTROL REGISTER
4196      : *TEST THAT BIT 7 SETS AFTER A RESET AND MRESET
4197      : *VERIFY THAT WRITING THE LOW BYTE OF
4198      : *THE TRANSMITTER BUFFER REGISTER CLEARS BIT 7
4199      : *****
4200
4201      : *****
4202      :
4203      : TEST 110
4204      :
4205      : *****
4206      : *****
4207 017464 012737 000110 001226 TST110: MOV #110,@#TSTNO
4208 017472 012737 017572 001216 MOV #TST111,NEXT
4209 017500 013702 001414 MOV TXDBUF,R2 ;LOAD SECOND REG
4210 017504 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
4211 017510 012705 000200 MOV #BIT7,R5 ;SET 'EXPECTED'
4212 017514 000005 RESET
4213 017516 011304 MOV (R3),R4 ;GET THE BIT
4214 017520 020504 CMP R5,R4 ;R5=GOOD,R4=?
4215 017522 001401 BEQ 1$ ;ARE THEY THE SAME?
4216 017524 104003 HLT 3 ;NO--REPORT THE ERROR
4217 017526 005005 1$: CLR R5 ;SET 'EXPECTED'
  
```

TRANSMITTER CONTROL REGISTER READ ONLY TEST BIT 7

```
4218 017530 005012 CLR (R2) ;WRITE THE LOW BYTE OF THE TXDBUF
4219 017532 011304 MOV (R3),R4 ;READ BACK BIT 7
4220 017534 020504 CMP R5,R4 ;R5=GOOD,R4=?
4221 017536 001401 BEQ 2$ ;ARE THEY THE SAME?
4222 017540 104003 HLT 3 ;NO-BIT 7 FAILED TO CLEAR
4223 017542 012705 000200 2$: MOV #BIT7,R5 ;SET 'EXPECTED'
4224 017546 052777 000400 161636 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4225 017554 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4226 017560 011304 MOV (R3),R4 ;READ BACK BIT 7
4227 017562 020504 CMP R5,R4 ;R5=GOOD,R4=?
4228 017564 001401 BEQ 3$ ;BRANCH IF BIT 7 OK
4229 017566 104003 HLT 3 ;BIT 7 FAILED TO SET AFTER A MRESET
4230 017570 104400 3$: SCOPE ;SCOPE THIS TEST
```

```
***** TEST 111 *****
*TRANSMITTER CONTROL REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
*****
*
* TEST 111
*
*****
```

```
4246 017572 012737 000111 001226 TST111: MOV #111,@TSTNO
4247 017600 012737 017714 001216 MOV #TST112,NEXT
4248 017606 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
4249 017612 012705 001000 MOV #BIT9,R5 ;GET BIT 9
4250 017616 010513 MOV R5,(R3) ;WRITE BIT 9 TO TRANSMITTER CONTROL REGISTER
4251 017620 011304 MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER BACK
4252 017622 005005 CLR R5 ;SET 'EXPECTED'
4253 017624 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4254 017630 020504 CMP R5,R4 ;R5=GOOD,R4=?
4255 017632 001401 BEQ 5$ ;BIT 9 IS OK
4256 017634 104003 HLT 3 ;BIT FAILED TO CLR
4257 017636 012705 001000 5$: MOV #BIT9,R5 ;RELOAD THE BIT
4258 017642 010513 MOV R5,(R3) ;WRITE BIT 9 TO THE REG
4259 017644 052777 000400 161540 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4260 017652 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4261 017656 011304 MOV (R3),R4 ;GET BIT 9
4262 017660 032704 001000 BIT #BIT9,R4 ;TEST BIT 9 FOR RESET CLR
4263 017664 001401 BEQ 7$ ;BIT 9 IS OK
4264 017666 104003 HLT 3 ;BIT FAILED TO CLEAR
4265 017670 010513 7$: MOV R5,(R3) ;SET BIT 9
4266 017672 005013 CLR (R3) ;CLR TRANSMITTER CONTROL REGISTER
4267 017674 011304 MOV (R3),R4 ;READ THE TRANSMITTER CONTROL REGISTER
4268 017676 005005 CLR R5 ;SET 'EXPECTED'
4269 017700 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4270 017704 020504 CMP R5,R4 ;R5=GOOD,R4=?
4271 017706 001401 BEQ 10$ ;BIT 9 IS OK
4272 017710 104003 HLT 3 ;BIT FAILED TO CLEAR
4273 017712 104400 10$: SCOPE ;SCOPE THIS TEST
```


4274
 4275
 4276
 4277
 4278
 4279
 4280
 4281
 4282
 4283
 4284
 4285
 4286
 4287
 4288
 4289
 4290
 4291
 4292
 4293
 4294
 4295
 4296
 4297
 4298
 4299
 4300
 4301
 4302
 4303
 4304
 4305
 4306
 4307
 4308
 4309
 4310
 4311
 4312
 4313
 4314
 4315
 4316
 4317
 4318
 4319
 4320
 4321
 4322
 4323
 4324
 4325
 4326
 4327
 4328
 4329

017714	012737	000112	001226
017722	012737	020036	001216
017730	013703	001412	
017734	012705	040000	
017740	010513		
017742	011304		
017744	005005		
017746	042704	000200	
017752	020504		
017754	001401		
017756	104003		
017760	012705	040000	
017764	010513		
017766	052777	000400	161416
017774	004737	005044	
020000	011304		
020002	032704	040000	
020006	001401		
020010	104003		
020012	010513		
020014	005013		
020016	011304		
020020	005005		
020022	042704	000200	
020026	020504		
020030	001401		
020032	104003		
020034	104400		

```

***** TEST 112 *****
*TRANSMITTER CONTROL REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
* TEST 112
*
*****
*****
TST112: MOV #112,@TSTNO
        MOV #TST113,NEXT
        MOV TXCSR,R3      ;GET THE TRANSMITTER CONTROL REGISTER
        MOV #BIT14,R5     ;GET BIT 14
        MOV R5,(R3)       ;WRITE BIT 14 TO TRANSMITTER CONTROL REGISTER
        MOV (R3),R4       ;READ TRANSMITTER CONTROL REGISTER BACK
        CLR R5            ;SET 'EXPECTED'
        BIC #BIT7,R4      ;CLEAR UNWANTED BITS
        CMP R5,R4         ;R5=GOOD,R4= ?
        BEQ 5$            ;BIT 14 IS OK
        HLT 3             ;BIT FAILED TO CLR
5$:      MOV #BIT14,R5     ;RELOAD THE BIT
        MOV R5,(R3)       ;WRITE BIT 14 TO THE REG
        BIS #MRESET,@TXCSR ;RESET THE DEVICE
        JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
        MOV (R3),R4       ;GET BIT 14
        BIT #BIT14,R4     ;TEST BIT 14 FOR RESET CLR
        BEQ 7$            ;BIT 14 IS OK
        HLT 3             ;BIT FAILED TO CLEAR
7$:      MOV R5,(R3)       ;SET BIT 14
        CLR (R3)          ;CLR TRANSMITTER CONTROL REGISTER
        MOV (R3),R4       ;READ THE TRANSMITTER CONTROL REGISTER
        CLR R5            ;SET 'EXPECTED'
        BIC #BIT7,R4      ;CLEAR UNWANTED BITS
        CMP R5,R4         ;R5=GOOD,R4= ?
        BEQ 10$           ;BIT 14 IS OK
        HLT 3             ;BIT FAILED TO CLEAR
10$:     SCOPE            ;SCOPE THIS TEST

```

```

***** TEST 113 *****
*TRANSMITTER CONTROL REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
* TEST 113
*
*****
*****

```

TRANSMITTER CONTROL REGISTER READ ONLY TEST BIT 15

```
4330 020036 012737 000113 001226 TST113: MOV #113,@#TSTNO
4331 020044 012737 020160 001216 MOV #TST114,NEXT
4332 020052 013703 001412 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
4333 020056 012705 100000 MOV #BIT15,R5 ;GET BIT 15
4334 020062 010513 MOV R5,(R3) ;WRITE BIT 15 TO TRANSMITTER CONTROL REGISTER
4335 020064 011304 MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER BACK
4336 020066 005005 CLR R5 ;SET 'EXPECTED'
4337 020070 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4338 020074 020504 CMP R5,R4 ;R5=GOOD,R4= ?
4339 020076 001401 BEQ 5$ ;BIT 15 IS OK
4340 020100 104003 HLT 3 ;BIT FAILED TO CLR
4341 020102 012705 100000 5$: MOV #BIT15,R5 ;RELOAD THE BIT
4342 020106 010513 MOV R5,(R3) ;WRITE BIT 15 TO THE REG
4343 020110 052777 000400 161274 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4344 020116 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4345 020122 011304 MOV (R3),R4 ;GET BIT 15
4346 020124 032704 100000 BIT #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
4347 020130 001401 BEQ 7$ ;BIT 15 IS OK
4348 020132 104003 HLT 3 ;BIT FAILED TO CLEAR
4349 020134 010513 7$: MOV R5,(R3) ;SET BIT 15
4350 020136 005013 CLR (R3) ;CLR TRANSMITTER CONTROL REGISTER
4351 020140 011304 MOV (R3),R4 ;READ THE TRANSMITTER CONTROL REGISTER
4352 020142 005005 CLR R5 ;SET 'EXPECTED'
4353 020144 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4354 020150 020504 CMP R5,R4 ;R5=GOOD,R4= ?
4355 020152 001401 BEQ 10$ ;BIT 15 IS OK
4356 020154 104003 HLT 3 ;BIT FAILED TO CLEAR
4357 020156 104400 10$: SCOPE ;SCOPE THIS TEST
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
```

```
***** TEST 114 *****
*TRANSMITTER BUFFER REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
::*****
:
: TEST 114
:
:*****
```

```
4372 020160 012737 000114 001226 TST114: MOV #114,@#TSTNO
4373 020166 012737 020272 001216 MOV #TST115,NEXT
4374 020174 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
4375 020200 012705 010000 MOV #BIT12,R5 ;GET BIT 12
4376 020204 010513 MOV R5,(R3) ;WRITE BIT 12 TO TRANSMITTER BUFFER REGISTER
4377 020206 011304 MOV (R3),R4 ;READ TRANSMITTER BUFFER REGISTER BACK
4378 020210 005005 CLR R5 ;SET 'EXPECTED'
4379 020212 020504 CMP R5,R4 ;R5=GOOD,R4= ?
4380 020214 001401 BEQ 5$ ;BIT 12 IS OK
4381 020216 104003 HLT 3 ;BIT FAILED TO CLR
4382 020220 012705 010000 5$: MOV #BIT12,R5 ;RELOAD THE BIT
4383 020224 010513 MOV R5,(R3) ;WRITE BIT 12 TO THE REG
4384 020226 052777 000400 161156 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4385 020234 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
```


TRANSMITTER BUFFER REGISTER READ ONLY TEST BIT 12

```
4386 020240 011304      MOV      (R3),R4      ;GET BIT 12
4387 020242 032704 010000 BIT      #BIT12,R4      ;TEST BIT 12 FOR RESET CLR
4388 020246 001401      BEQ      7$      ;BIT 12 IS OK
4389 020250 104003      HLT      3      ;BIT FAILED TO CLEAR
4390 020252 010513 7$:  MOV      R5,(R3)      ;SET BIT 12
4391 020254 005013      CLR      (R3)      ;CLR TRANSMITTER BUFFER REGISTER
4392 020256 011304      MOV      (R3),R4      ;READ THE TRANSMITTER BUFFER REGISTER
4393 020260 005005      CLR      R5      ;SET 'EXPECTED'
4394 020262 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4395 020264 001401      BEQ      10$      ;BIT 12 IS OK
4396 020266 104003      HLT      3      ;BIT FAILED TO CLEAR
4397 020270 104400 10$:  SCOPE      ;SCOPE THIS TEST
```

```
***** TEST 115 *****
*TRANSMITTER BUFFER REGISTER BIT 13 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 13 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****
```

```
*****
*
* TEST 115
*
```

```
*****
*****
4412 020272 012737 000115 001226 TST115: MOV      #115,@#TSTNO
4413 020300 012737 020404 001216      MOV      #TST116,NEXT
4414 020306 013703 001414      MOV      TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
4415 020312 012705 020000      MOV      #BIT13,R5      ;GET BIT 13
4416 020316 010513      MOV      R5,(R3)      ;WRITE BIT 13 TO TRANSMITTER BUFFER REGISTER
4417 020320 011304      MOV      (R3),R4      ;READ TRANSMITTER BUFFER REGISTER BACK
4418 020322 005005      CLR      R5      ;SET 'EXPECTED'
4419 020324 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4420 020326 001401      BEQ      5$      ;BIT 13 IS OK
4421 020330 104003      HLT      3      ;BIT FAILED TO CLR
4422 020332 012705 020000 5$:  MOV      #BIT13,R5      ;RELOAD THE BIT
4423 020336 010513      MOV      R5,(R3)      ;WRITE BIT 13 TO THE REG
4424 020340 052777 000400 161044      BIS      #MRESET,@TXCSR      ;RESET THE DEVICE
4425 020346 004737 005044      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
4426 020352 011304      MOV      (R3),R4      ;GET BIT 13
4427 020354 032704 020000      BIT      #BIT13,R4      ;TEST BIT 13 FOR RESET CLR
4428 020360 001401      BEQ      7$      ;BIT 13 IS OK
4429 020362 104003      HLT      3      ;BIT FAILED TO CLEAR
4430 020364 010513 7$:  MOV      R5,(R3)      ;SET BIT 13
4431 020366 005013      CLR      (R3)      ;CLR TRANSMITTER BUFFER REGISTER
4432 020370 011304      MOV      (R3),R4      ;READ THE TRANSMITTER BUFFER REGISTER
4433 020372 005005      CLR      R5      ;SET 'EXPECTED'
4434 020374 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4435 020376 001401      BEQ      10$      ;BIT 13 IS OK
4436 020400 104003      HLT      3      ;BIT FAILED TO CLEAR
4437 020402 104400 10$:  SCOPE      ;SCOPE THIS TEST
```

```
***** TEST 116 *****
*TRANSMITTER BUFFER REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST
```

TRANSMITTER BUFFER REGISTER READ ONLY TEST BIT 14

```
4442      ;*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK
4443      ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
4444      ;:*****
4445
4446      ;:*****
4447      ;*
4448      ;: TEST 116
4449      ;*
4450      ;:*****
4451      ;:*****
4452      020404 012737 000116 001226 TST116: MOV #116,@TSTNO
4453      020412 012737 020516 001216      MOV #TST117,NEXT
4454      020420 013703 001414      MOV TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
4455      020424 012705 040000      MOV #BIT14,R5      ;GET BIT 14
4456      020430 010513      MOV R5,(R3)      ;WRITE BIT 14 TO TRANSMITTER BUFFER REGISTER
4457      020432 011304      MOV (R3),R4      ;READ TRANSMITTER BUFFER REGISTER BACK
4458      020434 005005      CLR R5      ;SET 'EXPECTED'
4459      020436 020504      CMP R5,R4      ;R5=GOOD,R4= ?
4460      020440 001401      BEQ 5$      ;BIT 14 IS OK
4461      020442 104003      HLT 3      ;BIT FAILED TO CLR
4462      020444 012705 040000 5$: MOV #BIT14,R5      ;RELOAD THE BIT
4463      020450 010513      MOV R5,(R3)      ;WRITE BIT 14 TO THE REG
4464      020452 052777 000400 160732 BIS #MRESET,@TXCSR      ;RESET THE DEVICE
4465      020460 004737 005044      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
4466      020464 011304      MOV (R3),R4      ;GET BIT 14
4467      020466 032704 040000      BIT #BIT14,R4      ;TEST BIT 14 FOR RESET CLR
4468      020472 001401      BEQ 7$      ;BIT 14 IS OK
4469      020474 104003      HLT 3      ;BIT FAILED TO CLEAR
4470      020476 010513 7$: MOV R5,(R3)      ;SET BIT 14
4471      020500 005013      CLR (R3)      ;CLR TRANSMITTER BUFFER REGISTER
4472      020502 011304      MOV (R3),R4      ;READ THE TRANSMITTER BUFFER REGISTER
4473      020504 005005      CLR R5      ;SET 'EXPECTED'
4474      020506 020504      CMP R5,R4      ;R5=GOOD,R4= ?
4475      020510 001401      BEQ 10$      ;BIT 14 IS OK
4476      020512 104003      HLT 3      ;BIT FAILED TO CLEAR
4477      020514 104400 10$: SCOPE      ;SCOPE THIS TEST
4478
4479
4480      ;:***** TEST 117 *****
4481      ;*TRANSMITTER BUFFER REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
4482      ;*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
4483      ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
4484      ;:*****
4485
4486      ;:*****
4487      ;*
4488      ;: TEST 117
4489      ;*
4490      ;:*****
4491      ;:*****
4492      020516 012737 000117 001226 TST117: MOV #117,@TSTNO
4493      020524 012737 020630 001216      MOV #TST120,NEXT
4494      020532 013703 001414      MOV TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
4495      020536 012705 100000      MOV #BIT15,R5      ;GET BIT 15
4496      020542 010513      MOV R5,(R3)      ;WRITE BIT 15 TO TRANSMITTER BUFFER REGISTER
4497      020544 011304      MOV (R3),R4      ;READ TRANSMITTER BUFFER REGISTER BACK
```


TRANSMITTER BUFFER REGISTER READ ONLY TEST BIT 15

```
4498 020546 005005 CLR R5 ;SET 'EXPECTED'
4499 020550 020504 CMP R5,R4 ;R5=GOOD,R4=?
4500 020552 001401 BEQ 5$ ;BIT 15 IS OK
4501 020554 104003 HLT 3 ;BIT FAILED TO CLR
4502 020556 012705 100000 5$: MOV #BIT15,R5 ;RELOAD THE BIT
4503 020562 010513 MOV R5,(R3) ;WRITE BIT 15 TO THE REG
4504 020564 052777 000400 160620 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4505 020572 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4506 020576 011304 MOV (R3),R4 ;GET BIT 15
4507 020600 032704 100000 BIT #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
4508 020604 001401 BEQ 7$ ;BIT 15 IS OK
4509 020606 104003 HLT 3 ;BIT FAILED TO CLEAR
4510 020610 010513 7$: MOV R5,(R3) ;SET BIT 15
4511 020612 005013 CLR (R3) ;CLR TRANSMITTER BUFFER REGISTER
4512 020614 011304 MOV (R3),R4 ;READ THE TRANSMITTER BUFFER REGISTER
4513 020616 005005 CLR R5 ;SET 'EXPECTED'
4514 020620 020504 CMP R5,R4 ;R5=GOOD,R4=?
4515 020622 001401 BEQ 10$ ;BIT 15 IS OK
4516 020624 104003 HLT 3 ;BIT FAILED TO CLEAR
4517 020626 104400 10$: SCOPE ;SCOPE THIS TEST
```

```
***** TEST 120 *****
*PARAMETER STATUS REGISTER BIT 9 WRITE ONLY DEVICE RESET AND CLEAR TEST
*TEST THAT BIT 9 CANNOT BE WRITTEN AND READ
*BACK THE SAME. READ BIT 9,COMPLEMENT IT AND WRITE IT
*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 9,
*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
*REPEAT FOR A CLR INSTRUCTION.
*****
```

```
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535 020630 012737 000120 001226 TST120: MOV #120,@TSTNO
4536 020636 012737 020756 001216 MOV #TST121,NEXT
4537 020644 013703 001410 MOV PARCSR,R3 ;GET THE PARAMETER STATUS REGISTER
4538 020650 012705 001000 MOV #BIT9,R5 ;GET BIT 9
4539 020654 011305 MOV (R3),R5 ;READ THE REGISTER
4540 020656 010504 MOV R5,R4 ;SAVE THE BIT
4541 020660 032704 001000 BIT #BIT9,R4 ;CHECK BIT 9
4542 020664 001401 BEQ 1$ ;BIT 9 IS OK
4543 020666 104003 HLT 3 ;BIT FAILED TO CLEAR
4544 020670 005104 1$: COM R4 ;REVERSE THE BIT
4545 020672 010413 MOV R4,(R3) ;WRITE THE BIT TO THE REG
4546 020674 011304 MOV (R3),R4 ;READ IT BACK
4547 020676 020504 CMP R5,R4 ;R5=GOOD, R4=?
4548 020700 001401 BEQ 3$ ;BR IF OK
4549 020702 104003 HLT 3 ;COMPARE ERROR
4550 020704 012705 001000 3$: MOV #BIT9,R5 ;LOAD THE BIT
4551 020710 010513 MOV R5,(R3) ;WRITE THE BIT TO THE REG
4552 020712 052777 000400 160472 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4553 020720 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
```

```

4554 020724 011304      MOV      (R3),R4      ;GET BIT 9
4555 020726 032704 001000 BIT      #BIT9,R4      ;TEST BIT 9 FOR 0 AFTER RESET
4556 020732 001401      BEQ      4$      ;BIT 9 IS OK
4557 020734 104003      HLT      3      ;BIT IS NOT A 0
4558 020736 010513      4$: MOV      R5,(R3)      ;WRITE THE BIT TO THE REG
4559 020740 005013      CLR      (R3)      ;CLR THE PARAMETER STATUS REGISTER
4560 020742 011304      MOV      (R3),R4      ;GET THE PARAMETER STATUS REGISTER
4561 020744 005005      CLR      R5      ;SET 'EXPECTED'
4562 020746 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4563 020750 001401      BEQ      10$      ;BIT 9 IS OK
4564 020752 104003      HLT      3      ;BIT FAILED TO CLEAR
4565 020754 104400      10$: SCOPE      ;SCOPE THIS TEST

```

```

;***** TEST 121 *****
;*PARAMETER STATUS REGISTER BIT 12 WRITE ONLY DEVICE RESET AND CLEAR TEST
;*TEST THAT BIT 12 CANNOT BE WRITTEN AND READ
;*BACK THE SAME. READ BIT 12,COMPLEMENT IT AND WRITE IT
;*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 12,
;*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
;*REPEAT FOR A CLR INSTRUCTION.
;*****

```

```

;*****
; TEST 121
;*****

```

```

4583 020756 012737 000121 001226 TST121: MOV      #121,@TSTNO
4584 020764 012737 021104 001216 MOV      #TST122,NEXT
4585 020772 013703 001410 MOV      PARCSR,R3      ;GET THE PARAMETER STATUS REGISTER
4586 020776 012705 010000 MOV      #BIT12,R5      ;GET BIT 12
4587 021002 011305 MOV      (R3),R5      ;READ THE REGISTER
4588 021004 010504 MOV      R5,R4      ;SAVE THE BIT
4589 021006 032704 010000 BIT      #BIT12,R4      ;CHECK BIT 12
4590 021012 001401 BEQ      1$      ;BIT 12 IS OK
4591 021014 104003 HLT      3      ;BIT FAILED TO CLEAR
4592 021016 005104 1$: COM      R4      ;REVERSE THE BIT
4593 021020 010413 MOV      R4,(R3)      ;WRITE THE BIT TO THE REG
4594 021022 011304 MOV      (R3),R4      ;READ IT BACK
4595 021024 020504 CMP      R5,R4      ;R5=GOOD, R4= ?
4596 021026 001401 BEQ      3$      ;BR IF OK
4597 021030 104003 HLT      3      ;COMPARE ERROR
4598 021032 012705 010000 3$: MOV      #BIT12,R5      ;LOAD THE BIT
4599 021036 010513 MOV      R5,(R3)      ;WRITE THE BIT TO THE REG
4600 021040 052777 000400 160344 BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4601 021046 004737 005044 JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
4602 021052 011304 MOV      (R3),R4      ;GET BIT 12
4603 021054 032704 010000 BIT      #BIT12,R4      ;TEST BIT 12 FOR 0 AFTER RESET
4604 021060 001401 BEQ      4$      ;BIT 12 IS OK
4605 021062 104003 HLT      3      ;BIT IS NOT A 0
4606 021064 010513 4$: MOV      R5,(R3)      ;WRITE THE BIT TO THE REG
4607 021066 005013 CLR      (R3)      ;CLR THE PARAMETER STATUS REGISTER
4608 021070 011304 MOV      (R3),R4      ;GET THE PARAMETER STATUS REGISTER
4609 021072 005005 CLR      R5      ;SET 'EXPECTED'

```


PARAMETER STATUS REGISTER WRITE ONLY TEST BIT 12

```
4610 021074 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4611 021076 001401      BEQ      10$      ;BIT 12 IS OK
4612 021100 104003      HLT      3      ;BIT FAILED TO CLEAR
4613 021102 104400      10$: SCOPE      ;SCOPE THIS TEST
4614
4615
4616
```

```
***** TEST 122 *****
;PARAMETER STATUS REGISTER BIT 15 WRITE ONLY DEVICE RESET AND CLEAR TEST
;TEST THAT BIT 15 CANNOT BE WRITTEN AND READ
;BACK THE SAME. READ BIT 15, COMPLEMENT IT AND WRITE IT
;VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 15,
;DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
;REPEAT FOR A CLR INSTRUCTION.
*****
```

```
*****
;TEST 122
*****
```

```
4623
4624
4625
4626
4627
4628
4629
4630
4631 021104 012737 000122 001226 TST122: MOV      #122,@TSTNO
4632 021112 012737 021232 001216      MOV      #TST123,NEXT
4633 021120 013703 001410      MOV      PARCSR,R3      ;GET THE PARAMETER STATUS REGISTER
4634 021124 012705 100000      MOV      #BIT15,R5      ;GET BIT 15
4635 021130 011305      MOV      (R3),R5      ;READ THE REGISTER
4636 021132 010504      MOV      R5,R4      ;SAVE THE BIT
4637 021134 032704 100000      BIT      #BIT15,R4      ;CHECK BIT 15
4638 021140 001401      BEQ      1$      ;BIT 15 IS OK
4639 021142 104003      HLT      3      ;BIT FAILED TO CLEAR
4640 021144 005104      1$: COM      R4      ;REVERSE THE BIT
4641 021146 010413      MOV      R4,(R3)      ;WRITE THE BIT TO THE REG
4642 021150 011304      MOV      (R3),R4      ;READ IT BACK
4643 021152 020504      CMP      R5,R4      ;R5=GOOD, R4= ?
4644 021154 001401      BEQ      3$      ;BR IF OK
4645 021156 104003      HLT      3      ;COMPARE ERROR
4646 021160 012705 100000      3$: MOV      #BIT15,R5      ;LOAD THE BIT
4647 021164 010513      MOV      R5,(R3)      ;WRITE THE BIT TO THE REG
4648 021166 052777 000400 160216      BIS      #MRESET,@TXCSR      ;RESET THE DEVICE
4649 021174 004737 005044      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
4650 021200 011304      MOV      (R3),R4      ;GET BIT 15
4651 021202 032704 100000      BIT      #BIT15,R4      ;TEST BIT 15 FOR 0 AFTER RESET
4652 021206 001401      BEQ      4$      ;BIT 15 IS OK
4653 021210 104003      HLT      3      ;BIT IS NOT A 0
4654 021212 010513      4$: MOV      R5,(R3)      ;WRITE THE BIT TO THE REG
4655 021214 005013      CLR      (R3)      ;CLR THE PARAMETER STATUS REGISTER
4656 021216 011304      MOV      (R3),R4      ;GET THE PARAMETER STATUS REGISTER
4657 021220 005005      CLR      R5      ;SET 'EXPECTED'
4658 021222 020504      CMP      R5,R4      ;R5=GOOD,R4= ?
4659 021224 001401      BEQ      10$      ;BIT 15 IS OK
4660 021226 104003      HLT      3      ;BIT FAILED TO CLEAR
4661 021230 104400      10$: SCOPE      ;SCOPE THIS TEST
4662
4663
4664
4665
```

```
***** TEST 123 *****
;TRANSMITTER CONTROL REGISTER BIT 8 WRITE ONLY DEVICE RESET AND CLEAR TEST
```

4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721

```

; *TEST THAT BIT 8 CANNOT BE WRITTEN AND READ
; *BACK THE SAME.  READ BIT 8, COMPLEMENT IT AND WRITE IT
; *VERIFYING THAT IT DID NOT CHANGE.  WRITE BIT 8,
; *DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
; *REPEAT FOR A CLR INSTRUCTION.
; *****

```

```

: *****
:
: TEST 123
:
: *****

```

```

::*****
TST123: MOV      #123,@#TSTNO
        MOV      #TST124,NEXT
        MOV      TXCSR,R3          ;GET THE TRANSMITTER CONTROL REGISTER
        MOV      #BIT7,R5         ;SET EXPECTED BIT
        MOV      (R3),R4           ;READ THE REGISTER
        BIT      #BIT8,R4         ;CHECK BIT 8
        BEQ      1$               ;BIT 8 IS OK
        HLT      3                ;BIT FAILED TO CLEAR
1$:     COM       R4               ;REVERSE THE BIT
        MOV      R4,(R3)          ;WRITE THE BIT TO THE REG
        JSR      PC,SMALL         ;WAIT FOR DEVICE RESET TO FINISH
        MOV      (R3),R4          ;READ IT BACK
        CMP      R5,R4            ;R5=GOOD, R4= ?
        BEQ      3$               ;BR IF OK
        HLT      3                ;COMPARE ERROR
3$:     MOV      #BIT8,R4         ;LOAD THE BIT
        MOV      R4,(R3)          ;WRITE THE BIT TO THE REG
        BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
        JSR      PC,SMALL         ;WAIT FOR RESET TO FINISH
        MOV      (R3),R4          ;GET BIT 8
        BIT      #BIT8,R4         ;TEST BIT 8 FOR 0 AFTER RESET
        BEQ      4$               ;BIT 8 IS OK
        HLT      3                ;BIT IS NOT A 0
4$:     MOV      #BIT8,R4         ;LOAD THE BIT
        MOV      R4,(R3)          ;WRITE THE BIT TO THE REG
        CLR      (R3)             ;CLR THE TRANSMITTER CONTROL REGISTER
        MOV      (R3),R4          ;GET THE TRANSMITTER CONTROL REGISTER
        CMP      R5,R4            ;R5=GOOD,R4= ?
        BEQ      10$              ;BIT 8 IS OK
        HLT      3                ;BIT FAILED TO CLEAR
10$:    SCOPE                     ;SCOPE THIS TEST

```

```

***** TEST 124 *****
;*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
;*SET BIT8, VERIFY BIT8 WAS SET.
;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
;*****

```

TEST 124


```

4722
4723
4724 021364 012737 000124 001226
4725 021372 012737 021440 001216
4726 021400 013703 001404
4727 021404 012705 000400
4728 021410 010513
4729 021412 011304
4730 021414 020504
4731 021416 001401
4732 021420 104003
4733 021422 040513
4734 021424 011304
4735 021426 005005
4736 021430 020504
4737 021432 001401
4738 021434 104003
4739 021436 104400
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754 021440 012737 000125 001226
4755 021446 012737 021514 001216
4756 021454 013703 001404
4757 021460 012705 000100
4758 021464 010513
4759 021466 011304
4760 021470 020504
4761 021472 001401
4762 021474 104003
4763 021476 040513
4764 021500 011304
4765 021502 005005
4766 021504 020504
4767 021506 001401
4768 021510 104003
4769 021512 104400
4770
4771
4772
4773
4774
4775
4776
4777

;*****
;*****
TST124: MOV #124,@TSTNO
        MOV #TST125,NEXT
        MOV RXCSR,R3      ;SET REGISTER TO BE TESTED.
        MOV #BIT8,R5      ;SET 'EXPECTED'
        MOV R5,(R3)        ;WRITE THE REGISTER.
        MOV (R3),R4        ;READ THE REGISTER.
        CMP R5,R4          ;R5=GOOD; R4=UNKNOWN.
        BEQ 1$             ;ARE THEY THE SAME?
        HLT 3              ;COMPARISON ERROR.
1$:      BIC R5,(R3)        ;CLEAR BIT8
        MOV (R3),R4        ;READ THE REGISTER.
        CLR R5             ;SET 'EXPECTED'
        CMP R5,R4          ;R5=GOOD; R4=?
        BEQ 2$             ;BR IF OK
        HLT 3              ;COMPARISON ERROR
2$:      SCOPE             ;SCOPE THIS TEST

;***** TEST 125 *****
;*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
;*SET BIT6, VERIFY BIT6 WAS SET.
;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
;*****

;*****
;*****
; TEST 125
;*****
;*****
TST125: MOV #125,@TSTNO
        MOV #TST126,NEXT
        MOV RXCSR,R3      ;SET REGISTER TO BE TESTED.
        MOV #BIT6,R5      ;SET 'EXPECTED'
        MOV R5,(R3)        ;WRITE THE REGISTER.
        MOV (R3),R4        ;READ THE REGISTER.
        CMP R5,R4          ;R5=GOOD; R4=UNKNOWN.
        BEQ 1$             ;ARE THEY THE SAME?
        HLT 3              ;COMPARISON ERROR.
1$:      BIC R5,(R3)        ;CLEAR BIT6
        MOV (R3),R4        ;READ THE REGISTER.
        CLR R5             ;SET 'EXPECTED'
        CMP R5,R4          ;R5=GOOD; R4=?
        BEQ 2$             ;BR IF OK
        HLT 3              ;COMPARISON ERROR
2$:      SCOPE             ;SCOPE THIS TEST

;***** TEST 126 *****
;*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
;*SET BIT5, VERIFY BIT5 WAS SET.
;*CLEAR BIT5, VERIFY BIT5 WAS CLEARED.
;*****

```

RECEIVER CONTROL REGISTER READ/WRITE TEST BIT 5

```
4778 ;:*****
4779 ;:
4780 ;: TEST 126
4781 ;:
4782 ;:*****
4783 ;:*****
4784 021514 012737 000126 001226 TST126: MOV #126,@TSTNO
4785 021522 012737 021570 001216 MOV #TST127,NEXT
4786 021530 013703 001404 MOV RXCSR,R3 ;SET REGISTER TO BE TESTED.
4787 021534 012705 000040 MOV #BIT5,R5 ;SET 'EXPECTED'
4788 021540 010513 MOV R5,(R3) ;WRITE THE REGISTER.
4789 021542 011304 MOV (R3),R4 ;READ THE REGISTER.
4790 021544 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
4791 021546 001401 BEQ 1$ ;ARE THEY THE SAME?
4792 021550 104003 HLT 3 ;COMPARISON ERROR.
4793 021552 040513 1$: BIC R5,(R3) ;CLEAR BIT5
4794 021554 011304 MOV (R3),R4 ;READ THE REGISTER.
4795 021556 005005 CLR R5 ;SET 'EXPECTED'
4796 021560 020504 CMP R5,R4 ;R5=GOOD; R4=?
4797 021562 001401 BEQ 2$ ;BR IF OK
4798 021564 104003 HLT 3 ;COMPARISON ERROR
4799 021566 104400 2$: SCOPE ;SCOPE THIS TEST
4800
4801
4802 ;:***** TEST 127 *****
4803 ;:RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
4804 ;:SET BIT4, VERIFY BIT4 WAS SET.
4805 ;:CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
4806 ;:*****
4807
4808 ;:*****
4809 ;:
4810 ;: TEST 127
4811 ;:
4812 ;:*****
4813 ;:*****
4814 021570 012737 000127 001226 TST127: MOV #127,@TSTNO
4815 021576 012737 021644 001216 MOV #TST130,NEXT
4816 021604 013703 001404 MOV RXCSR,R3 ;SET REGISTER TO BE TESTED.
4817 021610 012705 000020 MOV #BIT4,R5 ;SET 'EXPECTED'
4818 021614 010513 MOV R5,(R3) ;WRITE THE REGISTER.
4819 021616 011304 MOV (R3),R4 ;READ THE REGISTER.
4820 021620 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
4821 021622 001401 BEQ 1$ ;ARE THEY THE SAME?
4822 021624 104003 HLT 3 ;COMPARISON ERROR.
4823 021626 040513 1$: BIC R5,(R3) ;CLEAR BIT4
4824 021630 011304 MOV (R3),R4 ;READ THE REGISTER.
4825 021632 005005 CLR R5 ;SET 'EXPECTED'
4826 021634 020504 CMP R5,R4 ;R5=GOOD; R4=?
4827 021636 001401 BEQ 2$ ;BR IF OK
4828 021640 104003 HLT 3 ;COMPARISON ERROR
4829 021642 104400 2$: SCOPE ;SCOPE THIS TEST
4830
4831
4832 ;:***** TEST 130 *****
4833 ;:TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
```


TRANSMITTER CONTROL REGIASTER READ/WRITE TEST BIT 6

```
4834 ;*SET BIT6, VERIFY BIT6 WAS SET.
4835 ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
4836 ;:*****
4837
4838 ;:*****
4839 ;:
4840 ;: TEST 130
4841 ;:
4842 ;:*****
4843 ;:*****
4844 021644 012737 000130 001226 TST130: MOV #130,@TSTNO
4845 021652 012737 021730 001216 MOV #TST131,NEXT
4846 021660 013703 001412 MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
4847 021664 012705 000100 MOV #BIT6,R5 ;SET 'EXPECTED'
4848 021670 010513 MOV R5,(R3) ;WRITE THE REGISTER.
4849 021672 011304 MOV (R3),R4 ;READ THE REGISTER.
4850 021674 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4851 021700 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
4852 021702 001401 BEQ 1$ ;ARE THEY THE SAME?
4853 021704 104003 HLT 3 ;COMPARISON ERROR.
4854 021706 040513 1$: BIC R5,(R3) ;CLEAR BIT6
4855 021710 011304 MOV (R3),R4 ;READ THE REGISTER.
4856 021712 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4857 021716 005005 CLR R5 ;SET 'EXPECTED'
4858 021720 020504 CMP R5,R4 ;R5=GOOD; R4=?
4859 021722 001401 BEQ 2$ ;BR IF OK
4860 021724 104003 HLT 3 ;COMPARISON ERROR
4861 021726 104400 2$: SCOPE ;SCOPE THIS TEST
4862
4863
```

```
4864 ;***** TEST 131 *****
4865 ;*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
4866 ;*SET BIT1, VERIFY BIT1 WAS SET.
4867 ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
4868 ;:*****
4869
```

```
4870 ;:*****
4871 ;:
4872 ;: TEST 131
4873 ;:
4874 ;:*****
4875 ;:*****
4876 021730 012737 000131 001226 TST131: MOV #131,@TSTNO
4877 021736 012737 022014 001216 MOV #TST132,NEXT
4878 021744 013703 001404 MOV RXCSR,R3 ;SET REGISTER TO BE TESTED.
4879 021750 012705 000002 MOV #BIT1,R5 ;SET 'EXPECTED'
4880 021754 010513 MOV R5,(R3) ;WRITE THE REGISTER.
4881 021756 011304 MOV (R3),R4 ;READ THE REGISTER.
4882 021760 042704 1770C1 BIC #177001,R4 ;CLEAR UNWANTED BITS
4883 021764 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
4884 021766 001401 BEQ 1$ ;ARE THEY THE SAME?
4885 021770 104003 HLT 3 ;COMPARISON ERROR.
4886 021772 040513 1$: BIC R5,(R3) ;CLEAR BIT1
4887 021774 011304 MOV (R3),R4 ;READ THE REGISTER.
4888 021776 042704 177001 BIC #177001,R4 ;CLEAR UNWANTED BITS
4889 022002 005005 CLR R5 ;SET 'EXPECTED'
```

4890 022004 020504 CMP R5,R4 ;R5=GOOD; R4=?
 4891 022006 001401 BEQ 2\$;BR IF OK
 4892 022010 104003 HLT 3 ;COMPARISON ERROR
 4893 022012 104400 2\$: SCOPE ;SCOPE THIS TEST

***** TEST 132 *****
 *RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
 *SET BIT2, VERIFY BIT2 WAS SET.
 *CLEAR BIT2, VERIFY BIT2 WAS CLEARED.

 *
 : TEST 132
 *

4907 *****
 4908 022014 012737 000132 001226 TST132: MOV #132,@TSTNO
 4909 022022 012737 022100 001216 MOV #TST133,NEXT
 4910 022030 013703 001404 MOV RXCSR,R3 ;SET REGISTER TO BE TESTED.
 4911 022034 012705 000004 MOV #BIT2,R5 ;SET 'EXPECTED'
 4912 022040 010513 MOV R5,(R3) ;WRITE THE REGISTER.
 4913 022042 011304 MOV (R3),R4 ;READ THE REGISTER.
 4914 022044 042704 177001 BIC #177001,R4 ;CLEAR UNWANTED BITS
 4915 022050 020504 CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
 4916 022052 001401 BEQ 1\$;ARE THEY THE SAME?
 4917 022054 104003 HLT 3 ;COMPARISON ERROR.
 4918 022056 040513 1\$: BIC R5,(R3) ;CLEAR BIT2
 4919 022060 011304 MOV (R3),R4 ;READ THE REGISTER.
 4920 022062 042704 177001 BIC #177001,R4 ;CLEAR UNWANTED BITS
 4921 022066 005005 CLR R5 ;SET 'EXPECTED'
 4922 022070 020504 CMP R5,R4 ;R5=GOOD; R4=?
 4923 022072 001401 BEQ 2\$;BR IF OK
 4924 022074 104003 HLT 3 ;COMPARISON ERROR
 4925 022076 104400 2\$: SCOPE ;SCOPE THIS TEST

***** TEST 133 *****
 *RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
 *SET BIT3, VERIFY BIT3 WAS SET.
 *CLEAR BIT3, VERIFY BIT3 WAS CLEARED.

 *
 : TEST 133
 *

4939 *****
 4940 022100 012737 000133 001226 TST133: MOV #133,@TSTNO
 4941 022106 012737 022164 001216 MOV #TST134,NEXT
 4942 022114 013703 001404 MOV RXCSR,R3 ;SET REGISTER TO BE TESTED.
 4943 022120 012705 000010 MOV #BIT3,R5 ;SET 'EXPECTED'
 4944 022124 010513 MOV R5,(R3) ;WRITE THE REGISTER.
 4945 022126 011304 MOV (R3),R4 ;READ THE REGISTER.

RECEIVER CONTROL REGISTER READ/WRITE TEST BIT 3

```
4946 022130 042704 177001      BIC    #177001,R4      ;CLEAR UNWANTED BITS
4947 022134 020504              CMP    R5,R4              ;R5=GOOD; R4=UNKNOWN.
4948 022136 001401              BEQ    1$                  ;ARE THEY THE SAME?
4949 022140 104003              HLT    3                    ;COMPARISON ERROR.
4950 022142 040513      1$:    BIC    R5,(R3)              ;CLEAR BIT3
4951 022144 011304              MOV    (R3),R4              ;READ THE REGISTER.
4952 022146 042704 177001      BIC    #177001,R4          ;CLEAR UNWANTED BITS
4953 022152 005005              CLR    R5                  ;SET 'EXPECTED'
4954 022154 020504              CMP    R5,R4              ;R5=GOOD; R4=?
4955 022156 001401              BEQ    2$                  ;BR IF OK
4956 022160 104003              HLT    3                    ;COMPARISON ERROR
4957 022162 104400      2$:    SCOPE                        ;SCOPE THIS TEST
4958
4959
```

```
***** TEST 134 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
*WRITE BIT 6,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

:*****

:*

:TEST 134

:*

:*****

:*****

```
4972 022164 012737 000134 001226 TST134: MOV    #134,@TSTNO
4973 022172 012737 022260 001216      MOV    #TST135,NEXT
4974 022200 013703 001404              MOV    RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
4975 022204 052713 000100              BIS    #BIT6,(R3)    ;SET BIT 6 AT RECEIVER CONTROL REGISTER
4976 022210 005005              CLR    R5                  ;SET 'EXPECTED'
4977 022212 052777 000400 157172      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
4978 022220 004737 005044              JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
4979 022224 032713 000100              BIT    #BIT6,(R3)    ;TEST BIT 6
4980 022230 001402              BEQ    1$                  ;BIT 6 IS CLEARED
4981 022232 011304              MOV    (R3),R4              ;LOAD 'FOUND'
4982 022234 104003              HLT    3                    ;BIT 6 IS SET AND SHOULDN'T BE
4983 022236 052713 000100      1$:    BIS    #BIT6,(R3)    ;SET BIT 6 AGAIN
4984 022242 005013              CLR    (R3)              ;CLEAR THE RECEIVER CONTROL REGISTER
4985 022244 032713 000100              BIT    #BIT6,(R3)    ;TEST TO SEE IF BIT 6 CLEARED
4986 022250 001402              BEQ    2$                  ;BIT 6 IS OK
4987 022252 011304              MOV    (R3),R4              ;LOAD 'FOUND'
4988 022254 104003              HLT    3                    ;BIT 6 FAILED TO CLEAR
4989 022256 104400      2$:    SCOPE                        ;SCOPE THIS TEST
4990
4991
```

```
***** TEST 135 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 5 RESET AND CLEAR TEST
*WRITE BIT 5,AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****
```

:*****

:*

:TEST 135

:*

5000

5001

RECEIVER CONTROL REGISTER RESET AND CLEAR TEST BIT 5

```
5002      ::*****
5003      ::*****
5004 022260 012737 000135 001226 TST135: MOV #135,@TSTNO
5005 022266 012737 022354 001216      MOV #TST136,NEXT
5006 022274 013703 001404      MOV RXCSR,R3      ;GET THE RECEIVER CONTROL REGISTER
5007 022300 052713 000040      BIS #BIT5,(R3)      ;SET BIT 5 AT RECEIVER CONTROL REGISTER
5008 022304 005005      CLR R5      ;SET 'EXPECTED'
5009 022306 052777 000400 157076      BIS #MRESET,@TXCSR      ;RESET THE DEVICE
5010 022314 004737 005044      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
5011 022320 032713 000040      BIT #BIT5,(R3)      ;TEST BIT 5
5012 022324 001402      BEQ 1$      ;BIT 5 IS CLEARED
5013 022326 011304      MOV (R3),R4      ;LOAD 'FOUND'
5014 022330 104003      HLT 3      ;BIT 5 IS SET AND SHOULDN'T BE
5015 022332 052713 000040 1$: BIS #BIT5,(R3)      ;SET BIT 5 AGAIN
5016 022336 005013      CLR (R3)      ;CLEAR THE RECEIVER CONTROL REGISTER
5017 022340 032713 000040      BIT #BIT5,(R3)      ;TEST TO SEE IF BIT 5 CLEARED
5018 022344 001402      BEQ 2$      ;BIT 5 IS OK
5019 022346 011304      MOV (R3),R4      ;LOAD 'FOUND'
5020 022350 104003      HLT 3      ;BIT 5 FAILED TO CLEAR
5021 022352 104400 2$: SCOPE      ;SCOPE THIS TEST
5022
5023
5024      ::***** TEST 136 *****
5025      ;*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
5026      ;*WRITE BIT 6,AND TEST THAT IT WILL BE CLEARED AFTER A
5027      ;*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
5028      ;******
5029
5030      ::*****
5031      ;
5032      ; TEST 136
5033      ;
5034      ::*****
5035      ::*****
5036 022354 012737 000136 001226 TST136: MOV #136,@TSTNO
5037 022362 012737 022450 001216      MOV #TST137,NEXT
5038 022370 013703 001412      MOV TXCSR,R3      ;GET THE TRANSMITTER CONTROL REGISTER
5039 022374 052713 000100      BIS #BIT6,(R3)      ;SET BIT 6 AT TRANSMITTER CONTROL REGISTER
5040 022400 005005      CLR R5      ;SET 'EXPECTED'
5041 022402 052777 000400 157002      BIS #MRESET,@TXCSR      ;RESET THE DEVICE
5042 022410 004737 005044      JSR PC,SMALL      ;WAIT FOR RESET TO FINISH
5043 022414 032713 000100      BIT #BIT6,(R3)      ;TEST BIT 6
5044 022420 001402      BEQ 1$      ;BIT 6 IS CLEARED
5045 022422 011304      MOV (R3),R4      ;LOAD 'FOUND'
5046 022424 104003      HLT 3      ;BIT 6 IS SET AND SHOULDN'T BE
5047 022426 052713 000100 1$: BIS #BIT6,(R3)      ;SET BIT 6 AGAIN
5048 022432 005013      CLR (R3)      ;CLEAR THE TRANSMITTER CONTROL REGISTER
5049 022434 032713 000100      BIT #BIT6,(R3)      ;TEST TO SEE IF BIT 6 CLEARED
5050 022440 001402      BEQ 2$      ;BIT 6 IS OK
5051 022442 011304      MOV (R3),R4      ;LOAD 'FOUND'
5052 022444 104003      HLT 3      ;BIT 6 FAILED TO CLEAR
5053 022446 104400 2$: SCOPE      ;SCOPE THIS TEST
5054
5055
5056      ::***** TEST 137 *****
5057      ;*THIS TEST CHECKS THE MAINTENANCE CLOCK
```



```

5058                                     ;*USED THROUGHOUT THE REMAINING DIAGNOSTICS
5059                                     ;:*****
5060
5061                                     ;:*****
5062                                     ;:
5063                                     ;: TEST 137
5064                                     ;:
5065                                     ;:*****
5066                                     ;:*****
5067 022450 012737 000137 001226 TST137: MOV #137,@TSTNO
5068 022456 012737 022604 001216      MOV #TST140,NEXT
5069 022464 052777 000400 156720      BIS #MRESET,@TXCSR ;RESET THE DEVICE
5070 022472 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
5071 022476 005037 001236          CLR TEMP1 ;TEST TIM SETUP
5072 022502 052777 004000 156702      BIS #SYSTST,@TXCSR ;ENTER SYSTEM TST MODE TO TURN ON CLOCK
5073 022510 032777 004000 156676 1$: BIT #TIMER,@TXDBUF ;CHECK THE CLOCK BIT
5074 022516 001407          BEQ 2$ ;BR IF OFF
5075 022520 005237 001236          INC TEMP1 ;INC WAIT LOOP
5076 022524 022737 177777 001236      CMP #-1,TEMP1 ;CHECK FOR LOOP TO BE DONE
5077 022532 001366          BNE 1$ ;BR IF MORE TIME TO WAIT
5078 022534 104000          HLT ;TIMER CLOCK BIT FAILED TO CLEAR
5079 022536 005037 001236 2$: CLR TEMP1 ;SECOND HALF SETUP
5080 022542 032777 004000 156644 3$: BIT #TIMER,@TXDBUF ;CHECK THE CLOCK BIT
5081 022550 001007          BNE 4$ ;BR IF ON
5082 022552 005237 001236          INC TEMP1 ;INC THE WAIT LOOP
5083 022556 022737 177777 001236      CMP #-1,TEMP1 ;CHECK FOR LOPP DONE
5084 022564 001366          BNE 3$ ;BR IF MORE TIME TO WAIT
5085 022566 104000          HLT ;TIMER BIT FAILED TO SET
5086 022570          4$:
5087 022570 052777 000400 156614      BIS #MRESET,@TXCSR ;RESET THE DEVICE
5088 022576 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
5089 022602 104400          SCOPE ;SCOPE THIS TEST
5090
5091
5092
5093                                     ;***** TEST 140 *****
5094                                     ;*THIS TEST WILL PERFORM STATIC TRANSMITTER FUNCTIONS
5095                                     ;*IN MAINTENANCE MODE. IT WILL PROVE THE INTERACTION
5096                                     ;*OF SEND, DONE AND TSOM.
5097                                     ;:*****
5098
5099                                     ;:*****
5100                                     ;:
5101                                     ;: TEST 140
5102                                     ;:
5103                                     ;:*****
5104                                     ;:*****
5105 022604 012737 000140 001226 TST140: MOV #140,@TSTNO
5106 022612 012737 022760 001216      MOV #TST141,NEXT
5107 022620 013.02 001414          MOV TXDBUF,R2 ;LOAD TX BUFFER
5108 022624 013703 001412          MOV TXCSR,R3 ;LOAD TX CONTROL REGISTER
5109 022630 005012          CLR (R2) ;CLEAR BUFFER
5110 022632 052777 000400 156552      BIS #MRESET,@TXCSR ;RESET THE DEVICE
5111 022640 004737 005044          JSR PC,SMALL ;WAIT FOR RESET TO FINISH
5112 022644 052777 001000 156536      BIS #CRCEN,@PARCSR ;TURN OFF CRC
5113 022652 052713 014000          BIS #MMODE,(R3) ;ENTER M/M - PROGRAM NOW CLOCKING

```

5114	022656	052713	000020		BIS	#SEND,(R3)	:ASSERT SEND
5115	022662	052705	014220		BIS	#SEND!TXDONE!MMODE,R5	:SET 'EXPECTED'
5116	022666	011304			MOV	(R3),R4	:READ TX CONTROL REGISTER
5117	022670	020504			CMP	R5,R4	:ARE THEY EQUAL?
5118	022672	001401			BEQ	1\$	:BR IF YES
5119	022674	104003			HLT	3	:NO
5120	022676	005005		1\$:	CLR	R5	:SET EXPECTED'
5121	022700	011204			MOV	(R2),R4	:READ TX BUFFER
5122	022702	042704	170000		BIC	#170000,R4	:MASK
5123	022706	020504			CMP	R5,R4	:R5=GOOD, R4=?
5124	022710	001405			BEQ	2\$	:ARE THEY EQUAL?
5125	022712	013703	001414		MOV	TXDBUF,R3	:ERROR MESSAGE SETUP
5126	022716	104003			HLT	3	:NO
5127	022720	013703	001412		MOV	TXCSR,R3	:RETURN TO NORMAL
5128	022724	010512		2\$:	MOV	R5,(R2)	:LOAD BUFFER
5129	022726	032713	000200		BIT	#TXDONE,(R3)	:TEST DONE
5130	022732	001401			BEQ	3\$	
5131	022734	104005			HLT	5	:BIT FAILED TO CLR
5132	022736	052712	000400	3\$:	BIS	#TSOM,(R2)	:SET TSOM
5133	022742	104412	000017		PKCLK	,15.	
5134	022746	032713	000200		BIT	#TXDONE,(R3)	:TEST DONE
5135	022752	001001			BNE	4\$	
5136	022754	104006			HLT	6	:BIT FAILED TO SET
5137	022756	104400		4\$:	SCOPE		:SCOPE THIS TEST

***** TEST 141 *****
 :*THIS TEST CHECKS THE STATIC FUNCTIONS OF THE TRANSMITTER
 :*IN MAINTENANCE MODE. THIS TEST PROVES THE INTERACTION OF
 :*TXDBUF, TXACT, TSOM,TRANSMITTED DATA AND DONE.
 :*****

 : TEST 141
 :*****

5151							
5152	022760	012737	000141	001226	TST141:	MOV	#141,@TSTNO
5153	022766	012737	023274	001216		MOV	#TST142,NEXT
5154	022774	013703	001412			MOV	TXCSR,R3
5155	023000	013702	001414			MOV	TXDBUF,R2
5156	023004	052777	000400	156400		BIS	#MRESET,@TXCSR
5157	023012	004737	005044			JSR	PC,SMALL
5158	023016	005012				CLR	(R2)
5159	023020	052713	014000			BIS	#MMODE,(R3)
5160	023024	052777	001000	156356		BIS	#CRCEN,@PARCSR
5161	023032	052713	000020			BIS	#SEND,(R3)
5162	023036	005005				CLR	R5
5163	023040	010512				MOV	R5,(R2)
5164	023042	012712	000400			MOV	#TSOM,(R2)
5165	023046	104412	000006			PKCLK	,6.
5166	023052	052777	020000	156332		BIS	#CLK,@TXCSR
5167	023060	032713	001000			BIT	#TXACT,(R3)
5168	023064	001001				BNE	1\$
5169	023066	104007				HLT	7

:LOAD CONTROL REGISTER
 :LOAD BUFFER
 :RESET THE DEVICE
 :WAIT FOR RESET TO FINISH
 :RESET TXDONE
 :ENTER M/M--PROGRAM CLOCKING
 :TURN OFF CRC
 :SET SEND
 :SET 'EXPECTED'.
 :LOAD TX BUFFER
 :TURN ON TSOM
 :SYNC UP DUP
 :POKE CLOCK UP
 :IS TXACT HIGH?
 :BR IF SET
 :TXACT FAILED TO SET

5170	023070				1\$:			
5171	023070	042777	020000	156314		BIC	#CLK,@TXCSR	:POKE CLOCK DOWN
5172	023076	104412	000015			PKCLK	,13.	:PUSH OUT DATA
5173	023102	032713	000020			BIT	#SEND,(R3)	:CHECK SEND
5174	023106	001001				BNE	2\$	:BR IF YES
5175	023110	104017				HLT	17	:BIT FAILED TO SET
5176	023112	012737	023112	001220	2\$:	MOV	#2\$,LOCK	:SETUPFOR SW 09
5177	023120	012705	000400			MOV	#TSOM,R5	:SET 'EXPECTED'.
5178	023124	011204				MOV	(R2),R4	:GET THE BUFFER REG
5179	023126	042704	170000			BIC	#170000,R4	:MASK CRC BITS
5180	023132	020504				CMP	R5,R4	:R5=GOOD, R4=?
5181	023134	001406				BEQ	3\$	:BR IF A MATCH
5182	023136	013703	001414			MOV	TXDBUF,R3	:ERROR MESSAGE SETUP
5183	023142	104003				HLT	3	:BIT FAILED TO SET
5184	023144	104401				SCOPI		:SW09=1?
5185	023146	012703	001412			MOV	#TXCSR,R3	:LOAD TRANSMITTER CSR
5186	023152	012737	023152	001220	3\$:	MOV	#3\$,LOCK	:SW09 SETUP
5187	023160	042712	000400			BIC	#TSOM,(R2)	:CLR TSOM
5188	023164	032713	000200			BIT	#TXDONE,(R3)	:TEST DONE
5189	023170	001402				BEQ	4\$	:BR IF CLEAR
5190	023172	104005				HLT	5	:DONE BIT IS SET AND SHOULD BE CLEARED
5191	023174	104401				SCOPI		:SW09=1?
5192	023176	012737	023176	001220	4\$:	MOV	#4\$,LOCK	:SW09 SETUP
5193	023204	032712	000400			BIT	#TSOM,(R2)	:TEST TSOM
5194	023210	001402				BEQ	5\$	:BR IF CLEAR
5195	023212	104010				HLT	10	:BIT FAILED TO CLR
5196	023214	104401				SCOPI		:SW09=1?
5197	023216	042713	000020		5\$:	BIC	#SEND,(R3)	:TURN OFF SEND
5198	023222	104412	000020			PKCLK	,16.	:POKE 8 BITS
5199	023226	032713	000200			BIT	#TXDONE,(R3)	:CHECK DONE
5200	023232	001401				BEQ	,+4	:BR IF OFF
5201	023234	104017				HLT	17	:DONE SET AND SHOULDN'T BE
5202	023236	104412	000002			PKCLK	,2	:POKE ONE FULL CLOCK
5203	023242	032713	001000			BIT	#TXACT,(R3)	:CHECK ACTIVE
5204	023246	001401				BEQ	6\$	:BR IF OFF
5205	023250	104011				HLT	11	:ACTIVE SETS AND SHOULDN'T BE
5206	023252	032713	000200		6\$:	BIT	#TXDONE,(R3)	:IS DONE UP?
5207	023256	001001				BNE	7\$	:BR IF YES
5208	023260	104006				HLT	6	:NO-REPORT ERROR
5209	023262	032713	040000		7\$:	BIT	#MTDATA,(R3)	:CHECK DATA OUT
5210	023266	001401				BEQ	10\$	:BR IF OFF
5211	023270	104012				HLT	12	:DATA SET SHOULD BE CLEAR
5212	023272	104400			10\$:	SCOPE		:SCOPE THIS TEST

5213
 5214
 5215
 5216
 5217
 5218
 5219
 5220
 5221
 5222
 5223
 5224
 5225

***** TEST 142 *****
 :*THIS TEST VERIFIES THAT THE DEVICE IDLES FLAGS
 :*IDLE A MINIMUM OF 64. FLAGS.
 :*****

:*****
 :*
 : TEST 142
 :*
 :*****
 :*****

5226	023274	012737	000142	001226	TST142: MOV	#142,@TSTNO	
5227	023302	012737	023364	001216	MOV	#TST143,NEXT	
5228	023310	013702	001414		MOV	TXDBUF,R2	;LOAD TX BUFFER
5229	023314	013703	001412		MOV	TXCSR,R3	;LOAD TX CONTROL REGISTER
5230	023320	052777	000400	156064	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
5231	023326	004737	005044		JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
5232	023332	005012			CLR	(R2)	;RESET TXDONE
5233	023334	052713	014000		BIS	#MMODE,(R3)	;ENTER M/MODE
5234	023340	052713	000020		BIS	#SEND,(R3)	;SET SEND
5235	023344	052712	000400		BIS	#TSOM,(R2)	;TURN ON START OF MSG
5236	023350	104412	000004		PKCLK	,4.	;SYNC UP DUP
5237	023354	004137	007052		JSR	R1,FLG	;SEND 64. FLAGS
5238	023360	000100			64.		;64. FLAGS
5239	023362	104400			SCOPE		;SCOPE THIS TEST

5240
 5241
 5242
 5243
 5244
 5245
 5246
 5247
 5248
 5249
 5250
 5251
 5252
 5253
 5254
 5255
 5256
 5257
 5258
 5259

```

***** TEST 143 *****
;THIS TEST PUSHES DATA THRU THE TRANSMITTER
;IN MAINTENANCE MODE. THE TEST SENDS A FLAG,
;AND TWO ALTERNATING ONES AND ZEROES CHARACTERS,
;AN ALL ZEROES CHARACTER AND AN ALL ONES
;CHARACTER TO VERIFY THE BIT STUFF CAPABILITY OF
;THE DUP WITHOUT A CRC CHECK.THE TEST ROTATES
;THE BITS THRU,SAMPLING THE DATA ON A BIT-BY-BIT
;BASIS,LSB FIRST. IT STORES THE BIT IN THE MSB OF
;THE SAVE LOCATION,COMPARES AND ROTATES RIGHT UNTIL
;THE CHARACTER IS ASSEMBLED.
*****
  
```

```

*****
;
; TEST 143
;
*****
*****
  
```

5260	023364	012737	000143	001226	TST143: MOV	#143,@TSTNO	
5261	023372	012737	023740	001216	MOV	#TST144,NEXT	
5262	023400	013703	001412		MOV	TXCSR,R3	;LOAD TRANSMITTER CONTROL REGISTER
5263	023404	013702	001414		MOV	TXDBUF,R2	;LOAD TRANSMITTER BUFFER
5264	023410	052777	000400	155774	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
5265	023416	004737	005044		JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
5266	023422	005012			CLR	(R2)	;RESET TXDONE
5267	023424	052713	014000		BIS	#MMODE,(R3)	;ENTER MINT MODE--PROGRAM CLOCKING
5268	023430	052777	001000	155752	BIS	#CRCEN,@PARCSR	;TURN OFF CRC
5269	023436	052713	000020		BIS	#SEND,(R3)	;ASSERT SEND
5270	023442	052712	000400		BIS	#TSOM,(R2)	
5271	023446	005037	001236		CLR	TEMP1	;CLEAR
5272	023452	005037	001240		CLR	TEMP2	;CLEAR
5273	023456	005037	001242		CLR	TEMP3	;CLEAR
5274	023462	005037	001244		CLR	TEMP4	;CLEAR
5275	023466	012704	000004		MOV	#4,R4	;LOAD THE # OF CHARACTERS TO DO
5276	023472	012705	023726		MOV	#TBL1,R5	;LOAD THE TABLE POINTER
5277	023476	012701	023726		MOV	#TBL1,R1	;DITTO
5278	023502	011137	001246		MOV	(R1),TEMP5	;DITTO
5279	023506	104412	000010		PKCLK	,8.	;START A FLAG
5280	023512	012512			MOV	(R5)+,(R2)	;LOAD THE FIRST CHARACTER AND CLR TSOM
5281	023514	104412	000014		PKCLK	,12.	;FINISH THE FLAG

DATA TEST OF A ZERO, ONE AND ALTERNATING ZERO-ONE CHARACTER

5282	023520	104412	000002	1\$:	PKCLK	,2	;PUSH OUT A BIT	
5283	023524	000241			CLC		;PUT CARRY IN A KNOWN STATE	
5284	023526	032713	040000		BIT	#MTDATA,(R3)	;TEST THE BIT	
5285	023532	001401			BEQ	2\$	;BR IF CLEAR	
5286	023534	000261			SEC		;SET CARRY FOR SOFTWARE	
5287	023536	106037	001236	2\$:	RORB	TEMP1	;STORE THE BIT	
5288	023542	000241			CLC		;PUT CARRY IN A KNOWN STATE	
5289	023544	106037	001246		RORB	TEMP5	;CHECK TO SEE WHAT THE BIT SHOULD BE	
5290	023550	103006			BCC	3\$	;BR IF CLEAR	
5291	023552	000261			SEC		;SET CARRY FOR SOFTWARE	
5292	023554	106037	001242		RORB	TEMP3	;SHIFT IN A ONE	
5293	023560	005237	001240		INC	TEMP2	;INC ONES COUNT	
5294	023564	000404			BR	4\$	;JUMP OVER	
5295	023566	106037	001242	3\$:	RORB	TEMP3	;LOAD A ZERO	
5296	023572	005037	001240		CLR	TEMP2	;CLEAR ONES COUNT	
5297	023576	023737	001236	001242	4\$:	CMP	TEMP1,TEMP3	;DOES HARDWARE = SOFTWARE
5298	023604	001401			BEQ	5\$	;BR IF YES	
5299	023606	104004			HLT	4	;HARDWARE AND SOFTWARE DON'T MATCH	
5300							;R1 HOLDS THE ADRS OF THE OUTPUT CHAR	
5301							;TEMP1,BIT7 = HARDWARE FOUND	
5302							;TEMP3,BIT7 = SOFTWARE CALCULATED	
5303							;TEMP4 GIVE THE BIT POSITION OUTPUT	
5304								
5305								
5306	023610	005704		5\$:	TST	R4	;CHECK FOR LAST CHAR	
5307	023612	001003			BNE	6\$	;BR IF NOT DONE	
5308	023614	052712	001000		BIS	#TEOM,(R2)	;LOAD END OF MESSAGE	
5309	023620	000405			BR	7\$	;FINISH TEST	
5310	023622	105777	155564	6\$:	TSTB	@TXCSR	;CHECK TO SEE IF EREADY FOR NEXT CHAR	
5311	023626	100002			BPL	7\$	;BR IF NO	
5312	023630	012512			MOV	(R5)+,(R2)	;LOAD NEXT CHAR	
5313	023632	005304			DEC	R4	;LOWER CHAR COUNT	
5314	023634	022737	000005	001240	7\$:	CMP	#5,TEMP2	;CHECK FOR STUFFED ZERO
5315	023642	001006			BNE	10\$	;BR IF NO	
5316	023644	104412	000002		PKCLK	,2	;PUSH OUT THE STUFFED ZERO	
5317	023650	032713	040000		BIT	#MTDATA,(R3)	;CHECK IT	
5318	023654	001401			BEQ	10\$	;BR IF OK	
5319	023656	104021			HLT	21	;FAILED TO BIT-STUFF!!	
5320	023660	005237	001244	10\$:	INC	TEMP4	;INC BIT COUNTER	
5321	023664	022737	000010	001244		CMP	#8.,TEMP4	;ARE WE DONE WITH THIS CHAR?
5322	023672	001312			BNE	1\$	;BR IF MORE TO GO	
5323	023674	005037	001236		CLR	TEMP1	;CLEAR OUT HARDWARE SAVE	
5324	023700	005037	001242		CLR	TEMP3	;CLEAR OUT SOFTWARE SAVE	
5325	023704	005037	001244		CLR	TEMP4	;CLEAR OU BIT COUNTER	
5326	023710	005721			TST	(R1)+	;POP TBL POINTER	
5327	023712	011137	001246		MOV	(R1),TEMP5		
5328	023716	032712	001000		BIT	#TEOM,(R2)	;CHECK FOR END OF TEST	
5329	023722	001676			BEQ	1\$	;BR IF MORE TO GO	
5330	023724	104400			SCOPE		;SCOPE THIS TEST	
5331								
5332	023726	000252		TBL1:	.WORD	252	;THE FIRST FOUR CHARACTERS	
5333	023730	000000			.WORD	000	;OF THIS TABLE ARE OUTPUT.	
5334	023732	000125			.WORD	125	;THE LAST CHARACTER IS	
5335	023734	000377			.WORD	377	;A PAD.	
5336	023736	000000			.WORD	000		
5337								

DATA TEST OF A ZERO,ONE AND ALTERNATING ZERO-ONE CHARACTER

5338 ;R1 = SOFTWARE TABLE POINTER
5339 ;R2 = TXDBUF
5340 ;R3 = TXCSR
5341 ;R4 = CHAR COUNTER
5342 ;R5 = HARDWARE TABLE POINTER
5343 ;TEMP1 = HARDWARE BIT
5344 ;TEMP2 = 1'S COUNT
5345 ;TEMP3 = SOFTWARE BIT
5346 ;TEMP4 = CHARACTER BIT COUNTER
5347 ;TEMP5 = SOFTWARE BYTE
5348
5349

***** TEST 144 *****
*THIS TEST VERIFIES THE ABORT SEQUENCE AND
*NORMAL DATA SEQUENCE OF FLAG, DATA, FLAG
*FOLLOWED BY IDLE LINE. THIS TEST ALSO PROVES
*THE FUNCTIONING OF ACTIVE, TSOM, TEOM, SEND AND DONE.
*WITHOUT USING A CRC CHECK.

*
: TEST 144
*

5363 *****
5364 023740 012737 000144 001226 TST144: MOV #144,@TSTNO
5365 023746 012737 024226 001216 MOV #TST145,NEXT
5366 023754 013703 001412 MOV TXCSR,R3 ;LOAD CONTROL REGISTER
5367 023760 013702 001414 MOV TXDBUF,R2 ;LOAD TRANSMITTER BUFFER
5368 023764 052777 000400 155420 BIS #MRESET,@TXCSR ;RESET THE DEVICE
5369 023772 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
5370 023776 005012 CLR (R2) ;RESET TXDONE
5371 024000 052713 014000 BIS #MMODE,(R3) ;ENTER M/MODE
5372 024004 052777 001000 155376 BIS #CRCEN,@PARCSR ;SHUT OFF CRC
5373 024012 052713 000020 BIS #SEND,(R3) ;SET SEND
5374 024016 052712 000400 BIS #TSOM,(R2) ;TURN ON START OF MSG
5375 024022 104412 000010 PKCLK ,8. ;SYNC UP DUP AND START PUSHING OUT A FLAG
5376 024026 052712 002000 BIS #TABORT,(R2) ;SET ABORT
5377 024032 104412 000014 PKCLK ,12. ;PUSH OUT THE ABORT
5378 024036 005037 001236 CLR TEMP1 ;CLEAR HOLD
5379 024042 104412 000002 1\$: PKCLK ,2 ;PUSH A BIT
5380 024046 032713 040000 BIT #MTDATA,(R3) ;CHECK DATA
5381 024052 001001 BNE 2\$;BR IF SET
5382 024054 104013 HLT 13 ;DATA FAILED TO SET
5383 024056 005237 001236 2\$: INC TEMP1 ;INC HOLD
5384 024062 022737 000002 001236 CMP #2,TEMP1 ;CHECK FOR FINISH
5385 024070 001364 BNE 1\$;BR IF NO
5386 024072 032713 001000 BIT #TXACT,(R3) ;TEST TRANSMITTER ACTIVE
5387 024076 001001 BNE 64\$;BR IF ACTIVE SET
5388 024100 104007 HLT 7 ;ACTIVE IS RESET AND SHOULDN'T BE
5389 024102 64\$:
5390 024102 042712 002000 BIC #TABORT,(R2) ;CLEAR ABORT
5391 024106 005037 001236 CLR TEMP1 ;CLEAR HOLD
5392 024112 104412 000002 3\$: PKCLK,2 ;PUSH A BIT
5393 024116 032713 040000 BIT #MTDATA,(R3) ;CHECK DATA

5394	024122	001001			BNE	4\$		:BR IF SET
5395	024124	104013			HLT	13		:DATA OUT FAILED TO SET
5396	024126	005237	001236		INC	TEMP1		:INC # TO DO
5397	024132	022737	000006	001236	CMP	#6,TEMP1		:AND CHECK IT
5398	024140	001364			BNE	3\$		:BR IF MORE TO GO
5399	024142	104412	000006		PKCLK	.6.		:POKE CLOCK
5400	024146	012712	000252		MOV	#252,(R2)		:CLEAR TSOM AND LOAD DATA
5401	024152	104412	000024		PKCLK	.20.		:FINISH THE FLAG AND DATA
5402	024156	052712	001000		BIS	#TEOM,(R2)		:SET TEOM
5403	024162	104412	000006		PKCLK	.6		:POKE CLOCK
5404	024166	004137	007052		JSR	R1,FLG		:SEND THREE FLAGS
5405	024172	000003			3			:DITTO
5406	024174	042713	000020		BIC	#SEND,(R3)		:CLEAR SEND
5407	024200	104412	000004		PKCLK	.4		:POKE CLOCK
5408	024204	032713	040000		BIT	#MTDATA,(R3)		:TEST TXDAT
5409	024210	001001			BNE	65\$		:MARK OUT
5410	024212	104013			HLT	13		:TXDAT A SPACE - SHOULD BE 1
5411	024214	032713	000200		BIT	#TXDONE,(R3)		:IS DONE UP
5412	024220	001001			BNE	66\$		:YES
5413	024222	104006			HLT	6		:NO - BUT IT SHOULD BE.
5414	024224	104400			66\$:	SCOPE		:SCOPE THIS TEST

***** TEST 145 *****
 *THIS TEST VERIFIES THAT A DATA
 *UNDER RUN CONDITION WILL CAUSE
 *THE TRANSMITTER DATA LATE BIT TO SET
 *AND THAT THE DEVICE WILL ABORT
 *UNTIL SEND IS CLEARED WHEN THE OUTPUT GOES TO A SPACE

 *
 : TEST 145
 *

5431	024226	012737	000145	001226	TST145:	MOV	#145,@TSTNO	
5432	024234	012737	024430	001216		MOV	#TST146,NEXT	
5433	024242	013703	001412			MOV	TXCSR,R3	:LOAD TRANSMITTER CONTROL REGISTER
5434	024246	013702	001414			MOV	TXDBUF,R2	:LOAD TRANSMITTER BUFFER
5435	024252	052777	000400	155132		BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5436	024260	004737	005044			JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5437	024264	005012				CLR	(R2)	:RESET TXDONE
5438	024266	052713	014000			BIS	#MODE,(R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5439	024272	052777	001000	155110		BIS	#CRCEN,@PARCSR	:TURN OFF CRC
5440	024300	052713	000020			BIS	#SEND,(R3)	:ASSERT SEND
5441	024304	052712	000400			BIS	#TSOM,(R2)	
5442	024310	104412	000006			PKCLK	.6.	:START OUTPUTING FLAG
5443	024314	012777	000000	155072		MOV	#0,@TXDBUF	:CLEAR TSOM-LOAD BUFFER WITH ZEROES
5444	024322	104412	000016			PKCLK	.14.	:FINISH FLAG
5445	024326	104412	000017			PKCLK	.15.	:OUTPUT UP TO 7 1/2 CLOCK TICKS
5446	024332	032777	100000	155052		BIT	#TXDLAT,@TXCSR	:MAKE SURE DNA IS NOT SET
5447	024340	001401				BEQ	1\$	:BR IF CLEARED
5448	024342	104014				HLT	14	:BIT IS SET TOO SOON
5449	024344	104412	000002		1\$:	PKCLK	.2	:FINISH LAST CLOCK

5450	024350	032777	100000	155034	BIT	#TXDLAT,@TXCSR	:NOW CHECK DNA
5451	024356	001001			BNE	2\$	:BRANCH IF SET
5452	024360	104015			HLT	15	:BIT IS CLEARED AND SHOULD BE SET
5453	024362						
5454	024362	042777	000020	155022	BIC	#SEND,@TXCSR	:TURN OFF SEND
5455	024370	104412	000020		PKCLK	,16.	:PUSH 8 BITS
5456	024374	012737	000010	001236	MOV	#8.,TEMP1	:SETUP FOR IDLE LINE
5457	024402	104412	000002		PKCLK	,2	:OUTPUT BIT
5458	024406	032777	040000	154776	BIT	#MTDATA,@TXCSR	:CHECK IT
5459	024414	001401			BEQ	4\$	:BRANCH IF ZERO
5460	024416	104012			HLT	12	:BIT IS A 1
5461	024420	005337	001236		DEC	TEMP1	:LOWER COUNT
5462	024424	001366			BNE	3\$	:NOT DONE? - GO BACK
5463	024426	104400			SCOPE		:SCOPE THIS TEST

***** TEST 146 *****
 :*THIS TEST VERIFIES THAT DROPPING OF
 :*SEND BEFORE SETTING TEOM CAUSES
 :*A SPACE TO BE OUTPUT AFTER COMPLETION OF
 :*A CHARACTER WITH AND WITHOUT BIT STUFF.
 :*****

:*****
 : *
 : TEST 146
 : *
 :*****

5479	024430	012737	000146	001226	TST146: MOV	#146,@TSTNO	
5480	024436	012737	025002	001216	MOV	#TST147,NEXT	
5481	024444	013703	001412		MOV	TXCSR,R3	:LOAD TRANSMITTER CONTROL REGISTER
5482	024450	013702	001414		MOV	TXDBUF,R2	:LOAD TRANSMITTER BUFFER
5483	024454	052777	000400	154730	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5484	024462	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5485	024466	005012			CLR	(R2)	:RESET TXDONE
5486	024470	052713	014000		BIS	#MMODE,(R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5487	024474	052777	001000	154706	BIS	#CRCEN,@PARCSR	:TURN OFF CRC
5488	024502	052713	000020		BIS	#SEND,(R3)	:ASSERT SEND
5489	024506	052712	000400		BIS	#TSOM,(R2)	
5490	024512	104412	000006		PKCLK	,6.	:PUSH 2 BITS
5491	024516	012777	000252	154670	MOV	#252,@TXDBUF	:LOAD DATA
5492	024524	104412	000014		PKCLK	,12.	:PUSH 6 BITS
5493	024530	042777	000020	154654	BIC	#SEND,@TXCSR	:TURN OFF TRANSMITTER
5494	024536	104412	000002		PKCLK	,2	:POKE A FULL CLOCK
5495	024542	012737	000010	001236	MOV	#8.,TEMP1	:LOAD TEMP1
5496	024550	104412	000012		PKCLK	,10.	:PUSH 5 BITS
5497	024554	032777	040000	154630	BIT	#MTDATA,@TXCSR	:CHECK DATA
5498	024562	001401			BEQ	2\$	:BR IF CLEAR
5499	024564	104012			HLT	12	:DATA IS SET - SHOULD BE CLEAR
5500	024566	005337	001236		DEC	TEMP1	:LOWER THE # OF TIMES TO REPEAT
5501	024572	001366			BNE	1\$	:BR IF TO GOE MSG
5502	024574	032777	001000	154610	BIT	#TXACT,@TXCSR	:CHECK ACTIVE
5503	024602	001401			BEQ	7\$	:BR IF OFF
5504	024604	104011			HLT	11	:ACTIVE FAILED TO CLEAR
5505	024606						

5506	024606	052777	000400	154576	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5507	024614	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5508	024620	013703	001412		MOV	TXCSR,R3	:LOAD TRANSMITTER CONTROL REGISTER
5509	024624	013702	001414		MOV	TXDBUF,R2	:LOAD TRANSMITTER BUFFER
5510	024630	052777	000400	154554	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5511	024636	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5512	024642	005012			CLR	(R2)	:RESET TXDONE
5513	024644	052713	014000		BIS	#MMODE,(R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5514	024650	052777	001000	154532	BIS	#CRCEN,@PARCSR	:TURN OFF CRC
5515	024656	052713	000020		BIS	#SEND,(R3)	:ASSERT SEND
5516	024662	052712	000400		BIS	#TSOM,(R2)	
5517	024666	104412	000006		PKCLK	,6.	:PUSH TWO BITS
5518	024672	012777	000177	154514	MOV	#177,@TXDBUF	:LOAD DATA
5519	024700	104412	000020		PKCLK	,16.	:PUSH 8 BITS
5520	024704	042777	000020	154500	BIC	#SEND,@TXCSR	:TURN OFF TRANSMITTER
5521	024712	104412	000014		PKCLK	,12.	:PUSH 6 BITS
5522	024716	032777	040000	154466	BIT	#MTDATA,@TXCSR	:CHECK DATA OUT FOR BIT STUFF FUNCTION
5523	024724	001001			BNE	3\$	:BR IF SET
5524	024726	104013			HLT	13	:BIT IS A 0, SHOULD BE A 1 - DEVICE
5525							:FAILED TO BIT STUFF
5526	024730	104412	000004		PKCLK	,4	:PUSH 2 BITS
5527	024734	012737	000010	001236	MOV	#8.,TEMP1	:LOAD TEMP1
5528	024742	104412	000002		PKCLK	,2	:PUSH A BIT
5529	024746	032777	040000	154436	BIT	#MTDATA,@TXCSR	:CHECK DATA
5530	024754	001401			BEQ	5\$	:BR IF OFF
5531	024756	104012			HLT	12	:DATA WINDOW SET - SHOULD BE CLEAR
5532	024760	005337	001236		DEC	TEMP1	:LOWER THE # OF TIMES TO CHECK
5533	024764	001366			BNE	4\$	:BR IF MORE TOGO
5534	024766	032777	001000	154416	BIT	#TXACT,@TXCSR	:CHECK ACTIVE
5535	024774	001401			BEQ	6\$	:BR IF OFF
5536	024776	104011			HLT	11	:ACTIVE FAILED TO CLEAR
5537	025000	104400			6\$:	SCOPE	:SCOPE THIS TEST

***** TEST 147 *****
 :*THIS TEST VERIFIES THAT A DATA UNDERRUN
 :*CONDITION WILL CAUSE THE TRANSMITTER DATA
 :*LATE BIT TO SET AND THAT THE DUP WILL GO TO
 :*A MARK OUTPUT
 :*****

:*****
 :*
 : TEST 147
 :*
 :*****

5551							
5552	025002	012737	000147	001226	TST147:	MOV	#147,@TSTNO
5553	025010	012737	025176	001216		MOV	#TST150,NEXT
5554	025016	013703	001412			MOV	TXCSR,R3
5555	025022	013702	001414			MOV	TXDBUF,R2
5556	025026	052777	000400	154356		BIS	#MRESET,@TXCSR
5557	025034	004737	005044			JSR	PC,SMALL
5558	025040	005012				CLR	(R2)
5559	025042	052713	014000			BIS	#MMODE,(R3)
5560	025046	052777	001000	154334		BIS	#CRCEN,@PARCSR
5561	025054	052713	000020			BIS	#SEND,(R3)

:LOAD TRANSMITTER CONTROL REGISTER
 :LOAD TRANSMITTER BUFFER
 :RESET THE DEVICE
 :WAIT FOR RESET TO FINISH
 :RESET TXDONE
 :ENTER MINT MODE--PROGRAM CLOCKING
 :TURN OFF CRC
 :ASSERT SEND

5562	025060	052712	000400		BIS	#TSOM,(R2)	
5563	025064	104412	000006		PKCLK	,6.	:START FLAG
5564	025070	112777	000176	154316	MOVB	#176,@TXDBUF	:LOAD CHARACTER TO BE BIT-STUFFED
5565	025076	042777	000400	154310	BIC	#TSOM,@TXDBUF	:SHUT OFF TSOM
5566	025104	104412	000016		PKCLK	,14.	:FINISH CLOCKING FLAG
5567	025110	104412	000017		PKCLK	,15.	:CLOCK TO WITHIN 1 1/2 CLOCKS
5568	025114	032777	100000	154270	BIT	#TXDLAT,@TXCSR	:CHECK DATA LATE
5569	025122	001401			BEQ	1\$	:BRANCH IF CLEAR
5570	025124	104014			HLT	14	:BIT IS SET TOO SOON
5571	025126	104412	000002	1\$:	PKCLK	,2	:CLOCK TO WITHIN A HALF-CLOCK OF DNA
5572	025132	032777	100000	154252	BIT	#TXDLAT,@TXCSR	:CHECK DNA
5573	025140	001401			BEQ	2\$	:BR IF OFF
5574	025142	104014			HLT	14	:BIT SET TOO SOON, DEVICE FAILED TO BIT-STUFF
5575	025144	104412	000004	2\$:	PKCLK	,4	:FINISH CHARACTER
5576	025150	032777	100000	154234	BIT	#TXDLAT,@TXCSR	:CHECK DNA
5577	025156	001001			BNE	3\$	:BRANCH IF SET
5578	025160	104016			HLT	16	:BIT SHOULD BE SET AND IS CLEARED
5579	025162	032777	040000	154222	3\$:	BIT	#MTDATA,@TXCSR
5580	025170	001001			BNE	4\$	:CHECK DATA
5581	025172	104013			HLT	13	:BR IF SET
5582	025174	104400		4\$:	SCOPE		:DATA WAS CLEAR - SHOULD BE SET
5583							:SCOPE THIS TEST

***** TEST 150 *****
 :THIS TEST VERIFIES THAT SETTING TEOM
 :AND TSOM AT THE SAME TIME WILL HOLD
 :ACTIVE UP AND SEND A FLAG
 :*****

:*****
 : *
 : TEST 150
 : *
 :*****

5597	025176	012737	000150	001226	TST150: MOV	#150,@TSTNO	
5598	025204	012737	025332	001216	MOV	#TST151,NEXT	
5599	025212	013703	001412		MOV	TXCSR,R3	:LOAD TRANSMITTER CONTROL REGISTER
5600	025216	013702	001414		MOV	TXDBUF,R2	:LOAD TRANSMITTER BUFFER
5601	025222	052777	000400	154162	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5602	025230	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5603	025234	005012			CLR	(R2)	:RESET TXDONE
5604	025236	052713	014000		BIS	#MMODE,(R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5605	025242	052777	001000	154140	BIS	#CRCEN,@PARCSR	:TURN OFF CRC
5606	025250	052713	000020		BIS	#SEND,(R3)	:ASSERT SEND
5607	025254	052712	000400		BIS	#TSOM,(R2)	
5608	025260	104412	000006		PKCLK	,6.	:SYNC UP DUP
5609	025264	012777	000252	154122	MOV	#252,@TXDBUF	:LOAD DATA
5610	025272	104412	000032		PKCLK	,26.	:FINISH THE FLAG AND DATA
5611	025276	052777	001400	154110	BIS	#TEOM!#TSOM,@TXDBUF	:TURN ON START AND END OF MSG
5612	025304	104412	000004		PKCLK	,4	:PUSH 2 BITS
5613	025310	032777	001000	154074	BIT	#TXACT,@TXCSR	:TEST ACTIVE
5614	025316	001001			BNE	1\$	:BR IF SET
5615	025320	104007			HLT	7	:ACTIVE SHOULD BE SET AND IS CLEAR
5616	025322	004137	007052	1\$:	JSR	R1,FLG	:PUSH OUT A FLAG
5617	025326	000001				1	:ONE FLAG

5618 025330 104400

SCOPE

;SCOPE THIS TEST

5619

5620

5621

5622

5623

5624

5625

5626

5627

5628

5629

5630

5631

5632 025332 012737 000151 001226

5633 025340 012737 026402 001216

5634 025346 012737 102010 026374

5635 025354 012737 026212 001244

5636 025362 012737 025370 001220

5637 025370

5638 025370 052777 000400 154014

5639 025376 004737 005044

5640 025402 005077 154006

5641 025406 052777 014000 153776

5642 025414 052777 000020 153770

5643 025422 052777 000400 153764

5644 025430 104412 000006

5645 025434 017777 153604 153752

5646 025442 104412 000020

5647 025446 005037 026400

5648 025452 004537 026222

5649 025456 000010

5650 025460 000000

5651 025462 177777

5652 025464 012777 001000 153722

5653 025472 104412 000016

5654 025476 004137 006750

5655 025502 100000

5656 025504 013705 026400

5657 025510 005105

5658 025512 020504

5659 025514 001401

5660 025516 104020

5661

5662

5663

5664 025520 104401

5665 025522 062737 000002 001244

5666 025530 012737 025536 001220

5667 025536

5668 025536 052777 000400 153646

5669 025544 004737 005044

5670 025550 005077 153640

5671 025554 052777 014000 153630

5672 025562 052777 000020 153622

5673 025570 052777 000400 153616

***** TEST 151 *****
 ;*TEST OF THE BCC OPERATION USING
 ;*CRC.CCITT FOR THE POLYNOMIAL. SPECIFIC
 ;*DATA PATTERNS ARE USED TO ISOLATE FAULTS.
 ;*****

TEST 151

TST151: MOV #151,@TSTNO
 MOV #TST152,NEXT
 MOV #CRC.CCITT,XPOLY ;SET THE POLYNOMIAL
 MOV #5\$,TEMP4 ;GET THE TABLE POINTER
 MOV #1\$,LOCK ;SETUP FOR SW09=1

1\$:

BIS #MRESET,@TXCSR ;RESET THE DEVICE
 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
 CLR @TXDBUF ;RESET TXDONE
 BIS #MMODE,@TXCSR ;ENTER M/MODE
 BIS #SEND,@TXCSR ;TURN ON DUP
 BIS #TSOM,@TXDBUF ;TURN ON START OF MSG
 PKCLK ,6. ;SYNC UP DUP
 MOV @TEMP4,@TXDBUF ;LOAD DATA
 PKCLK ,16. ;PUSH 8 BITS
 CLR CALBCC ;CLEAR OUT OLD BCC
 JSR R5,SIMBCC ;CALCULATE A SOFTWARE BCC
 8. ;BASED
 0 ;ON THESE
 -1 ;PARAMETERS
 MOV #TEOM,@TXDBUF ;CLEAR TSOM, SET TEOM
 PKCLK ,14. ;PUSHOUT DATA
 JSR R1,ACC ;DO A BCC CALCULATION (HDWR)
 BIT15 ;FOR 16 BITS
 MOV CALBCC,R5 ;MOV SOFTWARE BCC TO EXPECTED
 COM R5 ;INVERT IT
 CMP R5,R4 ;DOES EXPECTED=FOUND?
 BEQ 64\$;BR IF OK
 HLT 20 ;BCC CALCULATION ERROR
 ;TO DEBUG CRC USE THE CRC
 ;DEBUG AID TEST. SEE FRONT OF THE
 ;LISTING FOR TEST LOCATION
 ;SW09=1?

64\$:

SCOP1
 ADD #2,TEMP4
 MOV #2\$,LOCK ;MOVE THE TABLE POINTER
 ;SETUP FOR SW09=1

2\$:

BIS #MRESET,@TXCSR ;RESET THE DEVICE
 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
 CLR @TXDBUF ;RESET TXDONE
 BIS #MMODE,@TXCSR ;ENTER M/MODE
 BIS #SEND,@TXCSR ;TURN ON DUP
 BIS #TSOM,@TXDBUF ;TURN ON START OF MSG

5674	025576	104412	000006		PKCLK	,6.	:SYNC UP DUP
5675	025602	017777	153436	153604	MOV	@TEMP4,@TXDBUF	:LOAD DATA
5676	025610	104412	000020		PKCLK	,16.	:PUSH 8 BITS
5677	025614	005037	026400		CLR	CALBCC	:CLEAR OUT OLD BCC
5678	025620	004537	026222		JSR	R5,SIMBCC	:CALCULATE A SOFTWARE BCC
5679	025624	000010			8.		:BASED
5680	025626	000252			^B<10101010>		:ON THESE
5681	025630	177777			-1		:PARAMETERS
5682	025632	012777	001000	153554	MOV	#TEOM,@TXDBUF	:CLEAR TSOM, SET TEOM
5683	025640	104412	000016		PKCLK	,14.	:PUSHOUT DATA
5684	025644	004137	006750		JSR	R1,ACC	:DO A BCC CALCULATION (HDWR)
5685	025650	100000			BIT15		:FOR 16 BITS
5686	025652	013705	026400		MOV	CALBCC,R5	:MOV SOFTWARE BCC TO EXPECTED
5687	025656	005105			COM	R5	:INVERT IT
5688	025660	020504			CMP	R5,R4	:DOES EXPECTED=FOUND?
5689	025662	001401			BEQ	65\$	:BR IF OK
5690	025664	104020			HLT	20	:BCC CALCULATION ERROR
5691							:TO DEBUG CRC USE THE CRC
5692							:DEBUG AID TEST. SEE FRONT OF THE
5693							:LISTING FOR TEST LOCATION
5694	025666	104401			65\$: SCOP1		:SW09=1?
5695	025670	062737	000002	001244	ADD	#2,TEMP4	:MOVE THE TABLE POINTER
5696	025676	012737	025704	001220	MOV	#3\$,LOCK	:SETUP FOR SW09=1
5697	025704				3\$:		
5698	025704	052777	000400	153500	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5699	025712	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
5700	025716	005077	153472		CLR	@TXDBUF	:RESET TXDONE
5701	025722	052777	014000	153462	BIS	#MMODE,@TXCSR	:ENTER M/MODE
5702	025730	052777	000020	153454	BIS	#SEND,@TXCSR	:TURN ON DLP
5703	025736	052777	000400	153450	BIS	#TSOM,@TXDBUF	:TURN ON START OF MSG
5704	025744	104412	000006		PKCLK	,6.	:SYNC UP DUP
5705	025750	017777	153270	153436	MOV	@TEMP4,@TXDBUF	:LOAD DATA
5706	025756	104412	000020		PKCLK	,16.	:PUSH 8 BITS
5707	025762	005037	026400		CLR	CALBCC	:CLEAR OUT OLD BCC
5708	025766	004537	026222		JSR	R5,SIMBCC	:CALCULATE A SOFTWARE BCC
5709	025772	000010			8.		:BASED
5710	025774	000125			^B<01010101>		:ON THESE
5711	025776	177777			-1		:PARAMETERS
5712	026000	012777	001000	153406	MOV	#TEOM,@TXDBUF	:CLEAR TSOM, SET TEOM
5713	026006	104412	000016		PKCLK	,14.	:PUSHOUT DATA
5714	026012	004137	006750		JSR	R1,ACC	:DO A BCC CALCULATION (HDWR)
5715	026016	100000			BIT15		:FOR 16 BITS
5716	026020	013705	026400		MOV	CALBCC,R5	:MOV SOFTWARE BCC TO EXPECTED
5717	026024	005105			COM	R5	:INVERT IT
5718	026026	020504			CMP	R5,R4	:DOES EXPECTED=FOUND?
5719	026030	001401			BEQ	66\$	:BR IF OK
5720	026032	104020			HLT	20	:BCC CALCULATION ERROR
5721							:TO DEBUG CRC USE THE CRC
5722							:DEBUG AID TEST. SEE FRONT OF THE
5723							:LISTING FOR TEST LOCATION
5724	026034	104401			66\$: SCOP1		:SW09=1?
5725	026036	062737	000002	001244	ADD	#2,TEMP4	:MOVE THE TABLE POINTER
5726	026044	012737	026052	001220	MOV	#4\$,LOCK	:SETUP FOR SW09=1
5727	026052				4\$:		
5728	026052	052777	000400	153332	BIS	#MRESET,@TXCSR	:RESET THE DEVICE
5729	026060	004737	005044		JSR	PC,SMALL	:WAIT FOR RESET TO FINISH

5730	026064	005077	153324		CLR	@TXDBUF	:RESET TXDONE
5731	026070	052777	014000	153314	BIS	#MMODE,@TXCSR	:ENTER M/MODE
5732	026076	052777	000020	153306	BIS	#SEND,@TXCSR	:TURN ON DUP
5733	026104	052777	000400	153302	BIS	#TSOM,@TXDBUF	:TURN ON START OF MSG
5734	026112	104412	000006		PKCLK	.6.	:SYNC UP DUP
5735	026116	017777	153122	153270	MOV	@TEMP4,@TXDBUF	:LOAD DATA
5736	026124	104412	000020		PKCLK	.16.	:PUSH 8 BITS
5737	026130	005037	026400		CLR	CALBCC	:CLEAR OUT OLD BCC
5738	026134	004537	026222		JSR	R5,SIMBCC	:CALCULATE A SOFTWARE BCC
5739	026140	000010			8.		:BASED
5740	026142	000377			^B<11111111>		:ON THESE
5741	026144	177777			-1		:PARAMETERS
5742	026146	012777	001000	153240	MOV	#TEOM,@TXDBUF	:CLEAR TSOM, SET TEOM
5743	026154	104412	000020		PKCLK	.16.	:PUSH 8 BITS
5744	026160	004137	006750		JSR	R1,ACC	:DO A BCC CALCULATION (HDWR)
5745	026164	100000			BIT15		:FOR 16 BITS
5746	026166	012705	157400		MOV	#157400,R5	:LOAD THE BCC
5747	026172	020504			CMP	R5,R4	:DOES EXPECTED=FOUND?
5748	026174	001401			BEQ	67\$	:BR IF OK
5749	026176	104020			HLT	20	:BCC CALCULATION ERROR
5750							:TO DEBUG CRC USE THE CRC
5751							:DEBUG AID TEST. SEE FRONT OF THE
5752							:LISTING FOR TEST LOCATION
5753	026200	104401			67\$: SCOP1		:SW09=1?
5754	026202	062737	000002	001244	ADD	#2,TEMP4	:MOVE THE TABLE POINTER
5755	026210	104400			5\$: SCOPE		:SCOPE THIS TEST
5756	026212	000000			.WORD	0	
5757	026214	000252			.WORD	252	
5758	026216	000125			.WORD	125	
5759	026220	000377			.WORD	377	
5760	026222	010046			SIMBCC: MOV	R0,-(SP)	
5761	026224	010146			MOV	R1,-(SP)	
5762	026226	010246			MOV	R2,-(SP)	
5763	026230	012537	001236		MOV	(R5)+,TEMP1	
5764	026234	012537	001240		MOV	(R5)+,TEMP2	
5765	026240	012537	001242		MOV	(R5)+,TEMP3	
5766	026244	005037	026376		1\$: CLR	BCCFBK	
5767	026250	013700	001242		MOV	TEMP3,R0	
5768	026254	006037	001240		ROR	TEMP2	
5769	026260	005500			ADC	R0	
5770	026262	032700	000001		BIT	#BIT0,R0	
5771	026266	001402			BEQ	2\$	
5772	026270	005137	026376		COM	BCCFBK	
5773	026274	013700	026374		2\$: MOV	XPOLY,R0	
5774	026300	005100			COM	R0	
5775	026302	040037	026376		BIC	R0,BCCFBK	
5776	026306	000241			CLC		
5777	026310	006037	001242		ROR	TEMP3	
5778	026314	013700	026376		MOV	BCCFBK,R0	
5779	026320	013701	001242		MOV	TEMP3,R1	
5780	026324	010102			MOV	R1,R2	
5781	026326	040100			BIC	R1,R0	
5782	026330	043702	026376		BIC	BCCFBK,R2	
5783	026334	050200			BIS	R2,R0	
5784	026336	043737	026374	001242	BIC	XPOLY,TEMP3	
5785	026344	050037	001242		BIS	R0,TEMP3	

5786	026350	005337	001236	DEC	TEMP1
5787	026354	001333		BNE	1\$
5788	026356	013737	001242 026400	MOV	TEMP3,CALBCC
5789	026364	012602		MOV	(SP)+,R2
5790	026366	012601		MOV	(SP)+,R1
5791	026370	012600		MOV	(SP)+,R0
5792	026372	000205		RTS	R5
5793	026374	000000			
5794	026376	000000			
5795	026400	000000			
5796		120001			
5797		102010			
5798					
5799					

XPOLY: 0
 BCCFBK: 0
 CALBCC: 0
 CRC16=120001
 CRC.CCITT=102010

***** TEST 152 *****
 ;*THIS TEST IS AN AID FOR DEBUGGING CRC
 ;*ERRORS. A CHARACTER IS LOADED INTO THE
 ;*DUP AND PUSHED OUT BIT BY BIT WHILE
 ;*ALLOWING THE OPERATOR TO MONITOR THE CRC
 ;*CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
 ;*CAN ALSO BE CHANGED BY THE OPERATOR.
 ;*PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
 ;*ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
 ;NOTE: REMEMBER--IN SDLC A ONE IS A LOGIC LOW IN
 ;THE CRC GENERATOR.
 ;*****

 :
 :
 : TEST 152
 :
 :
 : *****

5818	026402	012737	000152	001226
5819	026410	012737	002764	001216
5820	026416	052777	000400	152766
5821	026424	004737	005044	
5822	026430	012737	102010	026374
5823	026436	012737	000125	026604
5824	026444	013737	026604	001252
5825	026452	012737	177777	026400
5826	026460	013737	026400	026606
5827	026466	005037	001242	
5828	026472	005037	001244	
5829	026476	005037	001246	
5830	026502	005077	152706	
5831	026506	052777	014000	152676
5832	026514	052777	000020	152662
5833	026522	052777	000020	152662
5834	026530	012777	000400	152656
5835	026536	104412	000044	
5836	026542	013777	026604	152644
5837	026550	104412	000020	
5838	026554	104412	000002	
5839	026560	013737	001244	001254
5840	026566	005237	001242	
5841	026572	005237	001244	

TST152: MOV #152,@TSTNO
 MOV #.EOP,NEXT
 BIS #MRESET,@TXCSR ;RESET THE DEVICE
 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
 MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
 MOV #125,3\$;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
 MOV 3\$,SAVR1
 MOV #-1,CALBCC ;CLEAR FOR SOFTWARE BCC
 MOV CALBCC,4\$
 CLR TEMP3 ;CLR SOFTWARE MEMORY
 CLR TEMP4 ;CLEAR BIT COUNTER
 CLR TEMP5
 CLR @TXDBUF ;RESET TXDONE
 BIS #MMODE,@TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
 MOV #TSOM,@TXDBUF ;
 PKCLK 36. ;PUSH OUT 2
 MOV 3\$,@TXDBUF ;LOAD DATA
 PKCLK 16. ;PUSH OUT ANOTHER
 PKCLK 2. ;PUSH OUT A BIT
 1\$: MOV TEMP4,SAVR2 ;SET UP TO TYPE
 INC TEMP3 ;UPDATE THE COUNT
 INC TEMP4 ;UPDATE BIT COUNTER

5842	026576	004537	026222	2\$:	JSR	R5,SIMBCC	:CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
5843	026602	000001			1		:SHIFTS
5844	026604	000000		3\$:	.WORD 0		:DATA
5845	026606	000000		4\$:	.WORD 0		:PREVIOUS BCC
5846	026610	004737	026706		JSR	PC,5\$	:CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
5847	026614	000241			CLC		:CLEAR FOR NEXT ROTATE
5848	026616	106037	026604		RORB	3\$	:SET UP THE NEXT BIT
5849	026622	013737	026400	026606	MOV	CALBCC,4\$	:FOR THE SOFTWARE BCC
5850	026630	022737	000006	001244	CMP	#6,TEMP4	
5851	026636	001002			BNE	.+6	
5852	026640	005077	152550		CLR	@TXDBUF	
5853	026644	022737	000014	001242	CMP	#12.,TEMP3	
5854	026652	001003			BNE	12\$	
5855	026654	012777	001000	152532	MOV	#TEOM,@TXDBUF	
5856	026662	022737	000020	001244	12\$:	CMP	#16.,TEMP4
5857							:ALL DONE WITH THE CHARACTER?
5858							:INCREASE THE COMPARE NUMBER TO
5859	026670	001331			BNE	1\$	:ALLOW CRC TO BE OUTPUT
5860	026672	052777	000400	152512	BIS	#MRESET,@TXCSR	:BR IF MORE TO GO
5861	026700	004737	005044		JSR	PC,SMALL	:RESET THE DEVICE
5862	026704	104400			SCOPE		:WAIT FOR RESET TO FINISH
5863							:SCOPE THIS TEST
5864	026706	032777	001000	152266	5\$:	BIT	#SW09,@SWR
5865	026714	001432			BEQ	6\$	:SW09=1?
5866	026716	013704	026400		MOV	CALBCC,R4	:BR IF NO
5867	026722	012737	000001	001256	MOV	#1,SAVR3	:THE DATA CHARACTER IS
5868	026730	000241			CLC		:FOLLOWED BY A ZERO CHARACTER.THE
5869	026732	006004			11\$:	ROR	:DATA BIT IN CRC SHOWS WHICH BIT
5870	026734	006137	001256		ROL	SAVR3	:OF THE TWO CHARACTERS IS BEING
5871	026740	103374			BCC	11\$	:GENERATED
5872	026742	105737	001246		TSTB	TEMP5	
5873	026746	001006			BNE	10\$	
5874	026750	104402	027503		TYPE	,EM1	:TYPE MSG
5875	026754	104402	027532		TYPE	,MH1	:TYPE HEADER
5876	026760	105137	001246		COMB	TEMP5	
5877	026764	104410			10\$:	CONVRT	
5878	026766	030174			DT1		
5879	026770	105777	152210		7\$:	TSTB	@TKCSR
5880	026774	100375			BPL	7\$	:CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
5881	026776	017701	152204		MOV	@TKDBR,R1	:BR IF NOT YET
5882	027002	000207			6\$:	RTS	:READ THE BUFFER
5883						PC	:RETURN
5884							

5885

(1)	027004	005015	051124	047101	EM2:	.ASCIZ	<15><12>/TRANSMITTER /	
(1)	027024	005015	042522	042503	EM3:	.ASCIZ	<15><12>/RECEIVER /	
(1)	027041	015	052412	042516	EM4:	.ASCIZ	<15><12>/UNEXPECTED INTERRUPT/	
(1)	027070	005015	051120	046511	EM6:	.ASCIZ	<15><12>^PRIMARY REGISTER TEST ^	
(1)	027121	377	051124	047101	EM10:	.ASCIZ	<377>/TRANSMITTER DATA ERROR. /	
(1)	027153	377	044103	041505	EM11:	.ASCIZ	<377>/CHECK CARRY AND OUTBIT. /	
(1)	027207	104	047117	020105	EM12:	.ASCIZ	/DONE BIT /	
(1)	027222	040506	046111	042105	EM13:	.ASCIZ	/FAILED TO CLEAR /	
(1)	027243	106	044501	042514	EM14:	.ASCIZ	/FAILED TO SET /	
(1)	027262	041501	044524	042526	EM15:	.ASCIZ	/ACTIVE BIT /	
(1)	027277	124	047523	020115	EM16:	.ASCIZ	/TSQM BIT /	
(1)	027312	052517	050124	052125	EM17:	.ASCIZ	/OUTPUT DATA /	
(1)	027330	042523	020124	047524	EM22:	.ASCIZ	/SET TOO SOON /	
(1)	027346	052377	042130	040514	EM23:	.ASCIZ	<377>/TXDLAT BIT /	
(1)	027364	041777	041522	041440	EM24:	.ASCIZ	<377>/CRC CALCULATION ERROR /	
(1)	027414	047111	042524	052522	DH2:	.ASCIZ	/INTERUPTED UNEXPECTEDLY./	
(1)	027445	377	054105	042520	DH6:	.ASCIZ	<377>/EXPECTED FOUND PRI REG /	
(1)	027503	377	051103	020103	EM1:	.ASCIZ	<377>/CRC GENERATOR STATUS /	
(1)	027532	042377	052101	020101	MH1:	.ASCIZ	<377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /	
(1)	027625	106	044501	042514	DH7:	.ASCIZ	/FAILED TO BIT-STUFF. /	
(1)	027653	015	050012	044522	EM5:	.ASCIZ	<15><12>/PRIMARY REGISTER ADDRESSING TIME-OUT /	
(1)	027723	015	050012	044522	EM7:	.ASCIZ	<15><12>/PRIMARY REGISTER RESET TEST /	
(1)	027763	377	042522	044507	DH5:	.ASCIZ	<377>/REGISTER REFERENCED TRAPPED FROM /	
(1)						.EVEN		
(1)	030026					.ERRTAB:		
(1)	030026	000000				0		
(1)	030030	000000				0		
(1)	030032	000000				0		
(1)	030034	027653				EM5		
(1)	030036	027763				DH5	:HALT 1	
(1)	030040	030212				DT5		
(1)								
(1)	030042	027723				EM7		
(1)	030044	027445				DH6	:HALT 2	
(1)	030046	030224				DT6		
(1)								
(1)	030050	027070				EM6		
(1)	030052	027445				DH6	:HALT 3	
(1)	030054	030224				DT6		
(1)								
(1)	030056	027121				EM10		
(1)	030060	000000				0	:HALT 4	
(1)	030062	000000				0		
(1)								
(1)	030064	027207				EM12		
(1)	030066	027222				EM13	:HALT 5	
(1)	030070	000000				0		
(1)								
(1)	030072	027207				EM12		
(1)	030074	027243				EM14	:HALT 6	
(1)	030076	000000				0		
(1)								
(1)	030100	027262				EM15		
(1)	030102	027243				EM14	:HALT 7	
(1)	030104	000000				0		

(1)					
(1)	030106	027277		EM16	
(1)	030110	027222		EM13	;HALT 10
(1)	030112	000000		0	
(1)					
(1)	030114	027262		EM15	
(1)	030116	027222		EM13	;HALT 11
(1)	030120	000000		0	
(1)					
(1)	030122	027312		EM17	
(1)	030124	027222		EM13	;HALT 12
(1)	030126	000000		0	
(1)					
(1)	030130	027312		EM17	
(1)	030132	027243		EM14	;HALT 13
(1)	030134	000000		0	
(1)					
(1)	030136	027346		EM23	
(1)	030140	027330		EM22	;HALT 14
(1)	030142	000000		0	
(1)					
(1)	030144	027346		EM23	
(1)	030146	027243		EM14	;HALT 15
(1)	030150	000000		0	
(1)					
(1)	030152	027346		EM23	
(1)	030154	027222		EM13	;HALT 16
(1)	030156	000000		0	
(1)					
(1)	030160	027207		EM12	
(1)	030162	027330		EM22	;HALT 17
(1)	030164	000000		0	
(1)					
(1)	030166	027364		EM24	
(1)	030170	027445		DH6	;HALT 20
(1)	030172	030224		DT6	
(1)					
(1)	030174	000003		DT1:	3
(1)	030176	006	021	.BYTE	6,17.
(1)	030200	001252		SAVR1	
(1)	030202	006	017	.BYTE	6,15.
(1)	030204	001254		SAVR2	
(1)	030206	006	002	.BYTE	6,2
(1)	030210	001256		SAVR3	
(1)	030212	000002		DT5:	2
(1)	030214	006	017	.BYTE	6,15.
(1)	030216	001252		SAVR1	
(1)	030220	006	002	.BYTE	6,2
(1)	030222	001254		SAVR2	
(1)	030224	000003		DT6:	3
(1)	030226	006	004	.BYTE	6,4
(1)	030230	001262		SAVR5	
(1)	030232	006	002	.BYTE	6,2
(1)	030234	001260		SAVR4	
(1)	030236	006	002	.BYTE	6,2
(1)	030240	001256		SAVR3	

CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 116
CZDPBC.P11 02-MAY-78 08:23 CRC DEBUGGING AID TEST

L 9

SEQ 0115

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

030242

000001

CORMAX:
.END

ABRT	007000	1720#												
ACC	006750	1711#	5654	5684	5714	5744								
ADRCNT=	003735	1300*	1336*	1345#										
BCCFBK	026376	5766*	5772*	5775*	5778	5782	5794#							
BINWRD	004240	1387*	1388	1425#										
BITW =	002000	777#												
BIT0 =	000001	639#	759	2200	2882	2886	2890	2892	3290	3297	3302	3650	3657	3662
		5770												
BIT1 =	000002	638#	758	2230	2530	2534	2538	2540	2914	2918	2922	2924	3690	3697
		3702	4879											
BIT10 =	002000	629#	749	764	777	797	2072	2500	2754	2758	2762	2764	3202	3206
		3210	3212	3410	3417	3422	4050	4057	4062					
BIT11 =	004000	628#	748	776	796	2104	2786	2790	2794	2796	3450	3457	3462	
BIT12 =	010000	627#	747	763	770	775	795	2136	2818	2822	2826	2828	3490	3497
		3502	4090	4097	4102	4375	4382	4387	4586	4589	4598	4603		
BIT13 =	020000	626#	746	774	794	2168	2850	2854	2858	2860	3530	3537	3542	4415
		4422	4427											
BIT14 =	040000	625#	745	762	773	793	1059	1207	3570	3577	3582	4130	4137	4142
		4291	4299	4304	4455	4462	4467							
BIT15 =	100000	624#	744	761	768	772	792	1056	3610	3617	3622	4170	4177	4182
		4333	4341	4346	4495	4502	4507	4634	4637	4646	4651	5655	5685	5715
		5745												
BIT2 =	000004	637#	757	1145	2260	2562	2566	2570	2572	2946	2950	2954	2956	3730
		3737	3742	4911										
BIT3 =	000010	636#	756	783	2008	2290	2594	2598	2602	2604	2690	2694	2698	2700
		2978	2982	2986	2988	3770	3777	3782	4943					
BIT4 =	000020	635#	755	782	2040	2320	2626	2630	2634	2636	2722	2726	2730	2732
		3010	3014	3018	3020	3810	3817	3822	4817					
BIT5 =	000040	634#	754	2350	3042	3046	3050	3052	3850	3857	3862	4787	5007	5011
		5015	5017											
BIT6 =	000100	633#	753	781	2380	3074	3078	3082	3084	3890	3897	3902	4757	4847
		4975	4979	4983	4985	5039	5043	5047	5049					
BIT7 =	000200	632#	752	780	1809	1810	1826	1849	1961	2011	2017	2043	2049	2075
		2081	2107	2113	2139	2145	2171	2177	2410	3106	3110	3114	3116	3330
		3337	3342	3930	3937	3942	4211	4223	4253	4269	4295	4311	4337	4353
		4682	4850	4856										
BIT8 =	000400	631#	751	766	779	799	2440	2658	2662	2666	2668	3138	3142	3146
		3148	3970	3977	3982	4684	4694	4699	4702	4727				
BIT9 =	001000	630#	750	765	769	778	798	2470	3170	3174	3178	3180	3370	3377
		3382	4010	4017	4022	4249	4257	4262	4538	4541	4550	4555		
BRW	003306	1151	1229#											
BRX	003310	1152	1230#											
CALBCC	026400	5647*	5656	5677*	5686	5707*	5716	5737*	5788*	5795#	5825*	5826	5849	5866
CARDET=	010000	747#												
CHRCNT	004236	1385*	1389	1405*	1423#	1424								
CLK =	020000	774#	1555	1558	5166	5171								
CLK.A	001433	885#	1671											
CLRVEC	005104	1580#												
CNVRT =	104411	852#	1168	1170	1172	1174	1498	1500						
CONVRT=	104410	850#	1104	1514	5877									
CORMAX	030242	5885#												
CRCEN =	001000	769#	5112	5160	5268	5372	5439	5487	5514	5560	5605			
CRCERR=	010000	763#												
CRC.CC=	102010	5634	5797#	5822										
CRC16 =	120001	5796#												
CREAM	001316	739#	966*	1625*	1626	1628*	1633	1634*	1635	1638*				

[illegible]

CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

CZDPBC.P11 02-MAY-78 08:23 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

	5281	5282	5316	5375	5377	5379	5392	5399	5401	5403	5407	5442	5444
	5445	5449	5455	5457	5490	5492	5494	5496	5517	5519	5521	5526	5528
	5563	5566	5567	5571	5575	5608	5610	5612	5644	5646	5653	5674	5676
	5683	5704	5706	5713	5734	5736	5743	5835	5837	5838			
POPRO = 012600	618#	1536											
POP1SP= 005726	616#												
POP2SP= 022626	620#	1227											
PRIPTY 001320	803#	1027*	1091*										
PRISEC= 010000	770#												
PS = 177776	609#	959*	1141*										
PUSHRO= 010046	617#	1521											
PUSH1S= 005746	615#												
PUSH2S= 024646	619#												
PWRUP 005060	1570	1572#											
QV.FLG 001343	821#	965*	1177*	1216									
RABORT= 002000	764#												
RCRCIN= 040000	793#												
RCRC7T= 100000	792#												
RCVEN = 000020	755#	5832											
REACT= 004000	748#												
REOM = 001000	765#												
RESREG 004614	1513	1516#											
RESTRT 003116	1176	1180	1188#										
RESOS = 104407	848#	1516											
RETURN 001214	697#	971*	1153*	1155	1188*	1225*	1228	1542*	1544	1577	1693*	1703*	1704
RING = 040000	745#												
RINTEN= 000100	753#												
RSOM = 000400	766#												
RTS = 000004	757#												
RUN 001314	737#	967*	1620	1623*	1624*	1631*	1632*						
RXCSR 001404	867#	1192	1639*	1643	1769	1803	1821	1843	1884	2529	2561	2593	2625
	2657	3289	3329	3369	3409	3449	3489	3529	3569	3609	4726	4756	4786
	4816	4878	4910	4942	4974	5006	5832*						
RXDBUF 001406	868#	1645*	1646*	1647	1648	1649	1908	3234	3649	3689	3729	3769	3809
	3849	3889	3929	3969	4009	4049	4089	4129	4169				
RXDERR= 100000	761#												
RXDONE= 000200	752#												
SAVACT 001312	735#	1060*	1066*	1068	1069	1095*	1121						
SAVNUM 001313	736#	962*	1062*	1096*	1175*	1178*							
SAVPC 001266	722#	1350*	1548										
SAVPCA 001306	731#												
SAVR0 001250	715#	1359*	1364										
SAVR0A 001270	724#												
SAVR1 001252	716#	1358*	1365	5824*	5885								
SAVR1A 001272	725#												
SAVR2 001254	717#	1357*	1366	5839*	5885								
SAVR2A 001274	726#												
SAVR3 001256	718#	1356*	1367	5867*	5870*	5885							
SAVR3A 001276	727#												
SAVR4 001260	719#	1355*	1368	5885									
SAVR4A 001300	728#												
SAVR5 001262	720#	1354*	1369	5885									
SAVR5A 001302	729#												
SAVSP 001264	721#												
SAVSPA 001304	730#												
SAV05 = 104406	846#	1476											

CROSS REFERENCE TABLE -- USER SYMBOLS

834#	1781	1857	1891	1914	1939	1965	1990	2022	2054	2086	2118	2150
2182	2212	2242	2272	2302	2332	2362	2392	2422	2452	2482	2512	254
2576	2608	2640	2672	2704	2736	2768	2800	2832	2864	2896	2928	2960
2992	3024	3056	3088	3120	3152	3184	3216	3242	3272	3312	3352	3392
3432	3472	3512	3552	3592	3632	3672	3712	3752	3792	3832	3872	3912
3952	3992	4032	4072	4112	4152	4192	4230	4273	4315	4357	4397	4437
4477	4517	4565	4613	4661	4709	4739	4769	4799	4829	4861	4893	4925
4957	4989	5021	5053	5089	5137	5212	5239	5330	5414	5463	5537	5582
5618	5755	5862										
836#	1775	1815	1830	1853	5184	5191	5196	5664	5694	5724	5753	
782#	5114	5115	5161	5173	5197	5234	5269	5373	5406	5440	5454	5488
5493	5515	5520	5561	5606	5642	5672	5702	5732	5833			
856#	1038	1042	1048	1052								
807#												
5648	5678	5708	5738	5760#	5842							
1565#	2533	2565	2597	2629	2661	2693	2725	2757	2789	2821	2853	2885
2917	2949	2981	3013	3045	3077	3109	3141	3173	3205	3300	3340	3380
3420	3460	3500	3540	3580	3620	3660	3700	3740	3780	3820	3860	3900
3940	3980	4020	4060	4100	4140	4180	4225	4260	4302	4344	4385	4425
4465	4505	4553	4601	4649	4689	4697	4978	5010	5042	5070	5088	5111
5157	5231	5265	5369	5436	5484	5507	5511	5557	5602	5639	5669	5699
5729	5821	5861										
1386*	1407	1410*	1424#									
749#												
811#	1053											
610#	960	1142	1543	1573								
756#												
810#	1049											
751#												
1354#												
684#	976	978*	990	994	1108	1116	1121	1126	1145	1207	1214	1235
1248	1465	1470	1519	1538	1540	1675	5864					
672#	978											
591#	994	1108										
590#	1675											
589#												
587#	1116											
586#												
585#												
584#												
583#												
582#	1538											
581#	1235	5864										
580#	1540											
579#	1214											
578#	1248	1465										

CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

CZDPBC MACV11 30A(1052) 02-MAY-78 13:58 PAGE 125

CZDPBC.P11 02-MAY-78 08:23

CROSS REFERENCE TABLE -- USER SYMBOLS

TST126	021514	4755	4784#	
TST127	021570	4785	4814#	
TST13	010244	2070	2101#	
TST130	021644	4815	4844#	
TST131	021730	4845	4876#	
TST132	022014	4877	4908#	
TST133	022100	4909	4940#	
TST134	022164	4941	4972#	
TST135	022260	4973	5004#	
TST136	022354	5005	5036#	
TST137	022450	5037	5067#	
TST14	010330	2102	2133#	
TST140	022604	5068	5105#	
TST141	022760	5106	5152#	
TST142	023274	5153	5226#	
TST143	023364	5227	5260#	
TST144	023740	5261	5364#	
TST145	024226	5365	5431#	
TST146	024430	5432	5479#	
TST147	025002	5480	5552#	
TST15	010414	2134	2165#	
TST150	025176	5553	5597#	
TST151	025332	5598	5632#	
TST152	026402	5633	5818#	5885
TST153=	***** U	5819		
TST16	010500	2166	2197#	
TST17	010554	2198	2227#	
TST2	007272	1766	1799#	
TST20	010630	2228	2257#	
TST21	010704	2258	2287#	
TST22	010760	2288	2317#	
TST23	011034	2318	2347#	
TST24	011110	2348	2377#	
TST25	011164	2378	2407#	
TST26	011240	2408	2437#	
TST27	011314	2438	2467#	
TST3	007536	1800	1881#	
TST30	011370	2468	2497#	
TST31	011444	2498	2527#	
TST32	011540	2528	2559#	
TST33	011634	2560	2591#	
TST34	011730	2592	2623#	
TST35	012024	2624	2655#	
TST36	012120	2656	2687#	
TST37	012214	2688	2719#	
TST4	007600	1882	1905#	
TST40	012310	2720	2751#	
TST41	012404	2752	2783#	
TST42	012500	2784	2815#	
TST43	012574	2816	2847#	
TST44	012670	2848	2879#	
TST45	012764	2880	2911#	
TST46	013060	2912	2943#	
TST47	013154	2944	2975#	
TST5	007636	1906	1929#	
TST50	013250	2976	3007#	

TST51	013344	3008	3039#											
TST52	013440	3040	3071#											
TST53	013534	3072	3103#											
TST54	013630	3104	3135#											
TST55	013724	3136	3167#											
TST56	014020	3168	3199#											
TST57	014114	3200	3232#											
TST6	007700	1930	1954#											
TST60	014154	3233	3260#											
TST61	014216	3261	3287#											
TST62	014330	3288	3327#											
TST63	014442	3328	3367#											
TST64	014554	3368	3407#											
TST65	014666	3408	3447#											
TST66	015000	3448	3487#											
TST67	015112	3488	3527#											
TST7	007746	1955	1980#											
TST70	015224	3528	3567#											
TST71	015336	3568	3607#											
TST72	015450	3608	3647#											
TST73	015562	3648	3687#											
TST74	015674	3688	3727#											
TST75	016006	3728	3767#											
TST76	016120	3768	3807#											
TST77	016232	3808	3847#											
TTST	003174	1148*	1149*	1151*	1152*	1208#								
TWOSYN=	000000	640#												
TXACT =	001000	778#	5167	5203	5386	5502	5534	5613						
TXCSR	001412	870#	1527*	1555*	1558*	1652*	1653*	1654	1714	1723	1735	1740	1747	1957
		2007	2039	2071	2103	2135	2167	2532*	2564*	2596*	2628*	2660*	2689	2692*
		2721	2724*	2753	2756*	2785	2788*	2817	2820*	2849	2852*	2884*	2916*	2948*
		2980*	3012*	3044*	3076*	3108*	3140*	3172*	3204*	3299*	3339*	3379*	3419*	3459*
		3499*	3539*	3579*	3619*	3659*	3699*	3739*	3779*	3819*	3859*	3899*	3939*	3979*
		4019*	4059*	4099*	4139*	4179*	4210	4224*	4248	4259*	4290	4301*	4332	4343*
		4384*	4424*	4464*	4504*	4552*	4600*	4648*	4681	4696*	4846	4977*	5009*	5038
		5041*	5069*	5072*	5087*	5108	5110*	5127	5154	5156*	5166*	5171*	5185	5229
		5230*	5262	5264*	5310	5366	5368*	5433	5435*	5446	5450	5454*	5458	5481
		5483*	5493*	5497	5502	5506*	5508	5510*	5520*	5522	5529	5534	5554	5556*
		5568	5572	5576	5579	5599	5601*	5613	5638*	5641*	5642*	5668*	5671*	5672*
		5698*	5701*	5702*	5728*	5731*	5732*	5820*	5831*	5833*	5860*			
TXDBUF	001414	871#	1656*	1657*	1658	1983	2199	2229	2259	2289	2319	2349	2379	2409
		2439	2469	2499										

WRKO.F 004570
 XBK 004374
 XCSR 003130
 XERR 003152
 XHEAD 006015
 XPASS 003144
 XPOLY 026374
 XSTATQ 006044
 XTSTN 005000
 XVEC 003136
 \$E = 000154

\$ENDAD 003104
 \$N = 000152

\$RAYO = 177777

1504	1507#												
1466	1468	1470#											
1168	1190#												
1174	1199#												
1100	1591#												
1172	1196#												
5634*	5773	5784	5793#	5822*									
1105	1591#												
1498	1549#												
1170	1193#												
1#	1766	1768#	1800	1802#	1882	1883#	1906	1907#	1930	1931#	1955	1956#	
1981	1982#	2006	2007#	2038	2039#	2070	2071#	2102	2103#	2134	2135#	2166	
2167#	2178	2199#	2228	2229#	2258	2259#	2288	2289#	2318	2319#	2348	2349#	
2378	2379#	2408	2409#	2438	2439#	2468	2469#	2498	2499#	2528	2529#	2560	
2561#	2592	2593#	2624	2625#	2656	2657#	2688	2689#	2720	2721#	2752	2753#	
2784	2785#	2816	2817#	2848	2849#	2880	2881#	2912	2913#	2944	2945#	2976	
2977#	3008	3009#	3040	3041#	3072	3073#	3104	3105#	3136	3137#	3168	3169#	
3200	3201#	3233	3234#	3261	3262#	3288	3289#	3328	3329#	3368	3369#	3408	
3409#	3448	3449#	3488	3489#	3528	3529#	3568	3569#	3608	3609#	3648	3649#	
3688	3689#	3728	3729#	3768	3769#	3808	3809#	3848	3849#	3888	3889#	3928	
3929#	3968	3969#	4008	4009#	4048	4049#	4088	4089#	4128	4129#	4168	4169#	
4208	4209#	4247	4248#	4289	4290#	4331	4332#	4373	4374#	4413	4414#	4453	
4454#	4493	4494#	4536	4537#	4584	4585#	4632	4633#	4680	4681#	4725	4726#	
4755	4756#	4785	4786#	4815	4816#	4845	4846#	4877	4878#	4909	4910#	4941	
4942#	4973	4974#	5005	5006#	5037	5038#	5068	5069#	5106	5107#	5153	5154#	
5227	5228#	5261	5262#	5365	5366#	5432	5433#	5480	5481#	5553	5554#	5598	
5599#	5633	5634#	5819	5820#									
666	986	1182#	1517										
1#	1754	1759	1768#	1786	1793	1802#	1869	1875	1883#	1894	1899	1907#	
1917	1923	1931#	1942	1948	1956#	1968	1974	1982#	1993	1999	2007#	2025	
2031	2039#	2057	2063	2071#	2089	2095	2103#	2121	2127	2135#	2153	2159	
2167#	2185	2191	2199#	2215	2221	2229#	2245	2251	2259#	2275	2281	2289#	
2305	2311	2319#	2335	2341	2349#	2365	2371	2379#	2395	2401	2409#	2425	
2431	2439#	2455	2461	2469#	2485	2491	2499#	2515	2521	2529#	2547	2553	
2561#	2579	2585	2593#	2611	2617	2625#	2643	2649	2657#	2675	2681	2689#	
2707	2713	2721#	2739	2745	2753#	2771	2777	2785#	2803	2809	2817#	2835	
2841	2849#	2867	2873	2881#	2899	2905	2913#	2931	2937	2945#	2963	2969	
2977#	2995	3001	3009#	3027	3033	3041#	3059	3065	3073#	3091	3097	3105#	
3123	3129	3137#	3155	3161	3169#	3187	3193	3201#	3219	3226	3234#	3245	
3254	3262#	3275	3281	3289#	3315	3321	3329#	3355	3361	3369#	3395	3401	
3409#	3435	3441	3449#	3475	3481	3489#	3515	3521	3529#	3555	3561	3569#	
3595	3601	3609#	3635	3641	3649#	3675	3681	3689#	3715	3721	3729#	3755	
3761	3769#	3795	3801	3809#	3835	3841	3849#	3875	3881	3889#	3915	3921	
3929#	3955	3961	3969#	3995	4001	4009#	4035	4041	4049#	4075	4081	4089#	
4115	4121	4129#	4155	4161	4169#	4194	4201	4209#	4234	4240	4248#	4276	
4282	4290#	4318	4324	4332#	4360	4366	4374#	4400	4406	4414#	4440	4446	
4454#	4480	4486	4494#	4520	4529	4537#	4568	4577	4585#	4616	4625	4633#	
4664	4673	4681#	4712	4718	4726#	4742	4748	4756#	4772	4778	4786#	4802	
4808	4816#	4832	4838	4846#	4864	4870	4878#	4896	4902	4910#	4928	4934	
4942#	4960	4966	4974#	4992	4998	5006#	5024	5030	5038#	5056	5061	5069#	
5093	5099	5107#	5140	5146	5154#	5215	5220	5228#	5241	5254	5262#	5350	
5358	5366#	5418	5425	5433#	5466	5473	5481#	5539	5546	5554#	5585	5591	
5599#	5620	5626	5634#	5800	5812	5820#	5885#						
1#	1754#	1755	1757#	1786#	1787	1791#	1869#	1870	1873#	1894#	1895	1897#	
1917#	1918	1921#	1942#	1943	1946#	1968#	1969	1972#	1993#	1994	1997#	2025#	
2026	2029#	2057#	2058	2061#	2089#	2090	2093#	2121#	2122	2125#	2153#	2154	

CZDPBC.P11 02-MAY-78 08:23

CROSS REFERENCE TABLE -- USER SYMBOLS

		2157#	2185#	2186	2189#	2215#	2216	2219#	2245#	2246	2249#	2275#	2276	2279#
		2305#	2306	2309#	2335#	2336	2339#	2365#	2366	2369#	2395#	2396	2399#	2425#
		2426	2429#	2455#	2456	2459#	2485#	2486	2489#	2515#	2516	2519#	2547#	2548
		2551#	2579#	2580	2583#	2611#	2612	2615#	2643#	2644	2647#	2675#	2676	2679#
		2707#	2708	2711#	2739#	2740	2743#	2771#	2772	2775#	2803#	2804	2807#	2835#
		2836	2839#	2867#	2868	2871#	2899#	2900	2903#	2931#	2932	2935#	2963#	2964
		2967#	2995#	2996	2999#	3027#	3028	3031#	3059#	3060	3063#	3091#	3092	3095#
		3123#	3124	3127#	3155#	3156	3159#	3187#	3188	3191#	3219#	3220	3224#	3245#
		3246	3252#	3275#	3276	3279#	3315#	3316	3319#	3355#	3356	3359#	3395#	3396
		3399#	3435#	3436	3439#	3475#	3476	3479#	3515#	3516	3519#	3555#	3556	3559#
		3595#	3596	3599#	3635#	3636	3639#	3675#	3676	3679#	3715#	3716	3719#	3755#
		3756	3759#	3795#	3796	3799#	3835#	3836	3839#	3875#	3876	3879#	3915#	3916
		3919#	3955#	3956	3959#	3995#	3996	3999#	4035#	4036	4039#	4075#	4076	4079#
		4115#	4116	4119#	4155#	4156	4159#	4194#	4195	4199#	4234#	4235	4238#	4276#
		4277	4280#	4318#	4319	4322#	4360#	4361	4364#	4400#	4401	4404#	4440#	4441
		4444#	4480#	4481	4484#	4520#	4521	4527#	4568#	4569	4575#	4616#	4617	4623#
		4664#	4665	4671#	4712#	4713	4716#	4742#	4743	4746#	4772#	4773	4776#	4802#
		4803	4806#	4832#	4833	4836#	4864#	4865	4868#	4896#	4897	4900#	4928#	4929
		4932#	4960#	4961	4964#	4992#	4993	4996#	5024#	5025	5028#	5056#	5057	5059#
		5093#	5094	5097#	5140#	5141	5144#	5215#	5216	5218#	5241#	5242	5252#	5350#
		5351	5356#	5418#	5419	5424#	5466#	5467	5471#	5539#	5540	5544#	5585#	5586
		5589#	5620#	5621	5624#	5800#	5801	5811#						
\$Y	= 000014	825#	834	836#	838#	840#	842#	844#	846#	848#	850#	852#	854#	856#
.	= 030242	858#												
		651#	652	655#	662#	667#	670#	673#	677#	679#	733#	734#	735#	736#
		737#	738#	804#	805#	810#	811#	892#	894#	895#	896#	898#	899#	900#
		902#	903#	904#	906#	907#	908#	910#	911#	912#	914#	915#	916#	918#
		919#	920#	922#	923#	924#	995	1125	1591#	1599#	1601#	1603#	1604#	1619
		1715	5200	5851										
.BEGIN	002660	1141#												
.CNVRT	004034	853	1377#											
.CONVR	004030	851	1376#											
.EOP	002764	1162#	5819											
.ERRTA	030026	1484	5885#											
.HLT	004350	658	1465#											
.INSTE	003516	843	1282#											
.INSTR	003412	841	1261#											
.INST1	003432	1265#	1285											
.MSG	003434	1263*	1266#											
.PARAM	003536	845	1293#											
.PFAIL	005050	656	961	1570#	1574									
.PKCLK	005006	855	1552#											
.RES05	003776	849	1364#											
.SAV05	003736	847	1350#											
.SCOPE	003160	835	1205#											
.SCOP1	003312	837	1235#											
.SETFL	004242	857	1433#	1445										
.START	001562	674	959#	971										
.TRPSR	004316	660	1453#											
.TRPTA	001344	833#	1458											
.TYPE	003336	839	1245#											

. ABS. 030242 000

CZDPBC MACY11 30A(1052) 02-MAY-78 13:58 PAGE 129

K 10

SEQ 0127

CZDPBC.P11 02-MAY-78 08:23

CROSS REFERENCE TABLE -- USER SYMBOLS

ERRORS DETECTED: 0

DSKZ:CZDPBC,DSKZ:CZDPBC.SEQ=DSKZ:DUP11.MAC,DSKZ:CZDPBC.P11

RUN-TIME: 20 29 1 SECONDS

RUN-TIME RATIO: 250/52=4.7

CORE USED: 24K (47 PAGES)

DOCUMENT PAGES: 127